Evolutionary Approaches for Network Coding Based Multicast Routing Problems

by

Huanlai Xing, BSc, MSc.

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

June 2013

Abstract

Network coding is an emerging technique in communication networks, where the intermediate nodes are allowed to combine (code) the data received from different incoming links if necessary. The thesis investigates a number of routing problems for network coding based multicast (NCM), which belong to combinatorial optimization problems (COPs). Evolutionary algorithms (EAs) are used to tackle the problems. The contributions of the thesis are described below.

We propose three EAs for the network coding resource minimization (NCRM) problem where the objective is to minimize the number of coding operations performed while meeting the data rate requirement based on NCM. The three EAs are population based incremental learning (PBIL), compact genetic algorithm (cGA) and path-oriented encoding EA (pEA), all specially developed for tackling the NCRM problem. PBIL adopts the binary link state (BLS) encoding which is an existing encoding approach. We find that PBIL is more suited to BLS encoding, compared with genetic algorithms (GAs). This is because the principle of updating the probability vector (PV) in PBIL can easily locate promising regions in the search space. An entropy-based restart scheme is developed, which can improve the global exploration ability of PBIL. cGA is also based on BLS encoding and fit for this encoding. Three improvement schemes are developed based on the features of the encoding and the structure of cGA, including all-one vector, a PV restart scheme and a problem specific local search operator. In particular, the local search operator is the very first domain-knowledge based operator for the problem concerned. We notice that BLS encoding and its variant have some weaknesses, namely high proportion of infeasible solutions in the search space and high computational costs spent on fitness evaluation. We therefore design a path-oriented encoding and develop pEA based on it, where a local search operator (based on the path-oriented encoding) is incorporated into the evolutionary framework to improve the local exploitation ability. Experimental results demonstrate that the three adapted EAs outperform existing EAs in the literature, in terms of the best results obtained and computational time.

To support real-time multimedia applications, we for the first time extend the NCRM problem by introducing the maximum transmission delay into the problem as a constraint, which is called the delay constrained NCRM problem. Benchmark datasets are created based on the datasets for the NCRM problem. Three EAs originally used for the NCRM problem are adapted for the delay constrained NCRM problem, including GAs and PBIL. We find that for the same dataset a severer delay constraint leads to a harder problem. We also find that the restart scheme in PBIL is not suited to the delay constrained NCRM problem, due to the setting of threshold value and the memory-less structure. We therefore design a substitute for the restart scheme to adapt PBIL for the delay constrained problem, namely the combination of a new PV update scheme and a PV mutation. It is noted that the PV update scheme is based on a set of historically best solutions. The combination improves the global exploration of PBIL and avoids local optima. PBIL with the new PV update scheme and PV mutation performs better than PBIL with restart scheme.

To study the conflicting interests of service providers and network users, we for the first time formulate a multi-objective NCM routing problem considering two objectives, cost and delay. The cost is the summation of the coding cost and link cost incurred in the NCM. The delay is the maximum transmission delay of paths in the NCM. This problem is referred to as the cost-delay bi-objective optimization (CDBO) problem. Benchmark datasets for the delay constrained NCRM problem are used to generate the datasets for the CDBO problem. Elitist nondominated sorting GA (NSGA-II) is adapted for the CDBO problem, where two problem specific schemes are proposed, namely an initialization scheme and an individual delegate scheme (IDS). The initialization scheme can generate a set of promising and diversified initial individuals, which initiates a diversified search at the beginning of the evolution. The IDS reduces the number of individuals having identical objective values in the population and leaves more space to accept other significant individuals. This scheme considers the crowding in both decision and objective spaces and helps to diversify the search during the evolution. The adapted NSGA-II performs better than a number of multi-objective EAs in terms of the inverted generational distance, generational distance and maximum spread.

List of Publications

- Xing H and Qu R (2011). A population based incremental learning for network coding resources minimization. *IEEE Communications Letters*, **15**(7): 698-700. This paper is reported in Section 4.1.

- Xing H and Qu R (2012). A compact genetic algorithm for the network coding based resource minimization problem. *Applied Intelligence*, **36**(4): 809-823. This paper is shown in Section 4.2.

- Xing H, Qu R, Kendall G, and Bai R. A path-oriented encoding evolutionary algorithm for network coding resource minimization. *Journal of Operational Research Society* (Accepted). This work is reported in Section 4.3.

- Xing H and Qu R (2011). A population based incremental learning for delay constrained network coding resource minimization. In *Proc. of EvoApplications*, Torino, Italy, pp. 51-60. This paper is shown in Chapter 5.

- Xing H and Qu R (2013). A nondominated sorting genetic algorithm for bi-objective network coding based multicast routing problems. *Information Sciences*, accepted, http://dx.doi.org/10.1016/j.ins.2013.01.014. This work is reported in Chapter 6.

Acknowledgements

I would like to express my sincere gratitude to my PhD supervisor, Dr. Rong Qu. Without her guidance and support, this thesis would never be complete. Her invaluable advices and suggestions are always sources of inspiration and make sure that my project is on the right track. I appreciate her help and encouragement throughout my PhD life which has greatly helped me overcome difficulties, gain confidence and move forward. I feel so lucky to be one of her students.

I would like to express my gratefulness to Dr. Dario Landa-Silva from University of Nottingham and Prof. Edward Tsang from University of Essex, for being my PhD viva examiners and providing insightful comments which has greatly helped to improve the quality of my thesis. Meanwhile, I would like to thank Dr. Landa-Silva for his great help and collaboration during my PhD study. I would also like to thank Dr. Jason Atkin for his valuable comments on the progress of my project and Dr. Ruibin Bai from University of Nottingham Ningbo Campus for academic discussion and cooperation.

I am very grateful to Prof. Wei Pan from Southwest Jiaotong University, Prof. Yuefeng Ji, Prof. Lin Bai and Prof. Yongmei Sun from Beijing University of Posts & Telecommunications, and Prof. Zhengping Wang from Harbin Engineering University for their valuable advices and consistent help. It is so nice to have them.

I would like to thank all members in ASAP group for contributing such an active and productive research environment in which I have had a pleasant experience. Special thanks to Fang, Mashael, Celina, Crystal, Khin, Yan, Jane, Grace, Bee, Jiawei, Yuqing, Amadeo, Saiful, Jufang, Hui and Yi (Wang) for their friendship, fruitful discussions and every moment we shared together. I would also like to thank Poayhoon, Tao, and Jianyong from IMA group.

My acknowledgement also goes to all my friends in UK and China. First, I would like to thank Haowen, Hao, Wenjuan (Yuan), Fangfang, Wenjuan (Deng), Bei, Ye, and Haiping for their help and encouragement, as well as many unforgetable moments of happiness, celebrations, and friendship. Special thanks also go to Patricia, Wei and Lin,

Alex and Maggie. It was really my pleasure and honour to be your housemate. I am sure after years I could still memorize the pleasant times we spent together shopping, having dinner (hot pot, barbecue, and dining out), watching movies and chatting. You are so nice housemates and close friends and have helped me a lot. I appreciate your friendship and all joyful moments we enjoyed. I would also like to take this opportunity to express my thankfulness to some old friends, Yingbin, Zhenfei, Xian, Lexi, Hanliang, Jia (Fu), Kunpeng, Bonan, and Zijian for their friendship and encouragement during my study.

I am grateful to China Scholarship Council (CSC) and University of Nottingham. My PhD would have not been possible without their joint international scholarship for Chinese students. I would also like to thank Dr. Dario Landa-Silva and Dr. Rong Qu for the internship they provided to ease my financial difficulities.

Last but not least, I am deeply grateful to my parents for their continuous love and endless support. Otherwise, I would not start the four years' journey pursuing my PhD in UK which is thousands of miles away from home.

Contents

List of Figures

List of Tables

List of Abbreviations

ACO:	Ant Colony Optimization
ACT:	Average Computational Time
AHD:	Average Hamming Distance
ATG:	Average Termination Generation
BGMP:	Border Gateway Multicast Protocol
BLS:	Binary Link State
BTS:	Block Transmission State
BU:	Basic Unit
CBT:	Core Based Tree
CDBO:	Cost-Delay Bi-objective Optimization
cGA:	Compact Genetic Algorithm
COPs:	Combinatorial Optimization Problems
DE:	Differential Evolution
DM:	decision maker
DVMRP:	Distance Vector Multicast Routing Protocol
EAs:	Evolutionary Algorithms
EC:	Evolutionary Computation
EDAs:	Estimation of Distribution Algorithms
ES:	Evolutionary Strategy
GAs:	Genetic Algorithms
GD:	Generational Distance
GF:	Galois Field (also called finite field)
hEA:	Hybrid Evolutionary Algorithm
IDS:	Individual Delegate Scheme
IGD:	Inverted Generational Distance
IGMP:	Internet Group Management Protocol
IP:	Internet Protocol
IPTV:	Internet Protocol Television
LS:	Local Search
MAs:	Memetic Algorithms
MOEAs:	Multiobjective Evolutionary Algorithms
MOPBIL:	Multiobjective Population Based Incremental Learning
MOPs:	Multiobjective Optimization Problems
MOSPF:	Multicast Open Shortest Path First
MS:	Maximum Spread
NCM:	Network Coding based Multicast
NCRM:	Network Coding Resource Minimization
NSGA:	Nondominated Sorting Genetic Algorithm
NSGA-II:	Elitist Nondominated Sorting Genetic Algorithm
ORX:	OR-based Crossover
PAES:	Pareto Archived Evolution Strategy
PBIL:	Population Based Incremental Learning
pEA:	Path-oriented encoding Evolutionary Algorithm
PESA:	Pareto Envelope-based Selection Algorithm
PF:	Pareto-optimal Front
PIM:	Protocol Independent Multicast
PIS:	Proportion of Infeasible Solutions
PMBGAs:	Probabilistic Model Building Genetic Algorithms
PS:	Pareto set
PSO:	Particle Swarm Optimization
PV:	Probability Vector
QEA:	Quantum-inspired Evolutionary Algorithm
QMP:	Quantum Mutation Probability
QoS:	Quality of Service
RAS:	Rotation Angle Step
SD:	Standard Deviation
SDC:	Structural Difference Coefficient
sGA:	Standard Genetic Algorithm
SPEA:	Strength Pareto Evolutionary Algorithm
SPEA2:	Improved Strength Pareto Evolutionary Algorithm
SR:	Successful Ratio
UMDA:	Univariate Marginal Distribution Algorithm
UX:	Uniform Crossover
XOR:	Exclusive OR Operation

Chapter 1 Introduction

Nowadays, communication networks are evolving rapidly due to the emergence of new technologies in information theory, data transmission, networking and switching. A communication network is a set of nodes connected by links, providing communication services, such as data and information exchanging, between users of the network located at various geographical points. A node in the network represents an individual hardware device, such as router or switch. A link is a communication channel connecting two or more nodes (Halsall, 1992; Elahi, 2001).

Depending on the switching technologies adopted, networks can be classified into two categories, i.e. circuit-switched networks and packet-switched networks (Bagad and Dhotre, 2010). Circuit-switched networks set up point-to-point path connection before the data transmission. The established path cannot be used by other traffic during the transmission. The connection is released only when the transmission is over. An example is Public Switched Telephone Network (PSTN), where hiring lines is reliable but quite expensive. On the other hand, packet-switched networks (e.g. Internet) transmit data using separate and small blocks (i.e. packets). With an address embedded in the header, each packet finds its own path to its destination. Compared with circuit-switching, packet-switching is cheaper and more flexible. It becomes the trend of the development of modern switching technologies (Bagad and Dhotre, 2010). In packet-switched networks, reliable data delivery between nodes can be realized by connection-oriented protocols, e.g. Transmission Control Protocol (TCP), where all data is sent over the same path during a communication session (Goralski, 2009). In this thesis, packet-switched networks and connection-oriented schemes are the underlying network model and routing schemes for conducting the research.

Regarding the design of communication networks, one fundamental issue considers how network traffic is delivered between network terminals (namely, users). This issue is referred to as network routing which is the process of selecting a number of paths in a communication network along which the traffic is transmitted (Halsall, 1992; Elahi, 2001;

Bagad and Dhotre, 2010). The objective of network routing is to enable communications among network terminals while efficiently utilizing network resources. Depending on the number of receivers, routing can be classified into three categories: unicast (one-to-one), broadcast (one-to-all) and multicast (one-to-many). Unicast delivers data from a source to a single receiver, where a path between the source and the receiver is established. Broadcast is that a source sends the same data to all other nodes in the network. On the other hand, multicast is that a certain source node delivers the same data to a subset of nodes in the network. To forward the data to each receiver, a subnetwork spanning the source and all receivers is found, where for each receiver there is a path originating from the source and terminating at the receiver. This subnetwork is referred to as multicast tree (Benslimane, 2007). Unicast and broadcast can be seen as two special cases of multicast.

1.1 Multicast Technologies

Multicast model was originally introduced by Steve Deering in 1988 for Internet Protocal (IP) networks. Compared with multiple unicasts, multicast can efficiently reduce the bandwidth consumption during the data transmission. Figure 1.1 shows a comparison between multiple unicasts and multicast in an example scenario where source node s wants to deliver information a to three receivers, t_1, t_2 and t_3. Both multiple unicasts and multicast can complete the task. However, compared with multiple unicasts, multicast is able to reduce server loads and traffic, saving nearly half (4/9 = 44.4%) of bandwidth utilization in the above example. One may imagine that multicast becomes more and more efficient than multiple unicasts with the growth of the number of receivers and the scale of the network.

With the rapid growth of real-time multimedia applications and services, multicast has become the key technology to support these applications and services in IP networks (Benslimane, 2007; Harte, 2008). The following summarizes some common multicast protocals used in IP networks and lists a number of multicast applications.

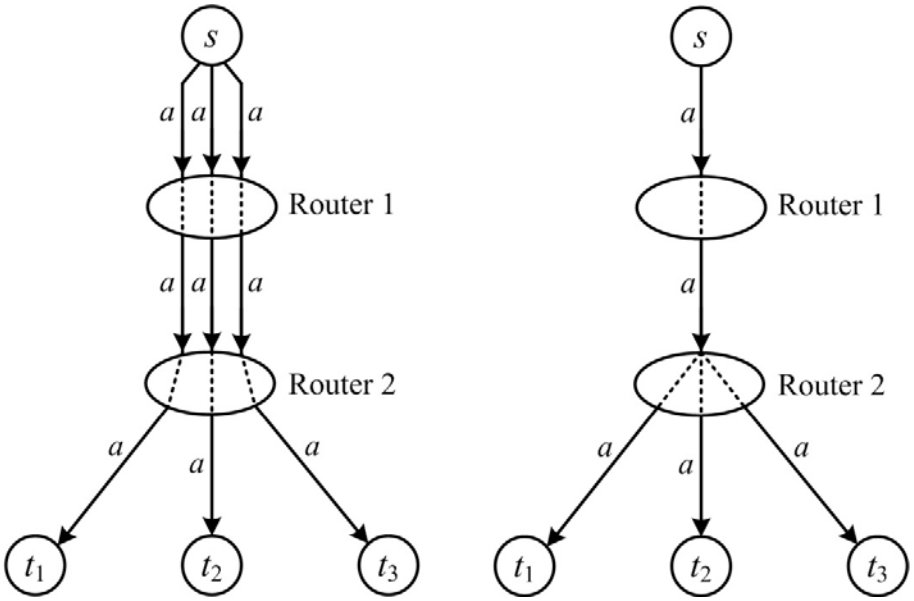


Figure 1. 1 An example of realizing one-to-many communication. (a) Multiple unicasts. (b) Multicast.

1.1.1 Multicast Protocols

A multicast protocol should be capable of collecting and updating the status information of a network, constructing and maintaining an appropriate multicast tree for a multicast group including a source node and a number of receivers when requested, and handling the changes caused by the Internet or nodes joining (or leaving) the multicast group (Benslimane, 2007; Harte, 2008). Depending on where they are deployed, multicast protocols can be categorized into three types. The first type of protocols allows end-user nodes to exchange messages with multicast routers for joining or leaving a multicast group, e.g. Internet Group Management Protocol (IGMP) (Deering, 1989). This type of protocols is installed at end-user nodes. The second type of protocols enables the multicast communications within the same autonomous system (which is one of the administrative domains in the Internet). This type of protocols are deployed on multicast routers and referred to as multicast interior gateway protocols, including Distance Vector Multicast Routing Protocol (DVMRP) (Waitzman *et al*, 1988), Core Based Tree (CBT)

(Ballardie *et al*, 1993), Multicast Open Shortest Path First (MOSPF) (Moy, 1994), and Protocol Independent Multicast (PIM) (Deering *et al*, 1996). The last type of protocols is deployed on border routers to enable the multicast between different domains on the Internet, e.g. Border Gateway Multicast Protocol (BGMP) (Thaler *et al*, 1998).

1.1.2 Multicast Applications

Nowadays, the development and commercialization on the Internet has resulted in increasing demand for multicast applications (Benslimane, 2007; Harte, 2008). Video distribution is one of the application fields, such as video conferencing and Internet Protocol Television (IPTV). Besides, remote education and periodic data delivery (e.g. stock quotes and sports scores) also requires multicast service over the Internet. Internet giants and equipment suppliers have developed their own products (software/hardware) to support increasing demand for multimedia content distribution. For example, Cisco Systems introduced TelePresence, the next generation of video conferencing, several years ago. TelePresence brings people together virtually and changes the way they work. In 2010, Microsoft unveiled Lync, a communications system that integrates audio/video conferencing, e-mail and instant messaging all into one application.

1.2 Max-Flow Min-Cut Theorem

When finding available routes to carry out data transmission in a communication network, one of the common concerns is the capacity of the network to accommodate information flows, e.g. how fast the information sent from a source can be received by its receiver(s). This concern is of particular importance to network designers because they expect to maximize the network capacity utilization to provide communication and data delivery services to the network users with the limited network resources.

To address the above concern, the Max-Flow Min-Cut Theorem was independently presented and proved by two groups of scientists, Elias, Feinstein and Shannon (1956), and Ford and Fulkerson (1956). This theorem defines the maximum flow capacity between a source node and its receiver(s) in a network.

A communication network can be modelled as a graph $G(V,E)$, where V and E are the sets of nodes and links, respectively. If G is an undirected graph, any link $e_{ik} \in E$ has no direction, where $i, k \in V$. Information can be sent from i to k and from k to i. If G is a directed graph, link e_{ik} can only forward information from i to k. Assume there are a source node s and an arbitrary non-source node t. A cut between s and t is a partition of V into two disjoint subsets S and T, where $S \cup T = V$, $S \cap T = \varnothing$, $s \in S$, and $t \in T$. Link $e_{ik} \in E$ is said to be in the cut if $i \in S$ and $k \in T$. The value of a cut is the summation of the capacity of all links in the cut.

The Max-Flow Min-Cut Theorem is stated as: given a source s and non-source t in network G, the minimum value of a cut between s and t is equal to the maximum volume of a flow from s to t (known as max-flow).

The above theorem provides a method to calculate the upper bound of the amount of information flows from s to t by simply finding the minimum cut between s and t. For any unicast session, the maximum data rate between the source and receiver is the same as the max-flow between them. For any multicast session, we use multicast throughput as the multicast data rate. Multicast throughput is defined as the minimum value of the max-flows obtained by all receivers. Figure 1.2 shows an example of the maximum data rate achieved by unicast and multicast, respectively.

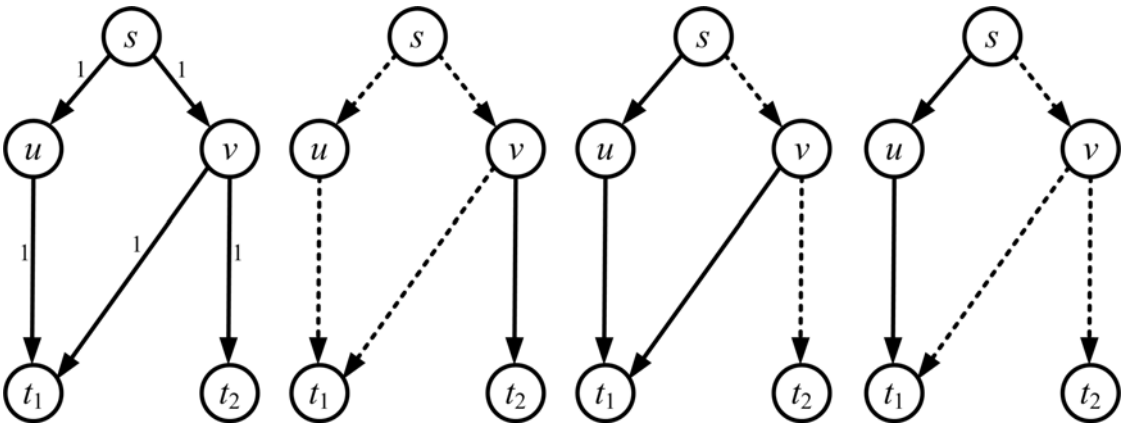


Figure 1. 2 An example of the maximum data rate. (a) the original network (b) unicast from s to t_1 (c) unicast from s to t_2 (d) multicast from s to t_1 and t_2

In the original network, there is a source s and two receivers, t_1 and t_2. For simplicity, assume each link has a unit capacity. Obviously, the max-flow from s to t_1 and t_2 is two units and one unit, respectively, which can be achieved using unicast, as shown in Figure 1.2(b)-(c), where dash lines illustrate the information flows in the network. On the other hand, if we use multicast to deliver the same data to t_1 and t_2, the multicast data rate is only one unit according to the definition of multicast throughput introduced above.

In traditional communication networks, information is delivered in the same way as cars share a highway or fluids share a pipe (Fragouli *et al*, 2006; Fragouli and Soljanin, 2007). Different data packets transmitted share the same limited network resources, but each data packet is separately processed and forwarded. Each intermediate node of a network simply replicates the incoming data packets and forwards a copy to the next node, adopting a method known as *store-and-forward* (Li *et al*, 2003). At any intermediate node, only data replication and forwarding are needed.

We refer to routing schemes which are based on store-and-forward data forwarding approach as traditional routing schemes. For example, in traditional multicast, any intermediate node simply forwards the incoming packets to the next node(s).

When investigating the capacity of traditional routing schemes, people found that traditional multicast cannot always reach the maximum throughput which is supported by the Max-Flow Min-Cut Theorem. Inspired by this deficiency, Ahlswede *et al* (2000) introduced the idea of network coding. When used in multicast, this new idea guarantees to achieve the theoretical maximum multicast throughput.

1.3 Network Coding

Network coding can be viewed as a generalized routing scheme and can be applied to any types of routing, i.e. unicast, multicast and broadcast, to enable a more efficient data transmission. In network coding, any intermediate node is allowed to not only forward but also combine (code) data packets received from different incoming links if necessary (Ahlswede *et al*, 2000; Li *et al*, 2003). The forwarding scheme in network coding is referred to as *code-and-forward* (Xing *et al*, 2010).

The following is an example comparing traditional multicast and *network coding based multicast* (NCM) in terms of the multicast throughput, as shown in Figure 1.2. Figure 1.2(*a*) is the example network, where source *s* multicasts two bits information (*a* and *b*) to receivers *y* and *z*. Assume each link is directed and has a capacity of one bit per time unit. The minimum cut C_{min} between the source and each receiver (*y* or *z*) is two bits per time unit. According to the MAX-FLOW MIN-CUT theorem, the maximum flow is equal to the minimum cut, and therefore the maximum multicast throughput is also two bits per time unit. By using traditional routing, however, the average data rate between *s* and each receiver is 1.5 bits per time unit and hence the multicast throughput is only 1 bit per time unit, as shown in Figure 1.2(*b*). This is because $w \rightarrow x$ is a bottleneck link which can only forward one bit (*a* or *b*) at a time to *x*, so *y* and *z* cannot simultaneously receive two different bits. However, if node *w* is allowed to combine *a* and *b* into one bit $a \oplus b$ ($\oplus$ denotes an Exclusive-OR operation) and to forward $a \oplus b$ to *x* as depicted in Figure 1.2(*c*), both *y* and *z* can receive two bits information, i.e. $\{a, a \oplus b\}$ in *y* and $\{b, a \oplus b\}$ in *z*. By decoding operations $b = a \oplus (a \oplus b)$ and $a = b \oplus (a \oplus b)$, *y* and *z* can then recover *b* and *a*, respectively. In this way, the theoretical maximum throughput is warranted.

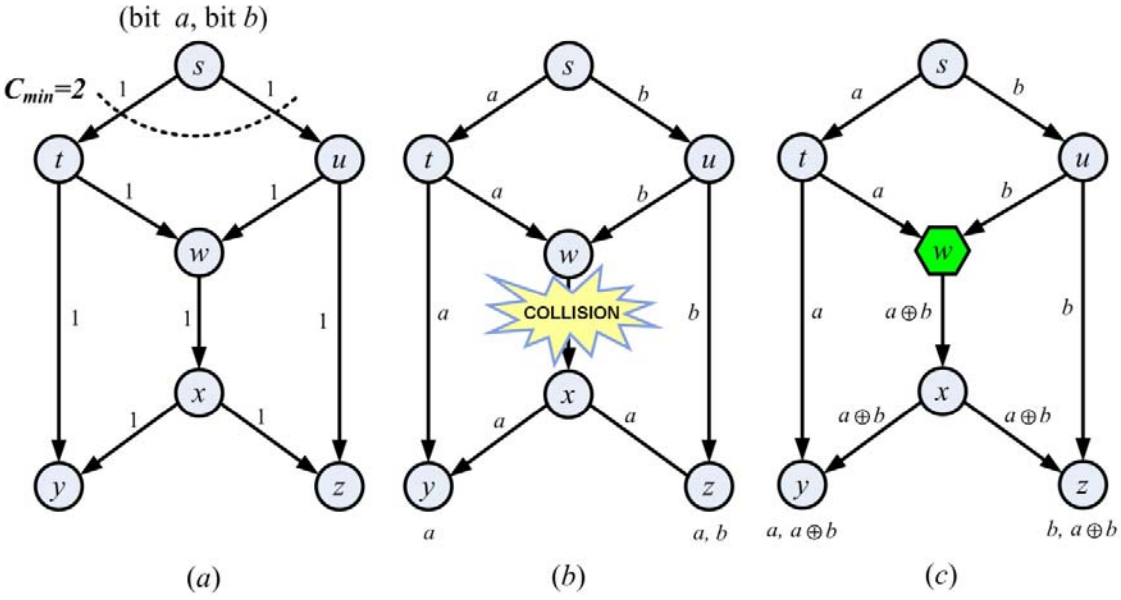


Figure 1. 3 Traditional multicast vs. network coding based multicast. (*a*) The example network. (*b*) Traditional multicast. (*c*) Network coding based multicast. (Xing and Qu, 2012)

There are two main streams of research in the field of network coding. One stream investigates efficient encoding-and-decoding algorithms to increase the data transmission rate while reducing the computational cost, and the other emphasizes the applications of network coding.

For the first stream, regarding the encoding functions to combine the received information at intermediate nodes, there have been a number of encoding-and-decoding algorithms, e.g. linear network coding (Li *et al*, 2003), algebraic network coding (Koetter and Médard, 2003), convolutional network coding (Li and Yeung, 2006) and so on. Among them, one of the most fundamental and widely used coding methods is linear network coding, where packets are linearly combined. Introduced in 2003, linear network coding always suffices to achieve the theoretical maximum throughput for multicast sessions (Li *et al*, 2003). In this thesis, linear network coding is used for network coding based data transmission (see Chapter 3).

1.3.1 Linear Network Coding

Linear network coding is similar to the example in Figure 1.3(c). Whereas the example uses Exclusive-OR operation to compress *a* and *b* into one bit, linear network coding will adopt linear combination of the incoming data which are interpreted as elements over some finite field (Fragouli *et al*, 2006). In abstract algebra, a finite field is a field that contains a finite number of elements and any operation performed in the field always results into an element within that field (Blake *et al*, 1993). Finite field is also known as Galois Field and usually denoted by $\mathbf{GF}(2^n)$ which contains 2^n elements, where n is a positive integer. Finite field has wide applications in many areas, such as coding theory, algebraic geometry and cryptography. The following briefly describes how linear network coding works, including the encoding approach performed at coding nodes and decoding approach performed at receivers. Details can be found in (Fragouli *et al*, 2006).

Assume each data packet in a communication network contains N binary bits. If we interpret every n consecutive bits of a packet as an element in the field $\mathbf{GF}(2^n)$, the packet consists of a vector of N/n elements. In linear network coding, the outgoing packets of a coding node are linear combinations of the incoming packets. Combining the packets requires two basic operations performed over the field $\mathbf{GF}(2^n)$: addition and

multiplication. Addition is the bitwise Exclusive-OR operation. For multiplication, every n consecutive bits, b_0, b_1, …, b_{n-1} can be interpreted as the polynomial $b_0 + b_1x + \ldots + b_{n-1}x^{n-1}$. Hence, multiplication is done by computing the product of two polynomials.

1.3.1.1 Encoding

Assume there are K original packets M_1, M_2,…,M_K to be delivered from the source to one or more receivers. Rather than simply delivering the original packets over the network as store-and-forward scheme, network coding allows packet combination at some intermediate nodes. Assume each packet (e.g. packet i) in the network consists of two parts, *encoding vector* $A_i = (\alpha_1^i,\ldots,\alpha_K^i)$ and *information vector* X_i, where α_k^i (k = 1,…,K) is selected in **GF**(2^n) and X_i is expressed as (1.1):

$$X_i = \sum_{k=1}^{K} \alpha_k^i M_k \,, \tag{1.1}$$

Assume there are m packets (A_1, X_1),…, (A_m, X_m) that need to be linearly coded at an intermediate node. The node first picks a set of coefficients (β_1,…,β_m) in **GF**(2^n) and then calculates the linear combination as below:

$$X_{new} = \sum_{i=1}^{m} \beta_i X_i \tag{1.2}$$

The new encoding vector A_{new} is obtained as:

$$A_{new} = (\sum_{i=1}^{m} \beta_i \alpha_1^i, \sum_{i=1}^{m} \beta_i \alpha_2^i, \ldots, \sum_{i=1}^{m} \beta_i \alpha_K^i) \tag{1.3}$$

After encoding, the new packet is forwarded to the next node along the route to one of the receivers. Note that coded packets can be recoded if necessary.

1.3.1.2 Decoding

As mentioned above, in linear network coding, each packet carries an encoding vector and an information vector. The packets received by receivers may not be original packets sent from the source. Hence, when packets arrive, receivers have to decode them

so as to obtain the original packets, i.e. $M_1, M_2, \ldots, M_K$. Assume a receiver gets n packets: $(A_1, X_1),\ldots, (A_n, X_n)$. The node needs to solve the following n linear equations:

$$\begin{cases} X_1 = \sum_{k=1}^{K} \alpha_k^1 M_k \\ X_2 = \sum_{k=1}^{K} \alpha_k^2 M_k \\ \vdots \\ X_n = \sum_{k=1}^{K} \alpha_k^n M_k \end{cases} \tag{1.4}$$

To successfully recover the original data, one needs to have: (1) $n \geq K$, i.e. the number of the received packets is no less than that of the original packets; and (2) all equations are linearly independent. The two requirements are easy to fulfill (Fragouli *et al*, 2006). By far, linear network coding has become one of the most popular coding methods in the field of network coding (Fragouli and Soljanin, 2007).

1.3.2 Advantages and Applications

The second stream of research in network coding focuses on its applications in communication networks. As a new communication paradigm, network coding is featured with a number of advantages. As mentioned above, network coding can increase the multicast throughput (as shown in Figure 1.3) (Ahlswede *et al*, 2000; Li *et al*, 2003). Besides, network coding can be used to balance network payload, reduce the energy consumption in wireless networks, increase the security level against network attackers, and improve the robustness against network failures.

Noguchi *et al* (2003) evaluated the capacity and load-balancing performance when applying network coding to multicast. They compared network coding based multicast with traditional IP multicast and showed that network coding had not only higher data transmission capacity but also a positive effect to balance the network traffic.

Wireless networks are becoming increasingly popular and indispensable to us. However, they in general suffer from low throughput, limited battery life and bandwidth, transmission failures, and so on. Network coding can optimize various kinds of wireless resources by making use of the broadcast nature of the wireless channel (Katti *et al*,

2006). In a wireless environment, it can be used to reduce transmitting energy and significantly reduce the delay (Fragouli *et al*, 2006; Katti *et al*, 2006; Chou and Wu, 2007; Wang *et al*, 2009a; Zeng *et al*, 2011; Wang *et al*, 2012).

As we know, the basic idea of network coding is to allow routers to recombine data packets they received (Ahlswede *et al*, 2000; Li *et al*, 2003). The combination (usually linear combination) of packets offers a natural way to avoid wiretapping attacks as the attackers cannot decode the original information sent as long as they cannot wiretap all multi-paths that transmit information for decoding (Cai and Yeung, 2002; Fragouli *et al*, 2006; Guo *et al*, 2010; Zhang *et al*, 2010d; Cai and Yeung, 2011).

Link failure usually happens in communication networks and may cause serious damage to them, especially in optical networks. How to design effective data protection scheme so that the transmission is affected as lightly as possible by link failure is of crucially importance. Network coding has been used for this purpose. Kamal carried out a series of studies on how to make data protection against link failure in optical networks by using network coding (Kamal, 2006, 2007a, 2007b, 2008, 2010; Kamal *et al*, 2011).

As shown above, network coding brings a lot of benefits to the current communication networks and is also foreseen as a core technology for future networks (Fragouli and Soljanin, 2007). Network coding has many network applications. The following lists some of them:

- Multimedia streaming and file distribution in peer-to-peer and wireless networks (Wang and Li, 2007a; Wang and Li, 2007b; Dimakis *et al*, 2007; Ma *et al*, 2007; Thomos and Frossard, 2009; Chu and Jiang, 2010; Li and Niu, 2011; Zhang *et al*, 2011; Kao *et al*, 2012)
- Reducing the consumption of network resources (such as energy and storage) while increasing the throughput in wireless and sensor networks (Katti *et al*, 2006; Bhadra and Shakkottai, 2006; Sengupta *et al*, 2007; Hou *et al*, 2008; Wang *et al*, 2010; Kasireddy *et al*, 2011; Shwe and Adachi, 2011)
- Secure data transmission against network attacks, e.g. eavesdropping and wiretapping (Cai and Yeung, 2002; Fragouli *et al*, 2006; Guo *et al*, 2010; Zhang *et al*, 2010d; Cai and Yeung, 2011)

- Data protection against network failures in optical networks (Kamal, 2006, 2007a, 2007b, 2008, 2010; Kamal *et al*, 2011)

1.4 Network Coding Based Multicast

Nowadays, a significant proportion of the Internet bandwidth has been consumed by the explosively growing multimedia applications and this trend is continuing. Among these applications, many require multicast to support data delivery, such as IPTV and video conferencing. Hence, the demand for a powerful and efficient multicast technology is becoming ever urgent.

As a new communication paradigm, network coding potentially offers many benefits to the communication network, such as, maximizing the multicast throughput which in general may not be achieved by traditional multicast (see Section 1.3). Therefore, using network coding in multicast could be a promising technical solution to support real-time and high-speed multicast data transmission (Wang and Li, 2007a, 2007b).

1.4.1 Two Tasks for Establishing A Multicast Session

Establishing a multicast session by using network coding could be divided into two almost independent tasks (Fragouli and Soljanin, 2006). The *first task* is to find a routing subgraph that meets the data rate requirement for each receiver, while the *second task* is to determine an appropriate encoding-and-decoding configuration over the subgraph so that any receiver is able to recover the original data sent from the source node. The two tasks are based on the graph theory and the classical coding theory, respectively (Fragouli and Soljanin, 2006).

This thesis emphasizes the first task, i.e. how to route multicast data from the source to all receivers at an expected data rate (see Chapters 3, 4, 5 and 6). For traditional multicast (with store-and-forward scheme), a multicast tree is constructed, where there is a single path connecting the source and each receiver (see Figure 1.1(b) as an example). The same data sent from the source are delivered through the multicast tree and received by each receiver. However, for network coding based multicast, we need to find a subgraph that consists of multiple link-disjoint paths (i.e. paths without common link)

between the source and each receiver. In this subgraph, different data flows may pass through different areas within the subgraph (see Figure 1.3(c) for instance). Such subgraph is referred to as *network coding based multicast* (NCM) subgraph (Xing and Qu, 2011a, 2011b, 2012) (see Chapter 3 for details).

1.4.2 Interesting Topics in the Literature

As mentioned in Section 1.3, network coding is an emerging technique. How to well integrate it with the existing network architecture to efficiently support multicast applications has raised a number of interesting issues. The following reviews some of these issues emerged in the literature, emphasizing the first task in Section 1.4.1:

- **Throughput optimization problems**. Given a network and a multicast request[1], how to design efficient algorithms to find a routing subgraph that achieves the maximum end-to-end throughput is of theoretical interest. Maximizing the throughput in undirected networks has been extensively investigated. Li *et al* (2005) found that for NCM, the difficulty of finding an optimal end-to-end throughput is reduced to polynomial time (while for traditional multicast, it is NP-complete). In 2006, Li *et al* developed an efficient algorithm working in a distributed fashion to compute the optimal data transmission scheme. Later, Li *et al* (2009) demonstrated that the achievable multicast throughput with network coding can at most double that without network coding.
- **Link-cost minimization problems**. When finding a routing subgraph for multicast with network coding, it is of practical interest to minimize the link usage while achieving an expected throughput (Lun *et al*, 2004). The cost comes from flows occupying links. Each link (i,k) within the network is associated with non-negative values a_{ik} and c_{ik} which are the cost per unit flow and how much capacity is used by flows, respectively. The total link cost is thus denoted as $\Sigma a_{ik}c_{ik}$, which is the objective to minimize. The minimum-cost multicast routing problems are formulated as a linear optimization problem (Lun *et al*, 2004).

[1] In communications networks, a multicast request is a request generated at a certain source node to build a multicast tree spanning this source and all its receivers (Benslimane, 2007). The same data is delivered from the source to each receiver through the constructed multicast tree.

Both static and dynamic multicast scenarios are considered, where in dynamic multicast, the membership of the multicast changes over time, i.e. nodes can join or leave the current multicast session whenever they want (Lun *et al*, 2004, 2005, 2006; Bhadra *et al*, 2006; Zhao *et al*, 2009).

- **Network coding resource minimization problems**. Applying network coding in multicast can guarantee optimal throughput, however, at the expenses of extra computational overhead at coding nodes (See Section 1.3). Hence, it is necessary to minimize the computational resource consumed (also known as *coding cost*) while achieving an acceptable data rate from the source to each destination. Ratnakar *et al* (2005) studied the problem of minimizing the number of packets that undergo network coding and adopted a linear programming formulation. Langberg (2006) and Fragouli and Soljanin (2006) independently investigated the minimization of the number of coding nodes involved in a multicast session and proposed two greedy algorithms. Later on, a number of evolutionary algorithms (EAs) have been developed to minimize the amount of coding operations while keeping a high throughput (Kim *et al*, 2006, 2007a, 2007b; Xing *et al*, 2010, 2011a, 2011b, 2012; Ji and Xing, 2011).
- **Coding and link costs minimization problems.** In a NCM data transmission, there are two types of costs, namely the link cost and the coding cost, where the coding cost represents the computational overhead incurred at coding nodes. One may be interested in simultaneously minimizing the two costs. If coding and link costs can be optimized at the same time, a minimum total cost can be obtained. If the two costs conflict with each other, a trade-off can be identified, e.g. occupying some extra links would eliminate the requirement of coding operations at some coding nodes. If so, the trade-off information can be used by network designers to properly deploy intermediate nodes which can perform coding operations. Kim *et al* (2007c) investigated the trade-off and adopted nondominated sorting genetic algorithm II (NSGA-II) to address it. In 2009, a two-stage method was proposed to simultaneously minimize coding and link costs for optical networks (Kim *et al*, 2009a).

- **Network coding based multicast with Quality-of-Service (QoS) issues**. QoS is one of the common terms in communication networks. It means that networks should be able to support data transmission services with certain performance requirements, such as constraints of delay, delay jitter, bandwidth and packet loss ratio, etc (Aurrecoechea *et al*, 1998). Real-time multimedia applications are often with stringent QoS requirements (Aurrecoechea *et al*, 1998; Chalmers and Sloman, 1999; Striegel and Manimaran, 2002). Hence, considering QoS issues is quite important, especially when network coding is used to support real-time multimedia multicast applications. An increasing amount of efforts has been dedicated to this research area (Pang *et al*, 2005; Walsh and Weber, 2008; Pu *et al*, 2009; Yeow *et al*, 2009a, 2009b; Zhang *et al*, 2009; Amir *et al*, 2010; Xuan and Lea, 2011; Anshelevich *et al*, 2011).

1.5 Motivations and Aims

As reviewed in Subsection 1.4.2, when using network coding based multicast (NCM), one of the most crucial and realistic issues comes from the considerable amount of computational resource consumed during the process of packets combinations at coding nodes, i.e. coding operations (see Section 1.3). This is because coding operations require intermediate nodes to perform mathematical operations and thus dramatically increases the computational complexity at coding nodes (Ahlswede *et al*, 2000; Li *et al*, 2003; Langberg *et al*, 2006). On the other hand, nodes that perform coding operations also need to have additional buffering as encoding cannot proceed until all data packets that need to be coded arrive at the coding node. Packets that arrive early at the coding node have to be stored, waiting for the others. Take the data transmission in Figure 1.3(c) as an example, coding node w is responsible for combining bits a and b into bit $a \oplus b$. However, XOR operation cannot start until the arrival of the two bits. Thus, additional buffer is needed to store the first bit b (or a) arrived.

Given a multicast request, the more the coding operations performed, the higher the computational and buffering costs incurred. The main concern in the thesis is how to route the data from the source to all receivers at the expected data rate while minimizing

the usage of computational and buffering resources consumed in the NCM data transmission. Such concern raises the network coding resource minimization problem (see Chapter 3).

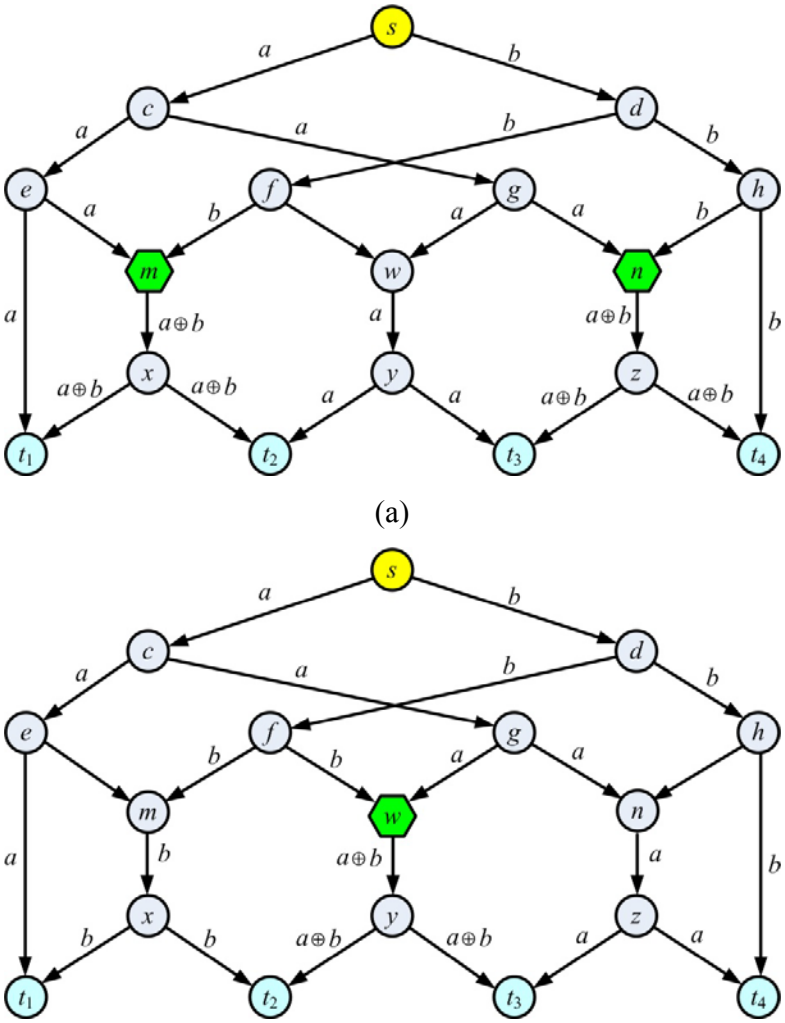
1.5.1 The Network Coding Resource Minimization Problem

In the literature, most of the research assumes that coding operations take place at all coding-possible nodes. However, it is often possible that coding operations only need to occur at a subset of those nodes to achieve an expected throughput (Kim *et al*, 2006, 2007a, 2007b). Figure 1.4 shows an example of two NCM schemes with different amount of coding operations involved (Xing and Qu, 2012). Source s expects to multicast two bits (a and b) to four receivers, t_1, t_2, t_3 and t_4 within the network, where each link has a capacity of one bit per time unit.

In scheme-A, coding operations occur at nodes m and n; while in scheme-B coding operation is only performed at node w. Both schemes achieve the multicast throughput of 2 bits per time unit, however, consuming different amount of network resources. Due to the mathematical operations involved (e.g. $\oplus$ in Figure 1.4), NCM not only incurs extra cost such as computational overhead and transmission delay, but also consumes public buffering resource. It is thus desired that the amount of coding operations incurred is minimized while retaining the benefits that NCM brings to us.

Given a network and a multicast request, the objective of the problem is to find a feasible NCM subgraph with coding operations minimized and all restrictions satisfied (see Chapter 3 for details). This is a minimization problem and has been proved as NP-hard (Kim *et al*, 2006). Different from a multicast tree where the source and each receiver are connected by a single path, in a NCM subgraph, there are multiple link-disjoint paths between the source and each receiver.

Minimizing network coding resource can save a huge amount of computational and buffering resources for the network. However, this problem is proved as NP-hard (Kim *et al*, 2006). Only a few methods such as greedy approaches and evolutionary algorithms (EAs) have been reported for solving the problem (Langberg *et al*, 2006; Fragouli and Soljanin, 2006; Kim *et al*, 2006, 2007a, 2007b).



(a)

(b)

Figure 1. 4 Two NCM schemes. (*a*) Scheme-A (*b*) Scheme-B (Xing and Qu, 2012)

1.5.2 The Delay Constrained Network Coding Resource Minimization

The network coding resource minimization is a fundamental problem which only takes into account the computational and buffering issues. Minimizing network coding resources may not be sufficient to well support real-time multimedia multicast applications. This is because these applications require the network to provide stringent QoS guarantees during the data transmission (see Subsection 1.4.2) (Aurrecoechea *et al*, 1998; Chalmers and Sloman, 1999; Striegel and Manimaran, 2002). For example, in video conferencing, the end-to-end delay should be below 300 miliseconds (*ms*) to allow

the participants to interact naturally; otherwise, the interactive session cannot be guaranteed (Lakshman *et al*, 1999; Borella, 2000; Pau *et al*, 2005). So, when applying NCM to support real-time multimedia applications, one needs to take into consideration not only the computational costs but also the QoS requirements. The end-to-end delay (i.e. transmission delay) is regarded as one of the most important QoS parameters for NCM (Walsh and Weber, 2008; Pu *et al*, 2009; Yeow *et al*, 2009a; Zhang *et al*, 2009; Amir *et al*, 2010). Hence, it should be bounded to gain an acceptable user experience.

In the thesis, we extend the network coding resource minimization (NCRM) problem by adding the end-to-end delay constraint between the source and its receivers (see Chapter 5 for details). This thesis, for the first time, formulates the above problem. Compared with the original NCRM problem, the extended problem is more realistic. By solving the new problem, one can obtain a NCM subgraph with coding operations minimized and end-to-end delay constraint satisfied.

1.5.3 The Cost-Delay Bi-objective Optimization Problem

For NCM routing problems, the cost of a multicast session is usually considered as a minimization objective (see Subsection 1.4.2). In the literature, two types of costs, i.e. coding cost of performing coding operations and link cost of employing transmission links, have been considered. There are three research streams studying the two costs: (1) minimizing the link cost (Lun *et al*, 2004, 2005, 2006; Bhadra *et al*, 2006; Zhao *et al*, 2009), (2) minimizing the coding cost (Ratnakar *et al*, 2005; Langberg *et al*, 2006; Fragouli and Soljanin, 2006; Kim *et al*, 2006, 2007a, 2007b), and (3) simultaneously minimizing both link and coding costs (Kim *et al*, 2007c; 2009).

When setting up multicast sessions, service providers expect to minimize the amount of network resources consumed by the multicast (Benslimane, 2007). In NCM routing, coding and link costs will both incur. The total cost (the sum of the coding and link costs) involved is more accurate to estimate the consumed resources, compared with coding cost or link cost (Kim *et al*, 2007c). Hence, it is worth studying the minimization of the total cost of a multicast. To the best of our knowledge, no research has been conducted concerning the total cost. On the other hand, network users feel uncomfortable if the transmission delay of an application (e.g. video conferencing) that they request is too

large, i.e. the users expect the transmission delays to be minimized. Reducing cost results into a cheaper multicast solution for network service providers, while decreasing delay improves the service quality for users. The two objectives may require a trade-off between the interests of service providers and network users. If so, the trade-off results could be investigated by the decision maker and a compromised NCM routing scheme can be obtained.

Therefore, in this thesis, we model the cost-delay bi-objective optimization (CDBO) problem for NCM, with one objective to minimize the total cost and the other to minimize the maximum transmission delay of the multicast (see Chapter 6 for details). This optimization problem is a multiobjective optimization problem, where two or more (often conflicting) objective functions are to be optimized simultaneously and the resulting trade-offs between them are collected for decision making (see Section 2.1.2).

1.5.4 Thesis Aims

The last decade has witnessed a new wave of evolution in communication networks. Network coding represents one of the breakthrough principles that are guiding the evolution. It has attracted increasing interests from the fields of information theory and communication networks. Due to its advantages such as increasing the throughput, network coding is an ideal choice for supporting multicast applications. However, to well adapt network coding for the existing network infrastructures and communication protocols, a number of issues need to be figured out. Among these issues, it is of vital importance to investigate three combinatorial optimization problems: (1) the NCRM problem, (2) the NCRM problem with delay constraint, and (3) the CDBO problem, where first problem has been studied in the literature while the others were for the first time formulated and investigated in the thesis (see Chapters 3, 4, 5 and 6). To support and facilitate the application of NCM, efforts need to be well spent in the fields of computational intelligence and operational research to provide good solutions to the above three problems.

Metaheuristics are a class of computational search approaches, optimizing a given problem by iteratively improving a candidate solution via a given measure of quality. Although metaheuristics do not guarantee to find an optimal solution, they are usually

the choice to provide good solutions for various NP-hard problems within a limited time (Guo *et al*, 2005). Evolutionary algorithms (EAs) are a class of metaheuristics that mimic the evolutionary procedure in nature, namely the survival of the fittest. Due to their robustness and effectiveness, EAs have been applied to a wide range of real world applications, including routing problems in communication networks (Puchinger and Raidl, 2005; Ribeiro *et al*, 2007). Therefore, this thesis is focused on using EAs to tackle the three optimization problems above.

To make NCM practically applicable, the aim of this research is to study the original NCRM problem (see Section 1.5.1), to model the new problems (see Subsections 1.5.2 and 1.5.3) and to develop effective EAs to solve the three problems concerned. In order to achieve this goal, the identified objectives are listed as follows:

- Investigate the NCRM problem and proposed efficient EAs to address it (see Chapter 4);
- Formulate the delay-constrained NCRM problem and study this problem by Eas (see Chapter 5);
- Formulate the CDBO problem and design appropriate multi-objective EAs to investigate the trade-off solutions for the problem (see Chapter 6);
- Evaluate the performance of the proposed algorithms by comparing it with that of the state-of-the-art algorithms on a number of NCM instances (see Chapters 4, 5, 6).

1.6 Thesis Outline

The reminder of the thesis is organized as follows. In Chapter 2, we review some of the most well-known EAs in the field of computational intelligence. Their principles and applications are summarized. Chapter 3 provides the problem description of the NCRM problem and reviews the related work. In Chapter 4, we adapt three EAs for the NCRM problem and study their performance. We extend the formulation of the NCRM problem by introducing delay constraint in Chapter 5. Several EAs are adapted for the new problem and their performance is evaluated. In Chapter 6, we formulate a multiobjective optimization version of the delay constrained NCRM problem and adapt a widely used

multiobjective EA for it. Finally, we summarize the thesis and discuss the future work in Chapter 7.

1.7 Contributions of the Thesis

The contributions of the thesis are listed below.

- Three EAs are presented to solve the NCRM problem (see Chapter 4). They are population based incremental learning (PBIL), compact genetic algorithm (cGA) and path-oriented encoding EA (pEA). In PBIL, an entropy-based restart scheme is for the first time proposed which is able to improve the global exploration of PBIL. In cGA, three enhancements are for the first time developed based on the features of the encoding and the structure of cGA, including the use of all-one solution, a probability vector (PV) restart scheme, and a local search operator based on domain knowledge. All-one solution can effectively guide the search towards feasible regions in the search space. The PV restart scheme can take advantage of the previously promising PV to escape local optima. The local search operator is based on incoming link removal and can improve the local exploitation ability of cGA. A new chromosome representation is for the first time proposed and referred to as path-oriented encoding. An EA based on this encoding (pEA) is presented, where initialization, crossover, mutation and a problem specific local search operator are specially developed. The mutation is in a greedy-based manner which helps to create fitter solutions. The local search is based on path reconstruction and shows strong local exploitation ability.
- User experience is for the first time introduced into the original NCRM problem, which leads to the formulation of delay constrained NCRM problem. PBIL is adapted for the new problem by two schemes. One is a new PV update scheme specially designed based on the features of the new problem. This new update scheme makes use of a set of historically best solutions. The other is a PV mutation from the literature. Combining the two schemes together can improve the global exploration and avoid local optima at the same time.

- An investigation into the conflicting interests of service providers and network users is presented in the context of NCM. A multi-objective optimization problem (MOP) is formulated based on the delay constrained NCRM problem and called the cost-delay bi-objective optimization (CDBO) problem. Two schemes are for the first time developed to adapt elitist nondominated sorting genetic algorithm (NSGA-II) for the problem, including an initialization scheme and an individual delegate scheme (IDS). The initialization can generate a set of promising and diversified individuals, which helps to initiate a diversified search at the beginning of the evolution. The IDS considers the diversity in both decision and objective spaces and helps to diversify the search during the evolution.

Chapter 2 Evolutionary Algorithms: An Overview

Evolutionary computation (EC) simulates the process of natural evolution to solve optimization problems. Because of its ability of tackling complex and real-world optimization problems, EC has gained significant amount of research attention over the last five decades and become one of the most attractive areas recently.

In the field of EC, one of the most important branches is known as evolutionary algorithms (EAs). In this chapter, we first introduce EAs and the related optimization problems and then review some of the well-known EAs in the literature. It is noted that the scope of the thesis is limited to population-based EAs. So, population-based EAs are of the main concern in this chapter.

2.1 Introduction

EAs are a family of stochastic optimization techniques that are inspired by biological evolution, namely selection, recombination and mutation. They maintain a population of solutions and optimize a given problem by iteratively improving the population via some quality measurements (Mitchell and Taylor, 1999). EAs are capable of searching a large solution space without any assumption about the problem being concerned. Although they do not guarantee to find an optimal solution, EAs are usually an ideal choice to provide good solutions for NP-hard problems within a short computational time.

For the past three decades, EC has been extensively studied and successfully applied to many areas, including telecommunications (Puchinger and Raidl, 2005; Ribeiro *et al*, 2007; Zhang *et al*, 2007; Xu, 2011), image processing (Mitchell and Taylor, 1999), industrial manufacturing (Oduguwa *et al*, 2005), financial market prediction (Michell and Taylor), scheduling and timetabling (Burke and Newall, 1999; Mitchell and Taylor, 1999), and bioinformatics (Pal *et al*, 2006).

As a class of practical and robust search algorithms, EAs can be used to address a variety of optimization problems. Among them, two types of problems are of practical importance in real-world applications, i.e. combinatorial optimization problems (COPs)

(Blum and Roli, 2003; Puchinger and Raidl, 2005) and multiobjective optimization problems (MOPs) (Mitchell and Taylor, 1999; Zitzler *et al*, 2004; Deb, 2011). Actually, among the three optimization problems concerned in the thesis, the network coding resource minimization (NCRM) problem and delay constrained NCRM problem are COPs while the cost-delay bi-objective optimization (CDBO) problem is MOP (see Section 1.5 and Chapters 3, 5 and 6 for details).

In the following subsections, we briefly review COPs and MOPs.

2.1.1 Combinatorial Optimization Problems

An optimization problem is the problem of finding the best configuration of a set of variables (i.e. best solution) to optimize some goal(s) (Blum and Roli, 2003). Optimization problems can be classified into two categories: those where solutions are encoded with real-valued variables and those where solutions are encoded with discrete variables. COPs belong to the second category. Many real world problems are COPs, such as scheduling and timetabling problems (Qu *et al*, 2009) and multicast routing problems (Xu, 2011).

Blum and Roli (2003) give the definition of COP $P = (S, f)$ as follows:

- a set of variables $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, where n is the number of variables;
- variable domains $D_1, D_2, \ldots D_n$;
- constraints among variables;
- an objective function f, where $f : D_1 \times \ldots \times D_n \rightarrow \mathbf{R}^+$;

All feasible configurations of the variables are shown in S as:

$$S = \{s = \{(x_1, v_1), \ldots, (x_n, v_n)\} \mid v_i \in D_i \text{ and } s \text{ satisfies all the constraints}\},$$

where, S and s are called search (or solution) space and candidate solution, respectively. Suppose the COP is a minimization problem. To tackle this COP, one needs to find a

solution $s^* \in S$ with the minimum objective function value so that $f(s^*) \leq f(s)$, $\forall\ s \in S$. Solution s^* is called optimal solution of (S, f).

2.1.2 Multiobjective Optimization Problems

Real world problems usually involve multiple measures of performance, i.e. so called objectives, which should be optimized simultaneously (Fonseca and Fleming, 1995). In certain cases, objectives may be optimized separately from each other and the obtained solution has the optimized value in each objective. However, this is very unlikely to happen in real world optimization scenarios. Instead, the objectives are often conflicting, i.e. improving one objective may harm the others, resulting into a set of trade-off solutions (Fonseca and Fleming, 1995; Zitzler *et al*, 2004; Deb, 2011). The problem of finding such a trade-off is a MOP. MOPs can be found in fields such as economics and finance (Mitchell and Taylor, 1999), engineering optimization (Marler and Arora, 2004), and Bioinformatics (Pal *et al*, 2006).

A general MOP can be formulated as follows (Marler and Arora, 2004; Zitzler *et al*, 2004; Deb, 2011; Zhou *et al*, 2011):

$$\begin{aligned} \text{Minimize / Maximize :} \quad & F(\mathbf{X}) = (f_1(\mathbf{X}),\ldots,f_{\mathrm{M}}(\mathbf{X}))^T \\ \text{Subject to :} \quad & g_j(\mathbf{X}) \geq 0, \qquad j = 1, \ldots, \mathrm{J}; \\ & h_k(\mathbf{X}) = 0, \qquad k = 1, \ldots, \mathrm{K}; \end{aligned} \tag{2.1}$$

where, solution $\mathbf{X} \in \mathbf{R}^n$ consists of n decision variables, i.e. $\mathbf{X} = (x_1, x_2,\ldots, x_n)^T$; M is the number of objectives, J is the number of inequality constraints, and K is the number of equality constraints. Solutions that satisfy all constraints and variable bounds constitute a feasible decision variable space $\boldsymbol{S}_{dec} \subset \mathbf{R}^n$. Besides, the M objective functions in MOP constitute an objective space $\boldsymbol{S}_{obj}$, where $\boldsymbol{S}_{obj} \subset \mathbf{R}^{\mathrm{M}}$. Any solution $\mathbf{X} \in \boldsymbol{S}_{dec}$ corresponds to a point $\mathbf{Z} \in \mathbf{R}^{\mathrm{M}}$, denoted by $F(\mathbf{X}) = \mathbf{Z} = (z_1,\ldots, z_{\mathrm{M}})^T$.

As mentioned above, objectives in MOPs often conflict with each other. Generally, there is no such a single solution that can optimize all objectives simultaneously. Rather, the trade-off solutions (known as nondominated solutions) are of vital importance to a

decision maker. The following introduces the definition of domination between two solutions ($\mathbf{X}_1$ and $\mathbf{X}_2$) and that of Pareto optimality.

Definition 2.1 Solution $\mathbf{X}_1$ is said to dominate solution $\mathbf{X}_2$ (or $\mathbf{X}_2$ is dominated by $\mathbf{X}_1$), if both of the following conditions hold:

- $\mathbf{X}_1$ is no worse than $\mathbf{X}_2$ in all objectives.
- $\mathbf{X}_1$ is strictly better than $\mathbf{X}_2$ in at least one objective.

Definition 2.2 A feasible solution $\mathbf{X}^*$ of problem (2.1) is called a Pareto optimal solution, if there is no $\mathbf{X} \in \boldsymbol{S}_{dec}$ such that $F(\mathbf{X})$ dominates $F(\mathbf{X}^*)$. All the Pareto optimal solutions constitute the Pareto set (PS) and the image of the PS in the objective space is called the Pareto optimal front (PF), where PF = $\{F(\mathbf{X}) \mid \mathbf{X} \in \text{PS}\}$.

2.2 EAs for COPs

This section reviews the principle and applications of a number of widely used EAs for addressing COPs, namely genetic algorithms (GAs), memetic algorithms (MAs), and estimation of distribution algorithms (EDAs). Meanwhile, these EAs are also employed to solve the three optimization problems concerned in the thesis.

2.2.1 Genetic Algorithms

GAs is a set of well investigated EAs in the literature. They operate on a population of potential solutions encoded to a special optimization problem, according to the theory of survival of the fittest (Holland, 1975; Mitchell, 1996; Mitchell and Taylor, 1999). Given a population of individuals (also known as chromosomes or solutions), the environmental pressure causes natural selection and this leads to a rise in the fitness of the population. Given a fitness function, each individual is measured and attached with a fitness based on which selection operation is implemented. In selection, the key idea is that better individuals are more likely to be selected to form the next generation. After

selection, crossover and mutation operations reproduce a partially or completely new population. Crossover, the regarded major evolutionary operator in GA, operates on two or more individuals to create at least one offspring which inherits the characteristics of its parents. Mutation is used to slightly change each selected individual, primarily aiming at avoiding pre-maturity. The evolution, i.e. selection, crossover and mutation, repeats in each generation and terminates when some stopping condition is met, e.g. GA stops after a predefined number of generations or the population converges to the same / similar individuals.

GAs have been applied to solve many real world COPs, such as routing and network design problems, travelling salesman problems, and supply chain management problems. More applications of GAs can be found in Mitchell and Taylor (1999), Reeves (2003), and Ribeiro *et al* (2007). Besides, GAs is also used to address the NCRM and CDBO problems (see Chapter 4 and 6).

2.2.1.1 Drawbacks

Making no assumptions to the problem being optimized, GAs are well adaptive and robust. However, GAs usually suffer from slow convergence and getting stuck at local optima (Mitchell, 1996). This is because the performance of GAs depends on the mechanism of balancing two conflicting objectives: global exploration and local exploitation (Mitchell, 1996; Ei-Mihoub *et al*, 2006). The former is the ability of exploring the entire search space and the latter the ability of exploiting the neighbourhood regions of the best solutions found so far. Global exploration requires a diversified population, undoubtedly restricting the local exploitation ability. On the other hand, intensifying the search around a small area in search space has to sacrifice the level of diversity. In practice, GAs cannot always well balance the exploration and exploitation, thus leading to slow convergence or being stuck at local optima. There are a number of factors that may cause the above problem, such as population size, selection pressure, crossover and mutation rates, etc (Mitchell, 1996). For example, if the population size is too small, the diversity of the population may be lost rapidly, leading to the prematurity and GAs getting stuck at local optima; if the population size is too large, the convergence becomes quite slow and a great amount of computational time may be consumed.

2.2.1.2 Improvements

We hereafter refer to GAs with basic components as *standard* GAs. To overcome the problem arisen such as slow convergence and prematurity and enhance the optimization performance, a number of techniques have been introduced into standard GAs. We review some of the well-known techniques, including elitism, adaptive parameters control, multi-population, and hybridization. When designing GA for solving a COP, one may use some/all of these techniques to improve the performance with respect to the problem feature.

- **Elitism**. In GAs, the best individuals in the population are important to guide the search during the evolution. However, these individuals could be lost if they are not chosen for producing the offspring or if they are destroyed in recombination and mutation operations (Mitchell, 1996). De Jong (1975) first introduced the term 'elitism'. The essence of elitism is that GA retains the best individual at each generation. The following discusses several methods to carry out elitism. One is to replace the worst individual of the current population with the best one of the parent population. Another is that the best individual of the current population is replaced with the best one of the parent population if the former is worse than the latter. Besides, elitism can be integrated into selection schemes. In standard GAs, all offspring individuals obtained from crossover and mutation form the next generation regardless of whether they are really 'good'. In elitist selection schemes, the offspring has to compete with the parents to be included in the new generation, showing the principle of the survival of the fittest (Thierens, 1998). GAs with elitism have been adopted for solving a number of COPs, including robust and nonparametric multivariate analysis (Chakraborty and Chaudhuri, 2003), printed circuit board inspection (Mashohor *et al*, 2005), assignment problem with imprecise goal (Majumdar and Bhunia, 2007), resource allocation in OFDM systems (Tang *et al*, 2008), dynamic optimization problem (Yang, 2008; Cheng and Yang, 2010a), routing scheme for wireless sensor networks (Singh and Sharma, 2012). It has been found that the

performance of GAs can be dramatically improved by elitism. However, elitism schemes should be carefully designed to avoid premature convergence.

- **Adaptive Parameters Control**. In GAs, controlling parameters have a great impact on the optimization performance. For example, the crossover and mutation probabilities (p_c and p_m), to a certain extent, determine the exploration and exploitation characteristics of GAs (Srinivas and Patnaik, 1994; Mitchell, 1996). Standard GAs use fixed parameters during the evolution. However, it is difficult to find an optimal parameter setting for an unknown problem (Grefenstette, 1986; Schaffer *et al*, 1989; Eiben *et al*, 1999). This motivates the idea of adaptively adjusting the controlling parameters of GA at each generation according to the population information, which maintains population diversity and sustains the convergence capacity at the same time. Srinivas and Patnaik (1994) investigated the dynamic adjustment of crossover and mutation probabilities. In their scheme, solutions with higher fitness values are assigned smaller p_c and p_m, which helps to preserve the good genes in the population; and those with lower fitness values are assigned larger p_c and p_m, which avoids GAs getting stuck at local optima. Marsili Libelli and Alba (2000) proposed an adaptive mutation probability scheme: (1) for high fitness solutions, the least significant bits are more likely to mutate, and (2) for low fitness solutions, they have an increased p_m along with the evolution. An adaptive GA was proposed for solving the financial knapsack problem (Szeto and Lo, 2004). Instead of tuning crossover and mutation probabilities, this GA adaptively controls the proportion of preserved (survived) individuals at each generation, where preserved individuals are individuals with fitness values larger than a dynamic threshold. Vellew (2008) investigated a dynamic population sizing scheme for GA. The basic idea is to increase the population size when there are high fitness individuals and abundant resources and to decrease the size otherwise. More adaptive schemes and applications of GAs can be found in Lobo and Lima (2005), Huang *et al* (2005), Law and Szeto (2007), Liu and Liu (2009) and Zhang *et al* (2010c).

- **Multi-population (Island models).** For GAs, a diversified population is necessary to avoid premature convergence and getting stuck at local optima. However, it is difficult for GAs to maintain the diversity by using a single population (Smith *et al*, 1993). This leads to the premature convergence, which may significantly deteriorate the performance of GAs. To alleviate the problems that single population GAs encounter, multi-population GAs have been studied (also known as island GAs) (Whitley *et al*, 1998). Multi-population GAs maintain a number of subpopulations which evolve independently and exchange their individuals periodically. The process of exchanging individuals among subpopulations is referred to as migration. There are mainly two parameters responsible for controlling the migration, i.e. migration interval and migration size. The former parameter is the number of generations between a migration and the latter the number (or proportion) of individuals for migration (Whitley *et al*, 1998). Compared to single population structure, multi-population is able to maintain the diversification and explore different areas of the search space at the same time. Meanwhile, subpopulations cooperate together to efficiently guide the search by sharing information via migration. Hence, multi-population GAs have often been reported to gain better optimization results than single population GAs. The following lists some of the applications of multi-population GAs: graph colouring (Kokosiński *et al*, 2005), feature selection (Zhu *et al*, 2006), the 0/1 knapsack problem (Hong *et al*, 2007), shortest path routing problems (Cheng and Yang, 2010b), and the job shop scheduling problem (Huang *et al*, 2010).
- **Hybridization**. As reviewed above, standard GAs are not good at handling the exploration and the exploitation at the same time, thus often leading to a weakened optimization performance. This can be alleviated by combining GAs and some local search techniques (maybe with domain-specific knowledge incorporated). These hybridized approaches are called hybrid GAs (Whitley, 1995; Preux and Talbi, 1999; De Jong, 2005) and also know as a branch of memetic algorithms (see Subsection 2.2.2) (Moscato *et al*, 2004). On the one hand, a significant strength that GAs own is the excellent global exploration

ability due to the population-based structure. Nevertheless, it is difficult for GAs to efficiently intensify the search within a small region of the search space where global optimum may reside (Preux and Talbi, 1999; De Jong, 2005). On the other hand, local search methods (such as, simulated annealing and tabu search) are effective to exploit the neighboring areas of a given solution (Reeves, 1995; Preux and Talbi, 1999). Integrating GAs with local search methods can make full use of the advantages of the both, i.e. GAs are responsible for maintaining the diversity and guiding the search to the most likely regions where optima are located; while local search methods carry out powerful local exploitation within these regions. Hence, both exploration and exploitation can be warranted so that an improved optimization performance is achieved. Thanks to their adaptability and robustness, hybrid GAs have been applied to many COPs, including timetabling (Burke *et al*, 1995), job shop scheduling (Preux and Talbi, 1999; Cai *et al*, 2011), feature selection (Oh *et al*, 2004), generalized assignment problem (Feltl and Raidl, 2004), travelling salesman problem (Whitley *et al*, 2010), and image processing (Abdullah *et al*, 2012). A detailed review of hybrid GAs can be found in Ei-Miboub *et al* (2006).

2.2.2 Memetic Algorithms

As introduced in late 1980s, memetic algorithms (MAs) denote a family of EAs that hybridize a broad class of metaheuristics, i.e. EAs with local search techniques embedded (Moscato, 1989; Moscato *et al*, 2004; Moscato and Cotta, 2007; Krasnogor, 2009). The essence of MAs is that EAs perform exploration and efficiently guide local search techniques to exploit promising areas of the search space, making use of available domain knowledge about the given problem. The idea of MAs partially inspires the work reported in Chapter 4.

MAs manipulate a population of individuals and iteratively improve the quality of the population by using variation operators such as combination and local search. Hao (2011) introduced a general template of MA, as shown in Figure 2.1. The following explains some of the important steps in the template.

- **Population initialization**. In standard EAs, the initial population is usually generated by randomly sampling the search space. Although it is easy to implement, the samples may not be of good quality, which could lead to poor optimization results. Hence, to produce high-quality and diversified population, MAs typically use more advanced initialization techniques, such as constructive heuristics or local improving method (Surry and Radcliffe, 1996; Moscato and Cotta, 2007).

Input: $|P|$ (size of the population P)

Output: s^* (the best solution found)

1. P = PopInitialization($|P|$); // generate an initial population
2. PopEvaluation(P); // fitness evaluation for each individual
3. s^* = BestInd(P); $f^* = f(s^*)$; // record the best individual and its fitness value
4. **while** *termination condition is not met* **do**
5. $(p_1 \ldots p_k)$ = ParentSelection(P); // $k \geq 2$, at least two individuals are selected
6. s = Recombination($p_1 \ldots p_k$); // one or more offspring is generated
7. s_{new} = OffspringImprovement(s); // improve s by local search
8. P = PopUpdate(s_{new}, P); // update the population
9. (s^*, f^*) = BestIndUpdate(s^*, f^*, P); // update the best individual

Figure 2. 1 A general template of memetic algorithm (Hao, 2011)

- **Parent selection**. This operator selects individuals from the population that survive for the reproduction of the next generation. In essence, high fitness individuals are more likely to be chosen, which is in consistence with the survival of the fittest. In GAs, roulette-wheel and tournament selections are two well-known selection schemes (Mitchell, 1996). Moreover, diversity should be considered in selection. This is important for balancing exploration and exploitation (Moscato *et al*, 2004; Hao, 2011).
- **Recombination**. Recombination is the process of blending two or more existing individuals to produce promising offspring individuals. For example, crossover in GA enables the cooperation among different individuals to discover new areas

in the search space. In MAs, the recombination is more general and aims to capture the semantics of the problem concerned to make sure good properties are passed from parents to offspring (Hao, 2011). In addition, MAs also consider diversifying offspring during the process of recombination. For example, Merz and Freisleben (2001) used the distance preserving crossover to generate an offspring with the same distance to both parents. Moreover, problem-specific recombination operators are welcome to effectively drive the search exploring promising regions of the search space (Krasnogor, 2009).

- **Offspring improvement.** This operator improves the quality of an offspring by employing a local search procedure. To be specific, such procedure starts with a given offspring and iteratively replaces the incumbent solution with another one taken from its neighbourhood, where the replacement criteria are based on the rule of the local search adopted (e.g. when hill-climbing is used, the incumbent solution is replaced by one of its neighbours only if the fitness value of the current solution is worse than that of the neighbour). This process continues until the stopping condition is met (Hao, 2011). Compared to recombination which mainly performs global exploration, local search is responsible for local exploitation. When designing local search procedures, domain knowledge about the underlying problem should be taken into account to make the search more efficient (Krasnogor, 2009).
- **Population update**. The population management scheme decides whether an offspring should be allowed to enter the population and if so which individual to replace. This scheme helps to maintain a population with a constant size (Moscato *et al*, 2004; Moscato and Cotta, 2007). One of the updating criteria considers the quality issue, e.g. the offspring replace the individual with the worst fitness in the population if the former has a lower fitness value than the latter; another criterion is concerned with the diversity issue, i.e. the offspring replace a similar individual according to some distance measurements. One is encouraged to consider these criteria at the same time to gain a better performance (Hao, 2011). Population update schemes are important to keep the diversification and avoid premature convergence.

- **Termination conditions**. There are a number of common stopping conditions, such as a predefined number of generations or evaluations, a predefined number of consecutive generations without improvement of the best-so-far solution, the emergence of a solution with acceptable quality, or a lower-bounded threshold for the population diversity (Hao, 2011).

Making full use of the advantages of EAs and local search metaheuristics (with domain knowledge exploited), MAs have been recognized as a powerful tool in EC and applied in many fields as diverse as operational research and management, electronics and engineering, machine learning and robotics, medicine and bioinformatics, economics, and oceanography. Detailed survey on MAs can be found in Moscato *et al* (2004), Moscato and Cotta (2007), Moscato and Cotta (2010), and Berretta *et al* (2012). Besides, two EAs incorporating local search techniques are also developed to solve the NCRM problem (see Chapter 4).

2.2.3 Estimation of Distribution Algorithms

Estimation of distribution algorithms (EDAs), also known as probabilistic model building genetic algorithms (PMBGAs), are often considered as an outgrowth of GAs (Baluja, 1994; Harik *et al*, 1999; Pelikan *et al*, 2002). In GAs, a population of candidate solutions is maintained and operated by crossover and mutation to drive the evolution process. However, EDAs do not manipulate such a population. Instead, they maintain a probability distribution which is extracted from a set of selected promising individuals from the previous generation. The probability distribution, sometimes called probability vector, in turn, is sampled to create a set of new and fitter individuals (Larranaga and Lozano, 2002). EDAs have a lot of advantages compared to traditional GAs. The most attractive ones are: (1) the absence of multiple controlling parameters (e.g. crossover and mutation probabilities), and (2) the expressiveness and transparency of the probabilistic model that guides the search process (Armananzas *et al*, 2008). In the thesis, we develop a number of EDAs to solve the optimization problem concerned (see Chapter 4 and 5). The basic procedure of EDAs is shown in Fig. 2.2 (Hauschild and Pelikan, 2011).

As outlined in Figure 2.2, EDAs start with an initial population of solutions which are randomly sampled from the search space. The population is then evaluated and each individual is assigned a fitness value. After that, a subset of the most promising solutions is selected and their statistical information is extracted to construct a probabilistic model ***PV*** which estimates the probability distribution of the selected solutions. By sampling the probabilistic model, a set of new solutions are generated and then used to update the old population. The above process continues until some termination condition is met (e.g. a predefined number of generations).

1. Set generation counter $t = 0$;
2. Generate initial population $S(t)$;
3. **while** *termination condition is not met* **do**
4. Select a set of promising solutions $s(t)$ from $S(t)$;
5. Build the probabilistic model $\boldsymbol{PV}(t)$ from the $s(t)$;
6. Sample $\boldsymbol{PV}(t)$ to generate new candidate solutions $O(t)$;
7. Incoporate $O(t)$ into $S(t)$;
8. Set $t = t + 1$;
9. **endwhile**

Figure 2. 2 The basic procedure of EDAs (Hauschild and Pelikan, 2011)

As being simple and effective, EDAs have been thoroughly investigated and successfully applied to address many COPs, such as nurse rostering (Aickelin *et al*, 2007; Aickelin and Li, 2007), job shop scheduling problem (Li and Jiang, 2009; Zhang and Li, 2011), hardware design (Gallagher *et al*, 2004), dynamic optimization problems (Yang and Yao, 2005; 2008), and so on. More discussions and applications can be found in Hauschild and Pelikan (2011) and Santana (2011). In addition, the thesis also adopts EDAs to tackle NCRM and delay-constrained NCRM problems (see Chapter 4 and 5).

The following reviews some of the well-known EDAs, including population based incremental learning (PBIL), compact genetic algorithm (cGA), univariate marginal distribution algorithm (UMDA), and quantum-inspired evolutionary algorithm (QEA).

2.2.3.1 Population Based Incremental Learning (PBIL)

Population based incremental learning (PBIL), a combination of evolutionary algorithm and competitive learning, was first introduced by Baluja (1994). Without using crossover and mutation, PBIL retains the stochastic search nature of GA by simply maintaining a single real-valued probability vector which generates promising solutions with high probabilities when sampled.

The procedure of the standard PBIL with elitism is shown in Figure 2.3 (Baluja, 1994). The probability vector at generation t is denoted by $\boldsymbol{PV}(t) = \{p_1^t, p_2^t, \ldots, p_L^t\}$, where L is the binary-encoding length. The value on each locus of $\boldsymbol{PV}(t)$, i.e. p_i^t, $i = 1, 2, \ldots, L$, is initialized as 0.5. At each generation, $\boldsymbol{PV}(t)$ is sampled to form a sampling set $S(t)$ of N solutions. Each solution in $S(t)$ is then evaluated and assigned a fitness value using a problem-specific fitness function. After fitness evaluation, the best so far solution $\boldsymbol{B}(t) = \{b_1^t, b_2^t, \ldots, b_L^t\}$ is selected and used to update $\boldsymbol{PV}(t)$ as follows:

$$p_i^t = (1.0 - \alpha) \cdot p_i^t + \alpha \cdot b_i^t, \qquad i = 1, 2, \ldots, L \tag{2.2}$$

where α is the learning rate specifying the distance $\boldsymbol{PV}(t)$ is shifted at each generation.

1. **Initialization**
2. Set $t = 0$;
3. **For** $i = 1$ **to** L **do** set $p_i^t = 0.5$;
4. Generate a sampling set $S(t)$ of N solutions from $\boldsymbol{PV}(t)$;
5. **Repeat**
6. Set $t = t + 1$;
7. Evaluate the samples in $S(t\text{-}1)$;
8. Find the best solution $\boldsymbol{B}(t)$ from $\boldsymbol{B}(t\text{-}1) \cup S(t\text{-}1)$;
9. Update $\boldsymbol{PV}(t)$ by Equation (2.2);
10. Mutate $\boldsymbol{PV}(t)$;
11. Generate a set $S(t)$ of N samples by $\boldsymbol{PV}(t)$;
12. **until** *termination condition is met*

Figure 2. 3 The basic procedure of PBIL (Baluja, 1994)

After learning $\boldsymbol{PV}(t)$ towards the best solution, a bit-wise mutation operation may be adopted to maintain diversity and avoid local optima. The mutation operation used by Baluja (1994) is to introduce a small amount of probability perturbation on each locus in $\boldsymbol{PV}(t)$ if mutation criteria is met. After mutation, a new sampling set is generated by the new $\boldsymbol{PV}(t)$. Steps 6-11 are repeated unless the termination condition is met. With the evolution progressing, the value on each locus is shifted to 0.0 or 1.0 and $\boldsymbol{PV}(t)$ gradually converges to an explicit solution.

The last decade has witnessed a significant growth in the number of applications of PBILs. The following lists some: dynamic optimization problems (Yang and Yao, 2005, 2008), electromagnetics optimization (Yang *et al*, 2007), iterated prisoners dilemma (Gosling *et al*, 2005), job shop scheduling problem (Kern, 2006), vehicle routing problem (Kern, 2006), travelling salesman problem (Kern, 2006; Ventresca and Tizhoosh, 2008), power system controller design (Sheetekela and Folly, 2010), profit-maximizing strategies in the payment card market (Alexandrova-Kabadjova *et al*, 2011), network routing in telecommunications (Khezri *et al*, 2011; Xing and Qu, 2011a, 2011b).

2.2.3.2 Compact Genetic Algorithm (cGA)

As one of estimation of distribution algorithms (EDAs), the compact genetic algorithm (cGA) was first introduced by Harik *et al* (1999). Whereas standard GA maintains a population of solutions, cGA simply employs a probability vector (PV) while still retaining the behaviour (Harik *et al*, 1999) of the standard GA with a uniform crossover. Contrary to the standard GA, cGA was found much faster, and requires far less memory so that significant amounts of computational time and memory are saved. Hence, cGA has drawn an increasing research attention and been successfully applied to a number of COPs including evolvable hardware implementation (Aporntewan and Chongstitvatana, 2001; Gallagher *et al*, 2004), multi-FPGA partitioning (Hidalgo *et al*, 2001), image recognition (Silva *et al*, 2008), TSK-type fuzzy model (Lin *et al*, 2010), telecommunications (Xing and Qu, 2012) and so on.

For cGA, at each generation, only two solutions are sampled from the probability vector PV and a single tournament is performed between them, i.e. a winner and a loser

are identified (Harik *et al*, 1999). The PV is then adjusted and shifted towards the winner. With cGA evolving, the PV converges to an explicit solution. Let the aforementioned PV at generation t denoted by $\boldsymbol{PV}(t) = \{p_1^t,\ldots,p_L^t\}$, where L is the encoding length. The value at each locus of $\boldsymbol{PV}(t)$, i.e. p_i^t, $i = 1,\ldots,L$, is initialized as 0.5 so that initially all solutions in the search space appear with the same probability. Let winner(i) and loser(i), $i = 1,\ldots,L$, be the i-th bit of the winner and the loser, respectively, and $1/N$ be the increment of the probability of the winning alleles after each competition, where N is an integer. Note that although cGA produces two solutions at each generation, it can mimic the convergence behaviour of a standard GA with a population size N (Harik *et al*, 1999). The basic procedure of the cGA is presented in Figure 2.4.

```
1.   Initialization
2.      Set t = 0;
3.      for i = 1 to L do set p_i^t = 0.5;
4.   Repeat
5.      Set t = t + 1;
        // Generate two individuals from the PV
6.      X_a = generate (PV(t)); X_b = generate (PV(t));
        // Let X_a and X_b compete
7.      winner, loser = compete (X_a, X_b);
        // The PV learns towards the winner
8.      for i = 1 to L do
9.         if winner(i) <> loser(i) then
10.           if winner(i) == 1 then set p_i^t = p_i^t + 1/N;
11.           else set p_i^t = p_i^t  – 1/N;
11.  until the PV has converged
12.  Output the converged PV as the final solution
```

Figure 2. 4 The basic procedure of cGA (Harik *et al*, 1999)

2.2.3.3 Univariate Marginal Distribution Algorithm (UMDA)

Mühlenbein and Paaß (1996) invented the univariate marginal distribution algorithm (UMDA). Like PBIL, UMDA also maintains a probability vector PV. At each generation,

UMDA selects a set of promising solutions from a population of samples and computes the frequencies of values on each position in these solutions. These frequencies are then used to generate new solutions which replace the old ones. The above process is repeated until the termination condition is met. The difference between PBIL and UMDA mainly lies in the way to update PV. Let $\boldsymbol{PV}(t)$ and $\boldsymbol{B}(t)$ be the PV and the best sample at generation t, respectively. In PBIL, $\boldsymbol{PV}(t)$ is updated by learning $\boldsymbol{PV}(t-1)$ towards $\boldsymbol{B}(t)$ while in UMDA, $\boldsymbol{PV}(t)$ is exactly determined by the statistics extracted from the selected promising solutions at generation t (Mühlenbein and Paaß, 1996; Pelikan *et al*, 2002). UMDAs have been applied to a number of COPs, such as dynamic optimization problems (Ghosh and Mühlenbein, 2004; Yang, 2005; Wu *et al*, 2010), communications systems (Bashir *et al*, 2010), and multi-processor scheduling (Hashemi and Meybodi, 2011).

2.2.3.4 Quantum-Inspired Evolutionary Algorithm (QEA)

By combining quantum computing and evolutionary algorithm, Han and Kim (2002) put forward the first quantum-inspired evolutionary algorithm (QEA). QEA maintains a population of Quantum-bit encoded individuals, each representing a linear superposition of all states in search space probabilistically, and adopts quantum gates, e.g. quantum rotation gate (Han and Kim, 2002) and quantum NOT gate (Xing *et al*, 2009a, 2009b, 2010), to change the probabilistic distribution of each individual in such a way that promising solutions have increasingly more chance to appear. As one of estimation of distribution algorithms (EDAs), QEA maintains and incrementally modifies multiple probabilistic models (Platel *et al*, 2008). QEA is characterized by maintaining a diversified population due to the Quantum-bit representation, being able to explore the search space with a smaller number of individuals and exploiting the search space for a global solution within a short computational time (Han and Kim, 2004). Hence, since its birth, QEA has drawn extensive research attention and been illustrated to be quite effective for solving complex optimization problems such as numerical and knapsack problems (Han and Kim, 2002; 2004; Wang *et al*, 2005), multi-objective flow shop scheduling problem (Li and Wang, 2007), unit commitment problem (Lau *et al*, 2009), parameter estimation of chaotic systems (Wang and Li, 2010), QoS multicast routing

problem in optical networks (Xing *et al*, 2009a; 2009b), network coding resource optimization problem (Xing *et al*, 2010; Ji and Xing, 2011) and so on.

2.2.3.5 EDAs for More Complicated Problems

The four EDAs above perform well for problems with no or light interactions among variables. However, they may not achieve descent performance when used to tackle problems where strong interdependencies are found among variables (Pelikan *et al*, 2002). Hence, a number of advanced EDAs have been developed for addressing more complicated COPs, taking into consideration the interactions among variables. The following lists some of them: the mutual information maximizing input clustering (MIMIC) algorithm (De Bonet *et al*, 1997), the bivariate marginal distribution algorithm (BMDA) (Pelikan and Mühlenbein, 1999), the extended compact genetic algorithm (ECGA) (Harik, 1999), the factorized distribution algorithm (FDA) (Mühlenbein *et al*, 1999), and the Bayesian optimization algorithm (BOA) (Pelikan *et al*, 2000). Detailed survey can be found in Pelikan *et al* (2002) and Hauschild and Pelikan (2011).

2.2.4 Other Algorithms

Evolutionary strategy (ES) and differential evolution (DE) are two branches of EC algorithms which typically aim at continuous optimization problems (Bäck *et al*, 1991; Storn and Price, 1997). However, they have also been adapted for COPs, e.g. survivable network design (Nissen and Gold, 2008), circle packing problem (Shi *et al*, 2010), travelling salesman problem (Prado *et al*, 2010) and graph coloring (Fister and Brest, 2011). In addition to EAs, there are two swarm intelligence methods that are commonly used in solving COPs, i.e. particle swarm optimization (PSO) algorithms and ant colony optimization (ACO) algorithms (Kennedy and Eberhart, 1995; Dorigo *et al*, 1996). Detailed descriptions and applications about PSOs and ACOs can be found in Poli (2008), Dorigo and Blum (2005), and Benyahia (2012). Guided local search is a metaheuristic (Voudouris and Tsang, 2003; Voudouris *et al*, 2010) that has been successfully applied to constrained optimization problems, e.g. the travelling salesman problem (Voudouris and Tsang, 1999), quadratic assignment problem (Mills *et al*, 2003), and workforce and empowerment scheduling problems (Tsang and Voudouris, 1997; Alsheddy, 2011).

2.3 EAs for MOPs

As reviewed in Subsection 2.1.2, the nature of the multiobjective optimization problems (MOPs) involves optimizing a number of objectives (that often conflict with each other) simultaneously. One may not find a single solution that is optimized in every objective, e.g. the improvement in one objective may lead to the deterioration in other objectives. To address a given MOP is to find a set of pareto-optimal solutions, known as Pareto set (PS), where the image of the PS in the objective space is called Pareto-optimal front (PF) (see definitions in Section 2.1.2). There are two goals when approximating the PF, namely, minimizing the distance between the set of obtained nondominated solutions and the PS, and maximizing the diversity of these obtained solutions in the set such that they can be evenly distributed along the PF (Zitzler *et al*, 2004; Deb, 2011).

EAs have been recognized as one of the ideal choices for solving MOPs (Fonseca and Fleming, 1995; Zitzler *et al*, 2004; Deb, 2011). Due to their population-based structure, EAs can search for multiple nondominated solutions in parallel and obtain a trade-off among objectives in a single run. EAs for tackling MOPs are referred to as *multiobjective evolutionary algorithms* (MOEAs) (Zhou *et al*, 2011). Chapter 6 develops a MOEA to study the CDBO problem.

This section first reviews some traditional MOEAs and then reviews several types of state-of-the-art MOEAs, including decomposition-based MOEAs, Indicator-based MOEAs, Preference-based MOEAs, Hybrid MOEAs and coevolution-based MOEAs.

2.3.1 Traditional MOEAs

The very first MOEA was introduced by Schaffer (1985), namely vector evaluation genetic algorithm (VEGA). After that, increasingly more efforts were spent in addressing MOPs and a number of traditional MOEAs were proposed a decade ago, for example, elitist nondominated sorting GA (NSGA-II) (Deb *et al*, 2002), improved strength Pareto EA (SPEA2) (Zitzler *et al*, 2001), Pareto archived evolution strategy (PAES) (Knowles and Corne, 2000a), Pareto envelope-based selection algorithm (PESA) (Corne *et al*,

2000), niched Pareto GA 2 (NPGA2) (Erickson *et al*, 2001) and micro-GA (Coello Coello and Toscano Pulido, 2001). Among them, some are still active and widely used nowadays, such as NSGA-II and SPEA2. The following briefly reviews NSGA-II and SPEA2.

- **Elitist nondominated sorting GA (NSGA-II).** NSGA-II is an improved version of NSGA which was proposed by Srinivas and Deb in 1995. Compared with its predecessor, NSGA-II is featured with three significant advantages: (1) a fast nondominated sorting approach that reduces the computational complexity from $O(MN^3)$ to $O(MN^2)$, where M is the number of objectives and N is the population size; (2) an elitism approach that speeds up the search and prevents the loss of the promising solutions already obtained; and (3) a parameter-less diversity-preservation scheme that eliminates the difficulty of setting an appropriate value for the sharing parameter, which is responsible for keeping the population diversified for NSGA. Since its introduction in 2002, NSGA-II and its variants have been widely and successfully applied to various areas, such as function optimization (Deb *et al*, 2002; Deb, 2008), engineering optimization (Coelho and Alotto, 2008; Kannan *et al*, 2009; Wang *et al*, 2009b), management (Kishor *et al*, 2007; Serrano *et al*, 2007), economic load dispatch (Xu *et al*, 2011), vehicle routing problem (Castro-Gutierrez *et al*, 2011; Jemai *et al*, 2012), and medical treatment (Heris and Khaloozadeh, 2011). Besides, NSGA-II is adapted for the CDBO problem in Chapter 6.
- **Improved strength Pareto EA (SPEA2).** This algorithm was proposed by Zitzler *et al* (2001). Compared with its predecessor, SPEA2 have three significant improvements, namely a fine-grained fitness assignment strategy, a density estimation technique and an enhanced archive truncation method. In the new fitness assignment scheme, for each individual, the number of individuals that it dominates and the number of individuals that dominates it are taken into consideration at the same time. This helps to incorporate more Pareto dominance information to effectively guide the search towards Pareto optimal front (Zitzler *et al*, 2001, 2004). The density estimation technique is an adaptation of the *k*-th nearest neighbour method, where the density around a

given point is a decreasing function of the distance to the *k*-th nearest point (Silverman, 1986). In addition, the enhanced archive truncation method performs well on avoiding boundary solutions being eliminated from the archive. Like NSGA-II, SPEA2 has been recognized as one of the versatile MOEAs and used in many MOPs, including multiobjective knapsack problem (Zitzler *et al*, 2001), multi-mission waveform design (Amuso and Enslin, 2007), broadcast optimization in MANET (Conzález *et al*, 2007), power systems (Khaleghi *et al*, 2009; Inglis *et al*, 2010), economic load dispatch (Xu *et al*, 2011), software architecture design (Li *et al*, 2011). This algorithm is also adapted for the CDBO problem (see Chapter 6).

2.3.2 Decomposition-based MOEAs

In 2007, a MOEA based on decomposition (MOEA/D) was proposed for the first time (Zhang and Li, 2007). Different from traditional MOEAs which are mainly based on Pareto dominance, this algorithm is based on weighted aggregation approaches, where a MOP is decomposed into a number of scalar optimization subproblems. In the course of evolution, all subproblems are optimized in parallel and each of them is addressed by making use of the information obtained from its neighboring subproblems. The definition of the neighbourhood relations among these subproblems is based on the distances between their aggregation weight vectors. In MOEA/D, each solution in the population is associated with a subproblem, where the solution is usually the best ever found for the corresponding subproblem. At each generation, MOEA/D generates a new solution for each subproblem via the recombination of a number of solutions from its neighboring subproblems. The current solution of the subproblem is replaced with the new one if the latter is better. Moreover, the new solution is compared with those of the neighboring subproblems and replacement is made if the new one is better. Compared with MOEAs without decomposition, such as NSGA-II and SPEA2, MOEA\D is featured with a significant advantage, namely scalar objective local search can be easily applied in each subproblem (Zhang and Li, 2007; Zhou *et al*, 2011).

Since Zhang and Li's seminal work, MOEAs with decomposition have received increasing research interests and already become state-of-the-art MOEAs. The following

reviews some variants and applications of MOEA/D. Li and Landa-Silva (2008, 2011) investigated the hybridization of MOEA/D and simulated annealing for addressing multiobjective combinatorial optimization problems. In their work, weight vectors (namely search directions) are adaptively adjusted to diversify the Pareto set (PS) during the evolution; novel competition schemes and external population updating schemes are introduced to ensure more effective search. Li and Zhang (2009) proposed a modified MOEA/D for continuous multiobjective optimization, where differential evolution and polynomial mutation are incorporated for balancing global exploration and local exploitation. Another MOEA/D with a dynamic computational resource allocation scheme was investigated (Zhang *et al*, 2009). This variant was reported to improve the algorithm performance with reduced overall cost. A MOEA/D with probabilistic representation based on pheromone trains was developed for the multiobjective travelling salesman problem, where high-quality solutions are easily generated and the evolutionary search can well approximate the Pareto front (PF) (Li *et al*, 2010). Zhang *et al* (2010b) hybridized MOEA/D with efficient global optimization algorithm, called MOEA/D-EGO, which was quite effective in solving expensive multiobjective optimization. Nebro and Durillo (2010) proposed a parallel version of MOEA/D which was run on modern multi-core processors in order to significantly reduce computational time. Ishibuchi *et al* (2010) presented an idea of simultaneously using different types of scalarizing functions to handle the problem where appropriate scalarizing functions are difficult to choose for different MOP. A hybridization of MOEA/D and particle swarm optimization was developed to solve continuous and unconstrained MOPs (Zapotecas Martínez and Coello Coello, 2011). Zhao *et al* (2012) proposed another MOEA/D that is based on the use of ensemble of different neighbourhood sizes with online self-adaptation. MOEA/Ds have been successfully applied in a number of real world applications, such as flowshop scheduling problem (Chang *et al*, 2008), portfolio management (Zhang *et al*, 2010a), multiobjective travelling salesman problem (Cheng *et al*, 2011; Shim *et al*, 2012), routing and deployment problems in sensor networks (Konstantinidis *et al*, 2010; Konstantinidis and Yang, 2012), microwave components synthesis (Bo *et al*, 2012) and antenna design (Carvalho *et al*, 2012). More information on MOEA/D can be found at

http://dces.essex.ac.uk/staff/zhang/webofmoead.htm which is maintained by Prof. Qingfu Zhang from University of Essex.

2.3.3 Indicator-based MOEAs

Indicator-based MOEAs is another group of state-of-the-art MOEAs. As we know, performance indicators (such as generational distance and hypervolume) are premarily used to measure the quality of an approximation of the PF in order to evaluate the performance of some MOEAs. However, in indicator-based MOEAs, those indicators are calculated at each generation and incorporated in evolutionary process such as selection in order to guide the search (Zitzler and Künzli, 2004; Zhou *et al*, 2011).

Zitzler and Künzli (2004) initiated the research on how to incorporate the preference information of the decision maker into the multiobjective search process. The authors proposed a general indicator-based MOEA, which is referred to as IBEA. This algorithm can be easily adapted to arbitrary preference information and does not require diversity preservation mechanism such as fitness sharing. IBEA was reported to gain better overall performance, compared with NSGA-II and SPEA2. Basseur and Zitzler (2006) focused on addressing MOPs with uncertainties and proposed a general indicator model that is suited to handle uncertainty. In their model, each solution is associated with a probability over the objective space. Meanwhile, a ϵ-indicator based model was also presented to improve the performance in environmental selection. In 2007, a general approach was proposed to incorporate objective reduction techniques into hypervolume based MOEAs (Brockhoff and Zitzler, 2007). Various objective reduction schemes were developed and tested. The obtained results showed that temporarily reducing the number of objectives could significantly improve the performance of hypervolume based MOEAs. Bader and Zitzler (2008) studied the high computational complexity caused by hypervolume calculation in many-objective optimization problem and proposed a fast hypervolume based MOEA (called HypE), where a technique based Monte Carlo simulations was developed to approximate the exact hypervolume values in order to reduce the runtime. Boonma and Suzuki (2011) presented a prospect indicator based MOEA, called PIBEA, where the new indicator measures the potential of each solution and helps to produce better offsprings. Basically, PIBEA has two advantages when solving many-objective

optimization problem, namely maintaining sufficient selection pressure and high level of diversity. PIBEA was reported to outperform well established MOEAs such as NSGA-II and SPEA2, regarding the convergence, diversity in the population, coverage of the PF, etc. A MOEA with boosted indicator (called BIBEA) was proposed, where a boosting approach aggregates existing quality indicators into a single indicator (Phan *et al*, 2011). The boosting approach is carried out with a training problem where the PS is known. A new performance indicator, namely Δ_p, was introduced, which is based on the averaged Hausdorff distance and composed of two well-known quality indicators, generational distance and inverted generational distance (Schutze et al, 2012). The Δ_p indicator was then incorporated into the selection mechanism in a differential evolution based MOEA (Rodríguez Villalobos and Coello Coello, 2012). The main advantage of the Δ_p indicator based MOEA is its significantly reduced computational cost.

2.3.4 Preference-based MOEAs

Depending on when the decision maker (DM) expresses his/her preferences, the multiobjective optimization methods can be classified into three categories: priori methods, posteriori methods and interactive methods (Miettinen, 1999). For the first category, preference information is given before the search and such information is used to guide the search towards the most interesting regions along the PF. In a posteriori method, the DM selects the most preferred solutions after the search when an approximated PF is ready for use. For interactive methods, the DM is involved in the whole process of search and interactively expresses the preference to guide the search.

There has been plenty amount of research concerning the use of preferences in solving MOPs. Fonseca and Fleming (1993) introduced a rank-based fitness assignment method to incorporate the preference information of the DM in an interactive manner, where the rank of individuals in the population is based on the Pareto dominance and the preferences of the DM. Greenwood *et al* (1996) designed a value function to rank the population and combined the preference information with Pareto rankings. Sakawa and Kato (2002) studied the preference-based multiobjective integer programming problems. They incorporated the fuzzy goals of the DM into the objective functions, where the reference membership values can be adjusted according to the preferences of DM.

Branke and Deb (2004) incorporated imprecise user preferences into NSGA-II via a guided dominance scheme and a biased crowding distance scheme. Deb *et al* (2006) investigated the use of reference point interactive approach in finding a set of preferred nondominated solutions for the DM. Deb and Chaudhuri (2005, 2007) suggested an interactive decision support system (I-MODE) that allows the DM to have an overview of the complete PF and can iteratively interact with the DM to guide the search towards the interesting regions along the PF. Thiele *et al* (2009) proposed a preference based MOEA, where the DM is asked at each generation to express his/her reference point consisting of desirable aspiration levels for objective functions. The preferred point is utilized to generate the offspring population by combining the fitness function and an achievement scalarizing function. Wagner and Trautmann (2010) introduced a concept for approximating preferred regions of the PF, where the preference information from the DM is incorporated into the desirability functions (DFs) of the objectives. The authors integrated the DFs into the *S*-metric selection based MOEA, which successfully restricts the search in the preferred regions of the PF. Friedrick *et al* (2011, 2013) incorporated the preferences of the DM by weighted information on the objective space. This method was successfully applied to well-known MOEAs such as NSGA-II and SPEA2. Pedro and Takahashi (2013) proposed an interactive territory defining EA (iTDEA) to address the MOPs with the user preference considered. In iTDEA, the preference model is constructed based on Neural Network technique, where the preference is incorporated into the training process. Other approaches that make use of the user preferences can be found in Zhou *et al* (2011) and Friedrich *et al* (2013).

2.3.5 Hybridization-based MOEAs

As reviewed in Subsections 2.2.1 and 2.2.2, incorporating local search techniques into EAs can enhance the optimization performance. Similarly, in MOEAs, one research stream considers the combination of different search techniques. A basic idea is to hybridize global search and local search, making use of their advantages (e.g. enhanced global exploration and local expoitation abilities). Ishibuchi and Murata (1996, 1998) introduced a hybrid MOEA, where a local search process is performed on each individual generated by genetic operators. In this algorithm, an aggregation function is used and its

weight values are randomly generated whenever parent solutions are to be selected for recombination. Ahn *et al* (2009, 2010) proposed a hybrid MOEA (HMEA) with an adaptive local search (ALS) incorporated. The ALS is based on weighted fitness function and knowledge based local search, being capable of efficiently exploiting the neighbourhoods of the most promising individuals. Santana-Quintero *et al* (2010) hybridized a MOEA based on differential evolution and a local search procedure based on rough set theory for addressing difficult constrained MOPs. MOEAs integrating global and local search techniques are sometimes referred to as multiobjective memetic algorithms (Knowles and Corne, 2000b; Jaszkiewicz, 2002; Ishibuchi *et al*, 2009; Qian *et al*, 2009; Lara *et al*, 2010; Tan *et al*, 2012). MOEA/Ds introduced in Section 2.3.2 can be viewed as multiobjective memetic algorithms since local search procedures can be adopted in MOEA/Ds to improve the quality of individuals at each generation. Another idea for hybridization is to combine the search operators coming from different MOEAs. For example, Li and Wang (2007) introduced a hybrid quantum-inspired genetic algorithm (QEA) (see Section 2.2.3 for the principle of QEA) to address multiobjective flow shop scheduling problem. In the proposed algorithm, quantum operators are applied to binary encoding individuals while genetic operators are applied to permutation encoding individuals. Chen *et al* (2010) proposed a hybrid immune MOEA, where a mutation operator is designed based on the combination of Gaussian and polynomial mutations. Tang and Wang (2013) developed a hybrid MOEA (HMOEA) for real-valued MOPs. The concepts of personal best and global best in particle swarm optimization (PSO) are used to update the population. Multiple crossover operators are utilized in the reproduction and a self-adaptive selection mechanism is used to select one of them, helping HMOEA gain a robust performance when adopted to solve different types of MOPs. Hybrid MOEAs have been applied in many real-world applications, such as the uncapacitated exam proximity problem (Côté and Wong, 2004), aircraft control system design (Adra *et al*, 2005), the design of compliant micro-actuators (Tai *et al*, 2008), space trajectory optimization (Vasile and Zuiani, 2010), machining parameters design (Deb and Datta, 2011), capacitated arc routing problem (Mei *et al*, 2011), etc.

2.3.6 Coevolution-based MOEAs

Coevolution can be interpreted as evolving multiple subpopulations which evolve independently and share information with each other. Kim *et al* (2009) investigated multiobjective population-based incremental learning (PBIL) for solving multiobjective routing problem in robot soccer systems. They maintained multiple probability vectors, each representing a subpopulation. Each probability vector is updated independently and the nondominated solutions found during the evolution are shared by the probability vectors. On the other hand, coevolution can be more challenging with regard to the idea of divide and conquer, where the targeted problem is split into a number of subproblems and each subpopulation addresses one of them (Zhou *et al*, 2011). Such subpopulations compete and cooperate with each other. Lohn *et al* (2002) presented a coevolutionary GA (CGA) based on competitive coevolution. In the fitness evaluation, the coevolutionary dynamics are adopted to adaptively regulate the difficulty level. Coello Coello and Sierra (2003) proposed a coevolutionary MOEA where the search space is divided in to a number of subregions according to the estimation of the importance of each decision variable. The size of each subpopulation changes with respect to their contribution to the approximated Pareto optimal front. A cooperative coevolutionary algorithm (CCEA) for MOPs was developed, where multiple solutions are evolved in the form of cooperative subpopulations (Tan *et al*, 2006). Meanwhile, CCEA was decentralized according to its inherent parallelism of cooperative coevolution, which was reported to efficiently reduce the computational time. Goh and Tan (2009) investigated a coevolutionary paradigm for MOPs in dynamic environment. The main idea is to design an adaptive decomposition process of the given MOP, where subpopulations compete to survive and the winners collaborate to reproduce offsprings. Zhan *et al* (2013) proposed another coevolutionary MOEA based on the idea of multiple populations for multiple objectives (MPMO). Each subpopulation is associated with only one objective and all subpopulations cooperate to evolve the population. More descriptions of coevolutionary MOEAs can be found in Cochran *et al* (2003), Chang *et al* (2007), Zhou *et al* (2011) and Coello (2013).

2.4 Summary

In this chapter, we review a number of well-known and commonly used EAs. First, we review the principles of three major branches of EAs, genetic algorithms, memetic algorithms and estimation of distribution algorithms, and their applications for COPs. Standard EAs may not handle well the exploration and the exploitation simultaneously. Incorporating local search techniques (maybe with domain knowledge considered) into EAs have been suggested to improve the performance of EAs. In nature, PBIL and cGA maintain a probability vector (PV) and gradually adjust it by the information extracted from promising solutions. The continuously changing PV helps to smoothly guide the search towards promising regions in the search space. Then, we review two traditional MOEAs, NSGA-II and SPEA2. They have many applications on MOPs and are still used recently. Finally, several state-of-the-art MOEAs are reviewed, including MOEAs with decomposition, indicator-based MOEAs, preference-based MOEAs, hybrid MOEAs and coevolutionary MOEAs. EAs have been widely applied to many real world optimization problems, e.g. routing, scheduling and timetabling, and bioinformatics.

As mentioned in Section 1.5, in this thesis, we consider three optimization problems, the network coding resource minimization (NCRM) problem, the NCRM problem with delay constraint, and the cost-delay bi-objective optimization (CDBO) problem. The first two problems are NP-hard COPs while the last one is a MOP. With near-optimal solutions and acceptable computational cost, EAs are ideal choices for solving the three problems above. The detailed description of these problems is given in the next chapter.

Chapter 3 The Network Coding Resource Minimization Problem: An Introduction

As an advanced communication paradigm, network coding can offer many attractive benefits to the modern communication network, e.g. maximized multicast throughput and balanced network payload. Hence, incorporating network coding into multicast may become an ideal technical solution for supporting real time and high speed multicast data transmission. However, a number of issues that hinders the immediate application of network coding based multicast (NCM) need to be addressed first. The network coding resource minimization (NCRM) problem is such an issue, as introduced in Section 1.5.1.

In this chapter, we first introduce the network model and basic concepts. We then describe the problem modelling, the related work and the encoding approaches that have been commonly used to solve the problem. Finally, the experimental network topologies and instances are introduced.

3.1 Network Model and Related Concepts

A communication network can be modelled as a directed acyclic graph $G = (V, E)$, where V and E are the node set and link set, respectively. Assume each link $e(u,v) \in E$ has a unit capacity, where $u,v \in V$. Only integer flows are allowed in G so a link is either idle or occupied by a flow of unit rate (Kim *et al*, 2007a, 2007b). Also, each link may be associated with some properties such as link cost and propagation delay.

A NCM routing problem can be expressed as a source $s \in V$ expects to send the same data to a group of receivers $T = \{t_1,\ldots,t_d\} \subseteq V - s$ at data rate R in the network G, where R is an integer and d is the number of receivers. R is no greater than the theoretical maximum throughput. Rate R (a capacity of R units) is achievable if any receiver $t_k \in T$ can receive the information sent from the source at rate R (Kim *et al*, 2007a, 2007b). As each link has a unit capacity, any single path connecting s and t_k ($k = 1, \ldots, d$) has a unit capacity. If we manage to set up R link-disjoint paths from s to each receiver t_k, denoted by $P_1(s, t_k),\ldots,P_R(s, t_k)$, data rate R can be achieved by linear network coding (see

Subsection 1.3.1 for details). This is because linear network coding is able to assign appropriate coding coefficients to the network so that for each receiver t_k the information received from the R link-disjoint paths is independent. Therefore, each receiver can decode and obtain R-unit original data sent from the source at each time unit.

It is noted that the data rate, R and the number of receivers, d are two key parameters in NCM. The following reviews the sizes of R and d used in those experiments based on real life networks, as listed in Table 3.1. We notice that R is from 2 to 4. Intuitively, these numbers are small. This is because R is a relative value showing how many times larger than a given data rate (Kim *et al*, 2007a). For example, if the given data rate is 2Mb/s and R = 2, NCM can achieve a data rate of 4Mb/s. To the best of our knowledge, the highest data rate achieved by NCM is 5GB/s, where NCM is implemented in an optical testbed and R = 2, d = 2(Qu *et al*, 2010). A larger R indicates a higher data rate that can be achieved by NCM. It is also seen that d is between 3 and 10, which reflects the size of a multicast group. A larger d means that more receivers request for the same information from the source in the same period.

Table 3. 1 The Sizes of R and d Investigated Based on Real Life Networks

Real life Networks	R	d	References
LATA-X	2	9	Kim *et al* (2006)
Ebone	2,3,4	4	Kim *et al* (2006), Ahn (2011), Luong *et al* (2012)
Exodus	2,3,4	10	Kim (2008)
ARPANET	4	3	Ramanathan *et al* (2010)
NSFNET	4	3	Ramanathan *et al* (2010)
NJLATA	4	3	Ramanathan *et al* (2010)

3.1.1 Network Coding Based Multicast Subgraph

Given a NCM request[2], the task is to find a connected subgraph in G to support a multicast session by using network coding. This subgraph is composed of a set of paths originating from s and terminating at one of the receivers $t_k \in T$. To be specific, there are R link-disjoint paths $P_i(s, t_k)$, i = 1,…,R, from s to receiver $t_k \in T$, in this subgraph. As

[2] NCM request is generated at a certain source node to set up the NCM data transmission between this node and all its receivers.

reviewed in Subsection 1.4.1, such subgraph is known as network coding based multicast subgraph (NCM subgraph, denoted by $G_{s\to T}$). Note that in a NCM subgraph paths to the same receiver never join together since they are link-disjoint. Only those to different receivers may partially overlap and form coding nodes. In the NCM subgraph, a node is called *coding node* if it performs mathematical operations to recombine the packets received from different incoming links. An outgoing link of a coding node is called a *coding link* if each packet sent via this link is a combination of at least two incoming packets of the coding node. For example, Figure 3.1 shows the NCM subgraph and all paths of the subgraph in Figure 1.3(*c*) (Section 1.3), where $d = 2$ and $R = 2$ (Xing and Qu, 2012). The NCM subgraph consists of four paths, i.e. $P_1(s, y)$, $P_2(s, y)$, $P_1(s, z)$ and $P_2(s, z)$, where paths to the same receiver are link-disjoint, e.g. $P_1(s, y)$ and $P_2(s, y)$.

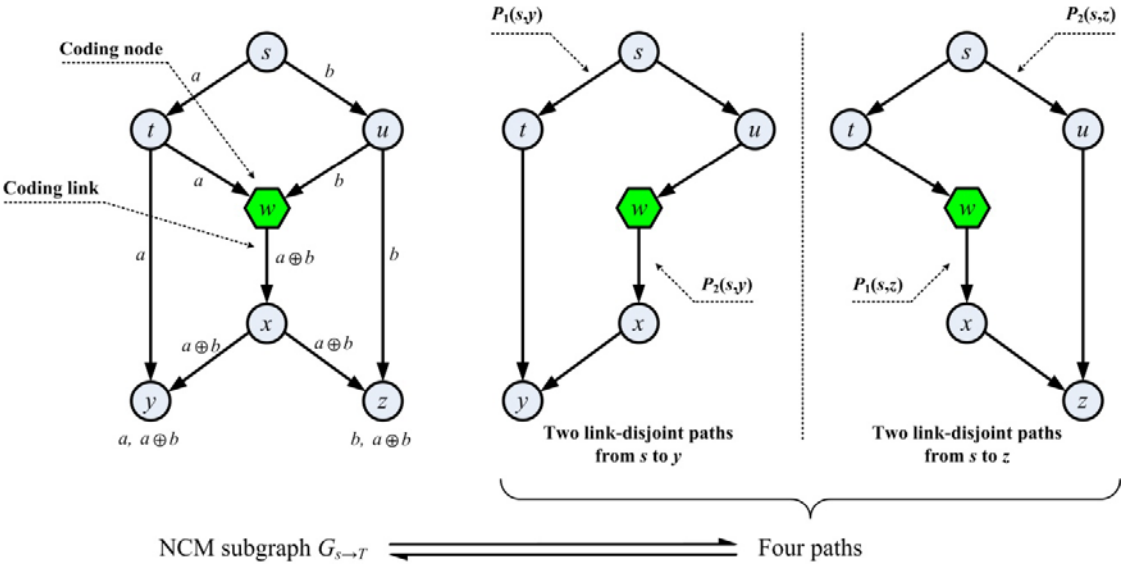


Figure 3. 1 A NCM subgraph and its link-disjoint paths.

3.1.2 Merging Nodes

According to the definition of network coding, coding operation never occurs at a node with single incoming link. This is because the packets that the node receives are from the same incoming link and they have nothing to combine. In network G, we refer to a non-receiver node with multiple incoming links as a *merging node* (Kim *et al*, 2007a,

2007b). Only merging nodes and their outgoing links are possible to become coding nodes and coding links, respectively. Langberg *et al* (2006) found that the number of coding links can well estimate the total amount of coding operations. Hence, this thesis uses the number of coding links as an estimate of the coding operations involved.

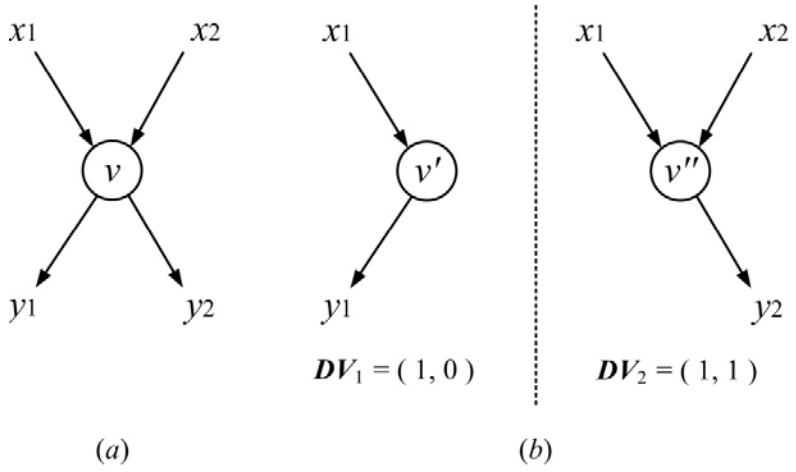


Figure 3. 2 Node v with two incoming and two outgoing links, described by decision vectors $\boldsymbol{DV}_1 = (a_{11}, a_{21})$ and $\boldsymbol{DV}_2 = (a_{12}, a_{22})$. (a) merging node v. (b) decision vectors.

To determine whether an outgoing link of a merging node serves as a coding link, one needs to check whether the information via this link is dependent on at least two incoming links of the merging node. Consider a merging node v with $In(i)$ incoming links and $Out(i)$ outgoing links, where $In(i) \geq 2$ and $Out(i) \geq 1$. A decision vector $\boldsymbol{DV}_k = \{a_{ik} \mid i = 1,\ldots, In(i), k = 1,\ldots, Out(i)\}$ can be used to represent information from how many incoming links of v contributing to the output over the k-th outgoing link of v (Kim *et al*, 2007a, 2007b). If the information from the i-th incoming link contributes to the coded output over the k-th outgoing link, $a_{ik} = 1$, otherwise $a_{ik} = 0$. The k-th outgoing link becomes coding link only if information from at least two incoming links contributes to the output over the k-th outgoing link. As mentioned in Subsection 1.3.1, this thesis only considers linear network coding which is sufficient for multicast (Li *et al*, 2003). Figure 3.2 shows an example of a merging node and two possible decision vectors, respectively (Xing *et al*, 2010). In Figure 3.2(a), x_1 and x_2 are incoming information while y_1 and y_2 are outgoing information. If v only forwards x_1 and x_2, y_1 and y_2 may be either x_1 or x_2,

respectively; otherwise, y_1 and y_2 may be different linear combinations of x_1 and x_2. In Figure 3.2(b), two decision vectors $\boldsymbol{DV}_1$ and $\boldsymbol{DV}_2$ reflect how x_1 and x_2 pass through v. Information x_1 is forwarded over the left outgoing link, so y_1 is exactly x_1. On the other hand, v expects to forward x_1 and x_2 over the right outgoing link. To avoid conflicts, a linear combination of x_1 and x_2 (i.e. y_2) is sent out via the right outgoing link. Hence, the right outgoing link is identified as a coding link.

3.2 Problem Description

As discussed in Subsection 1.5.1, when applying network coding in multicast, the theoretically maximized throughput can always be achieved, however, at the expense of additional computation tasks at coding nodes. Performing packet recombination not only incurs a considerable amount of computational overhead but also consumes extra buffer. The fewer the coding operations performed, the less the computational overhead and delay incurred. It is therefore important to minimize the number of coding operations while retaining the benefits of network coding. This problem is a combinatorial optimization problem and is NP-Hard (Kim *et al*, 2006).

The NCRM problem can be defined as to find a NCM subgraph $G_{s\rightarrow T}$ with the number of coding links minimized and the data rate R achieved. The following lists the notations used to formulate the NCRM problem:

n_{M}: the number of merging nodes in G.

$Out(i)$: the number of outgoing links that the i-th merging node owns.

R: the defined data rate (an integer) at which s expects to transmit information to each receiver t_k.

$\lambda(s, t_k)$: the data rate between s and t_k in $G_{s\rightarrow T}$.

$G_{s\rightarrow T}$: network coding based multicast (NCM) subgraph.

$\Phi(G_{s\rightarrow T})$: the number of coding links in $G_{s\rightarrow T}$.

σ_{ik}: a binary variable associated with the k-th outgoing link of the i-th merging node, $i = 1,\ldots, n_{\mathrm{M}}$, $k = 1,\ldots, n_{\mathrm{OL}i}$. $\sigma_{ik} = 1$ if the j-th outgoing link of the i-th node serves as a coding link; $\sigma_{ik} = 0$ otherwise.

$P_i(s, t_k)$: the i-th link-disjoint path found between s and t_k in $G_{s\rightarrow T}$. It is

represented by a chain of nodes through which the path passes.

$W_i(s, t_k)$: the link set of $P_i(s, t_k)$, i.e. $W_i(s, t_k) = \{e(u,v) \mid e(u,v) \in P_i(s, t_k)\}$.

Minimize:

$$\Phi(G_{s\to T}) = \sum_{i=1}^{n_M} \sum_{j=1}^{Out(i)} \sigma_{ij} \tag{3.1}$$

Subject to:

$$\lambda(s, t_k) = R, \forall\ t_k \in T \tag{3.2}$$

$$W_i(s, t_k) \cap W_j(s, t_k) = \emptyset, \forall i,j \in \{1,...,R\}, i \neq j, t_k \in T \tag{3.3}$$

Objective (3.1) defines the problem as to minimize the number of coding links; Constraint (3.2) defines the achievable rate between s and each receiver as R; Constraint (3.3) indicates that for an arbitrary receiver t_k the R constructed paths $P_i(s, t_k)$, $i = 1,\ldots,R$, must have no common link.

3.3 Related Work

Network coding has received significant amount of attention since its introduction in 2000, especially for the sake of developing efficient encoding-and-decoding network codes for NCM (Li *et al*, 2003; Koetter and Médard, 2003; Jaggi *et al*, 2005; Li and Yeung, 2006; Fong and Yeung, 2010). However, only a few efforts have been dedicated to the network coding resource minimization (NCRM) problem which, if addressed properly, may greatly facilitate the wide applications of NCM.

Bhattad *et al* (2005) investigated the problem of minimizing the number of packets that undergo network coding. They built linear programming formulation and proposed a distributed algorithm to solve it. Besides, the authors also considered a more realistic scenario, i.e. some nodes in the network could only perform store-and-forward schemes rather than coding operations. And another decentralized algorithm was presented. Due to the nature of the problem, the formulations have a complexity that grows dramatically with the number of receivers. Hence this approach is only limited to applications with relatively small multicast group.

Fragouli and Soljanin (2006) studied information flow decomposition for practical NCM. A given network was decomposed into a number of regions such that the same information was transferred within the same region. This method could help to derive the smallest code alphabet size for any NCM with two source nodes. Fragouli and Soljanin presented a greedy algorithm to specify which nodes within the network need to perform coding operations regardless of the overall network topology. This greedy algorithm aims to construct a subtree with the minimal coding operations by sequentially traversing the links in each decomposed graph and deleting those redundant links. In the same year, Langberg *et al* (2006) investigated the design of multicast coding networks with a limited number of nodes that can perform coding operations. The authors proved that in a directed acyclic coding network, the number of encoding nodes required was bounded by h^3k^2, where h is the number of total packets transmitted and k is the number of sinks. An algorithm was developed to minimize the number of required coding nodes by first transforming the given network to a graph where each node has a degree of at most three, and then removing links which make no contribution to the achievable data rate in the transformed graph. However, the performance of the two algorithms was dependent on the order of the traversed links. An inappropriate link traversal order might dramatically weaken the optimization results.

From then on, a few EAs were proposed. Kim *et al* carried out a series of research using GAs to optimize the required network coding resource (Kim *et al*, 2006, 2007a, 2007b). They proposed a GA working in an algebraic framework (Kim *et al*, 2006). Given a network G, they first constructed the corresponding labelled line graph G', using the information flow decomposition method in (Fragouli and Soljanin, 2006) and then associated each link $e \in G'$ with a link coefficient. All the coefficients resulted into a specific solution and the consumed network coding resource was calculated. This algebraic framework was the very first seminal work for mapping the NCRM problem to a GA framework. And the proposed GA performed outstandingly better than the two greedy algorithms mentioned above (Fragouli and Soljanin, 2006; Langberg *et al*, 2006) in terms of the best solutions achieved. However, this GA was only applicable to acyclic networks.

After that, Kim *et al* extended their previous GA (Kim *et al*, 2006) to a distributed version to significantly reduce the computational time. Instead of operating on the labelled line graph, the authors introduced a graph decomposition method which can be used in both acyclic and cyclic networks (Kim *et al*, 2007a). This decomposition method decomposed each merging node in *G* into a number of auxiliary nodes to show explicitly how a flow passes through this merging node. This method then became a popular means to map the NCRM problem to a GA framework (see Subsection 3.2.3 for details). The graph decomposition method is also adopted by the proposed algorithms in this thesis.

Kim *et al* (2007b) compared and analyzed GAs with two different genotype encoding approaches, i.e. the binary link state (BLS) and the block transmission state (BTS), and their associated genetic operators. Compared with the BLS encoding, the BTS encoding has smaller search space, however, at the cost of losing information which would be useful to guide the search towards the optimal solution(s). Currently, BLS encoding is the most popular for NCRM problem. Please see Subsection 3.2.4 for detailed description of BLS and BTS encoding approaches.

In addition, Ahn (2011) and Luong *et al* (2012) studied the NCRM problem using evolutionary approaches, where entropy-based evaluation relaxation techniques were introduced to EAs in order to reduce the computational cost incurred during the evolution. By making use of the inherent randomness feature of the individuals, the proposed EAs can rapidly recognize promising solutions with much fewer individuals needed to be evaluated.

Xing *et al* (2010) proposed a quantum-inspired evolutionary algorithm (QEA) for solving the NCRM problem. As reviewed in Subsection 2.2.3, QEA maintains a population of quantum-bit encoding individuals, each representing a probabilistic distribution model over the search space. When an individual is sampled, each solution in the search space has a certain probability to appear. Quantum rotation gate and quantum NOT gate are used to update the probabilistic distribution of each individual in such a way that promising solutions have increasingly more chance to appear. There are two important parameters to update an individual, the rotation angle step (RAS) for the quantum rotation gate and the quantum mutation probability (QMP) for the quantum NOT gate. The RAS value is randomly generated and QMP is adjusted according to the

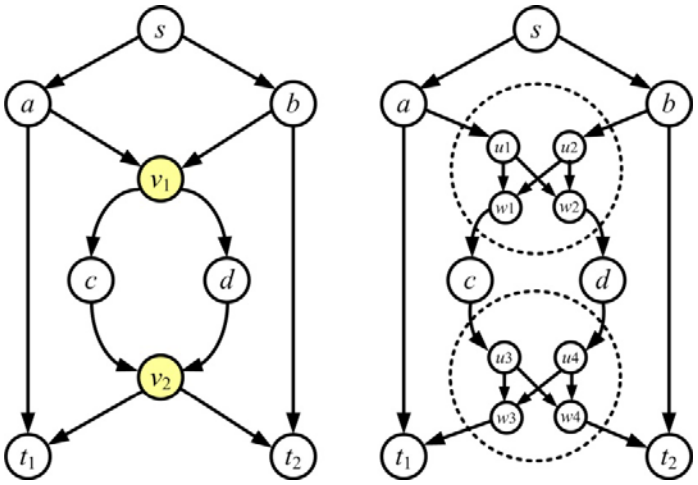
current fitness of an individual. Ji and Xing (2011) developed another QEA to minimize the amount of coding operations. Different from the QEA above, the new QEA adaptively tunes the RAS and QMP values according to both the current and previous fitness values of an individual. These two QEAs were reported to be superior over GA-based algorithms in a number of aspects such as fast convergence. However, as observed in the thesis, the evolution of the two QEAs is sometimes time-consuming and sensitive to different instances (see Chapter 4).

3.4 The Graph Decomposition Method

As mentioned in Subsection 3.1.2, merging nodes within network G are of particular importance as they are allowed to perform coding if necessary. To explicitly show all possible ways of how information flows passing through a particular merging node, the graph decomposition method is used to decompose each merging node in G into a number of auxiliary nodes (Kim *et al*, 2007a, 2007b). As a result, a decomposed graph G_D is created. The following describes the decomposition procedure.

For the i-th merging node, let $In(i)$ be the number of incoming links and $Out(i)$ be the number of outgoing links, respectively. The original merging node is decomposed into two sets of nodes: (1) $In(i)$ nodes, $u_1,\ldots, u_{In(i)}$, referred to as *incoming auxiliary nodes*, and (2) $Out(i)$ nodes, $w_1,\ldots, w_{Out(i)}$, referred to as *outgoing auxiliary nodes*. The j-th incoming link of the i-th original merging node is redirected to node u_j; and the k-th outgoing link of the i-th merging node is redirected to node w_k. Besides, a directed link $e(u_j, w_k)$ is inserted between u_j and w_k, $j = 1,\ldots, In(i)$, $k = 1,\ldots, Out(i)$. We hereafter call the inserted links as *auxiliary links*.

The graph decomposition method has been adopted by GAs (Kim *et al*, 2007a, 2007b) and QEAs (Xing *et al*, 2010; Ji and Xing, 2011) and is also employed in this thesis. Fig.3.3 shows how merging nodes in a graph are decomposed. Fig.3.3(a) is the original graph with source s and receivers t_1 and t_2, and v_1 and v_2 are two merging nodes. After graph decomposition, the decomposed graph is shown in Fig.3.3(b), where four auxiliary (newly inserted) links are inserted into each merging node, helping to show every possible route that the information may pass through v_1 and v_2.



(a) (b)

Figure 3. 3 An example of graph decomposition. (*a*) Original graph. (*b*) Decomposed graph.

3.5 Chromosome Representation and Evaluation

As reviewed in Section 3.3, binary link state (BLS) and block transmission state (BTS) are existing encoding approaches in the literature for the problem concerned. They are based on the graph decomposition method (Kim *et al*, 2007a, 2007b).

For an arbitrary merging node with $In(i)$ incoming links and $Out(i)$ outgoing links, there are $In(i)$ auxiliary links heading to each outgoing auxiliary node after graph decomposition, e.g. links $u_1 \rightarrow w_1$ and $u_2 \rightarrow w_1$ connect w_1 and links $u_1 \rightarrow w_2$ and $u_2 \rightarrow w_2$ connect w_2, as shown in Fig.3.3(*b*). There are two possible states for each auxiliary link, i.e. active or inactive. An active auxiliary link allows flow(s) to pass it while an inactive one does not. The state of each auxiliary link is associated with a one-bit binary variable. In this way, the states of the $In(i)$ auxiliary links heading to each outgoing auxiliary node can be determined by a $In(i)$-bit binary vector. Value '1' at a bit means the corresponding auxiliary link is active in G_D, and value '0' otherwise. For the above merging node, $Out(i)$ binary vectors (each with $In(i)$ bits) are enough to determine the states of all auxiliary links within the node. Note that there is no interdependency among the states of auxiliary links. The following reviews BLS and BTS encoding approaches and the chromosome

evaluation process.

3.5.1 BLS Encoding

Assume there are n_{OAN} outgoing auxiliary nodes in a decomposed graph G_D, where n_{OAN} is an integer. Specifically, n_{OAN} is the summation of $Out(i)$, $i = 1,\ldots,n_M$, the number of the outgoing links of the i-th merging node in original G, where n_M is the number of merging nodes in G (see Subsection 3.1.3 for details). In BLS encoding, an individual (solution) **X** consists of a number of binary vectors bv_i, $i = 1,\ldots,n_{OAN}$, each determining the states of the auxiliary links heading to a certain outgoing auxiliary node in G_D. BLS-based bv_i (e.g. of $In(i)$ bits) has $2^{In(i)}$ possible states from 0…00 to 1…11. Each chromosome corresponds to an explicit secondary graph G_S which may or may not support a valid network coding based multicast routing solution. BLS encoding is similar to the use of decision vectors (see Subsection 3.1.2).

Take Figure 3.3 as an example, a BLS-based chromosome and its corresponding secondary graph are shown in Figure 3.4. There are 8 auxiliary links in the decomposed graph. We associate each auxiliary link with a binary value (1 or 0) to determine if the link is active or inactive. Hence an 8-bit binary string is able to represent the states of all auxiliary links in the decomposed graph. Note that there are 4 auxiliary outgoing nodes, namely w_1, w_2, w_3 and w_4. Hence all auxiliary links are divided into 4 groups, where those heading to the same auxiliary node are in the same group. So, in this example, a chromosome is composed of four binary vectors, i.e. bv_1, bv_2, bv_3, and bv_4, each determining the states of two auxiliary links heading to an auxiliary node, e.g. bv_1 controls the states of e_1 and e_2. Chromosome 11010110 corresponds to a unique secondary graph G_S. In this thesis, BLS encoding is adopted by the majority of the proposed EAs (see Chapter 4, 5 and 6 for details).

3.5.2 BTS Encoding

In the BTS encoding, the chromosome representation is the same as that in the BLS encoding. However, for each vector bv_i in BTS based chromosome, once there are at least two 1's in bv_i, the remaining 0's in bv_i are replaced with 1's (Kim *et al*, 2007a,

2007b). In this way, BTS-based bv_i (e.g. of $In(i)$ bits) has $In(i)$ + 2 possible states, i.e. 0…00, 1…00, 0…10, 0…01, and 1…11. Since each merging node has at least two incoming links, i.e. $In(i) \geq 2$, we always have $In(i)$ +2 $\leq 2^{In(i)}$. Hence, for the same instance of the minimization problem, the size of the search space of BTS is usually smaller than that of BLS. However, using BTS may lose some guidance information that leads the search to the global optima (Kim *et al*, 2007b).

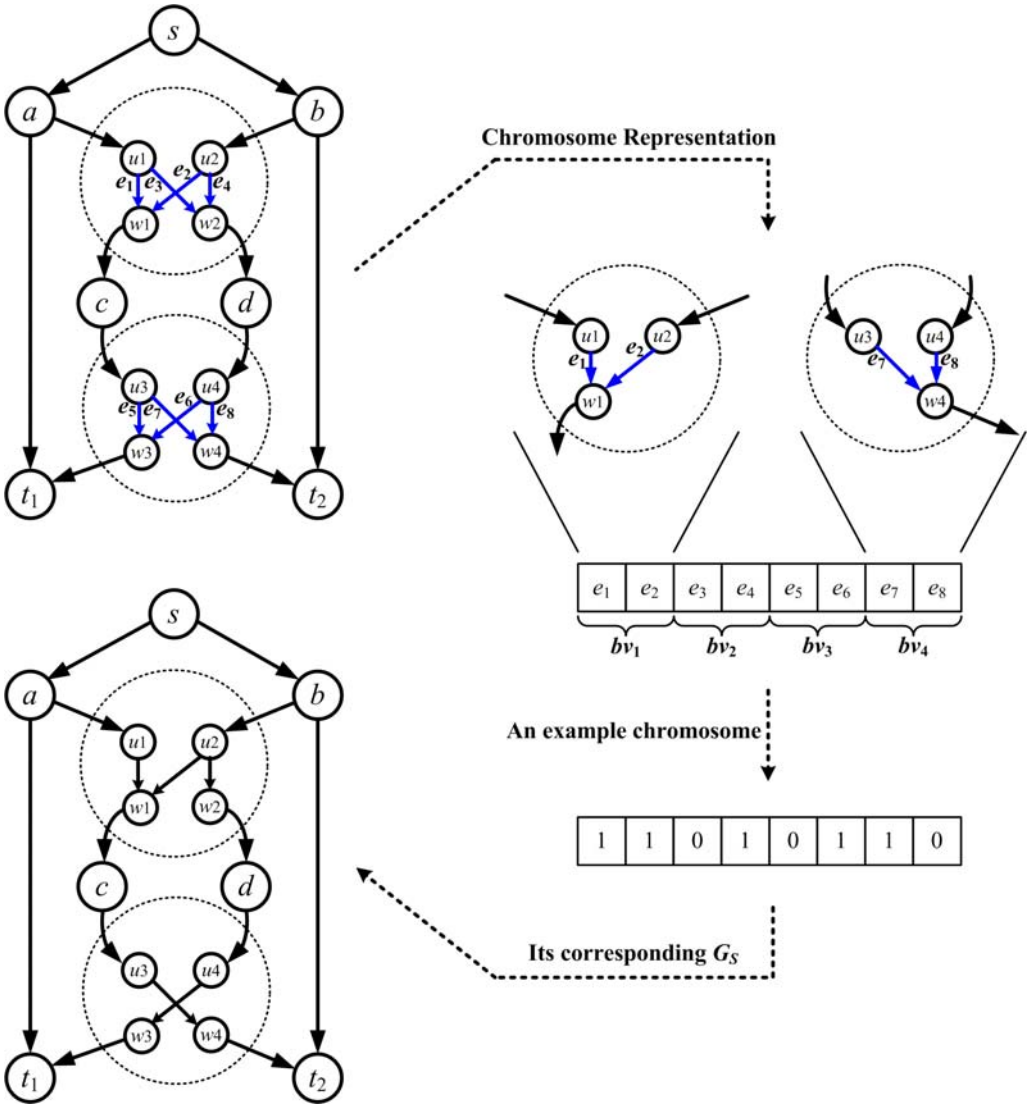


Figure 3. 4 Chromosome representation.

3.5.3 Fitness Evaluation

1. Obtain the corresponding secondary graph G_S of **X**;
   // Check if a NCM subgraph $G_{s\to T}$ can be resulted from G_S
2. Set flag = 1; // value 1 indicates **X** is feasible; value 0 otherwise.
3. **for** k = 1 **to** d **do**
4. Compute the max-flow from source s to receiver t_k;
5. **if** the max-flow is no smaller than R **then**
6. Select R paths from the resulting link-disjoint paths;
7. **else**
8. Set flag = 0 and break the loop; // indicating **X** is infeasible.
9. **if** flag == 1 **then**
10. $G_{s\to T}$ is constructed by all the selected paths;
11. Set $f(\mathbf{X}) = \Phi(G_{s\to T})$;
12. **else**
13. Set $f(\mathbf{X})$ = 50; // 50 is a sufficiently large number
14. Output the fitness value of **X**, i.e. $f(\mathbf{X})$.

Figure 3. 5 The procedure of fitness evaluation (Xing and Qu, 2011a)

BLS and BTS adopt the same evaluation procedure in EAs in the literature. For each individual **X**, first of all, its feasibility is checked. As mentioned above, **X** corresponds to a unique secondary graph G_S. The max-flow between the source s and an arbitrary receiver $t_k \in T$ in G_S is computed, where a number of classical max-flow algorithms can be used such as Ford-Fulkerson algorithm (Ford and Fulkerson, 1956) and Goldberg algorithm (Goldberg, 1985). As mentioned in Subsection 3.1.1, each link in G has a unit capacity. The max-flow between s and t_k is thus equivalent to the number of link-disjoint paths between s and t_k found by the max-flow algorithm. If all d max-flows are at least R, where d is the number of receivers, rate R is achievable and individual **X** is feasible. Otherwise, **X** is infeasible. If **X** is infeasible, its fitness $f(\mathbf{X})$ is assigned a sufficiently large value (50 in the thesis); otherwise, for each receiver $t_k \in T$, R paths are selected out of the link-disjoint paths from s to t_k obtained by max-flow algorithm (if the max-flow is R then all the link-disjoint paths are selected). Therefore, $R \cdot d$ paths in total, i.e. $P_i(s, t_k)$, i = 1,…,R, k = 1,…,d are obtained. The NCM subgraph $G_{s\to T}$ is composed of these $R \cdot d$

paths and $f(\mathbf{X})$ is set to the number of coding links in $G_{s\to T}$, i.e. $\Phi(G_{s\to T})$. The procedure of fitness evaluation is shown in Figure 3.5.

3.6 Benchmark Instances

As reviewed in Subsection 3.3, the NCRM problem is still emerging topic. Hence, only a few benchmark instances have been built. Among these instances, *n*-copy is a series of commonly used test networks (Kim *et al*, 2006, 2007a).

3.6.1 *n*-copy networks

Fig.3.5 illustrates an example of *n*-copy networks, where Fig.3.5(*b*) is a 3-copy network constructed by cascading 3 copies of the original network in Fig.3.5(*a*). As for *n*-copy network, the source is the node on the top and the receivers are at the bottom, e.g. *s* is the source and t_i, $i = 1\ldots4$, is the *i*-th receiver in the 3-copy network. The *n*-copy network has $n + 1$ receivers to which data rate from the source is 2. In general, 3-copy, 7-copy, 15-copy and 31-copy networks are of the most popularity. More information can be found at `http://www.cs.nott.ac.uk/~rxq/benchmarks.htm`.

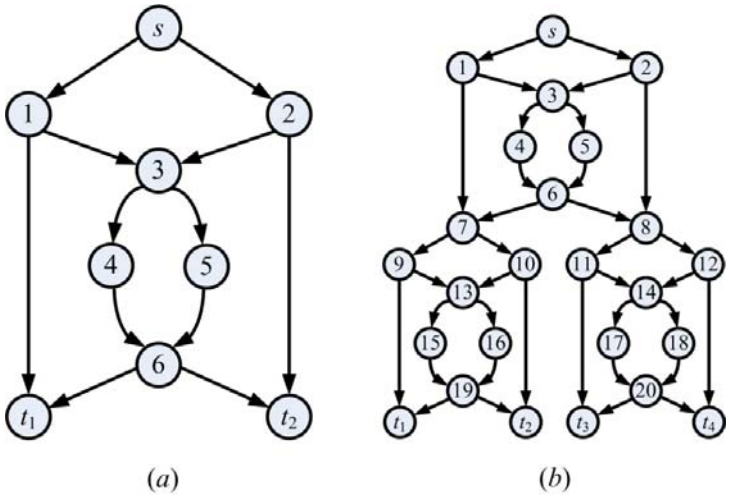


Figure 3. 6 An example of *n*-copy network. (*a*) original network. (*b*) 3-copy.

When n increases, n-copy networks grow exponentially in size, hence being able to represent different network and multicast scales. However, n-copy networks have similar topological structures, which may not reflect the performance of the proposed algorithms in other network topologies. This motivates us to develop random benchmark networks which may help to well estimate the overall performance of those algorithms in arbitrary networks.

3.6.2 Randomly Generated Networks

We construct ten randomly generated networks and corresponding NCM instances for performance evaluation. As described in Section 3.1, the underlying network model for the NCRM problem is directed acyclic graph. Let n_N and n_L be the number of nodes and links, respectively. By using the procedure provided in MATLAB software, we generate random network topologies as shown in Figure 3.6.

1. Randomly generate n_N nodes;
2. **Repeat do**
3. Randomly select two different nodes (v_i, v_k) from the n_N nodes;
4. Insert a directed link from v_i to v_k;
5. **If** the current graph is not directed acyclic graph **then**
6. Delete the link from v_i to v_k;
7. **Until** the number of links is equal to n_L
8. **If** the graph is connected **then**
9. Output the network topology;

Figure 3. 7 Procedure of generating random directed acyclic networks

Note that we may manually insert a small amount of directed links to the resulting topology to form more merging nodes if necessary. Then, we randomly determine a single source node and a number of receivers to constitute a multicast instance.

3.6.3 Description of Instances

Table 3.2 describes the 14 benchmark instances used (i.e. four fixed and ten random networks) in this thesis for the NCRM problem. Note that for BLS encoding, the number of auxiliary links is equal to the encoding length of chromosomes (see Section 3.5.1). For *n*-copy networks, the network size is from 25 to 249 while the receiver number is from 4 to 32 (see rows *3-copy* to *31-copy*). These networks were reported useful for verifying the performance of search algorithms on different network and multicast scales (Kim *et al*, 2007a, 2007b). As the encoding length reflects the size of the underlying search space and a larger search space would be more difficult for search algorithms to handle. With the number of copies *n* increasing, the search difficulity of algorithms is growing (Kim *et al*, 2007a). However, *n*-copy networks are similar with each other in terms of the network structure as they are composed of a number of copies of the same original network (as seen in Figure 3.6). The similarity of *n*-copy networks may not reflect the performance of EAs on networks with different structures. This is why we generate a number of random networks as a complement to the existing benchmark problems. The size of the random networks is set from 20 to 60, appropriate to represent different network scales in real communications networks (Xu, 2011). For each network size, we generate two instances with different network structures. This would also help to test the robustness of testing algorithms. In addition, the encoding length (see column *auxiliary links*) is between 32 and 368, indicating the instances covering a relatively wide range of problem sizes, namely from 2^{32} to 2^{368}. This could also help to test the performance of search algorithms on instances with different problem difficulties.

Detailed information of all instances is available at the website address below:

`http://www.cs.nott.ac.uk/~rxq/benchmarks.htm`.

Table 3. 2 Benchmark Instances for the NCRM problem

Networks	Original network G				Decomposed graph G_D		
	nodes	links	receivers	data rate	nodes	links	auxiliary links
3-copy	25	36	4	2	49	68	32
7-copy	57	84	8	2	117	164	80
15-copy	121	180	16	2	253	356	176
31-copy	249	372	32	2	617	740	368
Randomly Generated Instances	20	37	5	3	54	81	43
	20	39	5	3	65	89	50
	30	60	6	3	94	146	86
	30	69	6	3	113	181	112
	40	78	9	3	124	184	106
	40	85	9	4	91	149	64
	50	101	8	3	178	246	145
	50	118	10	4	194	307	189
	60	150	11	5	239	385	235
	60	156	10	4	262	453	297

3.7 Summary

This chapter introduces the detailed problem formulation of the NCRM problem, the delay constrained NCRM problem and the CDBO problem, respectively. Besides, the benchmark instances used for the three optimization problems are described.

The NCRM problem is of practical importance when deploying NCM (see Subsection 3.2.1). However, this problem has not received enough attention. Less than ten methods (mainly EAs) have been proposed in the literature. In terms of the coding resource involved, there remains a big gap between the obtained solutions and the optima. This is why the thesis investigates and offers high-performance EAs to provide high quality solutions, as introduced in Chapter 4.

On the other hand, the delay constrained NCRM problem and CDBO problem take into consideration not only the interests of service providers but also those of end users. They are more realistic, however, they have never been considered in the literature. This motivates the development of appropriate EAs used to solve the two problems, as shown in Chapter 5 and 6, respectively.

Chapter 4 EAs for the Network Coding Resource Minimization Problem

The network coding resource minimization (NCRM) problem is difficult to solve due to its NP-Hardness (Kim *et al*, 2006). As reviewed in Section 3.3, a number of search methods were developed to address the problem, including linear programming (Bhattad *et al*, 2005), greedy approaches (Fragouli and Soljanin, 2006; Langberg *et al*, 2006), and EAs (Kim *et al*, 2006, 2007a, 2007b; Xing *et al*, 2010; Ji and Xing, 2011). Linear programming is only suitable for small multicast groups as the computational complexity would increase significantly with the number of receivers. The performance of the greedy approaches is largely dependant on the link traversal orders. As the orders are randomly generated, their performance is not reliable. EAs do not suffer the problems that the above two techniques encounter. And EAs have been reported to outperform them with respect to the overall performance, e.g. GA developed by Kim *et al* (2006). However, the existing EAs cannot adapt well for the NCRM problem. As we observed, there was a big gap between the results obtained by these EAs and the global optima (Xing and Qu, 2011a, 2012). Also, some of them were time consuming and sometimes sensitive to instances (e.g. in some instances the resulting solutions were far worse than the best ones) (Xing and Qu, 2011a, 2012). The NCRM problem is a routing problem in the communications networks. Once a NCM request is generated, it is important to find a NCM subgraph for data transmission as soon as possible (Zhang *et al*, 2009). The drawbacks of the existing EAs motivate us to develop more effective and faster algorithms to optimize the NCRM problem, which will assist the practical deployment of NCM for supporting multicast applications.

In this chapter, we propose three EAs for solving the NCRM problem, emphasizing techniques such as estimation of distribution algorithms (see Section 2.2.3), memetic algorithms (see Section 2.2.2) and specifialized chromosome representation. The following briefly describes our work and contributions in this section:

- A population based incremental learning (PBIL) is adapted for the NCRM problem (see Section 4.1). An entropy-based restart scheme is proposed for the first time to improve the global exploration ability of PBIL.
- A compact genetic algorithm (cGA) is adapted for the above problem (see Section 4.2). Three new schemes are developed for enhancing the overall performance of cGA, including the use of all-one vector, a specialized probability vector restart scheme and a problem specific local search operator.
- A path-oriented encoding is adapted to represent the solutions to the NCRM problem and an EA based on the new encoding (called pEA) is developed (see Section 4.3). Algorithmic components such as initialization, crossover, mutation and a local search procedure are designed in compliance with the features of the new encoding.

The following introduces the new EAs and evaluates their performance via a large set of systematic experiments and analysis.

4.1 Population Based Incremental Learning

PBIL is a combination of GA and machine learning. It has many applications (see Section 2.2.3). Different from GAs maintaining a population of explicit solutions, PBIL evolves by operating a real-valued probability vector, thus yields a much lower memory requirement. With no complex genetic operator such as crossover, PBIL incurs much less computational cost. Besides, PBIL was reported to perform well on problems with no or light interdependencies among variables (Pelikan *et al*, 2002). This also matches the NCRM problem where there is no variable-dependency in BLS-based and BTS-based encodings (see Section 3.5).

This section presents an effective PBIL to minimize the amount of coding operations required. In this algorithm, a restart scheme is for the first time proposed to prevent the search from prematurity. The outcomes of this work have been reported in Xing and Qu (2011a).

4.1.1 Algorithm Design

PBIL maintains a real-valued probability vector which, when sampled, generates promising solutions with higher probability (Baluja, 1994). Let $\boldsymbol{PV}(t) = \{p_1^t, p_2^t, \ldots, p_L^t\}$ be the probability vector at generation t that generates binary solutions, where p_i^t is the probability of generating '1' at the i-th position and L is the solution length. At each generation, $\boldsymbol{PV}(t)$ is sampled N times to obtain N samples (i.e. solutions). Among them, high-quality samples are selected and their statistic information is used to adjust $\boldsymbol{PV}(t)$ for further evolution. Initially, the value of p_i^t is set to 0.5, $i = 1, \ldots, L$, and the first N samples are randomly created from the solution space. As search progresses, $\boldsymbol{PV}(t)$ is gradually shifted towards an explicit solution, i.e. p_i^t, $i = 1, \ldots, L$, is approaching to either 0.0 or 1.0. The description of basic PBIL can be found in Subsection 2.2.3.

4.1.1.1 Probability Vector Update Scheme

There are two probability vector update schemes in the literature. One is to use a single solution to update $\boldsymbol{PV}(t)$ (Baluja, 1994; Yang and Yao, 2005, 2008). $\boldsymbol{PV}(t)$ is learnt from the best solution in the samples at each generation. This scheme is easy to implement and could converge rapidly to an explicit solution. However, it may easily mislead the search to local optima unless the best solution is getting closer and closer to the global optimum. This is because the contribution to update the probability vector is dependant on the best solution and the diversity is lost rapidly.

The other scheme is to use a set of promising solutions selected from the samples to update $\boldsymbol{PV}(t)$, which is referred to as Hebbian-inspired rule (Gonzalez *et al*, 2000). This scheme limits the contribution of any single solution to $\boldsymbol{PV}(t)$, which helps to prevent $\boldsymbol{PV}(t)$ converging rapidly to local optima and thus to maintain an appropriate level of diversity. This scheme can be seen as an extension of the first scheme. The proposed PBIL adopts Hebbian-inspired rule to update the probability vector, as described below.

At generation t, the best ever found solution $\boldsymbol{B}(t\text{-}1)$ is inserted into the N obtained samples. The $N + 1$ solutions are then sorted by their fitness and the μ ($\mu \leq N + 1$) best solutions, $\mathbf{Y}_{1:N+1}, \ldots, \mathbf{Y}_{\mu:N+1}$, are selected. The statistic information of the μ solutions, i.e. $\boldsymbol{PV}_{select}$, is extracted and used to modify $\boldsymbol{PV}(t)$. Note that $\boldsymbol{PV}(t)$ produces feasible

solutions as well as infeasible ones (Kim *et al*, 2007a, 2007b). Let $\zeta(t)$ be the number of feasible solutions among the $N + 1$ solutions at generation t, $\mu(t)$ be the number of selected solutions at generation t, and α be the learning rate. The update scheme at generation t is as follows:

$$\boldsymbol{PV}(t) = (1.0 - \alpha) \cdot \boldsymbol{PV}(t) + \alpha \cdot \boldsymbol{PV}_{select} \tag{4.1}$$

$$\boldsymbol{PV}_{select} = \frac{1}{\mu(t)} \cdot \sum_{k=1}^{\mu(t)} \mathbf{Y}_{k:N+1} \tag{4.2}$$

$$\mu(t) = \begin{cases} \tau \cdot N, & if\ \zeta(t) \geq \tau \cdot N \\ \zeta(t), & otherwise \end{cases} \tag{4.3}$$

where τ indicates the maximum proportion of solutions to be selected. In our PBIL, τ = 20%.

4.1.1.2 Restart Scheme

During the search, $\boldsymbol{PV}(t)$ gradually converges to an explicit solution. As we know, $\boldsymbol{PV}(t)$ is a string of probabilities in nature. With the convergence of $\boldsymbol{PV}(t)$, its degree of uncertainty decreases. For $\boldsymbol{PV}(t)$, the degree of uncertainty to some extent reflects which stage the evolution is in (Pang *et al*, 2006). For example, in early stage of the evolution, the degree of uncertainty is high (e.g. the sampled solutions are diversified) and at the end of the evolution the degree is low (e.g. the sampled solutions have high similarity). If $\boldsymbol{PV}(t)$ is shifted to a local optimum and the degree of the uncertainty of $\boldsymbol{PV}(t)$ is lower enough, the search will be stuck to this local optimum. This encourages the idea to restart the search when the degree of uncertainty of $\boldsymbol{PV}(t)$ is lower than a threshold value.

In information theory, entropy is used to measure the degree of uncertainty (Ihara, 1993). Pang *et al* (2006) proposed a PBIL to solve flow shop and job shop scheduling problems, where learning rate and mutation probability are adaptively adjusted according to the entropy value of $\boldsymbol{PV}(t)$ at each generation. Inspired by Pang *et al* (2006), we use entropy to represent the degree of uncertainty. In our work, it is important to determine an appropriate threshold value of the entropy since this value tells PBIL when to restart.

The entropy value is expressed in equation (4.4), where $p_i^{(x=k)}$ denotes the probability of generating 'x' at the i-th position, $p_i^{(x=0)} = 1 - p_i^t$ and $p_i^{(x=1)} = p_i^t$ (Ihara, 1993).

$$E = -\sum_{i=1}^{L} \sum_{k=0}^{1} p_i^{(x=k)} \cdot \ln\left(p_i^{(x=k)}\right) \tag{4.4}$$

We can see the entropy value is determined by the string length L and the probability $p_i^{(x=k)}$ at each position. This means the entropy value may greatly vary with the problem size. If we use the entropy value as an indicator of the uncertainty, we need to assign an appropriate threshold value for each problem instance, which will be difficult and meaningless. In order to reduce the impact of the problem size on the entropy value, we introduce the average entropy per bit of $\boldsymbol{PV}(t)$, E_P, to normalize the uncertainty, as shown below. The average entropy makes it possible to determine a single threshold value for all instances.

$$E_P = -\frac{1}{L} \sum_{i=1}^{L} \sum_{k=0}^{1} p_i^{(x=k)} \cdot \ln\left(p_i^{(x=k)}\right) \tag{4.5}$$

Whilst $\boldsymbol{PV}(t)$ converges to an explicit solution during the evolution, E_P decreases and approaches to 0.00. Through empirical experiments we observed that $\boldsymbol{PV}(t)$ could hardly find a better solution when p_i^t was either higher than 0.97 or lower than 0.03. The critical value of E_p was thus calculated as 0.134, i.e. when each p_i^t was either 0.97 or 0.03. Let Θ denote the threshold value of E_P. In experiments, we set $\Theta = 0.14$. If $E_P < \Theta$, PBIL reinitializes $\boldsymbol{PV}(t)$ as (0.5,…,0.5).

4.1.1.3 The Procedure of the Proposed PBIL

The procedure of the proposed PBIL is shown in Figure 4.1. In the PBIL framework, we represent the problem being concerned by using the graph decomposition method (see Section 3.4). Binary link state (BLS) encoding is used as the chromosome representation scheme. Detailed description of BLS encoding and evaluation can be found in Section 3.5.1. In the fitness evaluation (see step 7), each infeasible solution is assigned a sufficiently large value (50 in our PBIL). If a solution is feasible, i.e. it corresponds to a valid NCM subgraph, its fitness value is assigned the number of coding links in the NCM

subgraph. To ensure that the algorithm begins with at least one feasible solution in the population, the initial best solution is set as an all-one vector where all merging nodes are active (Kim *et al*, 2007a, 2007b). The termination criteria are subject to two conditions: (1) a NCM subgraph without coding is obtained, or (2) the algorithm reaches a pre-defined number of generations.

1. Set generation $t = 0$
2. **For** $i = 1$ **to** L **do** set $p_i^t = 0.5$;
3. Set $\boldsymbol{B}(t)$ as an all-one vector and evaluate $\boldsymbol{B}(t)$;
4. Generate N samples from $\boldsymbol{PV}(t)$;
5. **repeat**
6. Set $t = t + 1$;
7. Evaluate the N samples;
8. Select $\mu(t)$ solutions from the N samples and $\boldsymbol{B}(t\text{-}1)$, the best solution so far;
9. Calculate $\boldsymbol{PV}_{select}$ by using Equation (4.2);
10. Update $\boldsymbol{PV}(t)$ by using Equation (4.1);
11. **If** $E_P < \Theta$ **then** // Θ is the threshold value for restart scheme
12. **for** $i = 1$ **to** L **do** reset $P_i = 0.5$;
13. Reset $\boldsymbol{B}(t)$ as an all-one vector and evaluate $\boldsymbol{B}(t)$;
14. Generate N samples from $\boldsymbol{PV}(t)$;
15. **Else**
16. Generate N samples from $\boldsymbol{PV}(t)$;
17. **until** *termination condition is met*

Figure 4. 1 Procedure of the Proposed PBIL (Xing and Qu, 2011a)

4.1.2 Performance Evaluation

To verify the effectiveness of the proposed PBIL, we compare two PBILs with three existing EAs regarding a number of evaluation criteria.

- GA-1: GA with binary link state (BLS) encoding and operators (Kim *et al*, 2007b). The description of BLS encoding can be found in Section 3.5.1.
- GA-2: GA with block transmission state (BTS) encoding and operators (Kim *et al*, 2007b). BTS encoding is described in Subsection 3.5.2.

- QEA: a quantum inspired evolutionary algorithm for solving the NCRM problem (Xing *et al*, 2010). Details can be found in Subsection 3.3.
- PBIL-1: the proposed PBIL without restart scheme.
- PBIL-2: the proposed PBIL with restart scheme.

Simulations have been carried out upon two fixed and six random networks (see Subsection 3.6 for details), where optimal solutions yield NCM subgraphs which do not require coding operations (also called coding-free NCM subgraphs). For all algorithms, the population size and pre-defined termination generation are set to 40 and 500, respectively. Their parameter settings are shown in Table 4.1. By running each algorithm 20 times, we collected the successful ratio of finding a coding-free NCM subgraph (SR), the mean and standard deviation (SD) of the best fitness values obtained, and the average computational time (ACT), as shown in Table 4.2. As the NCRM problem is a minimization problem, a lower fitness value indicates a fitter solution. All simulations were run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM.

Table 4. 1 Parameter Settings of the Five Algorithms

GA-1	GA-2	QEA	PBIL-1	PBIL-2
$p_c = 0.8$; $p_m = 0.006$; *tsize* = 4;	$p_c = 0.8$; $p_m = 0.012$; *tsize* = 12;	RAD = 0.1π;	$\alpha = 0.1$; $\tau = 0.2$;	$\alpha = 0.1$; $\tau = 0.2$; $\Theta = 0.14$

Note: p_c: crossover probability; p_m: mutation probability; t_{size}: tournament size; RAD: a constant number to calculate the rotation angle step.

It can be seen from Table 4.2 that in terms of the SR, mean and ACT, the two PBILs are the best algorithms, as discussed below. Looking at SR in each instance, The PBILs have higher values, e.g. in 15-copy instance, PBILs obtain 75% and 100% while the rest of the algorithms obtain 10%, 5% and 0%, respectively (see column *SR* in 15-copy). Also, if we compare the SR of PBILs in different instances, we can find that PBILs (especially PBIL-2) has consistent performance (see rows *PBIL-1*, *PBIL-2*, columns *SR*). The above analysis indicates the proposed PBILs are better in global exploration as they are more likely to find global optima in different instances than the others. If we compare the SR of QEA in different instances, we can find that QEA's performance is not stable (see

rows *QEA*, columns *SR* in each instance). This helps to reflect the sensitivity of QEA to instances.

Table 4. 2 Numerical Results (Best Results are in Bold) (Xing and Qu, 2011a)

Algorithm	7-copy 57nodes, 84links, 8sinks, $R = 2$				15-copy 121nodes, 180links, 16sinks, $R = 2$			
	SR (%)	Mean	SD	ACT (SEC.)	SR (%)	Mean	SD	ACT (SEC.)
GA-1	90	0.1	0.3	24.1	10	2.6	1.4	209.4
GA-2	95	0.05	0.2	35.7	5	2.2	1.5	213.2
QEA	50	1.2	2.4	45.5	0	7.3	6.6	272.2
PBIL-1	**100**	**0.0**	0.0	4.7	75	0.3	0.5	245.6
PBIL-2	**100**	**0.0**	0.0	**3.5**	**100**	**0.0**	0.0	**70.8**
Algorithm	Random Instance 1 40nodes, 78links, 9sinks, $R = 3$				Random Instance 2 40nodes, 85links, 9sinks, $R = 4$			
	SR (%)	Mean	SD	ACT (SEC.)	SR (%)	Mean	SD	ACT (SEC.)
GA-1	30	0.8	0.6	46.1	100	0.0	0.0	6.6
GA-2	30	0.9	0.7	48.6	100	0.0	0.0	5.3
QEA	55	0.5	0.6	32.0	100	0.0	0.0	1.1
PBIL-1	**100**	**0.0**	0.0	8.5	**100**	**0.0**	0.0	**0.2**
PBIL-2	**100**	**0.0**	0.0	**5.6**	**100**	**0.0**	0.0	0.4
Algorithm	Random Instance 3 50nodes, 101links, 8sinks, $R = 3$				Random Instance 4 50nodes, 118links, 10sinks, $R = 4$			
	SR (%)	Mean	SD	ACT (SEC.)	SR (%)	Mean	SD	ACT (SEC.)
GA-1	15	0.9	0.5	75.3	5	1.9	0.9	113.0
GA-2	25	0.8	0.5	72.6	10	1.7	1.0	108.9
QEA	50	0.5	0.5	55.1	50	0.6	0.6	161.2
PBIL-1	95	0.05	0.2	**25.5**	95	0.05	0.2	49.3
PBIL-2	**100**	**0.0**	0.0	25.6	**100**	**0.0**	0.0	**28.3**
Algorithm	Random Instance 5 60nodes, 150links, 11sinks, $R = 5$				Random Instance 6 60nodes, 156links, 10sinks, $R = 4$			
	SR (%)	Mean	SD	ACT (SEC.)	SR (%)	Mean	SD	ACT (SEC.)
GA-1	0	2.1	1.0	184.7	10	1.4	0.7	224.5
GA-2	10	1.4	0.8	176.9	10	1.2	0.6	188.2
QEA	10	1.6	1.1	204.2	70	0.3	0.4	113.8
PBIL-1	95	0.05	0.2	97.5	**100**	**0.0**	0.0	56.3
PBIL-2	**100**	**0.0**	0.0	**62.2**	**100**	**0.0**	0.0	**38.5**

Mean and SD together can reflect the overall performance of an algorithm. A smaller Mean indicates a better average performance while a smaller SD shows its performance is more consistent. Regarding the results of Mean and SD, one can also see that the PBILs in most cases have smaller Mean and smaller SD (see rows *PBIL-1*, *PBIL-2*, columns *Mean*, *SD* in each instance). Hence, these results illustrate PBILs have better overall performance than the others. On the other hand, the Mean and SD of QEA in

different instances also show the QEA's sensitivity to instances. In addition, it is also obvious that PBILs consume less average computational time in all instances than the others (see rows *PBIL-1*, *PBIL-2*, columns *ACT* in each instance). And in most of the instances, the time saving is substantial, e.g. in 7-copy instance, the ACT is reduced by at least 20 seconds (see rows *PBIL-1*, *PBIL-2*, column *ACT* in 7-copy instance).

The data analysis above indicates that PBIL performs better than existing GAs and QEA. The following explains why PBIL is more suited. All algorithms for comparison use binary encodings, i.e. either BLS or BTS (see Sections 3.5.1 and 3.5.2). In these encodings, a solution controls the states of all auxiliary links of all merging nodes. The detailed description of merging nodes can be found in Section 3.4. As being responsible for switching and diverting traffic, merging nodes are usually placed at key positions in the network (Kim *et al*, 2007a). So, if many merging nodes are disabled, it is highly possible that a feasible NCM subgraph cannot be found due to the violation of data rate requirement. In other words, a solution with many 0's (meaning many auxiliary links are deactivated) is very likely to be infeasible. Therefore, one can imagine that the search space contain many infeasible solutions (see Section 4.3). In contrast, a solution with many 1's is more likely to be feasible. The higher the proportion of 1's, the higher probability the solution is feasible. Instead of being evenly distributed over the search space, feasible solutions are more crowded around the all-one solution (Xing and Qu, 2013). This is why GAs include an all-one solution in the initial population to make sure they start with at least one feasible solution (Kim *et al*, 2007b). With many infeasible solutions in the search space, it may be difficult for EAs to locate feasible solutions (see Section 4.3). This motivates us to develop a novel EA that adopts another chromosome representation for solving the NCRM problem (see Section 4.3 for details).

Typically, in GAs, crossover is in charge of global exploration and explores different areas in the search space while mutation is responsible for local exploitation and avoiding prematurity by introducing small disturbances to solutions to exploit their neighbors (Mitchell, 1996). With too many infeasible solutions in the search space, if crossover fails to effectively locate feasible regions, mutation would not help much. In GA-1 and GA-2, uniform crossover is adopted. Unfortunately, uniform crossover is not good at producing feasible solutions (Kim *et al*, 2007a, 2007b). Hence, for BLS and BTS

encodings, traditional GAs (i.e. GA-1 and GA-2 in our experiments) cannot perform well and obtain promising results.

QEA maintains quantum-bit encoded individuals which are updated by quantum gates (see Section 2.2.3 for details). The performance of QEA is largely dependant on a constant value, called RAD, which is the key parameter in quantum gates (Xing *et al*, 2010). In our experiments, RAD may not be suited for some instances, which indicates the QEA could not well handle the trade-off between intensification and diversification in some instances. Therefore the performance of QEA turns out to vary greatly from instance to instance.

On the other hand, the BLS encoding is fitter for PBIL as explained below. As a combination of GA and machine learning, PBIL implicitly express the population using a probability vector (PV) (Baluja, 1994). The PV is gradually adjusted according to the statistical information extracted from promising solutions. This process shows the continuity of the changes in probability distribution, which helps to make full use of the promising solutions already obtained during the search. Once promising solutions appear in the samples, the PV learns from them (see PV update scheme in Section 4.1.1). It is very likely that the new samples inherit the good genes from them. Besides, the continuity is also reflected on the search smoothly moving from regions to regions. This is more suited to locate feasible regions, with the guidance of promising solutions. Therefore, it makes sense that PBIL performs better than GAs and QEA in the above experiments. The suitability of PBIL implies that EDAs with gradually changing PVs may be also suited for the BLS encoding. The next section introduces another BLS-based EDA for solving the NCRM problem (see Section 4.2).

Then, we compare the performance of PBIL-1 and PBIL-2. Regarding the SR, Mean and SD, PBIL-2 performs better than PBIL-1 in most cases (see rows *PBIL-1*, *PBIL-2*, columns *SR*, *Mean*, *SD* in each instance). The results show that the restart scheme can improve the global exploration ability of the proposed PBIL. The following explains why. As aforementioned, if without the restart scheme, the uncertainty of the PV gradually decreases as the evolution continues. The diversity of the generated samples becomes lower and lower and hence the search is confined in a smaller and smaller area in the search space. The PV would finally converge to an explicit solution. If this solution is not

close to the global optimum and the uncertainty (i.e. entropy in the thesis) of PV is lower enough, the search could not escape from the solution, which would cause prematurity. Therefore, if we restart the search once the uncertainty of PV is lower enough, we could avoid prematurity. Besides, if comparing the ACTs of PBIL-1 and PBIL-2, we see that the restart scheme helps to reduce the computational time (see rows *PBIL-1*, *PBIL-2*, columns *ACT* in each instance). According to the termination conditions, PBIL stops when either it finds a NCM subgraph without coding performed (i.e. a global optimum) or it reaches a predefined number of generations. With the restart scheme, PBIL-2 has better global exploration ability. Hence it consumes less ACT. The above explains why the restart scheme can enhance the overall performance of the proposed algorithm. On the other hand, the performance of the restart scheme largely depends on the threshold value of the average entropy. It is important to assign an appropriate threshold value as this determines when to restart the search. If the value is too large, the search is forced to restart before it converges and carries out effective local exploitation; if the value is too small, it is a waste of time exhaustively exploiting a very small region in the search space. In the thesis, the threshold value is calculated based on the observation of empirical experiments (explained in Restart Scheme, Section 4.1.1). The threshold value of the average entropy may need to be adjusted to adapt for other instances.

Although PBIL-2 is the fastest among the algorithms, one can also see that the ACT of PBIL-2 varies significantly, e.g. it is 3.5 seconds in 7-copy network and 70.8 seconds in 15-copy network, respectively. This makes sense as the ACT usually goes with the problem size and the problem solving difficulty. However, the dramatic change in ACT may restrict the application of the proposed algorithm in real communication networks in the sense that networks need to respond quickly to set up the NCM data transmission once there is a NCM request (Jaggi *et al*, 2005).

4.1.3 Summary of the Algorithm

A PBIL has been adopted for solving the NCRM problem. This work has shown that in nature the binary link state (BLS) encoding is more suitable for PBIL than the existing algorithms. The continuity of the changes in probability distribution helps PBIL to make use of promising solutions found and locate feasible and promising regions in the search

space. This, as discussed above, indicates that EDAs having similar PV update schemes may also match the BLS encoding. In our PBIL, an entropy based restart scheme is for the first time proposed. This scheme helps to avoid prematurity and improve the global exploration ability during the evolution. In the restart scheme, the threshold value of average entropy is an important parameter that affects the effectiveness of the scheme. It should be set properly to gain decent performance. Simulation results demonstrate that the proposed PBIL outperforms a number of existing EAs (including GAs and QEA) with respect to multiple performance evaluation criteria, namely success ratio, mean and standard deviation of the best fitness values and the average computational time. On the other hand, the computational costs to run PBIL may be quite expensive in some instances, which may limit the applicability of PBIL in real communications networks.

This research work has been published at IEEE Communications Letters (Xing and Qu, 2011a).

4.2 Compact Genetic Algorithm

Whenever a NCM request is generated, the network is expected to provide a feasible NCM routing scheme as quickly as possible (Jaggi *et al*, 2005). Hence, when designing NCM routing algorithms, apart from exploration and exploitation abilities, computational time is also an important issue to be considered. As discussed in Section 4.1, PBIL gains decent performance in terms of global exploration. However, it consumes considerable amount of computational time in some cases. Then an interesting question arises: is it possible to develop an algorithm that gains similar or even better performance than PBIL while requires less computational costs? This section answers the question.

As discussed in Section 4.1.2, the BLS encoding is suitable for PBIL because its PV update scheme can make use of the promising solutions to effectively locate and explore feasible regions. EDA with similar PV update scheme may also match the BLS encoding and if so this EDA may also obtain good optimization results. Therefore, regarding the choices of appropriate algorithms, the priority is given to EDAs. As discussed in Section 3.5, there is no interaction among variables in the solution. Apart from PBIL, there are two EDAs that are suitable for addressing problems with no variable-interdependencies,

namely cGA (see Section 2.2.3.2) and UMDA (2.2.3.3). They both maintain a single real-valued PV to implicitly represent the population. On the one hand, cGA generates two samples at each generation. They compete and the winner is used to update the PV. cGA and PBIL have similar PV update schemes where the PV of the previous generation is gradually shifted towards the promising solutions of the current generation. This shows the continuity in the changes of the probability distribution. Besides, cGA was reported to converge to a good solution within short computational time (Gallagher *et al*, 2004). On the other hand, UMDA generates a set of samples at each generation. Promising samples are selected and their statistical information is extracted as the PV. In other words, the PV is fully determined by the promising solutions of the current generation. If these promising solutions are not sufficiently good, the search would be guided to nonpromising regions. As aforementioned, the search space of BLS encoding contains large quantity of infeasible solutions. Once UMDA if misled to infeasible regions, it may be difficult for UMDA to escape. Due to the above analysis, we choose cGA as a candidate for addressing the NCRM problem. Detailed description of standard cGA can be found in Subsection 2.2.3.2.

This section proposes an elitism-based cGA for the NCRM problem. Three novel schemes have been developed, as listed below.

- **The use of an all-one vector.** This vector helps the PV to locate feasible regions in the search space. This scheme not only warrantees that the cGA starts with a feasible solution at the beginning of the evolution but also adjusts the PV in such a way that feasible solutions appear with increasingly higher probabilities (see Section 4.2.1.1).
- **A novel PV restart scheme**. Different from the entropy based restart scheme introduced in PBIL (see Subsection 4.1.1.2), the new PV restart scheme resets the PV whenever the incumbent best solution cannot be improved within a given number of consecutive generations. This scheme stops ineffective evolution and helps to escape from local optima and increase the chance to hit an optimal solution (see Section 4.2.1.2).
- **A novel local search operator**. It is devised to exploit the neighbourhood of each feasible solution so as to enhance the local exploitation of our cGA. As far

as we know, this operator is the first attempt to incorporate domain knowledge to strengthen the local exploitation ability of the proposed cGA (see Section 4.2.1.3).

Simulation experiments have been conducted over a number of fixed and randomly generated NCM scenarios. Results demonstrate that all the adopted schemes are effective and the proposed cGA outperforms existing EAs in obtaining optimal solutions within a reduced computational time.

4.2.1 Algorithm Design

In our cGA, BLS encoding is used to represent chromosomes (see Subsection 3.2.4). The following first introduces the use of all-one vector, the restart scheme and the local search operator, respectively, and then gives the overall structure of the proposed cGA.

4.2.1.1 The Use of an All-one Vector

The NCRM problem is highly constrained (see Subsection 3.2.1) and infeasible solutions form a large proportion of the solution space (experimental results can be found in Subsection 4.3.2). Hence, the PV may not efficiently evolve with a limited number of feasible individuals, i.e. the optimization of cGA could be seriously weakened due to the lack of feasible individuals. Kim *et al* (2007a, 2007b) noticed this problem and inserted an all-one vector, i.e. '11…1', into the initial population to warrantee that their GAs start with at least one feasible individual (the all-one vector ensures that all auxiliary links in G_D are active and guarantees a feasible NCM subgraph to be found). This method shows to improve the optimization performance.

Inspired by the idea of Kim *et al* (2007a, 2007b), we simply set an all-one vector as the elite individual in the initialization to ensure that our cGA begins with at least one feasible individual. It is easy to know that individuals containing more 1s are more likely to be feasible (see Subsection 3.1.2 and 3.2.3). The PV which gradually shifts towards the all-one vector thus gets increasingly higher chance to produce feasible individuals. This method shows to significantly accelerate the convergence speed of the cGA in the early stage of evolution (see Subsection 4.2.2.1 for details).

4.2.1.2 The Probability Vector Restart Scheme

In the literature of EDAs, restart schemes have been introduced into PBILs to avoid premature evolution and enhance the global search capability. The essence of these schemes is to restart (re-initialize) the search under certain circumstances. For example, in applying PBILs to dynamic optimization problems, the PV can be reset as soon as the environment changes (Yang and Yao, 2005, 2008). As described in Section 4.1.1.2, we also propose an entropy based restart scheme. However, the performance of the entropy based scheme is dependent on the threshould value of the average entropy which is calculated based on the observation of empirical experiments (see Section 4.1.2). Ahn and Ramakrishna (2003) present a restart scheme for cGA. In their scheme, the elite solution is regenerated using the initial PV once the current elite solution dominates in a predefined number of generations. Nevertheless, this restart scheme does not fit for the BLS encoding because feasible solutions are not evenly distributed over the search space and the initial PV would not effectively produce feasible solutions (see Section 4.1.2).

Considering the features of BLS encoding and cGA, we design a novel PV restart scheme for our cGA. In the scheme, the current PV is replaced with a previously recorded PV when the current elite solution dominates in a predefined number of generations, namely g_c generations, where g_c is an integer. An appropriate PV to be recorded is a PV that should generate feasible individuals with a high probability so that the cGA retains an effective evolution. In addition, the PV should have an appropriate probability distribution to maintain a good diversity, where the current elite individual has less chance to appear again.

Let $\mathbf{X}_e$ denote the current elite individual. The PV restart scheme is shown below.

1. Record the PV that generates the very first feasible individual during the evolution. For example, if the PV produces this first feasible individual at generation t_i, we record $\boldsymbol{PV}(t_i)$ for the proposed restart scheme. Here, assume that the PV can produce at least one feasible individual during the evolution of cGA. As far as we know, to record PV has not been considered in the literature.
2. After $\boldsymbol{PV}(t_i)$ is recorded, the restart scheme is launched. We set *counter* = 0 at generation t_i. Note that initially the value of *counter* is −1, which implies the restart scheme has not started.

3. If $\mathbf{X}_e$ is not changed in a new generation, set *counter* = *counter* + 1. If *counter* = g_c, which means $\mathbf{X}_e$ stays for g_c consecutive generations, set $\boldsymbol{PV}(t) = \boldsymbol{PV}(t_i)$ and *counter* = 0. Besides, whenever $\mathbf{X}_e$ is changed, set *counter* = 0. The restart condition is the same as the one in the work of Ahn and Ramakrishna (2003).

This scheme shows to effectively improve the global exploration capability of our cGA (see Section 4.2.2.2 for details).

4.2.1.3 The Local Search Operator

As we know, an effective search algorithm can well handle the trade-off between exploration and exploitation during the search. However, EAs with pure evolutionary structure are often not good at the local exploitation (see Section 2.2.1). This is why local search techniques are encouraged to be included in EAs to strengthen their exploitation ability (Whitley, 1995; De Jong, 2005; Moscato *et al*, 2004). On the other hand, at each generation, cGA produces much fewer solutions (at most two) than population-based EAs, as reviewed in Section 2.2.3.2. It is hence meaningful to study how to make full use of each promising solution to improve the overall performance of the cGA. The reasons above motivate us to consider a local search operator to be integrated with the cGA. Unfortunately, all existing EAs for the above problem are based on pure evolutionary framework (see Section 3.3). In other words, no local search enhancement has been studied in the literature.

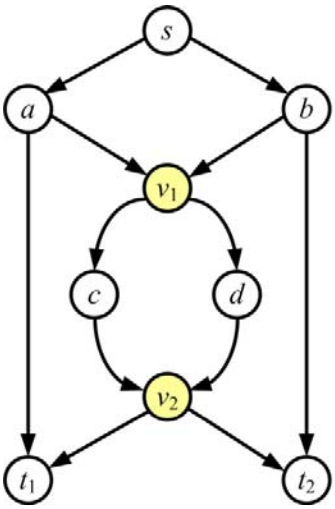
As mentioned in Section 3.5, each feasible individual corresponds to a unique secondary graph G_S based on which a NCM subgraph $G_{s\rightarrow T}$ can be found. However, within the same G_S, the NCM subgraph found may not be unique, i.e. we could possibly find more than one feasible NCM subgraph from the given G_S. We call these feasible NCM subgraphs the neighbours of $G_{s\rightarrow T}$ in G_S. The better the NCM subgraph found (i.e. one with fewer coding operations performed), the higher the quality of the corresponding solution to the problem. This makes it possible to design a local search operator (L-operator) to explore the neighbours of $G_{s\rightarrow T}$ and hopefully find a new NCM subgraph with fewer coding operations involved. This operator makes use of the problem-specific domain knowledge and is used to enhance the local exploitation ability of the proposed cGA.

Assume $G_{s\to T}$ has n_{CN} coding nodes, where the i-th coding node v_i has $In(v_i)$ incoming links. We denote by e_{ik} the k-th incoming link of node v_i in $G_{s\to T}$. Obviously, there is also an identical e_{ik} in G_S since $G_{s\to T} \subseteq G_S$. Let $G_{s\to T}$(inc) and $G_{s\to T}$(new) be the incumbent NCM subgraph and a new NCM subgraph, respectively. The procedure of the L-operator is shown as follows:

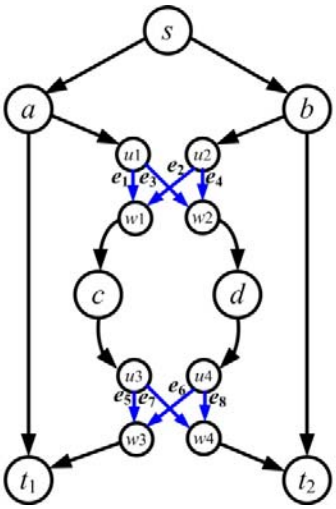
1. Set $G_{s\to T}$(inc) = $G_{s\to T}$; $i = 1$; $k = 1$;
2. Remove link e_{ik} from G_S. Use the max-flow algorithm (Goldberg, 1985) to calculate the max-flow between s and each sink in G_S. If the d max-flows found are at least R, go to step 3. Otherwise, reinsert link e_{ik} to G_S and go to step 4.
3. For each sink t_k, select R paths from the obtained link-disjoint paths from s to t_k (if there are R paths we then select all of them), and map all the selected paths to G_S to obtain a new NCM subgraph $G_{s\to T}$(new). If the number of coding links in $G_{s\to T}$(new) is less than that in $G_{s\to T}$(inc), set $G_{s\to T}$(inc) = $G_{s\to T}$(new); otherwise, reinsert the link e_{ik}.
4. If $k = In(v_i)$, proceed to the next coding node, i.e. set $i = i + 1$ and go to step 5; otherwise, proceed to the next incoming link of the same coding node, i.e. set $k = k + 1$, and go to step 2.
5. If $i = n_{\mathrm{CN}} + 1$, stop the procedure and output $G_{s\to T}$(inc); otherwise, proceed to the first incoming link of the i-th coding node, i.e. set $k = 1$, and go to step 2.

Figure 4.2 shows an example of the local search procedure. The original graph with the source s and receivers t_1 and t_2, as shown in Figure 4.2(a), has two merging nodes, i.e. v_1 and v_2. In Figure 4.2(b), v_1 and v_2 are decomposed into two groups of auxiliary nodes connected by auxiliary links, i.e. e_1, …, e_8 (the graph decomposition method is described in Subsection 3.2.3). We assume the i-th bit of each individual is associated with link e_i, i = 1,…,8 (see Subsection 3.2.4). Given an individual '11101011', its corresponding secondary graph G_S is shown in Figure 4.2 (c). Based on G_S, we find a NCM subgraph $G_{s\to T}$ with only one coding node w_1, as shown in Figure 4.2 (d), where links e_{11} and e_{12} are two incoming links of w_1. In Figure 4.2(e), the L-operator removes e_{11} from G_S, resulting a new G_S. Based on the new G_S, we obtain a better NCM subgraph in Figure 4.2(f), where the subgraph is coding-free (i.e. with no coding operation performed). It can be seen that removing e_{12} from Figure 4.2(e) produces an infeasible NCM subgraph thus

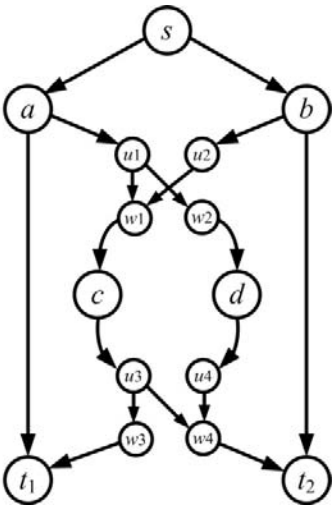
e_{12} remains. Since all incoming links of coding nodes are checked, the L-operator stops and outputs the NCM subgraph shown in Figure 4.2(*f*). In this example, the fitness of individual '11101011' is set to zero.



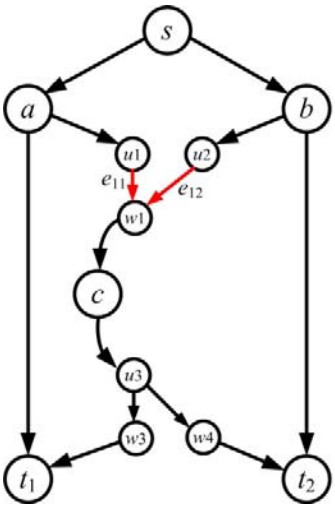
(*a*) Original network *G*



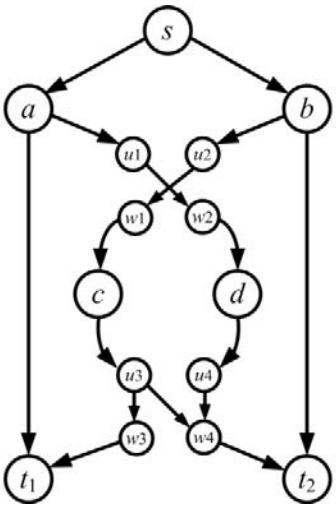
(*b*) Decomposed graph G_D



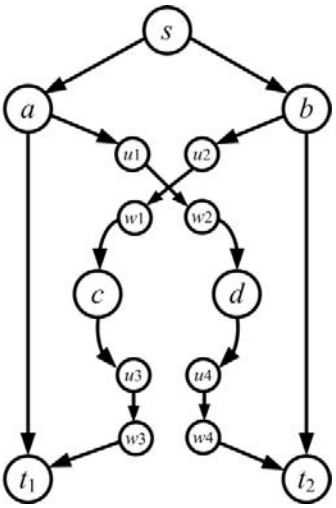
(*c*) Secondary graph G_S of individual '11101011'



(*d*) NCM subgraph $G_{s\to T}$



(*e*) New G_S without link e_1



(*f*) A new $G_{s\to T}$

Figure 4. 2 An example of the local search procedure.

In our cGA, L-operator is used in fitness evaluation to improve the NCM subgraph of each feasible individual **X** (see Section 4.2.1.4). As the first operator exploiting the problem-specific domain knowledge for the NCRM problem, L-operator is effective to improve the quality of solutions obtained by cGA (see Subsection 4.2.2.3).

4.2.1.4 The Overall Procedure of the Proposed cGA

1) **Initialization**
2) Set $t = 0$; *counter* $= -1$;
3) **for** $i = 1$ **to** L **do** set $p_i^t = 0.5$ // initialize PV
4) Set $\mathbf{X}_e = 11\ldots1$; // Initialize the elite individual with an all-one vector
5) $f(\mathbf{X}_e)$ = **evaluate** ($\mathbf{X}_e$); // Evaluate the elite individual
6) **repeat**
7) Set $t = t + 1$;
8) $\mathbf{X}$ = **generate** ($\boldsymbol{PV}(t)$); // Generate a single individual from the PV
9) $f(\mathbf{X})$ = **evaluate** ($\mathbf{X}$); // Evaluate the individual
10) // Record the PV for the restart scheme
if $\mathbf{X}$ *is the very first feasible individual* **then**
Set *counter* $= 0$; $\boldsymbol{PV}_{record} = \boldsymbol{PV}(t)$;
11) // The PV restart scheme
if $f(\mathbf{X}_e) \leq f(\mathbf{X})$ && *counter* ≥ 0 **then**
Set *counter* = *counter* + 1;
if *counter* == g_c **then**
Set $\boldsymbol{PV}(t) = \boldsymbol{PV}_{record}$; *counter* $= 0$;
12) // Record better individuals
if $f(\mathbf{X}_e) > f(\mathbf{X})$ **then**
Set $\mathbf{X}_e = \mathbf{X}$; $f(\mathbf{X}_e) = f(\mathbf{X})$;
if *counter* > 0 **then**
Set *counter* $= 0$;
13) // The PV learns towards the elite individual
for $i = 1$ **to** L **do**
if $\mathbf{X}_e(i) \neq \mathbf{X}(i)$ **then**
if $\mathbf{X}_e(i) == 1$ **then** $p_i^t := p_i^t + 1/N$;
else $p_i^t := p_i^t - 1/N$;
14) **until** *the termination condition is met*

Figure 4. 3 Procedure of the proposed cGA (Xing and Qu, 2012)

The pseudo-code of the proposed cGA is shown in Figure 4.3 with the following three extensions: (1) In the initialization the elite individual is set to an all-one vector (see Section 4.2.1.1); (2) the PV is reset when the elite individual is not changed for a number of consecutive generations (see Section 4.2.1.2); (3) a local search operator is

integrated to exploit the neighbours of each feasible NCM subgraph found during the evolution (see Section 4.2.1.3).

BLS encoding (see Subsection 3.5.1) is used to represent chromosomes in cGA. The elite individual $\mathbf{X}_e$ is set to an all-one vector to ensure our cGA begins with a feasible individual (see Step 4). In steps 5 and 9, a single individual $\mathbf{X}$ is sampled from the PV, and its feasibility is checked (see Section 3.5.3). If $\mathbf{X}$ corresponds to a G_S where a feasible NCM subgraph $G_{s\rightarrow T}$ can be found, we use the L-operator to search the neighbours of $G_{s\rightarrow T}$ and obtain hopefully a new and better NCM subgraph (see Section 4.2.1.3). The number of coding links in the new NCM subgraph is set to the fitness of $\mathbf{X}$, i.e. $f(\mathbf{X})$. Otherwise, a sufficiently large fitness value (50 in the thesis) is set to $f(\mathbf{X})$ for an infeasible solution; If $\mathbf{X}$ is the very first feasible individual during the evolution, we record the PV from which $\mathbf{X}$ is created, and launch the PV restart scheme by setting *counter* = 0 (see Step 10). In step 11, the restart scheme is triggered when no better individual appears within g_c generations. In step 12, we replace the elite individual $\mathbf{X}_e$ with $\mathbf{X}$ if the latter is better. After that, the PV is shifted towards $\mathbf{X}_e$, as seen in step 13. The procedure terminates subject to two conditions: (1) a coding-free NCM subgraph is obtained, or (2) the algorithm reaches a pre-defined number of generations.

4.2.2 Performance Evaluation

We first investigate the effectiveness of the three extensions in cGA, i.e. the use of an all-one vector, the PV restart scheme and the local search operator on two instances, i.e. 3-copy and 20-node. The description of the 3-copy and 20-node networks can be found in Subsection 3.6. We then evaluate the overall performance of the proposed cGA on 11 intances (see Section 4.2.2.4). The increment of each winning allele is set to 0.05 in cGA, which mimics the converging behaviour of a standard GA with population size 20 (= 1/0.05) (Harik *et al*, 1999). All experimental results are collected by running each algorithm 50 times.

4.2.2.1 The Effectiveness of the All-one Vector

As mentioned above, the all-one vector not only enables cGA to begin with a feasible individual but also adjusts the PV to produce more and more feasible individuals.

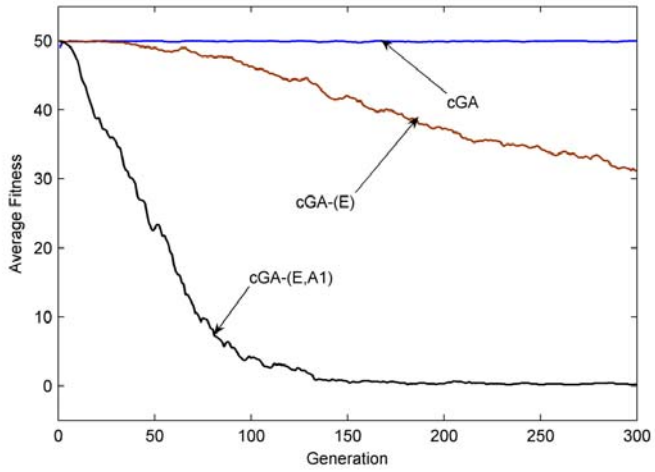
To evaluate the performance of the all-one vector, we compare the following three variants of cGA on the 3-copy and 20-node networks:

- cGA: the standard cGA (see Subsection 2.2.3).
- cGA-(E): cGA with elitism. At each generation, a single solution is generated and competes with the best ever found solution (Ahn and Ramakrishna, 2003).
- cGA-(E,A1): cGA-(E) with the use of the all-one vector.

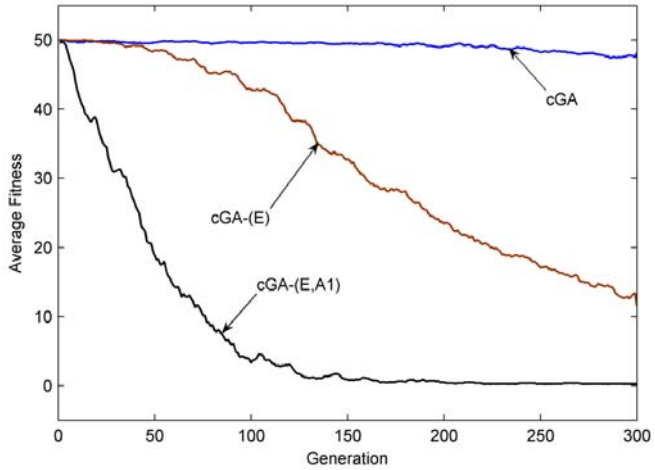
We compare the above algorithms by using the following evaluation criteria:

- The evolution of the average fitness. The termination condition here is a pre-defined number of generations, i.e. algorithms stop after 300 generations.
- The successful ratio (SR) of finding a coding-free NCM subgraph (i.e. one without coding operation performed) in 50 runs.
- The mean and standard deviation (SD) of the best fitness values obtained in 50 runs.
- The average termination generation (ATG) over 50 runs. The termination condition here is a coding-free NCM subgraph has been found or the algorithms have reached 300 generations.

Figure 4.4 shows the comparisons of the three algorithms in terms of the average fitness during the evolution. Compared with cGA-(E) and cGA-(E,A1), cGA is the worst one for both networks. In the 3-copy network, cGA has no evolution. In 20-node network, the convergence of cGA can be hardly observed before the 200th generation. cGA-(E) performs better than cGA on both networks, which once again demonstrates the effectiveness of the elitist scheme over the traditional tournament scheme (Ahn and Ramakrishna, 2003). With the use of the all-one vector, cGA-(E,A1) performs the best with respect to the convergence. For example, cGA-(E,A1) converges to a stable state after the 150-th generation for the 3-copy network and the 200-th generation for the 20-node network, respectively. This is because, with more feasible solutions quickly generated from the PV, cGA-(E,A1) is able to converge much faster to better solutions.



(*a*) 3-copy network



(*b*) 20-node network

Figure 4. 4 Average fitness vs. generations in variants of cGA.

Experimental results of different evaluation criteria for each algorithm are presented in Table 4.3. Similarly, we see that cGA-(E,A1) is the best, with highest SR, and smallest mean and ATG (see column *cGA-(E,A1)* in each instance). The obtained results clearly show that the all-one vector can improve the optimization performance of cGA. This is due to search space features of BLS encoding, namely feasible solutions are not evenly distributed over the search space. Instead, those closer to the all-one vector are more likely to be feasible. Hence, all-one vector helps cGA to locate feasible solutions at the beginning of the evolution and guide the search towards feasible and promising regions

(see Section 4.1.2). Therefore, the all-one vector contributes to an enhanced performance of cGA.

Table 4. 3 Experimental Results (Best results are in bold)

Scenarios	Criteria	cGA	cGA-(E)	cGA-(E,A1)
3-copy	SR (%)	6	24	**86**
	Mean(SD)	40.24(19.72)	28.20(24.84)	**0.14**(0.35)
	ATG	292.16	264.66	**78.78**
20-node	SR (%)	6	38	**78**
	Mean(SD)	20.82(24.08)	8.58(18.27)	**0.22**(0.41)
	ATG	283.94	243.20	**111.86**

Note: SR: successful ratio; ATG: average termination generation; Mean: average best fitness values; SD: standard deviation.

4.2.2.2 The Effectiveness of the PV Restart Scheme

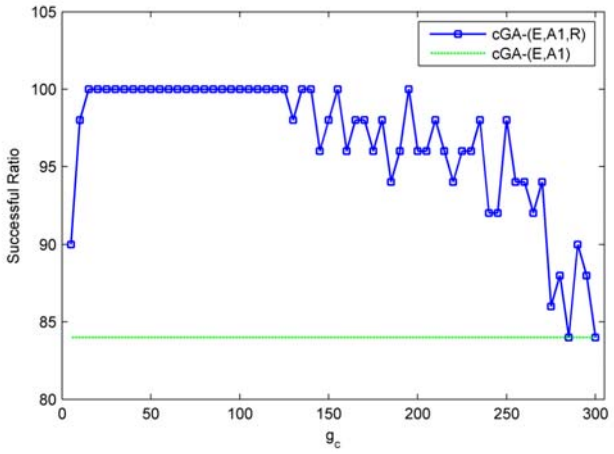
To illustrate the effectiveness of the PV restart scheme, we compare the following two variants of algorithms on the 3-copy and 20-node networks:

- cGA-(E,A1)
- cGA-(E,A1,R): cGA-(E,A1) with the PV restart scheme.

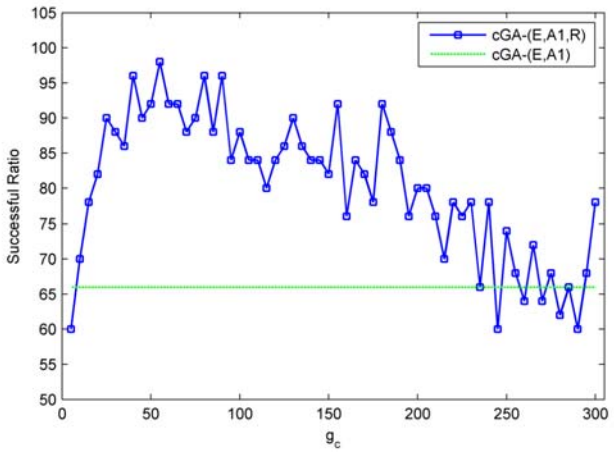
As introduced in Section 4.2.1.2, the PV will be replaced once the elite solution does not change for a predefined number of consecutive generations, denoted by g_c. We study the impact of g_c on the performance of the cGA (see Subsection 4.2.1.2). Note that g_c is not the predefined termination generation. The SR of cGA-(E,A1,R) has been reported with g_c set to 5, 10, 15, …, 295, and 300, respectively (increment of 5 generations). For the sake of fair comparison, we also collect the SR of cGA-(E,A1), by running it 50 times. The SR is 84% for 3-copy network, and 66% for 20-node network, respectively.

Figure 4.5 shows the SRs obtained by cGA-(E,A1,R) with different g_c. With g_c growing, the SR of cGA-(E,A1,R) first increases, and then falls down. As described in Section 4.2.1.2, a previously recorded PV is used to replace the current PV when the restart condition is met. After replacement, the PV is shifted towards the elite solution and promising solutions appear with increasingly higher probabilities. If g_c is too small, the PV is restarted too frequently and the search may not effectively locate promising regions before the next restart; if g_c is too big, the number of the restart times is limited and the search may already get stuck at local optimum awaiting the next restart. Note that the termination generation is 300 in our experiments. If $g_c > 150$, there are at most 2

times to restart the PV. Hence, for both cases (too small g_c or too big g_c), the global exploration ability of cGA is relatively weak. For both networks, the SR of cGA-(E,A1,R) is better than that of cGA-(E,A1) in most cases, e.g. from g_c = 5 to 250 on the 3-copy network and from g_c = 10 to 200 on the 20-node network. We can see that g_c, when properly set, contributes to a better SR for cGA-(E,A1,R). Besides, instead of an exact value, g_c can be selected from a wide range of values, e.g. roughly from 20 to 120 in 3-copy network and from 25 to 110 in 20-node network (see Figure 4.5). Hence, the selection of the value g_c is easier than the threshold value in the entropy-based restart scheme. In the following experiments, we set g_c = 50 in the PV restart scheme since with this value cGA-(E,A1,R) significantly outperforms cGA-(E,A1).



(*a*) 3-copy network



(*b*) 20-node network

Figure 4. 5 SR vs. g_c in variants of cGA.

4.2.2.3 The Effectiveness of the Local Search Operator

As analyzed before, the all-one vector and the PV restart scheme both show to be effective for solving the two testing problems, i.e. the 3-copy and 20-node networks. Here, we evaluate the effectiveness of the L-operator and verify whether the three improvements can be cascaded, by running the following three variants of cGA on the 3-copy and 20-node networks:

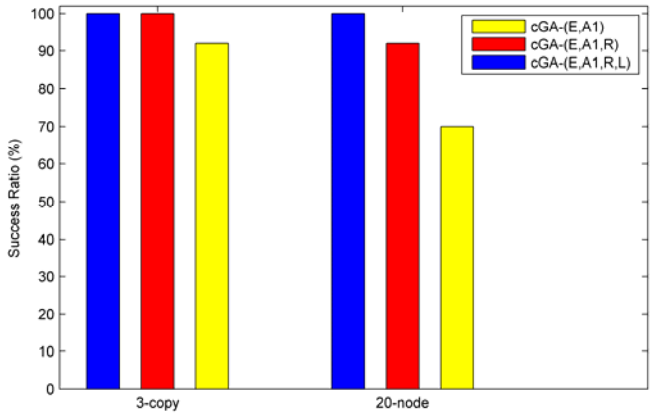
- cGA-(E,A1)
- cGA-(E,A1,R)
- cGA-(E,A1,R,L): cGA-(E,A1,R) with the L-operator.

Table 4. 4 Experimental Results of the Three Algorithms (Best Results are in Bold)

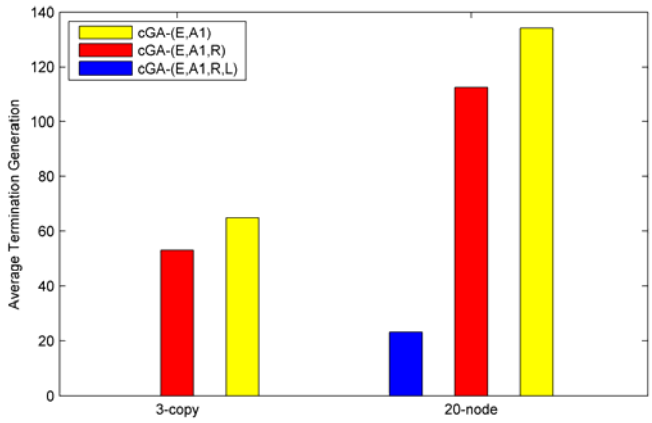
Scenarios	Criteria	cGA-(E,A1)	cGA-(E,A1,R)	cGA-(E,A1,R,L)
3-copy	SR(%)	92	**100**	**100**
	Mean(SD)	0.08(0.27)	**0.00**(0.00)	**0.00**(0.00)
	ATG	64.90	53.2	**0**
20-node	SR(%)	70	**92**	**100**
	Mean(SD)	0.30(0.46)	0.08(0.27)	**0.00**(0.00)
	ATG	134.16	112.56	**23.20**

Table 4.4 shows the experimental results of the three algorithms above. We can see that cGA-(E,A1,R,L) performs the best, obtaining the highest SR and the smallest mean and ATG (see column *cGA-(E,A1,R,L)* in each instance). The second best algorithm is cGA-(E,A1,R). To assist easy comparisons, we also provide the results of SR and ATG in Figure 4.6. Clearly, L-operator improves the performance of cGA-(E,A1,R). Note that the ATG of cGA-(E,A1,R,L) is zero on the 3-copy network, meaning a coding-free NCM subgraph can be found at the initialization of the algorithm with the L-operator. This is not only because 3-copy instance is simpler but also due to that its network structure is more suitable for implementing L-operator. According to Section 4.2.1.3, a feasible solution corresponds to a secondary graph G_S and there may be more than one feasible NCM subgraph in G_S. L-operator is designed to exploit G_S and find a NCM subgraph with few coding operations. L-operator performs well on networks where each merging node has sufficient number of outgoing links. This is because L-operator is based on the removal of auxiliary links in merging nodes. Sufficient number of outgoing links helps to

construct alternative paths which can avoid coding operations. Regarding the structure of *n*-copy networks, each merging node has two outgoing links (see Section 3.6.1). Once an auxiliary link is removed, an alternative path can be built. Hence, coding operations can be avoided. On the other hand, as mentioned in Section 4.2.1.1, the elite solution is set to an all-one vector, which means all auxiliary links are activated in the network and enough alternative paths are available. Therefore, L-operator can find an optimal solution when evaluating the all-one vector.



(a) SR results



(b) ATG results

Figure 4. 6 Results comparison of the three variants of cGA

4.2.2.4 The Overall Performance Analysis

In order to thoroughly analyze the overall performance of the proposed cGA, we compare the following algorithms in terms of the aforementioned evaluation criteria and the average computational time (ACT):

- QEA: a quantum-inspired evolutionary algorithm for NCRM problem (Xing *et al*, 2010).
- sGA-1: a standard genetic algorithm with BLS encoding and operators (Kim *et al*, 2007b). A greedy sweep operator is employed after the evolution to improve the quality of the best individual found.
- sGA-2: a standard genetic algorithm with BTS encoding and operators (Kim *et al*, 2007b). The same greedy sweep operator is adopted.
- cGA-1: cGA-(E,A1).
- cGA-2: cGA-(E,A1,R).
- cGA-3: cGA-(E,A1,R,L).

Note that the above cGAs are also based on BLS encoding (see Subsection 3.5.1). The population size for QEA, sGA-1 and sGA-2 is set to 20. The parameter settings of the six algorithms are listed in Table 4.5. Experiments have been conducted on three fixed and eight randomly-generated directed networks. To ensure a fair comparison, QEA, sGA-1 and sGA-2 have been re-implemented on the same machine and evaluated on the same NCM scenarios. Table 4.6 shows the experimental networks and parameter setup (see Subsection 3.5.1 for details). All experiments have been run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM. The results achieved by each algorithm are averaged over 50 runs.

Table 4. 5 Parameter Settings of the Six Algorithms

QEA	sGA-1	sGA-2	cGA-1	cGA-2	cGA-3
RAD = 0.1π;	p_c = 0.8; p_m = 0.01; *tsize* = 4;	p_c = 0.8; p_m = 0.012; *tsize* = 12;	N = 20;	N = 20; g_c = 50;	N = 20; g_c = 50;

Note: p_c: crossover probability; p_m: mutation probability; t_{size}: tournament size; RAD: a constant number to calculate the rotation angle step (Xing et al, 2010); N: the parameter that determines the learning rate; g_c: the predefined number of consecutive generations.

Table 4.7 compares the six algorithms with respect to the SR. Obviously, cGA-3 is the best algorithm, achieving the highest SRs (see column *cGA-3* in each instance). This demonstrates that the three extensions can strengthen the global exploration and local exploitation of cGA-3. sGA-2 performs the second best (see column *cGA-2* for details). Apart from the Rnd-5, Rnd-6 and Rnd-8 networks, the SRs obtained by sGA-2 are at least no worse and usually higher than those obtained by the other four algorithms. Without the local search operator, cGA-2 shows to be weak on local exploitation, however, is still able to obtain decent results compared with sGA-1.

Table 4. 6 Experimental Networks and Parameter Setup

	Multicast Scenario Description				Parameters	
Networks	nodes	links	sinks	rate	LI	DTG
7-copy	57	84	8	2	80	500
15-copy	121	180	16	2	176	500
31-copy	249	372	32	2	368	1000
Rnd-1	30	60	6	3	86	500
Rnd-2	30	69	6	3	112	500
Rnd-3	40	78	9	3	106	500
Rnd-4	40	85	9	4	64	500
Rnd-5	50	101	8	3	145	500
Rnd-6	50	118	10	4	189	500
Rnd-7	60	150	11	5	235	1000
Rnd-8	60	156	10	4	297	1000

Note: LI: the length of an individual; DTG: the defined termination generation.

Table 4. 7 Comparisons of Successful Ratio (%) (Best Results are in Bold)

Scenarios	QEA	sGA-1	sGA-2	cGA-1	cGA-2	cGA-3
7-copy	45	80	96	42	**100**	**100**
15-copy	0	0	50	0	4	**100**
31-copy	0	0	0	0	0	**100**
Rnd-1	**100**	98	**100**	94	**100**	**100**
Rnd-2	**100**	**100**	**100**	**100**	**100**	**100**
Rnd-3	66	50	70	24	68	**100**
Rnd-4	**100**	**100**	**100**	**100**	**100**	**100**
Rnd-5	46	56	40	14	26	**98**
Rnd-6	42	16	32	18	30	**96**
Rnd-7	25	6	60	2	14	**100**
Rnd-8	84	14	60	14	80	**100**

Experimental results of the mean and SD of the best fitness values are shown in Table 4.8. Obviously, cGA-3 outperforms all the other algorithms (see column *cGA-3*). The results also show that the three extensions significantly improve the optimization performance of cGA.

Table 4.9 illustrates the comparisons of the ATG obtained by each algorithm. cGA-3 performs outstandingly better than the others, terminating in the initial generation in five networks and in a significantly reduced number of generations in the other networks except for the Rnd-2 network (see column *cGA-3*). As mentioned, cGA-3 also terminates in the initial generation on the 3-copy network. Note that termination in the initial generation occurs only when a coding-free NCM subgraph is found in the initialization of cGA-3. This phenomenon shows that combining the L-operator with the all-one vector is particularly effective to solve *n*-copy networks, Rnd-1 and Rnd-4 networks. The following explains why. On the one hand, the all-one solution activates all auxiliary links of merging nodes and hence the corresponding secondary graph has the highest probability to produce an optimal NCM subgraph, compared with other solutions. As discussed in Section 4.2.2.3, L-operator gains better performance in networks where merging nodes have alternative paths to forward the incoming data. On the other hand, *n*-copy, Rnd-1 and Rnd-4 networks have enough outgoing links to divert the traffic (see Section 3.6). Therefore, cGA-3 can find optimal NCM subgraph in the initialization in the above instances.

Table 4. 8 Comparisons of Mean and Standard Deviation of the Best Fitness Values (Best Results are in Bold)

Scenarios	QEA	sGA-1	sGA-2	cGA-1	cGA-2	cGA-3
7-copy	0.95(1.09)	0.70(1.83)	0.04(0.19)	0.82(0.87)	**0.00**(0.00)	**0.00**(0.00)
15-copy	10.2(7.09)	4.55(3.85)	0.60(0.68)	5.46(1.85)	2.42(1.23)	**0.00**(0.00)
31-copy	18.8(5.35)	22.5(6.36)	3.85(1.13)	17.64(2.68)	7.60(2.39)	**0.00**(0.00)
Rnd-1	**0.00**(0.00)	0.02(0.14)	**0.00**(0.00)	0.06(0.23)	**0.00**(0.00)	**0.00**(0.00)
Rnd-2	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)
Rnd-3	0.32(0.47)	0.50(0.51)	0.30(0.47)	1.10(0.76)	0.32(0.47)	**0.00**(0.00)
Rnd-4	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)	**0.00**(0.00)
Rnd-5	0.55(0.51)	0.50(0.51)	0.64(0.48)	1.04(0.56)	0.74(0.44)	**0.02**(0.14)
Rnd-6	0.60(0.59)	1.15(0.74)	0.94(0.84)	1.34(0.93)	0.94(0.73)	**0.04**(0.19)
Rnd-7	1.50(1.23)	1.00(0.32)	0.35(0.48)	2.22(0.95)	1.48(0.93)	**0.00**(0.00)
Rnd-8	0.16(0.37)	0.90(0.44)	0.35(0.48)	1.24(0.74)	0.20(0.40)	**0.00**(0.00)

The computational time is of vital importance for evaluating an algorithm. Table 4.10 illustrates that cGA-3 spends less time than QEA, sGA-1 and sGA-2 on all networks and sometimes the time reduction can be substantial. For example, in the case of 31-copy network, the average computational time of cGA-3 is around 30 sec, which is

significantly shorter than 3993 sec by QEA and 2406 sec by sGA-1. This demonstrates that, integrated with intelligent schemes, our proposed cGA consumes less computational time while obtaining better solutions compared with the existing algorithms. The reason for cGA-3 consuming less computational time is that the number of fitness evaluations is far reduced as it only produces one individual at each generation. Although more time may be spent in the local search procedure, the high quality solutions found by the L-operator and the less number of fitness evaluations can well compromise and lead to less computational expenses.

Table 4. 9 Comparisons of Average Termination Generation (Best Results are in Bold)

Scenarios	QEA	sGA-1	sGA-2	cGA-1	cGA-2	cGA-3
7-copy	301.2	289.6	228.3	328.9	233.6	**0.0**
15-copy	500.0	500.0	458.2	500.0	497.5	**0.0**
31-copy	1000.0	1000.0	1000.0	1000.0	1000.0	**0.0**
Rnd-1	9.7	66.7	85.3	60.5	33.7	**0.0**
Rnd-2	**6.5**	51.5	44.7	34.0	46.2	19.5
Rnd-3	225.0	398.1	338.6	405.4	309.8	**72.2**
Rnd-4	7.4	32.0	36.8	29.0	32.0	**0.0**
Rnd-5	349.4	355.6	393.3	443.3	420.6	**152.34**
Rnd-6	338.6	457.7	436.4	435.4	425.6	**136.3**
Rnd-7	832.2	989.9	755.0	982.5	924.1	**183.2**
Rnd-8	300.5	891.5	753.1	875.9	507.4	**114.1**

Table 4. 10 Comparisons of Average Computational Time (sec.)

Scenarios	QEA	sGA-1	sGA-2	cGA-1	cGA-2	cGA-3
7-copy	25.15	14.14	16.82	3.06	1.38	0.11
15-copy	195.57	112.68	158.24	23.03	16.28	2.04
31-copy	3903.5	436.85	2406.2	399.05	269.71	28.32
Rnd-1	0.51	3.40	6.22	0.35	0.10	0.06
Rnd-2	0.62	2.26	3.25	0.15	0.15	0.16
Rnd-3	27.97	31.55	31.74	5.17	2.56	3.60
Rnd-4	0.68	2.58	3.37	0.09	0.11	0.04
Rnd-5	56.73	41.72	57.69	7.39	4.64	9.33
Rnd-6	75.20	63.83	78.14	10.77	8.14	16.90
Rnd-7	292.28	272.79	225.32	43.05	36.61	46.00
Rnd-8	120.90	224.23	229.32	37.83	14.35	22.83

In summary, with regard to the successful ratio, the average best fitness and standard deviation, the average termination generation and the average computational time, cGA with the three improvement schemes outperforms all other opponents.

4.2.3 Summary of the Algorithm

This section presents a novel compact genetic algorithm (cGA) for addressing the NCRM problem. As discussed in Section 4.1, the proposed PBIL consumes plenty of computational time in some network instances. This may restrict its application in real communications networks, which is the motivation of the work in this section. There are three improvements in the developed cGA. The first one is to set the initial elite individual as an all-one solution, which not only makes sure that cGA starts with a feasible solution but also gradually tunes the PV such that feasible solutions appear with increasingly higher probabilities. This scheme is important to locate feasible regions for cGA at the beginning of the evolution. The second improvement is to reset the PV once the current elite solution dominates in a predefined number of consecutive generations (denoted by g_c). It avoids cGA getting stuck at local optima, hence being able to improve the global exploration capability of cGA. The parameter g_c needs to be set properly so that cGA can locate different promising regions during the search. Besides, a wide range of values are suitable for g_c. Hence the selection of an appropriate value for g_c is easier than the selection of threshold value in the entropy-based restart scheme. As the cGA produces only one solution at each generation, it is necessary to make use of each feasible solution generated. The third improvement is a local search operator (L-operator) exploring the secondary graph of each feasible solution so as to improve the quality of the NCM subgraph. L-operator gains decent performance on networks in which there are sufficient number of alternative paths diverting the incoming traffic of merging nodes. The second and third improvements are proposed for the first time. In particular, the third improvement is the very first attempt to incorperate problem-specific domain knowledge into the evolution to facilitate the local exploitation ability of EAs for solving the NCRM problem. The three improvements, when employed together, can significantly improve the performance of the standard cGA. Compared with the existing EAs and PBIL (see Section 4.1), the proposed cGA gains better overall optimization performance with lower computational expenses. This is important since a lower computational time may offer a possibility of applying our cGA to real-time and dynamic communications networks.

This work has been published at Applied Intelligence (Xing and Qu, 2012).

4.3 Path-Oriented Encoding Evolutionary Algorithm

As reviewed in Section 3.3, all existing EAs for the NCRM problem adopt BLS or BTS encoding to represent chromosomes. Also, the proposed PBIL and cGA are based on BLS encoding (see Section 4.1 and 4.2). However, using BLS or BTS encoding has two disadvantages. First, the search space contains too many infeasible solutions. The following explains why. Merging nodes are usually placed at key positions of a network, responsible for switching and diverting flows (see Subsection 3.1.2). If all merging nodes are deactivated, it is highly likely to lead to an infeasible NCM subgraph due to the violation of the data rate restriction. As stated in Subsection 3.4, how flows passing through merging nodes is determined by the states of the incoming auxiliary links of secondary graph G_S. If many incoming auxiliary links are inactive in G_S (i.e. many 0's in chromosome), an infeasible solution is very likely to appear. This is why the search space contains too many infeasible solutions (Section 4.3.2.1 demonstrate this finding). Infeasible solutions may disconnect feasible areas in the search space and dramatically increase the difficulty of finding an optimal solution by EAs. Second, the evaluation of BLS and BTS is complex and indirect, requiring a number of processing steps, i.e. chromosome $\mathbf{X} \rightarrow G_S \rightarrow d$ max-flows (d sets of link-disjoint paths) $\rightarrow f(\mathbf{X})$ (see Subsection 3.5.3). Meanwhile, the computational overhead of the evaluation is relatively high since d max-flows have to be computed for each chromosome. The two drawbacks above motivate us to explore an alternative encoding to represent the solutions to the NCRM problem.

In telecommunications, EAs are widely used as an optimizer to select appropriate routes within limited time. When designing EAs, path-oriented encoding is a direct and natural choice since routing itself is the process of selecting paths in a network along which the traffic is delivered. In the literature, path-oriented encoding has been adopted by EAs for solving shortest path routing and multicast routing problems. A number of GAs (Ahn and Ramakrishna, 2002; Cheng and Yang, 2010; Yang *et al*, 2010a) have been employed to find the cost-optimal path connecting the given source and receiver. Each chromosome is represented by a path containing a string of IDs of nodes through which the path passes. Also, EAs have been used to construct least-cost spanning trees, where

each chromosome is represented by a set of paths from the source to receivers (Palmer and Kershenbaum, 1994; Siregar *et al*, 2005; Oh *et al*, 2006; Cheng and Yang, 2008, 2010b). Similar to constructing a spanning tree, NCM finds a routing subgraph which owns multiple paths. Hence, path-oriented encoding could be a potential choice as the chromosome representation to the NCRM problem. However, to our knowledge no research in the literature concerns path-oriented encoding for the problem concerned.

On the other hand, a NCM subgraph is different from a spanning tree. In a spanning tree, there is a single path connecting the source and each receiver; while in a NCM subgraph there are multiple link-disjoint paths (i.e. paths without common link) between the source and each receiver. In addition, the same data flow is transmitted through the spanning tree; while different data flows pass through different areas of the NCM subgraph. So, when designing path-oriented encoding for our problem, one should carefully design the associated genetic operators according to the characteristics of NCM subgraphs.

In this section, we propose an EA using path-oriented encoding (denoted by pEA) to address the NCRM problem. Our contribution is summarized as below.

- **A path-oriented encoding scheme.** In this scheme, a chromosome is comprised of d basic units (BUs), where d is the number of receivers. Each BU consists of a set of link-disjoint paths connecting the source and a certain receiver (see Section 4.3.1.1). This encoding is, for the first time, used to adapt for the NCRM problem.
- **The associated genetic operators**. We develop three genetic operators, namely initialization, crossover and mutation based on the path-oriented encoding. In the initialization, an allelic BU pool is generated for each receiver. Then, each chromosome of the population is created by randomly selecting one BU for each receiver. To explore the search space we use a single-point crossover which operates upon BUs. Mutation is performed on selected BUs, where the mutation probability is associated with the number of receivers. The design of the above operators takes into account the features of the NCRM problem.
- **A problem-specific local search operator**. Different from the one developed for cGA (see Section 4.2.1.3), this operator is designed for the path-oriented

encoding and aims to revise BUs to avoid performing coding operations. It helps to improve solution quality and avoid prematurity. The new operator is proposed for the first time.

4.3.1 Algorithm Design

We first describe the new encoding approach and its evaluation procedure. After that, we introduce other necessary components, namely initialization, crossover, mutation and local search operator. Finally, the overall procedure of pEA is given.

4.3.1.1 The Path-Oriented Encoding and Evaluation

We adopt the path-oriented encoding in pEA. Each chromosome consists of a set of paths originating from the source and terminating at one of the receivers. Each path is encoded as a string of positive integers representing the IDs of nodes through which the path passes. The set of paths is classified into d subsets, i.e. d basic unit (BU), where paths in BU connect to the same receiver and they do not share any common link (i.e. they are link-disjoint). Besides, there are R paths in each BU, where R is the expected data rate. Each chromosome is feasible since each BU of the chromosome satisfies the data rate requirement. Each BU can be easily obtained by max-flow algorithms. For example, we find a NCM subgraph from Figure 4.2(b) which consists of four paths, as shown below.

$P_1(s,t_1) = s \rightarrow a \rightarrow t_1$; $P_2(s,t_1) = s \rightarrow b \rightarrow u_2 \rightarrow w_1 \rightarrow c \rightarrow u_3 \rightarrow w_3 \rightarrow t_1$;

$P_1(s,t_2) = s \rightarrow a \rightarrow u_1 \rightarrow w_1 \rightarrow c \rightarrow u_3 \rightarrow w_4 \rightarrow t_2$; $P_2(s,t_2) = s \rightarrow b \rightarrow t_2$;

The corresponding chromosome is illustrated in Figure 4.7.

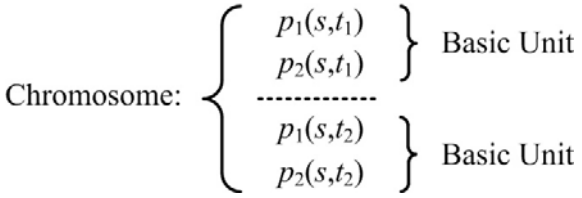


Figure 4. 7 An example chromosome.

Based on the path-oriented encoding, the evaluation of a chromosome is simple. For chromosome **X**, the union of all paths in **X** forms the corresponding NCM subgraph. The

fitness of **X**, $f(\mathbf{X})$, is the number of coding links used in the NCM subgraph. So, the computation complexity here is significantly lower than that of BLS and BTS encodings since the latter has to compute multiple max-flows.

Compared with BLS and BTS, path-oriented encoding has two advantages. First, all solutions in the search space are feasible, which leads to a globally connected search space. Hence, the new encoding fundamentally resolves the infeasibility problem caused by BLS and BTS encodings. Second, the evaluation procedure of the new encoding is less time-consuming than that of BLS and BTS.

4.3.1.2 Initialization

It is widely recognized that, for EAs, a good initial population is more likely to lead to a better optimization result. For the proposed algorithm, we initialize the population in the following way. First, we create an allelic BU pool (pool-i) for each receiver t_i, where $i = 1,\ldots,d$. Second, we randomly choose one BU from pool-i, $i = 1,\ldots,d$, and combine the selected BUs as a chromosome. The second step is repeated to create a population of a predefined size.

Let N be the population size of the algorithm. Let G_D be the decomposed graph after graph decomposition (see Subsection 3.2.3). Note that in pEA we operate on the decomposed graph instead of the secondary graph. Let R denote the expected data rate and hence each BU contains R link-disjoint paths. Let $Flow(s,t_i)$ and $Vol(s,t_i)$ be the max-flow (made of link-disjoint paths) and its volume from s to receiver t_i, respectively. The max-flow algorithm (Goldberg, 1985) is used to calculate $Flow(s,t_i)$ and $Vol(s,t_i)$. Figure 4.8 shows the initialization procedure in our EA.

For a specific graph G_{temp}, only one BU can be obtained by the max-flow algorithm. To obtain an allelic BU pool for receiver t_i, we have to change the structure of G_{temp} by deleting different links from G_D at each time. As aforementioned, how the information flows pass through a particular merging node is dependent on the states of the auxiliary links in the node, and how the information flows pass through a network depends on the states of all auxiliary links in the network. So, only auxiliary links are considered for deletion in pEA. To generate a new BU for receiver t_i, we randomly pick up a BU from pool-i and randomly select an auxiliary link owned by the BU, as shown in steps 9 and

10. The selected link is then removed from G_{temp} to make sure the new G_{temp} is a different graph, in step 11.

```
      // Generation of BU pools
1.    for i = 1 to d do
2.        Set G_temp = G_D and pool-i = ∅
3.        for j = 1 to pop do
3.            Find Flow(s,t_i) from G_temp by the max-flow algorithm (Goldberg, 1985)
4.            if Vol(s,t_i) ≥ R then
5.                Randomly select R paths from Flow(s,t_i) as a new BU
6.                if the new BU is not in pool-i then
7.                    Put this BU into pool-i
8.            Set G_temp = G_D
9.            Randomly select a BU (with at least one auxiliary link) from pool-i
10.           Randomly choose an auxiliary link owned by the selected BU
11.           Delete this auxiliary link from G_temp
      // Generation of the population
12.   for j = 1 to N do
13.       for i = 1 to N do
14.           Randomly select a BU from pool-i
15.           Include the BU in the j-th chromosome
16.   Output the initial population
```

Figure 4. 8 The procedure of initialization

4.3.1.3 Crossover

In pEA, we use single-point crossover to each pair of selected chromosomes with a crossover probability p_c. As aforementioned, there are d BUs in a chromosome. The crossover point is randomly chosen from the $(d - 1)$ positions between two consecutive BUs. Two offspring are created by swapping the BUs of the two parents behind the crossover point. The proposed crossover does not destroy the structure of any BU. So, after crossover, the offspring are still feasible to warrantee a connected search space. An example crossover operation is illustrated in Figure 4.9, where each parent consists of four BUs and the crossover point is between the second and third BUs.

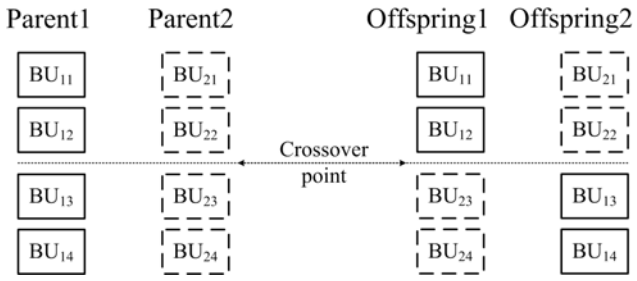


Figure 4. 9 An example of the crossover operator.

4.3.1.4 Mutation

Mutation is another important genetic operator to help the local exploitation and avoid the prematurity of EAs. As aforementioned, each BU is a set of R link-disjoint paths from the source to a particular receiver. Mutation upon a BU leads to another set of R link-disjoint paths. The idea behind the mutation is that some auxiliary links owned by the chosen BU are deleted from the decomposed graph G_D. Then, a new BU is generated by implementing the max-flow algorithm on the new G_D. We propose a mutation operator, where each BU of a chromosome is to be mutated at a mutation probability p_m. We only concern the removal of auxiliary links since they determine the amount of coding resources required.

In the mutation, for a chosen BU, we randomly select an auxiliary link owned by the BU and delete the link from the decomposed graph G_D. This is to make sure a different BU can be reconstructed. Besides, we also delete those unoccupied auxiliary links which connect to one of the outgoing auxiliary nodes being occupied by the remaining $(d - 1)$ BUs, to ensure that no additional coding links are introduced after implementing M. After that, we compute the max-flow, i.e. $Flow(s,t_i)$, by using the max-flow algorithm on G_D (Goldberg, 1985). If the volume of $Flow(s,t_i)$, namely $Vol(s,t_i)$, is no smaller than the expected data rate R, a new BU can be obtained by randomly selecting R paths in $Flow(s,t_i)$. The new BU then replaces the old BU. If $Vol(s,t_i)$ is smaller than R, the data rate requirement cannot be met and the old BU remains.

The procedure of the proposed mutation is shown in Figure 4.10, where *rnd*() is a function that generates a uniformly distributed random value in the range [0,1]. One advantage of the mutaiton is that the fitness value of a chromosome tends to be smaller after mutation.

```
1.  for j = 1 to N do
2.      for i = 1 to d do
3.          if rnd() < p_m then
                // the i-th BU of the j-th chromosome is chosen
4.              Set G_temp = G_D;
5.              if the i-th BU owns at least one auxiliary link then
6.                  Randomly select an auxiliary link owned by the i-th BU;
7.                  Delete the link from G_temp;
8.                  Delete those unoccupied auxiliary links connecting to one of the
                    outgoing auxiliary nodes being occupied by the remaining BUs;
9.                  Find Flow(s,t_i) from G_temp using the max-flow algorithm;
10.                 if Vol(s,t_i) ≥ R then
11.                     Randomly select R paths from Flow(s,t_i) and replace the old BU
                        with them;
12. Output the mutated population.
```

Figure 4. 10 The procedure of the proposed mutation.

Regarding the mutation probability p_m, a fixed value may not be a wise choice since the number of BUs in a chromosome changes according to d, i.e. the number of receivers in different multicast scenarios. A fixed p_m, e.g. 0.1, could lead to a dramatically different number of mutation operations during the evolution, thus is not generally applicable to different NCM sessions. In pEA, we set $p_m = 1/d$, thus the amount of mutation operations involved does not change too much in different NCM sessions, and more likely to lead to a stable optimization performance of EA.

4.3.1.5 The Local Search Operator

As introduced in Section 4.2.1.3, a local search operator (L-operator) is developed to improve the NCM subgraph of each feasible solution. This operator makes use of domain knowledge to reduce the number of coding operations involved in the NCM subgraph in order to improve the quality of solutions of each generation. When incorporated in cGA, L-operator helps to greatly strengthen the local exploitation ability and hence the overall optimization performance of cGA (see Section 4.2.2.3). This indicates the importance of combining pure evolutionary framework and local search procedures. As it is developed

for BLS/BTS encoding, L-operator is not suitable for path-oriented encoding which is composed of BUs. This motivates us to design an appropriate local search operator for the proposed EA, taking the features of the new encoding into account.

This section proposes a local search (LS) operator which is performed on a randomly selected chromosome at each generation.

The aim of this operator is to revise some BUs of the selected chromosome to gradually reduce the number of coding links involved in a multicast. Note that each outgoing link of a merging node is redirected to an outgoing auxiliary node after the graph decomposition, as discussed in Subsection 3.2.3. So in a NCM subgraph, each coding link corresponds to a certain coding node (i.e. an outgoing auxiliary node that performs coding). In G_D, to reduce the number of coding nodes is equal to decrease the number of coding links. Assume there is a chromosome $\mathbf{X}$ of which the NCM subgraph contains K coding nodes, where K is an integer. The LS operator aims to remove the occurrence of coding operation at each coding node. The K coding nodes will be processed one by one, in an ascending order according to their node IDs.

Assume the k-th coding node (denoted by v_k, $k = 1, 2, \ldots, K$) is to be processed by the LS operator and there are $In(v_k)$ auxiliary links connecting to v_k in the NCM subgraph**,** meaning information via these links is involved in the coding at v_k. To remove the coding from v_k, one simple idea is to delete arbitrarily $In(v_k) - 1$ auxiliary links from the NCM subgraph. However, directly removing these links leads to an infeasible $\mathbf{X}$ since those BUs which occupy these $In(v_k) - 1$ links will be damaged. To overcome this problem, our LS operator reconstructs the affected BUs so that they bypass the use of the $In(v_k) - 1$ auxiliary links mentioned above, explained as follows.

First of all, we randomly select $In(v_k) - 1$ auxiliary links connecting to v_k and check which BUs are occupying these links. The affected BUs will be reconstructed, while the others remain in the NCM subgraph. Next, we delete the selected $In(v_k) - 1$ auxiliary links from the decomposed graph G_D. Besides, we also delete those currently unoccupied auxiliary links from G_D which connect to one of the outgoing auxiliary nodes being occupied by the unaffected BUs. The reason to remove those unoccupied auxiliary links is that we expect to eliminate the chance of removing one coding node at the expense of introducing other coding node(s). Finally, we reconstruct the affected BUs by using the

max-flow algorithm over G_D. If all affected BUs are successfully constructed, we obtain a new chromosome $\mathbf{X}_{new}$. If $\mathbf{X}_{new}$ owns less coding links than $\mathbf{X}$, we replace the incumbent $\mathbf{X}$ with $\mathbf{X}_{new}$ (i.e. the LS moves to an improved solution $\mathbf{X}_{new}$) and repeat the LS operator to improve the new incumbent $\mathbf{X}_{new}$. Otherwise, we retain $\mathbf{X}$ and proceed to the next coding node of $\mathbf{X}$. The LS operator stops when either no improvement is made to the incumbent chromosome after checking all its coding nodes, or a new chromosome with no coding involved (i.e. optimal) is found.

The LS operator is useful to improve the quality of the selected chromosome. Also, it changes the structure of the chromosome. Sometimes the changes can be significant, meaning the new chromosome may be far away from the original one in the search space. Hence, the new chromosome may help to increase the population diversity (see Section 4.3.2.6).

4.3.1.6 The Overall Procedure of the Proposed EA

1. **Initialization**
2. Set $t = 0$;
3. Initialize population $\{\mathbf{X}_1(t), \ldots, \mathbf{X}_N(t)\}$ by the proposed initialization operator;
4. Evaluate each chromosome $\mathbf{X}_i(t)$, $i = 1,\ldots, N$;
5. Randomly select one chromosome and perform LS operator on it;
6. **Repeat**
7. Set $t = t + 1$;
8. Select a new population $\{\mathbf{X}_1(t), \ldots, \mathbf{X}_N(t)\}$ by the tournament selection;
9. Replace a random chromosome with the best chromosome of the previous generation, e.g. $\mathbf{X}_{best}(t - 1)$;
10. Execute crossover to each pair of selected chromosomes with probability p_c;
11. Execute mutation to each BU of each chromosome with probability p_m;
12. Evaluate each chromosome $\mathbf{X}_i(t)$, $i = 1,\ldots, N$;
13. Randomly select one chromosome and perform the LS operator on it;
14. **until** *the termination condition is met*

Figure 4. 11 The procedure of pEA (a standard EA framework).

Based on the path-oriented encoding, we develop an EA for solving the NCRM problem. To efficiently drive the evolution, we design the associated genetic operators as

explained above concerning characteristics of the problem concerned. The procedure of pEA is shown in Figure 4.11. Note that pEA is a standard EA with specialized encoding.

The evaluation of chromosome $\mathbf{X}_i(t)$ (in step 4) is simple. We mark those nodes and links in G_D which are being occupied by at least one of the BUs in $\mathbf{X}_i(t)$. The union of the marked nodes and links in G_D forms the NCM subgraph $G_{s\to T}$ of $\mathbf{X}_i(t)$. The number of coding links in $G_{s\to T}$, $\Phi(G_{s\to T})$, is assigned to $\mathbf{X}_i(t)$ as its fitness.

In step 8, tournament selection (Mitchell, 1996) is adopted in pEA. The tournament size is set to 2. In step 9, the elitism scheme is used to preserve the best chromosome of previous generations, which helps to guide the search towards global optima. In step 11, either ordinary mutation or greedy mutation can be used. The termination conditions are, either the EA has found a chromosome of which the NCM subgraph is coding-free, or EA has evolved a predefined number of generations.

4.3.2 Performance Evaluation

We first investigate the deficiency of BLS and BTS encodings in terms of the distribution of infeasible solutions over search space. After that we study the effectiveness of the crossover and mutation operators of pEA, and compare EAs with path-oriented, BLS and BTS encodings. The LS operator is studied next. Finally, we compare pEA with the state-of-the-art EAs and hEA (see Subsection 4.3.1) in terms of optimization performance and computational time.

We consider 14 different test instances, four on fixed networks and 10 on randomly generated networks. The four fixed networks are 3-copy, 7-copy, 15-copy and 31-copy networks and the 10 random networks (Rnd-i, i = 1,…,10) are directed networks with 20 to 60 nodes. Table 4.11 shows the 14 instances and their parameters (see Subsection 3.5.1 for details). The predefined number of generations for all algorithms tested is set to 200. All experiments were run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM. The results were collected by running each algorithm 50 times.

Table 4. 11 Experimental Networks and Instance Parameters

Network	nodes	links	sinks	rate	*L*
3-copy	25	36	4	2	32
7-copy	57	84	8	2	80
15-copy	121	180	16	2	176
31-copy	249	372	32	2	368
Rnd-1	20	37	5	3	43
Rnd-2	20	39	5	3	50
Rnd-3	30	60	6	3	86
Rnd-4	30	69	6	3	112
Rnd-5	40	78	9	3	106
Rnd-6	40	85	9	4	64
Rnd-7	50	101	8	3	145
Rnd-8	50	118	10	4	189
Rnd-9	60	150	11	5	235
Rnd-10	60	156	10	4	297

Note: *L*: the encoding length of individuals

4.3.2.1 Deficiency of BLS and BTS encodings

Using different encoding approaches may greatly affect the performance of EAs (Mitchell, 1996). The resulting search spaces might be significantly different with respect to not only the size but also the structure and connectivity of the underlying landscape. As discussed in Section 4.3, in theory, the search space of BLS or BTS encoding may contain many infeasible solutions which may disconnect feasible regions in the search space. The connectivity between feasible solutions may be so weak that EAs finding optimal solution(s) becomes extremely difficult.

We roughly estimate the proportion of infeasible solutions by randomly sampling the search space. In our experiment, the sampling size is in proportion to the encoding length *L* of each instance, namely $100 \cdot L$ (Xu and Qu, 2012). The larger the problem size, the larger the sampling size. For example, the sampling size is 3,200 in 3-copy instance and 29,700 in Rnd-10 instance, respectively. Table 4.12 shows the results of the proportion of infeasible solutions (PIS) of the three encodings. Regarding the BLS and BTS encodings, the PIS values are more than 99% in all instances (see columns *BLS*, *BTS*). In particular, in 7-copy, 15-copy, 31-copy and Rnd-5,7,8,9,10, the PISs are always 100%, meaning that all random samples are infeasible solutions (see rows *7,15,31-copy, Rnd-5,7,8,9,10,* columns *BLS*, *BTS*). Large amount of infeasible solutions may disconnect feasible regions in the search space and dramatically increase the problem difficulty for search algorithms. Hence, BLS and BTS encodings may not be appropriate choices for the

optimization problem concerned. On the other hand, path-oriented encoding results into 0% PIS, meaning that all samples are feasible (see column *Path-Oriented*). This is not a surprise because in theory the search space of the path-oriented encoding only contains feasible solutions (see Section 4.3.1.1).

Table 4. 12 Comparison of PIS (%) of the BLS, BTS and Path-oriented encodings (Best Results are in Bold)

Networks	BLS	BTS	Path-Oriented	Networks	BLS	BTS	Path-Oriented
3-copy	99.84	99.75	**0.00**	Rnd-4	99.77	99.33	**0.00**
7-copy	100.00	100.00	**0.00**	Rnd-5	100.00	100.00	**0.00**
15-copy	100.00	100.00	**0.00**	Rnd-6	99.98	99.90	**0.00**
31-copy	100.00	100.00	**0.00**	Rnd-7	100.00	100.00	**0.00**
Rnd-1	99.53	99.13	**0.00**	Rnd-8	100.00	100.00	**0.00**
Rnd-2	99.98	99.98	**0.00**	Rnd-9	100.00	100.00	**0.00**
Rnd-3	99.88	99.89	**0.00**	Rnd-10	100.00	100.00	**0.00**

4.3.2.2 Performance Measures

To show the performance of pEA in various aspects, such as the optimal solution obtained, the convergence characteristic, and the consumed running time, the following performance metrics are used.

- Mean and standard deviation (SD) of the best solutions found over 50 runs. One best solution is obtained in one run. The mean and SD are important metrics to show the overall performance of a search algorithm.
- Student's *t*-test (Walpole *et al*, 2007; Yang and Yao, 2008) to compare two algorithms (e.g. Alg-1 and Alg-2) in terms of the fitness values of the 50 best solutions obtained. Two-tailed *t*-test with 98 degrees of freedom at a 0.05 level of significance is used. The *t*-test result can show statistically if the performance of Alg-1 is better than, worse than, or equivalent to that of Alg-2.
- Successful ratio of finding an optimal solution in 50 runs. This metric reflects the global exploration ability of an EA to find optimality.
- Evolution of the best fitness over generations in 50 runs. The plot of the evolution illustrates the convergence process of an algorithm.
- Average computational time consumed by an algorithm over 50 runs.

4.3.2.3 The Effectiveness of Crossover in pEA

As mentioned in Section 4.3.1.3, a single-point crossover is developed for pEA. We investigate the feasibility of this operator and the impact of different settings of the crossover probability p_c on the performance of pEA. Mutation and LS operator is excluded in pEA in this experiment. We set the population size $N = 20$ and compare the performance of pEA with four different p_c, i.e. 0.0, 0.3, 0.6, and 0.9, where $p_c = 0.0$ means the algorithm stops after initialization since crossover is not involved. By comparing the results of different p_c and those of $p_c = 0.0$, one could see the effectiveness of crossover itself.

Table 4. 13 Comparisons of pEA with Different Crossover Probabilities (Best Results are in Bold)

	$p_c = 0.0$		$p_c = 0.3$		$p_c = 0.6$		$p_c = 0.9$	
Networks	Mean	SD	Mean	SD	Mean	SD	Mean	SD
3-copy	2.84	0.37	1.70	0.61	1.32	0.51	**1.08**	0.27
7-copy	9.58	1.45	7.64	1.43	6.78	1.35	**6.16**	1.23
15-copy	22.88	0.47	20.68	1.92	19.74	2.00	**17.54**	1.98
31-copy	46.94	0.42	45.32	1.89	44.72	1.79	**43.20**	2.26
Rnd-1	2.44	0.64	1.70	0.64	1.18	0.66	**0.96**	0.66
Rnd-2	0.62	0.56	0.12	0.32	0.04	0.19	**0.02**	0.14
Rnd-3	2.64	0.56	1.86	0.70	1.40	0.72	**1.22**	0.64
Rnd-4	0.72	0.45	0.38	0.49	0.22	0.41	**0.10**	0.30
Rnd-5	7.58	0.81	5.60	1.08	5.06	1.13	**4.46**	1.32
Rnd-6	0.40	0.49	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-7	3.86	1.01	3.06	0.79	2.96	1.02	**2.34**	0.77
Rnd-8	6.84	0.42	5.76	1.04	5.28	1.10	**4.64**	1.10
Rnd-9	6.00	0.00	5.42	0.67	5.14	0.63	**4.98**	0.58
Rnd-10	7.94	1.39	6.40	1.22	5.54	1.51	**5.18**	1.17

The results of the mean and SD of the obtained best fitness values are shown in Table 4.13. It can be seen that pEA with crossover performs better than pEA without crossover in each instance, indicating crossover can properly drive the evolution process (compare *Mean* of $p_c = 0.0$ and *Mean* of $p_c = 0.3$, 0.6, 0.9 in the same row). Besides, we find that with larger p_c the mean becomes increasingly better (compare *Mean* of $p_c = 0.3$, 0.6, 0.9 in the same row). The variant of pEA with $p_c = 0.9$ performs the best, showing that rapid exchange of genetic information over different chromosomes helps to explore different areas in the search space. However, one can also find that there remain big gaps

between the best solutions obtained by pEA with only crossover and the optimal solutions (with fitness values of 0.00) in each instance. This is mainly because employing crossover only is not enough to guide pEA to escape from local optima. We therefore need mutation to enhance local exploitation and avoid prematurity.

4.3.2.4 The Effectiveness of Mutation in pEA

As introduced in Section 4.3.1.4., we design a mutation operator with probability $p_m = 1/d$, where d is the number of receivers. To mutate a BU, the mutation operator deletes a random auxiliary link of the BU and a number of unoccupied auxiliary links from G_D. The removal of the random link is to make sure that the mutated BU is different from the old one. The removal of those unoccupied links is to ensure no extra coding link will be introduced after mutation. In the following experiment, we study the performance of pEA with the proposed crossover and mutation. The mutation probability p_m is set to $2/d$, $1/d$, and $0.5/d$. Let In the experiment, LS operator is excluded. We set $N = 20$ and $p_c = 0.9$.

Table 4. 14 Comparison of pEA with Different Mutation Probabilities (Best Results are in Bold)

	$p_m = 2/d$		$p_m = 1/d$		$p_m = 0.5/d$	
Networks	Mean	SD	Mean	SD	Mean	SD
3-copy	**0.04**	0.19	0.14	0.35	0.26	0.44
7-copy	**1.26**	0.59	1.64	0.80	1.94	0.79
15-copy	**5.72**	1.22	6.04	0.92	6.68	0.84
31-copy	**17.60**	1.50	18.20	1.19	18.30	1.55
Rnd-1	**0.00**	0.00	0.04	0.19	0.06	0.23
Rnd-2	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-3	**0.00**	0.00	0.04	0.19	0.02	0.14
Rnd-4	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-5	**0.00**	0.00	0.02	0.14	0.12	0.32
Rnd-6	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-7	**0.12**	0.32	0.34	0.47	0.56	0.50
Rnd-8	**0.02**	0.14	0.04	0.19	0.30	0.46
Rnd-9	**0.80**	0.40	0.86	0.35	0.94	0.23
Rnd-10	**0.00**	0.00	**0.00**	0.00	0.06	0.23

Table 4.14 shows the results of mean and SD of the obtained best fitness values by pEA with different p_m. We find that the mean is getting better and better with p_m changing from $0.5/d$ to $2/d$ (compare columns *Mean* in the same row). This is because the mutation operator ensures that the rebuilt BU does not increase the number of coding

operations to the corresponding chromosome. On the contrary, it is quite possible that coding at one or several nodes of a chromosome is eliminated after mutation. Hence, imposing reasonably more mutation operations to the evolving population is more likely to obtain a better optimization performance of pEA.

To further support our findings, we compare pEA with different p_m by using *t*-test. Results are given in Table 4.15. pEA with a larger p_m performs better than pEA with a smaller p_m. However, their performance does not differ too much. For example, between $2/d$ and $1/d$, the former only wins 2 instances (see column *2/d↔1/d*).

Table 4. 15 *t*-Test Results for pEA with Different Mutation Probabilities

Networks	$2/d \leftrightarrow 1/d$	$2/d \leftrightarrow 0.5/d$	$1/d \leftrightarrow 0.5/d$
3-copy	~	+	~
7-copy	+	+	~
15-copy	~	+	+
31-copy	~	~	~
Rnd-1	~	~	~
Rnd-2	~	~	~
Rnd-3	~	~	~
Rnd-4	~	~	~
Rnd-5	~	+	~
Rnd-6	~	~	~
Rnd-7	+	+	+
Rnd-8	~	+	+
Rnd-9	~	+	~
Rnd-10	~	~	~

Note: the result of $p_{m1} \leftrightarrow p_{m2}$ is shown as "+", "–", or "~" when pEA with p_{m1} is significantly better than, significantly worse than, or statistically equivalent to pEA with p_{m2}, respectively.

4.3.2.5 Comparisons of Different Encoding Approaches

This subsection compares three EAs, namely pEA, GA with BLS encoding (BLSGA) and GA with BTS encoding (BTSGA). For BLS and BTS encoding approaches please see (Kim *et al*, 2007b) and Section 3.5 for details. Note that an all-one vector is inserted into the initial population of BLSGA and BTSGA to make sure they begin with at least one feasible solution; otherwise, the two GAs may never converge since no feasible solution may be obtained during the search (Kim *et al*, 2007b). This has showed to be an effective method in previous work (Kim *et al*, 2007a, 2007b; Xing and Qu, 2011a, 2011b, 2012).

The comparison is based on a standard GA framework, where genetic operators in each EA include selection, crossover and mutation. The population size is set to 20 for each algorithm. The parameter settings of the three GAs are listed in Table 4.16.

Table 4. 16 Parameter Settings of BLSGA, BTSGA and pEA

BLSGA	BTSGA	pEA
$p_c = 0.8$;	$p_c = 0.8$;	$p_c = 0.9$;
$p_m = 0.006$;	$p_m = 0.012$;	$p_m = 1/d$;
$tsize = 2$;	$tsize = 2$;	$tsize = 2$;

Note: p_c: crossover probability; p_m: mutation probability; t_{size}: tournament size.

Table 4. 17 Comparisons of GA with Different Encodings (Best results are in bold)

	Mean and Standard Deviation					
	BLSGA		BTSGA		pEA	
Networks	Mean	SD	Mean	SD	Mean	SD
3-copy	0.46	1.01	0.74	1.20	**0.14**	0.35
7-copy	3.82	4.26	3.86	3.93	**1.64**	0.80
15-copy	7.92	5.64	11.92	6.00	**6.04**	0.92
31-copy	37.60	9.19	43.22	4.47	**18.20**	1.19
Rnd-1	0.96	1.29	1.00	1.48	**0.04**	0.19
Rnd-2	0.44	0.83	0.38	0.75	**0.00**	0.00
Rnd-3	0.40	0.98	0.66	1.20	**0.04**	0.19
Rnd-4	0.28	0.45	0.08	0.27	**0.00**	0.00
Rnd-5	2.98	4.01	4.22	4.70	**0.02**	0.14
Rnd-6	0.42	0.81	0.36	0.77	**0.00**	0.00
Rnd-7	2.14	1.95	2.72	2.16	**0.34**	0.47
Rnd-8	3.04	1.94	3.88	1.96	**0.04**	0.19
Rnd-9	3.68	1.40	4.24	1.59	**0.86**	0.35
Rnd-10	3.52	3.40	3.76	3.50	**0.00**	0.00

	Successful Ratio (%)			Average Computational Time (sec.)		
Networks	BLSGA	BTSGA	pEA	BLSGA	BTSGA	pEA
3-copy	80	68	**86**	1.13	1.47	**0.61**
7-copy	8	2	**8**	11.47	11.85	**10.72**
15-copy	0	0	0	54.57	51.19	**38.52**
31-copy	0	0	0	98.51	**72.55**	180.47
Rnd-1	46	54	**96**	3.17	2.86	**0.39**
Rnd-2	78	78	**100**	0.91	1.12	**0.09**
Rnd-3	84	74	**96**	4.02	4.21	**0.77**
Rnd-4	72	92	**100**	2.95	1.98	**0.18**
Rnd-5	8	10	**98**	15.75	13.45	**2.85**
Rnd-6	78	82	**100**	3.05	2.67	**0.20**
Rnd-7	10	6	**66**	21.11	19.15	**11.31**
Rnd-8	2	0	**96**	32.60	29.23	**10.00**
Rnd-9	0	2	**14**	51.49	45.81	**38.86**
Rnd-10	4	0	**100**	62.04	57.25	**9.01**

The comparisons of EAs with different encodings are shown in Table 4.17, where

the mean and standard deviation of the obtained best fitness values, successful ratio and average computational time are given. In terms of the mean and SR, pEA outperforms BLSGA and BTSGA in all instances (compare columns *Mean*, *SR* in the same row). On the one hand, the search space of the path-oriented encoding is comprised of feasible solutions only. The developed genetic operators are devised in compliance with the nature of the new encoding and they also incorporate domain knowledge to facilitate the search. On the other hand, the search space of BLS/BTS contains too many infeasible solutions which are potential barriers against smooth search. Moreover, feasible solutions are not evenly distributed but more close to the all-one solution. However, the above characteristic has not been considered in the design of crossover and mutation in BLSGA and BTSGA (Kim *et al*, 2007b). Hence, pEA gains better optimization results than BLSGA and BTSGA. From Table 4.17, we also see that pEA consumes less average computational time than BLSGA and BTSGA in almost all instances except the 31-copy network. In addition, the *t*-test results comparing pEA, BLSGA and BTSGA are shown in Table 4.18.

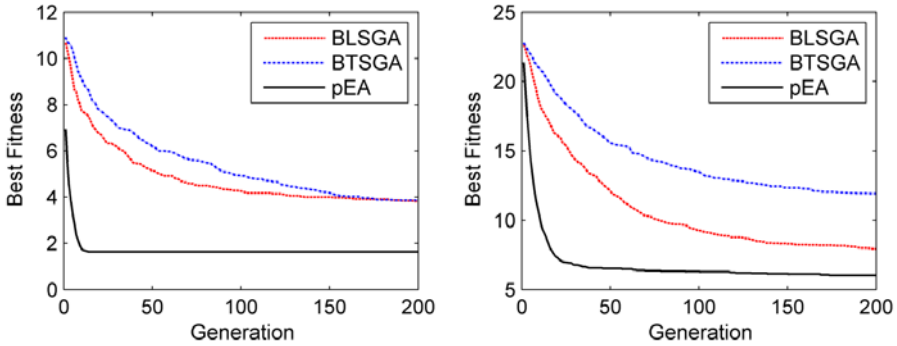
Table 4. 18 *t*-Test Comparison of Different GAs

Networks	pEA↔BLSGA	pEA↔BTSGA	Networks	pEA↔BLSGA	pEA↔BTSGA
3-copy	+	+	Rnd-4	+	+
7-copy	+	+	Rnd-5	+	+
15-copy	+	+	Rnd-6	+	+
31-copy	+	+	Rnd-7	+	+
Rnd-1	+	+	Rnd-8	+	+
Rnd-2	+	+	Rnd-9	+	+
Rnd-3	+	+	Rnd-10	+	+

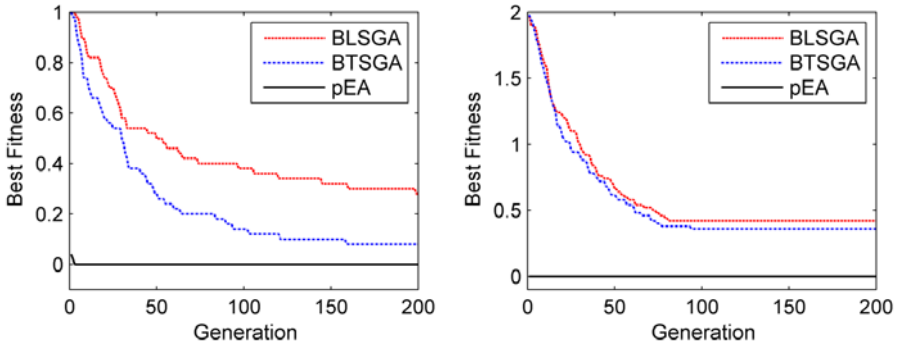
Note: the result of Algorithm1↔Algorithm2 is shown as "+", "−", or "~" when Algorithm1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm2, respectively.

To show the convergence of the three EAs, we plot the evolution of the best fitness in each generation, averaged over 50 runs for two fixed and four random instances, as shown in Figure 4.12. First, we see that pEA always obtains better initial solutions than BLSGA and BTSGA. For example, in Figure 4.12(*a*), at the beginning of the evolution, the average best fitness for pEA is around 7 while those of BLSGA and BTSGA are both 11. Moreover, we find that pEA converges fast especially in the early generations. For example, in Figure 4.12(*a*)-(*b*), pEA converges after around 10 and 25 generations. This

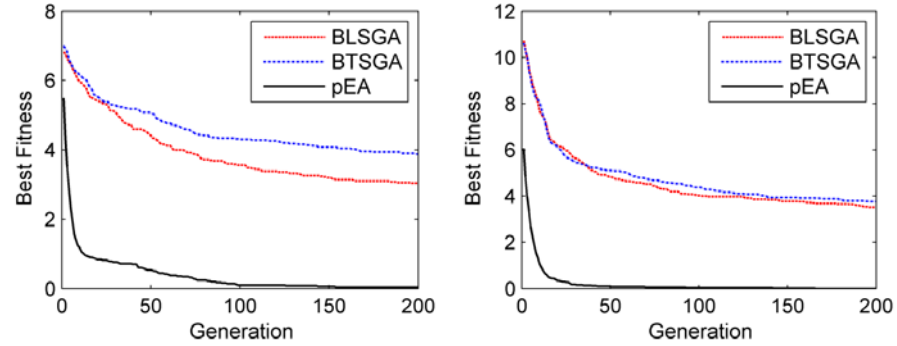
is an outstanding advantage especially in real-time and dynamic applications, where a decent solution should be found within limited time.



(*a*) 7-copy (*b*) 15-copy



(*c*) Rnd-4 (*d*) Rnd-6



(*e*) Rnd-8 (*f*) Rnd-10

Figure 4. 12 Best fitness vs. generation for the following instances. (*a*) 7-copy (*b*) 15-copy (*c*) Rnd-4 (*d*) Rnd-6 (*e*) Rnd-8 (*f*) Rnd-10

According to the results and analysis above, we find that pEA is more efficient than BLSGA and BTSGA with respect to global optimization, convergence, and running time.

4.3.2.6 The Effectiveness of the LS Operator

As discussed in Subsection 4.3.1.5, a LS operator is applied to a randomly chosen chromosome at each generation to improve solution regarding the fitness. To verify this operator, we generate five random chromosomes for each instance using the initialization method in Section 4.3.1.2. We apply the LS operator on each chromosome and compare the fitness values i.e. Φ_{BEF} and Φ_{AFT}, of the chromosome before and after implementing the LS operator. Let $\mathbf{X}$ and $\mathbf{X}'$ denote the chromosome before and after the LS, and $E_A(\mathbf{X})$ and $E_A(\mathbf{X}')$ be the set of auxiliary links owned by $\mathbf{X}$ and $\mathbf{X}'$, respectively. We define the structural difference coefficient (SDC) between $\mathbf{X}$ and $\mathbf{X}'$ according to the Marczewski-Steinhaus concept of distance (Marczewski and Steinhaus, 1958), as follows:

$$SDC = \frac{|E_A(\mathbf{X}) \cup E_A(\mathbf{X}')| - |E_A(\mathbf{X}) \cap E_A(\mathbf{X}')|}{|E_A(\mathbf{X}) \cup E_A(\mathbf{X}')|} \cdot 100\% \tag{4.6}$$

The value of SDC is between 0.0 and 1.0, which tells us to what degree $\mathbf{X}$ and $\mathbf{X}'$ are different, showing the effect of LS operator on the structure change of the solutions. A larger SDC indicates a severer structural change caused by the LS operator.

Table 4. 19 Results of Φ_{BEF}, Φ_{AFT} and SDC before/after Implementing LS Operator

	Solution 1			Solution 2			Solution 3			Solution 4			Solution 5		
Networks	Φ_{BEF}	Φ_{AFT}	SDC (%)	Φ_{BEF}	Φ_{AFT}	SDC (%)	Φ_{BEF}	Φ_{AFT}	SDC (%)	Φ_{BEF}	Φ_{AFT}	SDC (%)	Φ_{BEF}	Φ_{AFT}	SDC (%)
3-copy	3	0	54.5	4	0	20.0	3	0	30.0	5	0	36.3	6	0	50.0
7-copy	12	0	51.7	18	0	53.1	13	0	54.8	16	0	56.2	14	0	53.3
15-copy	30	0	54.5	35	0	56.3	27	0	54.5	29	0	55.2	40	0	52.8
31-copy	47	0	53.4	69	0	53.8	60	0	53.9	71	0	54.9	70	0	57.3
Rnd-1	4	2	36.0	5	3	17.3	2	0	30.0	7	3	26.9	6	1	38.4
Rnd-2	2	0	8.33	3	2	18.5	5	3	7.41	4	2	11.5	3	1	33.3
Rnd-3	3	0	38.8	3	0	19.3	5	1	34.1	7	0	45.2	6	0	50.0
Rnd-4	4	1	25.7	5	2	25.6	4	0	22.8	1	0	13.7	2	1	42.1
Rnd-5	12	7	20.0	9	3	17.8	11	3	21.8	8	4	23.6	10	2	39.6
Rnd-6	2	0	21.7	1	0	14.2	1	0	44.0	3	0	24.0	2	0	25.0
Rnd-7	8	4	20.0	6	2	28.3	4	1	10.2	9	3	13.5	6	5	8.33
Rnd-8	8	5	10.5	12	7	15.8	11	3	29.8	15	8	20.4	14	5	22.0
Rnd-9	13	7	15.7	18	5	21.7	8	4	12.3	14	4	19.7	12	7	15.9
Rnd-10	12	3	30.8	15	5	24.5	8	5	8.89	9	5	10.9	7	3	31.1

The experimental results of Φ_{BEF}, Φ_{AFT} and SDC are shown in Table 4.19. First, it is seen that Φ_{AFT} is significantly better than Φ_{BEF}, demonstrating that the LS operator can improve the quality of chromosomes (see Φ_{BEF}, Φ_{AFT} of the same solution). This is

because the LS operator is designed to reduce the number of coding operations in the NCM subgraph. It makes use of the domain knowledge to reconstruct BUs that cause coding operations (see Section 4.3.1.5). On the other hand, regarding the values of SDC in all instances, 32 chromosomes (45% of the 70 chromosomes) are at least 30% different on the structure, meaning the LS operator may also help to introduce additional diversity to the population (see columns *SDC*). This is because the BU reconstruction process causes the structural change of a chromosome. The more BUs reconstructed, the severer the structural change.

4.3.2.7 Overall performance Evaluation

In this section, we evaluate the overall performance of pEA by comparing it with six algorithms. The following explains the algorithms for comparison.

- GA1: BLS based GA with greedy sweep operator (Kim *et al*, 2007b).
- GA2: BTS based GA with greedy sweep operator (Kim *et al*, 2007b).
- QEA1: A quantum-inspired evolutionary algorithm (QEA) (Xing *et al*, 2010).
- QEA2: Another QEA proposed by Ji and Xing (2011).
- PBIL: PBIL proposed in the thesis (see Section 4.1) (Xing and Qu, 2011a).
- cGA: cGA proposed in the thesis (see Secton 4.2) (Xing and Qu, 2012).
- pEA: the path-oriented encoding based EA (see Subsection 4.3).

The population size is set to 20 for each algorithm. The parameter settings of the seven algorithms are shown in Table 4.20.

Table 4. 20 Parameter Settings of the Seven Algorithms

GA1	GA2	QEA1	QEA2	PBIL	cGA	pEA
$p_c = 0.8$; $p_m = 0.006$; *tsize* = 2;	$p_c = 0.8$; $p_m = 0.012$; *tsize* = 2;	RAD = 0.1π;	$\sigma_0 = 0.04\pi$; $\Delta\sigma = 0.004\pi$; $\Gamma = 20$;	$\alpha = 0.1$; $\tau = 0.2$; $\Theta = 0.14$;	$N = 20$; $g_c = 50$;	$p_c = 0.9$; $p_m = 1/d$; *tsize* = 2;

Note: p_c: crossover probability; p_m: mutation probability; t_{size}: tournament size; RAD: a constant number to calculate the rotation angle step (RAS); σ_0: initial RAS; $\Delta\sigma$: incremental RAS; Γ: the parameter in penalty function; α: learning rate; τ: the percentage of the selected solutions; Θ: entropy threshold value; N: the parameter that determines the learning rate; g_c: the predefined number of consecutive generations.

The results of the mean and standard deviation of the best fitness values obtained and results of the successful ratio are shown in Table 4.21 and 4.22, respectively.

Regarding the mean value (compare columns *Mean* in the same row), we can see that pEA always perform the best in each instance, while cGA is the second best. The third best algorithm is PBIL. PBIL outperforms QEA1 and QEA2 in 6 instances (see rows *7,15-copy, Rnd-5,7,9,10*) and is beaten by QEAs in 2 instances (see rows *31-copy*, *Rnd-8*). QEA1 and QEA2 perform better than GA1 and GA2 taking into account all instances (see columns *Mean* of GA1, GA2, QEA1, QEA2). In addition, Table 4.22 also demonstrates that pEA has the highest SR (see column *pEA*). The following explains why pEA has better global optimization performance than the others. First, pEA is based on path-oriented encoding which is fundamentally appropriate to be used in EAs for the NCRM problem (see Sections 4.3.1.1 and 4.3.2.5). Besides, the associated genetic operators and the problem specific local search procedure are designed in compliance with the features of the new encoding. Therefore, pEA gains decent performance in terms of global exploration and local exploitation at the same time.

Table 4. 21 Results of Mean and Standard Deviation (Best Results are in Bold)

	GA1		GA2		QEA1		QEA2		PBIL		cGA		pEA	
Networks	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
3-copy	0.36	0.74	0.08	0.27	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
7-copy	1.96	1.92	0.68	0.84	0.18	0.62	0.48	0.70	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
15-copy	7.48	5.12	3.66	2.13	3.10	4.18	5.80	1.62	2.14	4.31	**0.00**	0.00	**0.00**	0.00
31-copy	28.75	7.97	18.66	22.58	19.10	5.76	20.00	0.00	28.90	10.30	**0.00**	0.00	**0.00**	0.00
Rnd-1	0.52	0.88	0.44	0.50	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-2	0.26	0.66	0.02	0.14	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-3	0.44	0.83	0.02	0.14	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-4	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-5	2.78	2.71	1.16	0.61	0.46	0.50	0.48	0.54	0.04	0.28	0.04	0.19	**0.00**	0.00
Rnd-6	0.22	0.41	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-7	1.58	0.92	1.36	0.66	0.66	0.47	0.58	0.53	0.38	0.60	0.22	0.41	**0.00**	0.00
Rnd-8	2.52	1.44	2.28	0.94	0.98	0.82	0.48	0.61	0.60	1.56	0.24	0.43	**0.00**	0.00
Rnd-9	2.82	1.22	2.34	1.34	1.64	0.98	1.94	1.16	0.06	0.23	0.04	0.19	**0.00**	0.00
Rnd-10	3.26	2.68	1.38	0.69	0.66	0.68	0.42	0.64	**0.00**	0.00	0.08	0.27	**0.00**	0.00

Now, we compare the average computational time, which is given in Table 4.23. Before analyzing the data, we roughly divide the 14 instances into two groups, easy instances and hard instances. In easy instances, more than half of the algorithms obtain a successful ratio of 100%. The rest are hard instances. Easy instances includes 3-copy and Rnd-1,2,3,4,6 (see rows *3-copy, Rnd-1,2,3,4,6* in Table 4.22) while hard instances contains 7,15,31-copy and Rnd-5,7,8,9,10 (see rows *7,15,31-copy, Rnd-5,7,8,9,10* in

Table 4.22). For each easy instance, most algorithms can obtain an optimal solution within a very short time (e.g. less than 1 second) (see rows *3-copy, Rnd-1,2,3,4,6* in Table 4.23). However, for each hard instance, the average computational time of each algorithm may differ significantly (see rows *7,15,31-copy, Rnd-5,7,8,9,10* in Table 4.23).

Table 4. 22 Results of Successful Ratio (%) (Best Results are in Bold)

Networks	GA1	GA2	QEA1	QEA2	PBIL	cGA	pEA
3-copy	80	92	**100**	**100**	**100**	**100**	**100**
7-copy	14	52	88	62	100	**100**	**100**
15-copy	0	4	26	0	58	**100**	**100**
31-copy	0	0	0	0	0	**100**	**100**
Rnd-1	62	56	**100**	**100**	**100**	**100**	**100**
Rnd-2	86	98	**100**	**100**	**100**	**100**	**100**
Rnd-3	76	98	**100**	**100**	**100**	**100**	**100**
Rnd-4	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Rnd-5	4	10	54	54	98	96	**100**
Rnd-6	78	**100**	**100**	**100**	**100**	**100**	**100**
Rnd-7	8	8	34	44	68	78	**100**
Rnd-8	2	0	30	58	82	76	**100**
Rnd-9	4	8	14	10	94	96	**100**
Rnd-10	4	6	46	64	100	92	**100**

Table 4. 23 Results of Average Computational Time (sec.) (Best Results are in Bold)

Networks	GA1	GA2	QEA1	QEA2	PBIL	cGA	pEA
3-copy	0.99	1.61	0.24	0.21	0.10	**0.02**	0.09
7-copy	12.42	11.98	8.54	10.41	2.20	**0.15**	0.33
15-copy	55.85	49.27	89.88	91.61	66.14	2.09	**1.57**
31-copy	232.92	200.73	728.13	750.70	543.64	29.55	**20.79**
Rnd-1	2.95	3.30	0.73	0.50	0.29	0.23	**0.16**
Rnd-2	1.14	1.33	0.37	0.40	0.13	**0.02**	0.11
Rnd-3	5.13	5.07	0.68	0.75	0.23	**0.06**	0.27
Rnd-4	3.19	3.13	0.57	0.81	0.26	**0.16**	0.23
Rnd-5	16.57	14.52	13.82	14.38	6.09	3.14	**0.63**
Rnd-6	3.54	3.34	0.72	0.84	0.17	**0.03**	0.23
Rnd-7	24.13	20.78	24.35	22.52	24.29	6.83	**2.10**
Rnd-8	38.37	30.89	38.04	31.47	27.43	20.11	**0.95**
Rnd-9	62.46	50.73	73.73	73.94	47.29	16.40	**1.93**
Rnd-10	71.25	55.46	64.12	52.39	31.81	17.42	**1.15**

A robust algorithm should spend a short computational time to find the optima for not only easy instances but also hard instances. For the easy instances in Table 4.23, pEA, cGA, PBIL, QEA1 and QEA2 all consume similar average computational time (i.e. less than 1 second) while GA1 and GA2 are the two worst ones. For hard instances, pEA is the best algorithm. This is because: (1) the evaluation process of path-oriented encoding

is simpler than that of BLS/BTS encoding (see Section 4.3.1.1), and (2) pEA is powerful in finding global optima and it stops once an optimum is found (see Section 4.3.1.6).

Regarding the overall performance in Table 4.21, 4.22 and 4.23, pEA performs the best. To further support this finding, we show the *t*-test results comparing pEA with the others in Table 4.24. We can see that for each instance pEA performs at least statistically equivalent to but usually significantly better than the other algorithms.

Table 4. 24 *t*-Test Results for Different EAs

Networks	pEA↔GA1	pEA↔GA2	pEA↔QEA1	pEA↔QEA2	pEA ↔PBIL	pEA ↔cGA
3-copy	+	+	~	~	~	~
7-copy	+	+	+	+	~	~
15-copy	+	+	+	+	+	~
31-copy	+	+	+	+	+	~
Rnd-1	+	+	~	~	~	~
Rnd-2	+	~	~	~	~	~
Rnd-3	+	~	~	~	~	~
Rnd-4	~	~	~	~	~	~
Rnd-5	+	+	+	+	~	~
Rnd-6	+	~	~	~	~	~
Rnd-7	+	+	+	+	+	~
Rnd-8	+	+	+	+	+	+
Rnd-9	+	+	+	+	~	~
Rnd-10	+	+	+	+	~	+

Note: the result of Algorithm1↔Algorithm2 is shown as "+", "–", or "~" when Algorithm1 is significantly better than, significantly worse than, or statistically equivalent to Algorithm2, respectively.

4.3.3 Summary of the Algorithm

We develop a new evolutionary algorithm (pEA) for the NCRM problem. pEA is based on path-oriented encoding. The use of path-oriented encoding is motivated by that the existing BLS/BTS encoding has two drawbacks, namely large number of infeasible solutions in the search space and high computational overhead in the fitness evaluation. The drawbacks may harm the performance of EAs. Our contribution in this section includes the design of path-oriented encoding for the NCRM problem and pEA based on it. The new encoding and its fitness evaluation do not suffer the drawbacks of BLS/BTS encoding. Besides, the features of the new encoding are considered in the process of designing algorithm components.

In pEA, each chromosome consists of a number of basic units (BUs), each of which contains a set of link-disjoint paths from the source to the same receiver. In accordance to

the new encoding approach, we develop the associated initialization, crossover and mutation operators. Experiments demonstrate that pEA performs better than BLS/BTS based EAs with respect to the pure evolutionary search ability (see Section 4.3.2.5). pEA obtains better global exploration, faster convergence, and lower computational time. Meanwhile, a problem-specific local search operator is developed to improve the quality of a selected chromosome at each generation. Different from the one introduced in Section 4.2.2.3, the local search in pEA is specially designed for path-oriented encoding. Though, the aims of the two local search operators are similar, namely they attempt to reduce the coding operations in the NCM subgraph in order to improve the quality of the chosen solution. pEA with LS operator is reported to outperform existing EAs and the proposed PBIL and cGA in terms of the best solutions obtained and the computational time consumed.

This work is accepted by Journal of Operational Research Society.

4.4 Summary

This chapter presents three EAs developed for solving the NCRM problem, i.e. PBIL in Section 4.1, cGA in Section 4.2 and path-oriented encoding EA (pEA) in Section 4.3.

The proposed PBIL is more suited for the BLS encoding, compared with the existing EAs. This is because of the nature of PBIL. The continuously changing probability vector (PV) exploits the promising solutions found in order to locate promising regions in the search space. Moreover, the entropy-based restart scheme, as a contribution, helps to escape the local optima and improve the global exploration of PBIL. There are two problems with the proposed PBIL. One is that although it outperforms the existing EAs in terms of the global exploration, PBIL's computational overhead is relatively high in some instances, which may limit its application. The other is in the restart scheme the threshold value is based on the observation of empirical experiments, which may not be always reliable to unknown networks.

Later on, we adapt cGA for the NCRM problem which outperforms PBIL and other existing EAs in terms of the best results and the computational time. This algorithm is also suitable for the BLS encoding because of the continuity reflected in the process of

updating PV. We have three contributions when developing the cGA, namely the use of all-one vector, a PV restart scheme and a problem specific local search operator. The PV restart scheme is designed according to the features of the encoding and cGA. The key parameter in the restart scheme is the predefined number of consecutive generations (g_c) that the current elite solution dominates. The selection of g_c is easier than that of the parameter in PBIL's restart scheme. The local search operator is the very first problem specific local search operator designed for the problem concerned.

However, BLS/BTS encoding has a drawback with respect to the infeasibility issue of the search space. Besides, the corresponding fitness evaluation requires relatively high computational costs. Therefore, we look for an alternative to the BLS/BTS encoding and then contribute a path-oriented encoding EA (called pEA). The new encoding is designed according to the features of the NCRM problem. Besides, the associated components are developed to build up the evolutionary framework. Experimental results demonstrate that pEA is effective to solve the NCRM problem and its performance is relatively better than the proposed cGA and PBIL with respect to the overall optimization performance.

Addressing the NCRM problem can be seen as an early step towards the practical deployment of NCM in communication networks with existing infrastructures. However, this problem does not take into account service quality for end users, which may not be appropriate to support real-time multimedia multicast applications. In the next chapter, we investiage a variant of the NCRM problem which not only considers the minimization of network coding resources but also the user experience, i.e. quality-of-service (QoS) issues.

Chapter 5 PBIL for the Delay Constrained Network Coding Resource Minimization

5.1 Motivation

As discussed in Subsection 1.5.2, to support real-time multimedia applications usually requires stringent quality-of-service (QoS) guarantees (Aurrecoechea *et al*, 1998; Chalmers and Sloman, 1999; Striegel and Manimaran, 2002). Hence, QoS requirements should be taken into account when NCM is applied to real-time applications (Walsh and Weber, 2008). Among those QoS parameters, transmission delay (i.e. end-to-end delay) is of vital importance, especially for delay sensitive applications, e.g. video conferencing and distributed game (Striegel and Manimaran, 2002). In the study of traditional QoS multicast routing problems (based on the store-and-forward scheme), the transmission delay is usually considered as a constraint so as to deliver the multimedia service with acceptable quality to users (Xu, 2011). Therefore, whilst studying the problem of minimizing the amount of the required coding operations, we should consider the transmission delay bound. Unfortunately, this has never been studied in the literature. Researchers focused on either the NCRM problem without concerning the delay constraint (Bhattad *et al*, 2005; Fragouli and Soljanin, 2006; Langberg *et al*, 2006; Kim *et al*, 2006, 2007a, 2007b; Ahn, 2011; Luong *et al*, 2012), or dealing with delay related issues without concerning the minimization of the amount of coding operations (Walsh and Weber, 2008; Pu *et al*, 2009; Yeow *et al*, 2009a; Zhang *et al*, 2009; Amir *et al*, 2010). This motivates us to extend the NCRM problem by considering the delay bound, which has resulted in new problem, namely the delay-constrained NCRM problem (Xing and Qu, 2011b).

In this chapter, we formulate the delay-constrained NCRM problem and study it by using a population based incremental learning algorithm (PBIL) (Xing and Qu, 2011b).

5.2 Problem Description

This thesis models the delay constrained NCRM problem as how to construct a NCM subgraph $G_{s\to T}$ with the minimal number of coding links while achieving the expected data rate and satisfying the end-to-end delay constraint Ω. This problem is a variant of the NCRM problem. The following lists the notations to define the problem:

n_{M}: the number of merging nodes in G.

$Out(i)$: the number of outgoing links that the i-th merging node owns.

R: the defined data rate (an integer) at which s expects to transmit information to each receiver t_k.

Ω: the transmission delay bound for $G_{s\to T}$.

$\lambda(s, t_k)$: the data rate between s and t_k in $G_{s\to T}$.

$e(u, v)$: directed link from node u to node v.

$G_{s\to T}$: network coding based multicast (NCM) subgraph.

$D(\omega)$: the delay of term ω. If ω is a link, $D(\omega)$ represents its propagation delay; if ω is a node, $D(\omega)$ denotes its data processing delay; If ω is a path $P(u,v)$, $D(\omega)$ is the end-to-end transmission time from node u to node v; if ω is a $G_{s\to T}$, $D(\omega)$ denotes the maximal path delay in $G_{s\to T}$.

$\Phi(G_{s\to T})$: the number of coding links in $G_{s\to T}$.

σ_{ik}: a binary variable associated with the k-th outgoing link of the i-th merging node, $i = 1,\ldots, n_{\mathrm{M}}$, $k = 1,\ldots, n_{\mathrm{OL}i}$. $\sigma_{ik} = 1$ if the j-th outgoing link of the i-th node serves as a coding link; $\sigma_{ik} = 0$ otherwise.

$P_i(s, t_k)$: the i-th link-disjoint path found between s and t_k in $G_{s\to T}$. It is represented by a chain of nodes through which the path passes.

$W_i(s, t_k)$: the link set of $P_i(s, t_k)$, i.e. $W_i(s, t_k) = \{e(u,v) \mid e(u,v) \in P_i(s, t_k)\}$.

$P_i^{s\to t_k}(u, v)$: the subpath from node u to node v along path $P_i(s, t_k)$, where $u,v \in P_i(s, t_k)$.

For the relevant network model and definitions, please refer to Chapter 3 for details. The following shows the delay constrained NCRM problem, which is a combinatorial optimization problem:

Minimize:

$$\Phi(G_{s\to T}) = \sum_{i=1}^{n_M} \sum_{j=1}^{Out(i)} \sigma_{ij} \tag{5.1}$$

Subject to:

$$\lambda(s, t_k) = R, \forall\ t_k \in T \tag{5.2}$$

$$W_i(s, t_k) \cap W_j(s, t_k) = \emptyset, \forall i,j \in \{1,...,R\}, i \neq j, t_k \in T \tag{5.3}$$

$$D(G_{s\to T}) \leq \Omega \tag{5.4}$$

where we have,

$$D(G_{s\to T}) = \max\{D(P_i(s, t_k)) \mid i = 1,\ldots,R,\ t_k \in T\} \tag{5.5}$$

Objective (5.1) defines the optimization problem as to find a NCM subgraph with the minimal number of coding links; Constraint (5.2) defines that the data rate from s to every receiver is exactly R; Constraint (5.3) restricts that for each t_k arbitrary two paths have no common link; Constraint (5.4) defines that the maximal delay[3] of the constructed NCM subgraph cannot be greater than the pre-defined delay bound Ω; The delay of the NCM subgraph is defined in (5.5) as the maximal delay among all paths. In this thesis, we only consider the transmission delay from s to t_k along each path $P_i(s, t_k)$. The decoding delay to obtain the original information in each receiver is omitted. This is because compared with the transmission delays along paths, decoding delays at receivers are trivial and hence negligible (Zhang *et al*, 2009). Note that the problem formulated here is based on the same network model in Section 3.1.

In general, there are four types of nodes along each path $P_i(s, t_k)$: the source, the receiver, forwarding node(s) and coding node(s), where a forwarding node can only forward the incoming information to the next node. Compared with coding operation, forwarding operation incurs negligible data processing delay. Hence, we assume $D(v_j) = 0$ for any forwarding node v_j, where $D(v_j)$ is the data processing delay incurred at node v_j.

[3] Note that the term 'delay' considered in the thesis is in transmission time. Given a path from a source to a receiver, the transmission delay of the path can be estimated in advance since the propagation time of each link along the path can be estimated according to length of the link and the propogation medium, such as optical fibre or wireless environment.

For a receiver or a forwarding node v_j, the transmission delay of subpath $P_i^{s\to t_k}(s,v_j)=(s,...,v_{j-1},v_j)$ from s to v_j can be defined as follows:

$$D(P_i^{s\to t_k}(s,v_j))=D(P_i^{s\to t_k}(s,v_{j-1}))+D(e(v_{j-1},v_j)) \tag{5.6}$$

where $v_{j\text{-}1}$ is the upstream node next to v_j along subpath $P_i^{s\to t_k}(s,v_j)$.

If a number of data packets are required to be coded together at a coding node, coding operation cannot start until the arrival of the last data packet. For a coding node v_j, we assume that among the $R{\cdot}d$ paths in $G_{s\to T}$ there are Y paths that pass through v_j. Obviously, along each of the Y paths there is a subpath from s to v_j. We denote the transmission delays of the Y subpaths by $D_1(s, v_j)$, $D_2(s, v_j)$, …, $D_Y(s, v_j)$, respectively. The transmission delay of the subpath from s to coding node n_j is defined as follows:

$$D(P_i^{s\to t_k}(s,v_j))=\max\{D_r(s,v_j)\mid r=1,...,Y\}+\Delta_c \tag{5.7}$$

where Δ_c is the time consumed by the coding operation. We assume any coding operation consumes the same processing time Δ_c.

5.3 Selection of EAs

The delay-constrained NCRM problem is an extension of the NCRM problem (see Chapter 3). So, those algorithms used for solving the NCRM problem could be potential candidates for addressing the new problem.

As introduced and discussed in Chapter 4, this thesis has developed three EAs for solving the NCRM problem, including PBIL, cGA and pEA (see Section 4.1, 4.2 and 4.3 for details). They gain better overall performance than other existing algorithms, such as GAs and QEAs (Kim et al, 2007a, 2007b; Xing *et al*, 2010; Ji and Xing, 2011). So, the three EAs above may be the most appropriate algorithms for addressing a variant of the NCRM problem on which they perform very well. Therefore, we first discuss their suitabilities and applicabilities to the new problem. Compared with the NCRM problem, the new one bounds the maximal path delay in the NCM subgraph in order to provide

better user experience. The two problems have the same objective function, namely finding a feasible NCM subgraph with the least number of coding operations performed. There is only an extra constraint (i.e. delay) in the new problem. From this point of view, the existing chromosome representations used in EAs for the NCRM problem can also be used here. So, the binary encodings (namely BLS and BTS) and path-oriented encoding used in Chapter 4 are also applicable to the new problem. However, the fitness evaluation process and those problem specific operators (which are based on the exploitation of domain knowledge) for the NCRM problem cannot be applied to the new problem directly because the definition of the feasibility of a NCM subgraph has changed due to the additional constraint. In the new problem, there are more infeasible solutions in the search space. Hence, when adapting the existing EAs for the new problem, modifications have to be made to the fitness evaluation and the problem specific operators already developed. On the one hand, PBIL in Section 4.1 is based on a pure evolutionary framework. It is simple and fast to adapt it for the new problem since only a few changes need to be made to the fitness evaluation and the whole algorithmic structure (which is discussed later). Besides, this algorithm has been reported to have decent performance on the NCRM problem. So, PBIL could be used as an initial attempt to study the features and properties of the new problem. On the other hand, cGA and pEA both incorporate problem specific local search operators which require significant changes in order to adapt for the new problem. This is because the two operators are designed based on the definition of the feasibility of solution. For each move in them, only feasible NCM subgraphs are accepted. As the new problem will have a different feasibility definition (because of the delay constraint), the local search operators have to be carefully modified if used in the new problem. Hence cGA and pEA are not suitable to initiate the study on the new problem. According to the above discussion, PBIL in Section 4.1 is chosen to be adapted for the delay constrained NCRM problem. But a single EA may not be enough to investigate the features and properties of the new problem. GAs (based on BLS/BTS encodings) have been applied to the NCRM problem and also used for evaluating PBIL, cGA and pEA (see Chapter 4). Besides, the fitness evaluation of GAs is the same as the one used in PBIL in Section 4.1, which means there is no difficulty adapting GAs (BLSGA and BTSGA) for the new problem after the adaption of PBIL. So, in this

chapter, BLSGA and BTSGA are also selected to be adapted for the delay constrained NCRM problem. It is more likely to get insights into the new problem via multiple EAs.

The following introduces the new fitness evaluation procedure, as shown in Figure 5.1. As mentioned in Section 3.5.1, each BLS-based individual corresponds to a unique secondary graph G_S. The fitness evaluation measures each G_S. Given individual **X**, we first check if a feasible NCM subgraph $G_{s\to T}$ can be found from the corresponding G_S. For each receiver $t_k \in T$, we use a max-flow algorithm (Goldberg, 1985) to compute the max-flow between the source s and t_k in the corresponding G_S. If all d max-flows are at least R (here, d is the number of receivers and R is the expected data rate), for each receiver t_k we select R least-delay paths from all link-disjoint paths obtained from s to t_k. All the selected paths form the NCM subgraph $G_{s\to T}$. If the maximal path delay in $G_{s\to T}$ satisfies the delay constraint, i.e. $D(G_{s\to T}) \le \Omega$, we set $\Phi(G_{s\to T})$ to $f(\mathbf{X})$, where $\Phi(G_{s\to T})$ is the number of coding links in $G_{s\to T}$. If $G_{s\to T}$ cannot be found or it violates the delay constraint, **X** is regarded as infeasible solution and we set a very large fitness value to $f(\mathbf{X})$ (50 in the thesis).

```
1.  Obtain the corresponding secondary graph G_S of X;
    // Check if a NCM subgraph G_{s→T} can be resulted from G_S
2.  Set flag = 1; // value 1 indicates X is feasible; value 0 otherwise.
3.  for k = 1 to d do
4.      Compute the max-flow from source s to receiver t_k;
5.      if the max-flow is no smaller than R then
6.          Select R least-delay paths from the resulting link-disjoint paths;
7.      else
8.          Set flag = 0 and break the loop; // indicating X is infeasible.
9.  if flag == 1 then
10.     G_{s→T} is constructed by all the selected paths;
11.     if D(G_{s→T}) ≤ Ω then // G_{s→T} meets the delay requirement
12.         Set f(X) = Φ(G_{s→T});
13. else
14.     Set f(X) = 50; // 50 is a sufficiently large number
15. Output the fitness value of X, i.e. f(X).
```

Figure 5. 1 The fitness evaluation for the delay constrained NCRM problem (Xing and Qu, 2011b).

In the fitness evaluation for the NCRM problem (see Section 3.5), a solution is referred to as a feasible solution as long as a NCM subgraph with data rate constraint met is found. However, the definition of feasibility is crispier in the new problem. A feasible solution must result into a NCM subgraph meeting the data rate requirement and delay constraint. According to this, the search space of the new problem will contain more infeasible solutions. The difference between the two fitness evaluations is Step 6 and 11 (as shown Figure 5.1) which are only minor changes. With the new fitness evaluation, the PBIL in Section 4.1 is adapted for the new problem. Hence the adaptation is simple and fast. The experimental results on the new problem are collected in Section 5.5.

5.4 An Improved PBIL (iPBIL)

PBIL in Section 4.1 is featured with an entropy-based restart scheme. The scheme, as demonstrated in Section 4.1.2, can greatly enhance the global exploration ability of the PBIL when solving the NCRM problem. However, the restart scheme potentially has some deficiencies when used in the new problem due to the following reasons. First, the key parameter of the restart scheme, namely the threshold value of the average entropy, is calculated based on empirical experiments and observations. So, the value may not be suitable for the new problem (see Section 4.1.2 for more discussion). Second, once the restart condition is met, the search of the algorithm restarts from scratch, which means no memory of the promising solutions previously found is incorporated to facilitate the next round search. This is not a big problem in the NCRM problem as the all-one vector in the initial population is always feasible and it can help to guide the search towards feasible regions. However, the all-one vector may be infeasible if the delay constraint is set severely. In this case, the restarted search may lose the guidance and hence the contribution of the restart scheme may be weakened. The above analysis motivates us to find a substitute for the restart scheme which helps PBIL gain strengthened global exploration ability.

The following introduces an alternative approach to the restart scheme in PBIL. The new approach is a combination of two schemes, namely a new probability vector (PV) update scheme and a PV mutation scheme.

5.4.1 A New Probability Vector Update Scheme

As we know, for the NCRM problem, the search space of BLS encoding contains high proportion of infeasible solutions (see Section 4.3 for details). For the delay constrained NCRM problem, this proportion should be higher due to the extra constraint. Therefore, when PBIL is adapted for the extended problem, it would be more difficult for the PV to generate promising solutions and locate promising regions in the search space. To gain adequate exploration and exploitation abilities, the PV needs necessary guidance to effectively lead the search (Baluja, 1994). The guidance information comes from the promising solution(s) used in the PV update scheme. There are mainly two PV update schemes for PBIL in the literature. One is that the PV is learnt towards a single best solution (Baluja, 1994; Yang and Yao, 2005, 2008). It is featured with fast convergence. But the search is easily misled to local optima since the PV is updated by a single solution. The other is based on the Hebbian inspired rule that incorporates a best ever found solution and a set of promising solutions of the current generation (Gonzalez *et al*, 2000). This scheme helps to maintain the diversity level and alleviates the prematurity tendency (see Secton 4.1.1 for detailed discussion). However, under the second scheme, the PV is mainly dependent on the statistical information of the promising solutions of each generation, namely the local search information. This could be a problem in the delay constrained NCRM problem as it would be difficult to generate promising solutions at some generations. Hence, the second scheme may also mislead the search to unpromising areas. Taking into account the global exploration ability and the problem features, we propose a new PV update scheme to be used in PBIL, as described below.

The new PV update scheme concerns a set of best so far individuals (solutions) $S_{BSF} = \{\boldsymbol{B}_1,\ldots, \boldsymbol{B}_H\}$, where $H \geq 1$ is a constant number ($H = 10$ in the experiment). Let $\boldsymbol{PV}(t)$ be the PV at generation t. Initially, $\boldsymbol{PV}(t)$ is sampled H times to create H individuals to form S_{BSF}. At each generation, when a number of fitter individuals appear, we update S_{BSF} by replacing those worst individuals in S_{BSF} with the fitter ones. Then, the statistical

information of S_{BSF}, namely $\boldsymbol{PV}_{BSF}$, is extracted and used to update $\boldsymbol{P}(t)$, as shown in Equations (5.8) and (5.9), where α is the learning rate.

$$\boldsymbol{PV}_{BSF} = \frac{1}{\mathrm{H}} \cdot \sum_{k=1}^{\mathrm{H}} \boldsymbol{B}_k \tag{5.8}$$

$$\boldsymbol{PV}(t) = (1.0 - \alpha) \cdot \boldsymbol{PV}(t) + \alpha \cdot \boldsymbol{PV}_{BSF} \tag{5.9}$$

Equation (5.8) shows how to extract the statistical information from S_{BSF}. We assume each individual is of length L. For position j (j = 1,…,L), the corresponding values of all individuals in S_{BSF} are summed up and the result is then divided by H. For $\boldsymbol{PV}_{BSF}$, the value in each position is between 0.0 and 1.0 since all individuals in S_{BSF} are binary strings. Given a position in $\boldsymbol{PV}_{BSF}$, a value larger than 0.5 means bit 1 has higher chance to appear then bit 0 and vice versa. Equation (5.9) illustrates how to update $\boldsymbol{PV}(t)$ using $\boldsymbol{PV}_{BSF}$. In each position, the value of $\boldsymbol{PV}(t)$ is shifted towards the value of $\boldsymbol{PV}_{BSF}$ at speed α, where α is smaller than 1.0.

The procedure of the new probability vector update scheme at generation t is shown in Figure 5.2. Note that this update scheme generalizes the update scheme in the standard PBIL. When H = 1, it is equivalent to a standard PBIL where only a single best so far individual is maintained in S_{BSF}.

1. Find H best individuals from generation t and sort them in sequence $\{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_{\mathrm{H}}\}$, where $f(\boldsymbol{C}_1) \le f(\boldsymbol{C}_2) \le \ldots \le f(\boldsymbol{C}_{\mathrm{H}})$
2. **for** i = 1 **to** H **do**
3. Find the worst individual in S_{BSF}, e.g. $\boldsymbol{B}_{wst}$, where $f(\boldsymbol{B}_{wst}) = \max\{f(\boldsymbol{B}_1), f(\boldsymbol{B}_2), \ldots, f(\boldsymbol{B}_{\mathrm{H}})\}$
4. **if** $f(\boldsymbol{C}_i) \le f(\boldsymbol{B}_{wst})$ **then**
5. Set $\boldsymbol{B}_{wst} := \boldsymbol{C}_i$; $f(\boldsymbol{B}_{wst}) = :f(\boldsymbol{C}_i)$; // S_{BSF} is updated
6. Update $\boldsymbol{PV}(t)$ by Equations (5.1) and (5.2)

Figure 5. 2 The new probability update scheme (Xing and Qu, 2011b)

As the problem concerned is highly constrained, $\boldsymbol{PV}(t)$ in the initialization of PBIL may not be able to create feasible individuals, and thus deteriorates the effectiveness and efficiency of PBIL. In the study of the NCRM problem, all-one vector has been used to

guide the search towards feasible regions in the search space (see Section 4.1). Inspired by this, we employ all-one vector(s) in the probability vector update scheme to improve the performance of PBIL. The all-one vector compensates for the absence of feasible individuals in S_{BSF} in the initialization of our algorithm. For example, if there are u ($0 < u \leq H$) infeasible individuals in S_{BSF}, these individuals are replaced by u all-one vectors. Note that, different from the NCRM problem, the extended problem considers the delay constraint. Therefore, all-one vector may be infeasible as the delay constraint may not be met.

The new PV update scheme maintains a set of historically best individuals to adjust the PV. The PV is updated based on the global exploration information obtained during the search. So the search is less likely to get stuck at a single local optimum and the global exploration ability could be enhanced. However, the scheme may have a slow convergence as those best ever found individuals may reside in different areas in the search space. On the other hand, it is still possible that the search is stuck at some local optima if the best ever found individuals are too close to each other. Hence, some diversity preservation mechanism should be introduced to avoid prematurity. This is why a probability vector mutation operator is adopted, as explained below.

5.4.2 Probability Vector Mutation

Proposed by Baluja in 1994, PV mutation introduces a small amount of probability perturbation on each locus of $\boldsymbol{PV}(t)$ at each generation. Its aim is to maintain an appropriate level of diversity and avoid local optima. This operator is not included in PBIL in Section 4.1 since the entropy-based restart scheme already has the ability to prevent PBIL getting stuck at local optima. In the improved PBIL (called iPBIL), the new PV update scheme and PV mutation are used to replace the restart scheme in Section 4.1.

The procedure of PV mutation is described below. The PV is implemented after learning $\boldsymbol{PV}(t) = \{p_1^t, p_2^t, \ldots, p_L^t\}$ towards $\boldsymbol{PV}_{BSF}$. We denote by p_m and ε the mutation probability and the mutation amount in each locus of $\boldsymbol{PV}(t)$, respectively. For locus i, $i = 1, 2, \ldots, L$, we randomly generate a uniformly distributed number rnd_i between 0.0 and 1.0. If $rnd_i < p_m$, p_i^t is mutated by the following formula:

$$p_i^t = (1.0 - \varepsilon) \cdot p_i^t + rnd\{0.0,1.0\} \cdot \varepsilon \tag{5.10}$$

where $rnd\{0.0,1.0\}$ is a two-value random number that generates 0.0 or 1.0 both with a probability of 0.5.

5.4.3 The Structure of iPBIL

The procedure of iPBIL is shown in Figure 5.3. This algorithm is characterized by a new PV update scheme and a mutation operator. There are three differences between PBIL in Section 4.1 and iPBIL. The first difference is the PV update scheme. In PBIL in Section 4.1, the PV is updated using the best ever found solution (i.e. global search information) and a proportion of promising solutions of the each generation (i.e. local search information). In iPBIL, the PV is updated via a set of historically best solutions, which is focused on improving the global exploration ability of iPBIL. The second difference is that the entropy-based restart scheme (used in PBIL in Section 4.1) is not adopted in iPBIL due to the feature of the new problem. The third difference is that PV mutation is incorporated into iPBIL to collaborate with the new PV scheme to balance exploration and exploitation.

1. **Initialization**
2. Set $t = 0$;
3. **for** $i = 1$ **to** L **do**
4. Set $p_i^t = 0.5$;
5. Generate a sampling set $S(t)$ of N individuals from $\boldsymbol{PV}(t)$;
6. Generate a set S_{BSF} of H individuals by sampling $\boldsymbol{PV}(t)$;
7. Replace infeasible individuals in S_{BSF} by all-one vectors;
8. **repeat**
9. Set $t = t + 1$;
10. Evaluate the samples in $S(t\text{-}1)$;
11. Update $\boldsymbol{PV}(t)$ by using the PV update scheme (Figure 5.2);
12. Mutate $\boldsymbol{PV}(t)$ by Equation (5.10);
13. Generate a set $S(t)$ of N samples by $\boldsymbol{PV}(t)$;
14. **until** *termination condition is met*

Figure 5. 3 Procedure of iPBIL (Xing and Qu, 2011b)

5.5 Experimental Results and Discussion

This section first introduces the experimental datasets and then studies the problem features using evolutionary approaches. Finally, the performance of iPBIL is evaluated.

5.5.1 The Experimental Datasets

As aforementioned, the delay constrained NCRM problem has not been considered in the literature. No benchmark dataset is exclusively available. Fortunately, this problem is a variant of the NCRM problem (see Section 5.2 and 5.3). The difference between them is whether the delay constraint is included as a constraint. So, we can modify the benchmark instances for the NCRM problem to adapt for the delay constrained NCRM problem. As discussed in Section 5.2, each link $e \in G$ is associated with a propagation delay and each coding operation is associated with a processing delay in order to calculate the transmission delay of each path from source s to receiver $t_k \in T$ in the constructed NCM subgraph $G_{s \to T}$. In the thesis, the propagation delay of link e is uniformly distributed in the range [2ms, 10ms] (Xu, 2011). The computing time for doing a coding operation should be less than any link delay (Benslimane, 2007; Harte, 2008). For simplicity, we assume any coding operation consumes processing time of 1ms. The experimental networks and instance parameters are given in Table 5.1. Detailed instance description can be found at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm.

5.5.2 Problem Investigation

The delay constrained NCRM problem is a new optimization problem, where the value of delay constraint is an important issue. Xu (2011) studies the traditional delay constrained multicast routing problem (where store-and-forward data forwarding scheme is used) and set two different delay constraints to each instance. A reference solution is chosen and its corresponding delay ($Delay_{ref}$) is used to determine the two constraint values which are $0.9 \cdot Delay_{ref}$ and $1.1 \cdot Delay_{ref}$, respectively. In our experiments, the all-one solution is used as a reference as it heavily affects the performance of EAs. If the all-one solution is regarded as feasible solution, the evolutionary search is guaranteed to

start with at least one feasible solution; otherwise, the search may end up with infeasible solutions. To investigate the delay constrained NCRM problem, we set two delay bounds for each instance, namely $0.9 \cdot Delay_{ref}$ and $1.1 \cdot Delay_{ref}$. Intuitively, a smaller value indicates a severer delay bound. So, we hereafter call $0.9 \cdot Delay_{ref}$ as the severe constraint and $1.1 \cdot Delay_{ref}$ as the loose constraint. In order to investigate problem features, we adapt three EAs for this problem, namely PBIL in Section 4.1 (we hereafter simply call it PBIL), BLSGA and BTSGA (see Section 4.3.2). The predefined number of generation is set to 200. The parameter settings of the three algorithms are shown in Table 5.2. All experiments were run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM. The results were collected by running each algorithm 50 times. A number of performance metrics are used, as listed below.

Table 5. 1 Experimental Networks and Instance Parameters

Network	nodes	links	sinks	rate	L
3-copy	25	36	4	2	32
7-copy	57	84	8	2	80
15-copy	121	180	16	2	176
31-copy	249	372	32	2	368
Rnd-1	20	37	5	3	43
Rnd-2	20	39	5	3	50
Rnd-3	30	60	6	3	86
Rnd-4	30	69	6	3	112
Rnd-5	40	78	9	3	106
Rnd-6	40	85	9	4	64
Rnd-7	50	101	8	3	145
Rnd-8	50	118	10	4	189
Rnd-9	60	150	11	5	235
Rnd-10	60	156	10	4	297

Note: L: the encoding length of individuals

Table 5. 2 Parameter Settings of BLSGA, BTSGA and PBIL

BLSGA	BTSGA	PBIL
$p_c = 0.8$;	$p_c = 0.8$;	$\alpha = 0.1$;
$p_m = 0.006$;	$p_m = 0.012$;	$\tau = 0.2$;
$tsize = 2$;	$tsize = 2$;	$\Theta = 0.14$;

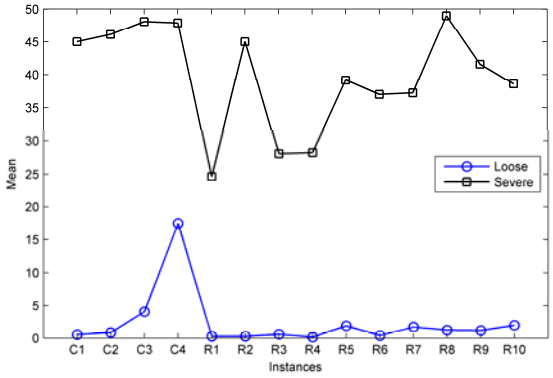
Note: p_c: crossover probability; p_m: mutation probability; t_{size}: tournament size; α: learning rate; τ: the percentage of the selected solutions; Θ: entropy threshold value.

- Mean and standard deviation (SD) of the best solutions found over 50 runs. One best solution is obtained in one run. The mean and SD are important metrics to show the overall performance of a search algorithm.
- Successful ratio (SR) of finding an optimal solution in 50 runs. This metric reflects the global exploration ability of an EA to find optimality.
- Evolution of the average fitness and the best fitness over generations in 50 runs. The plots illustrate the convergence features of an algorithm.
- Average computational time (ACT) consumed by an algorithm over 50 runs.

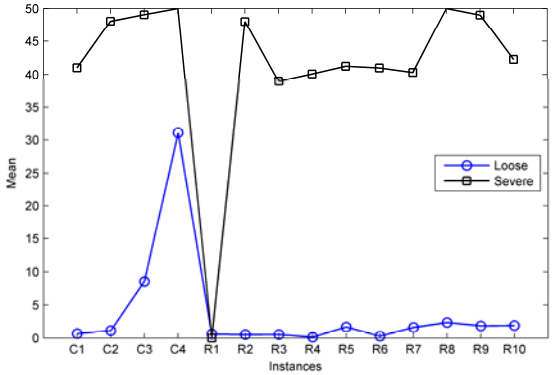
Table 5. 3 Experimental Results of BLSGA, BTSGA and PBIL (The Best Results are in Bold)

	BLSGA				BTSGA				PBIL			
	Loose		Severe		Loose		Severe		Loose		Severe	
Instances	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
3-copy	0.54	1.16	45.04	15.03	0.66	1.25	41.04	19.31	**0.00**	0.00	**0.00**	0.00
7-copy	0.84	2.29	46.16	13.16	1.12	2.59	48.04	9.70	**0.00**	0.00	**0.02**	0.14
15-copy	4.04	7.45	48.12	9.30	8.52	9.35	49.06	6.64	**0.52**	0.70	**0.58**	0.73
31-copy	17.44	9.75	47.92	8.35	31.06	8.52	50.00	0.00	**8.90**	2.20	**9.88**	2.42
Rnd-1	0.28	0.96	24.62	33.16	0.60	1.21	**0.00**	0.00	0.04	0.19	0.04	0.19
Rnd-2	0.28	0.70	45.04	15.03	0.52	0.88	48.00	9.89	**0.00**	0.00	**0.00**	0.00
Rnd-3	0.58	1.17	28.06	25.00	0.54	1.16	39.00	20.92	**0.00**	0.00	**0.00**	0.00
Rnd-4	0.16	0.37	28.24	24.80	0.14	0.35	40.10	20.00	**0.00**	0.00	**0.00**	0.00
Rnd-5	1.86	3.26	39.16	20.63	1.66	3.16	41.26	18.86	**0.10**	0.30	**0.10**	0.30
Rnd-6	0.38	0.49	37.00	22.15	0.26	0.44	41.00	19.40	**0.00**	0.00	**0.00**	0.00
Rnd-7	1.68	2.14	37.22	21.78	1.60	2.15	40.32	19.57	**0.36**	0.48	**0.38**	0.49
Rnd-8	1.22	2.07	49.00	7.07	2.36	2.57	50.00	0.00	**0.30**	0.54	**0.50**	0.54
Rnd-9	1.16	1.51	41.60	18.17	1.84	1.74	49.00	7.07	**0.28**	0.88	**3.22**	11.95
Rnd-10	1.96	3.93	38.58	20.59	1.88	3.44	42.28	17.89	**0.04**	0.28	**0.10**	0.36

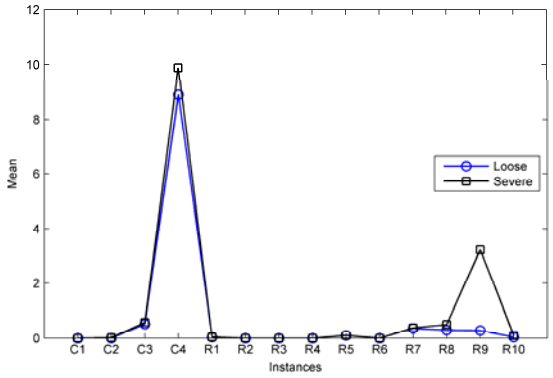
Table 5.3 shows the mean and SD values of BLSGA, BTSGA and PBIL, where columns ‘Loose’ and ‘Severe’ correspond to loose and severe constraints, respectively. If we compare the mean values in Loose and Severe cases, we can find that the value of delay bound does affect the performance of each algorithm, namely the severe case is more difficult to address (see columns *Loose*, *Severe* of the same algorithm).



(a) BLSGA

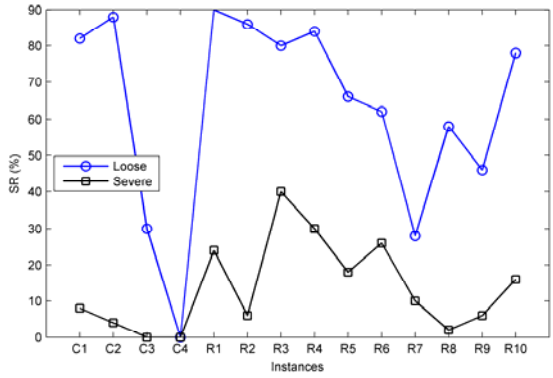


(b) BTSGA

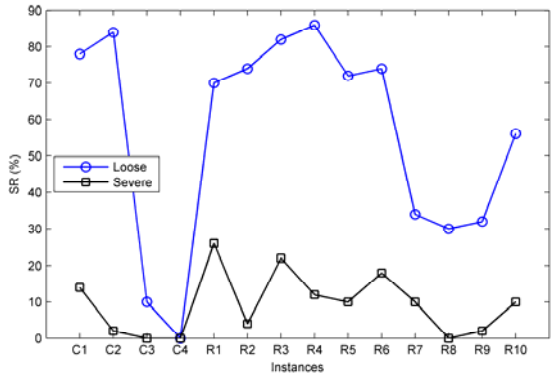


(c) PBIL

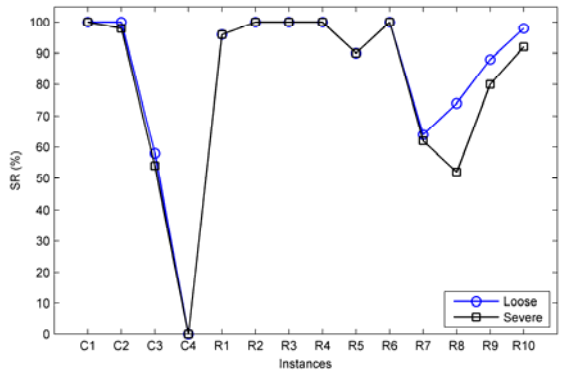
Figure 5. 4 Mean values obtained in Loose and Severe cases. A smaller mean value indicates a better performance. Note that along X axis, C1 to C4 represent 3,7,15,31-copy, and R1 to R10 denote Rnd-1 to Rnd-10, respectively. The instance order along X axis is the same as in Table 5.1.



(a) BLSGA



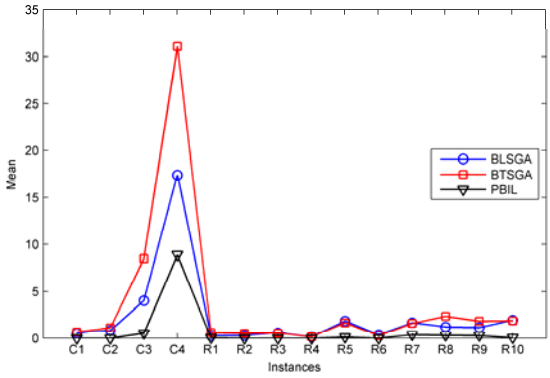
(b) BTSGA



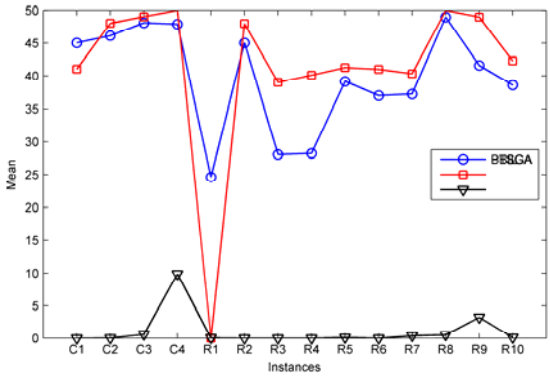
(c) PBIL

Figure 5. 5 Successful ratios (SRs) obtained in Loose and Severe cases. A higher SR indicates a better performance. Note that along X axis, C1 to C4 represent 3,7,15,31-copy, and R1 to R10 denote Rnd-1 to Rnd-10, respectively. The instance order along X axis is the same as in Table 5.1.

Figure 5.4 and 5.5 provide a visible comparison on the mean values and successful ratios obtained in Loose and Severe cases. The mean values and SRs in Loose case are usually better than that in Severe case, which indicates that the problem with severer delay bound is harder. Moreover, we observe that the performance of PBIL is much more stable than BLSGA and BTSGA (see Figure 5.4(c) and 5.5(c)). As in severer cases, there are less feasible solutions in the search space. But, BLSGA and BTSGA are not good at locating and producing feasible solutions. So, once the delay bound is set severely, the performance of BLSGA and BTSGA is seriously weakened. On the other hand, PBIL can effectively locate feasible solutions and make use of them to generate promising solutions. So, PBIL is less affected than the GAs.

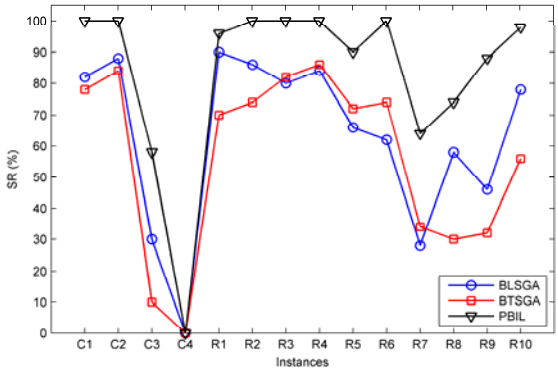


(a) Loose case

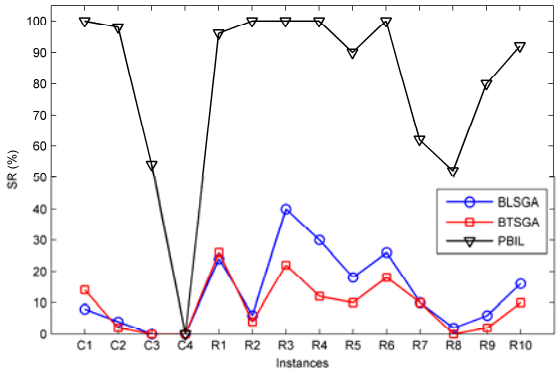


(b) Severe case

Figure 5. 6 Comparison of the mean values of BLSGA, BTSGA and PBIL.



(a) Loose case



(b) Severe case

Figure 5. 7 Comparison of the successful ratios of BLSGA, BTSGA and PBIL.

Figure 5.6 and 5.7 show the mean values and successful ratios of BLSGA, BTSGA and PBIL. In Loose case, the three algorithms have similar performance, although PBIL performs slightly better in all instances. In Severe case, PBIL outperforms the others. This is mainly because PBIL locates promising regions more easily, as discussed above. Moreover, the adopted restart scheme also helps to improve the global exploration ability of PBIL. So, when adapted, PBIL also reflects a relatively decent performance. Moreover, one may also notice that some instances are harder than the others, namely 15,31-copy and Rnd-7,8,9 on which the three algorithms obtain relatively larger mean values and lower successful ratios. This is observed by the mean values and successful ratios of each algorithm in Loose/Severe case (see the mean curve and successful ratio curve of the same algorithm). PBIL also gains relatively worse results in the above five instances. As

discussed in Section 5.4, PBIL with the entropy-based restart scheme may expose some weaknesses when used to address the delay constrained NCRM problem, mainly due to two reasons. The first one is the threshold value in the restart scheme is set empirically and deliberately for the NCRM problem, which may not suitable for all instances of the new problem. The other reason is that the restart scheme would force the search to start from scratch each time, without the guidance of the promising solutions previously found. This would be a problem as in severe case the all-one solution (which helps the search to locate feasible regions) becomes infeasible and so the search may start with infeasible solutions only. Therefore, another approach is developed to replace the restart scheme, as introduced in Section 5.4. The following subsection evaluates the PBIL based the new approach (namely iPBIL).

5.5.3 Performance Evaluation of iPBIL

iPBIL is an adaptation of PBIL in Section 5.3 for the delay constrained NCRM problem (see Section 5.4). In iPBIL, the restart scheme is replaced with a combination of two schemes, namely a new PV update scheme and a PV mutation. We evaluate the performance of iPBIL by comparing it and PBIL (in Section 5.3) on all instances in Table 5.1 where there are two delay bounds (Loose and Severe) in each instance. The maximal number of generations and population size are set to 200 and 20, respectively. The parameter settings are given in Table 5.4. All experiments were run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM. The results were collected by running each algorithm 50 times.

Table 5. 4 Parameter Settings of BLSGA, BTSGA and PBIL

PBIL	PBIL
$\alpha = 0.1$;	$\alpha = 0.1$;
$\tau = 0.2$;	H = 10;
$\Theta = 0.14$;	$p_m = 0.02$;
	$\varepsilon = 0.05$

Note: α: learning rate; τ: the percentage of the selected solutions; Θ: entropy threshold value; H: the number of the best ever found solutions maintained; p_m: mutation probability; ε: the mutation amount in each position in the PV.

Table 5. 5 The Mean and SD values of PBIL and iPBIL (Best Results are in Bold)

	Loose				Severe			
	PBIL		iPBIL		PBIL		iPBIL	
Instances	Mean	SD	Mean	SD	Mean	SD	Mean	SD
3-copy	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
7-copy	**0.00**	0.00	**0.00**	0.00	0.02	0.14	**0.00**	0.00
15-copy	0.52	0.70	**0.04**	0.19	0.58	0.73	**0.06**	0.23
31-copy	8.90	2.20	**0.62**	0.87	9.88	2.42	**0.86**	0.92
Rnd-1	0.04	0.19	**0.00**	0.00	0.04	0.19	**0.00**	0.00
Rnd-2	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-3	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-4	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-5	0.10	0.30	**0.00**	0.00	0.10	0.30	**0.00**	0.00
Rnd-6	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00
Rnd-7	0.36	0.48	**0.08**	0.27	0.38	0.49	**0.12**	0.32
Rnd-8	0.30	0.54	**0.06**	0.23	0.50	0.54	**0.30**	0.46
Rnd-9	0.28	0.88	**0.06**	0.23	3.22	11.95	**0.12**	0.32
Rnd-10	0.04	0.28	**0.00**	0.00	0.10	0.36	**0.04**	0.19

Table 5. 6 The Successful Ratios (SRs) of PBIL and iPBIL (Best Results are in Bold)

	Loose		Severe	
Instances	PBIL	iPBIL	PBIL	iPBIL
3-copy	**100**	**100**	**100**	**100**
7-copy	**100**	**100**	98	**100**
15-copy	58	**98**	54	**94**
31-copy	0	**58**	0	**44**
Rnd-1	96	**100**	96	**100**
Rnd-2	**100**	**100**	**100**	**100**
Rnd-3	**100**	**100**	**100**	**100**
Rnd-4	**100**	**100**	**100**	**100**
Rnd-5	90	**100**	90	**100**
Rnd-6	**100**	**100**	**100**	**100**
Rnd-7	64	**92**	62	**88**
Rnd-8	74	**94**	52	**70**
Rnd-9	88	**94**	80	**88**
Rnd-10	98	**100**	92	**96**

The comparison of mean and SD values are shown in Table 5.5. It is easily seen that iPBIL gains better overall performance than PBIL. In Loose case, iPBIL outperforms PBIL in eight instances (compare mean values in rows *15,31-copy, Rnd-7,8,9,10*, column *Loose*) while in Severe case, the former beats the latter in nine instances (compare mean values in rows *7,15,31-copy, Rnd-5,7,8,9,10*, column *Severe*). Similarly, the results of successful ratio (SR) can also reflect the superiority of iPBIL over PBIL, as given in Table 5.6. The following explains the underlying reason of the above results. As discussed in Section 5.4, the adapted PBIL in Section 5.3 may not be fit for the problem concerned in this chapter due to the adopted restart scheme. This scheme is designed to

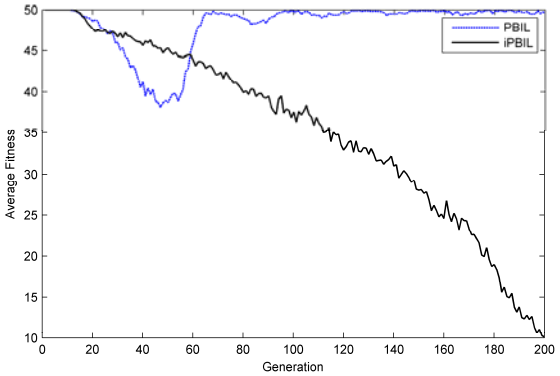
improve the global exploration ability of PBIL for the NCRM problem, where the restart threshold value is set based on initial tests and observations. So, the setting of threshold value may not match the situation in delay constrained NCRM problem. Besides, that restart scheme is start-from-scratch based and completely ignores the previously found promising solutions which may provide useful guidance to facilitate effective search. In this way, the evolution is divided into several separate search stages where neither coherence nor continuity is reflected. Because of the delay constraint, it would be more difficult for the search to locate feasible solutions. Hence the promising solutions found during the evolution should be well exploited. On the other hand, iPBIL is featured with a new PV update scheme and a PV mutation operation. The new update scheme makes full use of the historically best solutions (i.e. promising solutions during the evolution), which is designed to fit for the new optimization problem. The PV mutation introduces small disturbance to the PV. The combination of the two schemes not only explores the most possible promising areas based on the global search information but also avoid the search getting stuck into local optima. Therefore iPBIL performs better than PBIL.

Figure 5.8 illustrates an example that compares the evolutionary features of PBIL and iPBIL. We choose 31-copy instance because it is the most difficult problem and clearly shows the differences between PBIL and iPBIL. In the figure, the evolution of the average fitness (AF) and the best ever found fitness (BF) in 50 runs is reported. First, we analyze the evolution of the average fitness (see subfigures (a) and (b)). In Loose case, PBIL and iPBIL have the similar performance before 15th generation, namely a horizon line. In both algorithms, PV is initialized as {0.5,…,0.5} which means it generates 0 and 1 with equal probability at each position. As previously discussed, solutions closer to the all-one solution are more likely to be feasible. So, the majority of the solutions in the early generations are infeasible because the PV has to be gradually adjusted before producing feasible solutions. After that the PVs in both algorithms are able to produce feasible solutions and hence the AFs both decrease until around 40th generation. In this period, PBIL converges faster than iPBIL. This is because of the underlying difference between the two PV update schemes adopted. For PBIL, the best ever found solution and a proportion of the promising solutions of the current generation are used to update the PV, which is mainly based on the local search information. For iPBIL, the best ever

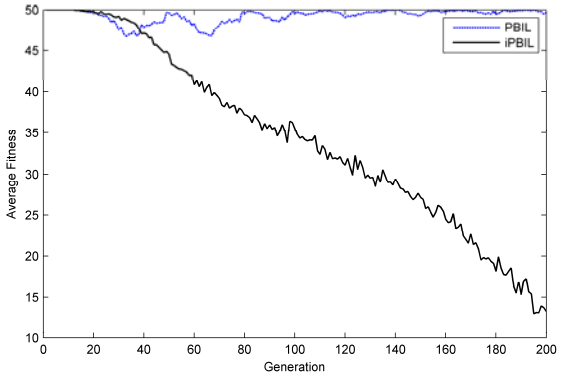
found solutions are used to adjust the PV, which is based on the global search information. So, the search of PBIL is easier to intensify the search in a smaller and smaller area, which causes the rapid convergence. From the 40th generation, the AF of PBIL increases to 50 and fluctuates. This is because once the restart criteria is met the PV is re-set to {0.5,…,0.5}, where infeasible solutions are more likely to be sampled first. As the restart points are not fixed during the evolution and in 50 runs, the AF of PBIL fluctuates around the value of 50. Note that the fitness value of any infeasible solution is set to 50 in this chapter. On the other hand, the AF of iPBIL gradually goes down along generations, which means the focus of the search is gradually changing from global exploration to local explotation. The evolutionary features of the AFs of PBIL and iPBIL in Severe case are similar to those in Loose case. However, there are some differences between the Loose and Severe cases. This time, we compare the AFs of the same algorithm in two different cases. For PBIL and iPBIL, the first AF decline in Severe case happens around 20th generation. This is later than in Loose case as it is more difficult to generate feasible solutions with severer delay bound. For PBIL, the AF in Severe case has smaller fluctuations than that in Loose case. The above is also because the severer delay bound introduces extra difficulty in locating feasible regions. On the other hand, for iPBIL, the AF is similar to that in Loose case after the 20th generation. This is mainly because the exploitation of the best ever found solutions which provide useful guidance to the search.

Then, we analyze the evolution of the best ever found fitness (see Figure 5.8 (c) and (d)). We start from the Loose case, where the all-one solution is feasible. In the early generations, the BF of PBIL and iPBIL is a constant value around 39. As aforementioned, the PV cannot produce feasible solutions in early generations because it needs time (i.e. evolution) to learn the PV from the all-one solution. So, before the PV is able to produce feasible solutions, the BF is the fitness value of the all-one solution which is the only feasible solution at the beginning of the evolution. Once the search is guided into feasible regions, fitter solutions are generated and hence the BF of PBIL and that of iPBIL both decline. The BF of PBIL has a faster convergence until around 80th generation and then it suffers a slower convergence, compared with that of iPBIL. The following explains why. On the one hand, The PV in PBIL is updated mainly based on the local search

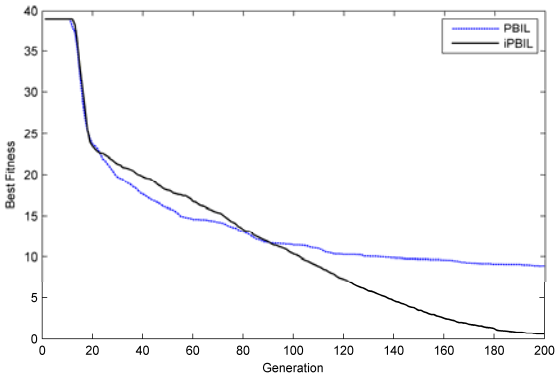
information so the search easily converges to a small region in the search space. And the local exploitation helps to find a locally promising solution relatively quickly. Then the restart scheme helps the search escape from the region being searched. This process is repeated until the end of the evolution. However, the restart scheme cannot effectively improve the BF due to its deficiencies such as the ignorance of previously found promising solutions. On the other hand, the collaboration between the new PV update scheme and PV mutation scheme contributes to a better global exploration and improves the BF consistently. Therefore, iPBIL finally achieves better BF than PBIL. In Severe case, the evolutionary features of the BFs are similar to those in Loose case, which also indicates the superiority of iPBIL over PBIL.



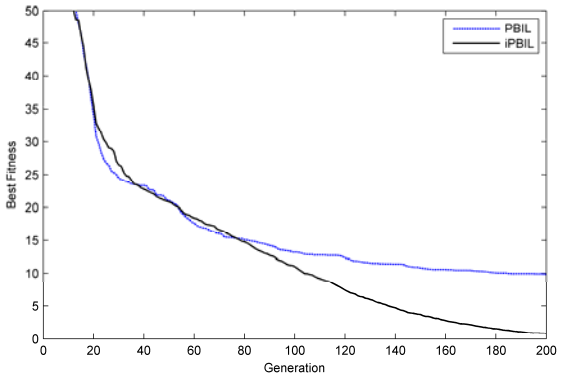
(a) Average fitness vs. generation in Loose case



(b) Average fitness vs. generation in Severe case



(c) Best fitness vs. generation in Loose case



(d) Best fitness vs. generation in Severe case

Figure 5. 8 Evolution of PBIL and iPBIL in 31-copy instance.

Table 5. 7 The Average Computational Time (ACT) of PBIL and iPBIL (sec.) (The Best Results are in Bold)

	Loose		Severe	
Instances	PBIL	iPBIL	PBIL	iPBIL
3-copy	0.10	**0.09**	**0.16**	**0.16**
7-copy	**2.82**	3.58	**4.01**	7.60
15-copy	**40.06**	53.00	**39.62**	66.74
31-copy	**233.83**	863.86	**212.82**	689.77
Rnd-1	0.94	**0.39**	1.34	**0.57**
Rnd-2	**0.12**	0.16	0.15	**0.13**
Rnd-3	**0.19**	0.24	0.23	**0.21**
Rnd-4	**0.25**	0.52	0.49	**0.30**
Rnd-5	8.37	**7.13**	**7.49**	8.24
Rnd-6	**0.16**	0.19	**0.17**	0.19
Rnd-7	21.08	**15.10**	20.73	**16.32**
Rnd-8	30.18	**28.98**	**38.19**	50.12
Rnd-9	61.57	**60.83**	**61.11**	68.64
Rnd-10	**40.66**	42.91	**41.36**	53.09

The results of the average computational time (ACT) are collected in Table 5.7. It is seen that iPBIL does not show advantage in ACT. Rather, it consumes much more ACT in some instances, such as 15-copy and 31-copy. There are mainly two issues that relate to the ACT, namely the fitness evaluation and the termination condition. In the fitness evaluation, a solution is evaluated based on three steps, i.e. check if the data rate requirement is met, and if so check if the delay constraint is met, and if so compute the number of coding operations performed. In the first step, the receivers are checked one by one and the evaluation stops if some receiver is found that does not meet the data rate requirement. There is no need to check the rest of the receivers if some receiver already violates the data rate requirement. So, the time spent on fitness evaluation may vary dramatically. In the termination condition, each algorithm stops either a coding-free NCM subgraph is found or a predefined number of generations are reached. A higher successful ratio may indicate the fewer fitness evaluations are used. In Loose and Severe cases, PBIL spends much less ACT in 15-copy, 31-copy than iPBIL. For PBIL, the fitness evaluation in the short period after each restart consumes less time. This is because most of the solutions generated by the PV do not meet the data rate requirement. On the other hand, iPBIL produce more feasible solutions and the fitness evaluation is hence more expensive.

5.6 Summary

This chapter formulates the delay constrained NCRM problem which is a variant of the NCRM problem. Delay constraint is for the first time introduced into the NCRM problem because the transmission delay is one of the most important QoS parameters in supporting real time multimedia multicast applications. The benchmark instances for the NCRM problem are adapted for the new problem. We study the new problem by adapting three EAs for it, namely GAs based on binary encodings and PBIL in Section 4.1. The adaptation process is easy as only minor modifications are made to the fitness evaluation. Two delay bounds are considered in the experiments (namely the Loose and Severe cases). We find that for the same instance, a severer delay constraint indicates a more difficult problem. Besides, PBIL gains better performance than GAs in terms of the best solutions obtained and successful ratio. Meanwhile, we notice that the restart scheme used in PBIL has two deficiencies when used in the delay constrained problem. One is that the threshold value is deliberately designed for the NCRM problem. The other is the memory-less structure that cannot make use of the promising solutions previously found to effectively guide the search. Hence we develop a substitute for the restart scheme, namely the combination of a new PV update scheme and a PV mutation. In the new PV update scheme, the PV is adjusted based on a group of historically best solutions, which improves the global exploration ability of PBIL. The PV mutation is adopted to avoid local optima. The experimental results demonstrate that PBIL with the new PV update scheme and PV mutation outperforms PBIL with restart scheme in terms of the global exploration ability.

The work has been partially published at EvoStar 2011 (Xing and Qu, 2011b).

Chapter 6 NSGA-II for the Cost-Delay Bi-objective Optimization Problem

6.1 Motivation

As discussed in Chapter 5, the delay constrained NCRM problem considers not only the computational cost but also the user experience. The problem, to some extent, reflects the expectation of service providers and network users (Xing and Qu, 2011b). Intuitively, service providers and network users have conflicting interests, i.e. the former for cheap data delivery and communications while the latter for better service quality (Pu *et al*, 2009). When investigating the interests of service providers and network users, the decision maker (or network administrator) may need to compromise between the two sides (Wang *et al*, 2006; Xu, 2011).

As reviewed in Sections 1.4.2 and 1.5.3, in the course of NCM data transmission, two types of costs will incur, coding cost and link cost (Kim *et al*, 2007c). Coding cost denotes the amount of coding operations performed while link cost represents the cost of those links leased by the NCM. From the point of view of service providers of a network, the total cost (i.e. sum of coding and link costs) can estimate the network resources consumed for NCM more precisely than coding cost or link cost individually (Kim *et al*, 2007c). A smaller total cost indicates that less network resources are consumed. Therefore, it is quite important to minimize the total cost of a NCM. However, this issue has not been taken into account in the literature. On the other hand, the transmission delay is another important issue in NCM that needs to be carefully concerned (see Sections 1.5.2 and Chapter 5). From the point of view of end users of NCM, especially those requiring real-time multimedia applications, the transmission delay should be kept as low as possible (i.e. minimized) (Zhang *et al*, 2009). Simultaneously minimizing total cost and minimizing transmission delay are two natural objectives representing the interests of the both, respectively. Between the two objectives, one is to obtain cheaper multicast solutions which may degrade the service quality; while the other is to improve

service quality that might increase the cost. Intuitively, the two objectives may reflect a trade-off between the interest of service providers and that of network users. If so, the trade-off results can be used to achieve a well compromised NCM solution. This motivates us to formulate a cost-delay bi-objective optimization (CDBO) problem in this thesis (Xing and Qu, 2013). By studying the above problem, one may obtain a trade-off between the interests of service providers and network users. The resulting trade-off can help the decision maker to balance the two sides. In addition to the problem modeling, we also initiate the investigation into the problem using adapted MOEAs.

In this chapter, we first introduce the detailed modelling of the CDBO problem and then study it from the view of evolutionary computation.

6.2 Problem Description

Based on the network model introduced in Section 3.1, we for the first time define the bi-objective combinatorial optimization problem as to build a NCM subgraph $G_{s\rightarrow T}$ while simultaneously minimizing two objectives (the total cost C_{total} and the maximum transmission delay D_{max}) and achieving the demanded data rate. This problem is a variant of the NCRM problem. They are based on the same network model in Section 3.1. The following lists the notations used:

R: the defined data rate (an integer) at which s expects to transmit information to each receiver t_k.

$\lambda(s, t_k)$: the data rate between s and t_k in $G_{s\rightarrow T}$.

$e(u, v)$: directed link from node u to node v.

$G_{s\rightarrow T}$: network coding based multicast (NCM) subgraph.

n_{CN}: the number of coding nodes in $G_{s\rightarrow T}$.

v_c^i: the i-th coding node in $G_{s\rightarrow T}$.

$c_{code}(v_c^i)$: cost incurred at v_c^i, $i = 1,\ldots, n_{\mathrm{CN}}$. The more data streams involved in the coding operation in n, the larger $c_{code}(v_c^i)$.

$c_{link}(e(u,v))$: cost of link $e(u,v)$ if $e(u,v) \in G_{s\rightarrow T}$.

$P_i(s, t_k)$: the i-th link-disjoint path found between s and t_k in $G_{s\rightarrow T}$. It is represented by a chain of nodes through which the path passes.

$W_i(s, t_k)$: the link set of $P_i(s, t_k)$, i.e. $W_i(s, t_k) = \{e(u,v) \mid e(u,v) \in P_i(s, t_k)\}$.

$D(P_i(s, t_k))$: the end-to-end transmission time of path $P_i(s, t_k)$.

Based on the notations above, we define the CDBO optimization problem below. It is noted that

Minimize:

$$C_{total} = \sum_{i=1}^{n_{CN}} c_{code}(v_c^i) + \sum_{\forall e(u,v) \in G_{s \to T}} c_{link}(e(u,v)) \tag{6.1}$$

$$D_{\max} = \max\{D(P_i(s, t_k)) \mid i = 1,\ldots,R, \ \forall t_k \in T\} \tag{6.2}$$

Subject to:

$$\lambda(s, t_k) = R, \ \forall \ t_k \in T \tag{6.3}$$

$$W_i(s, t_k) \cap W_j(s, t_k) = \emptyset, \ \forall i,j \in \{1,...,R\}, \ i \neq j, \ t_k \in T \tag{6.4}$$

Objective (6.1) is to minimize the total cost of the NCM subgraph, which consists of the coding cost and link cost; Objective (6.2) is to minimize the transmission delay of the NCM subgraph. Please see Section 5.2 for the calculation of $D_{\max}$. Constraint (6.3) defines that in $G_{s \to T}$ the data rate from s to each receiver is R; Constraint (6.4) restricts that the R paths from s to an arbitrary receiver are link-disjoint.

The CDBO problem belongs to MOPs. Given two solutions $\mathbf{X}_1$ and $\mathbf{X}_2$ to the CDBO problem, we say $\mathbf{X}_1$ dominates $\mathbf{X}_2$ (or $\mathbf{X}_2$ is dominated by $\mathbf{X}_1$) if either of the following two conditions holds (according to the Dominance definition in Section 2.1.2):

- $\mathbf{X}_1$ is no worse than $\mathbf{X}_2$ in C_{total} and in D_{max};
- $\mathbf{X}_1$ is strictly better than $\mathbf{X}_2$ in at least one objective, namely C_{total} or D_{max}.

6.3 MOEAs for Multi-Objective Multicast Routing Problems

The nature of the CDBO problem is a multi-objective multicast routing problem (MOMRP) in communications networks since its ultimate goal is to provide high-quality multicast (i.e. one-to-many) services from the source to a number of receivers using

network coding. This section reviews the MOEAs adapted for multi-objective multicast routing problems (MOMRPs) and selects a MOEA for studying the CDBO problem.

NSGA-II has been applied to a number of MOMRPs. Kim *et al* (2007c) studied the trade-off between coding and link costs in NCM data transmission. Like the CDBO problem, this problem can also be viewed as a variant of the NCRM problem (Kim *et al*, 2007a) (also see Section 1.4.2). The similarity between the two problems is that both of them are NCM routing problems and based on the same network model, where NCM subgraphs are to be constructed for data delivery. The difference between them is that the total cost and maximum transmission delay are two objectives in the CDBO problem while the coding cost and link cost are two objectives in the work of Kim *et al* (2007c). NSGA-II was successfully adapted for the coding and link costs trade-off problem. Montoya *et al* (2008) formulated a multi-objective routing model for multicast overlay networks and investigated the model by NSGA-II, where the total end-to-end delay and the link usage are two objectives to be minimized at the same time. Han *et al* (2008) incorporated the security metric into the Quality-of-Service (QoS) multicast routing and modelled the problem in a multiobjective manner, where the transmission delay, bandwidth utilization and security parameters are objectives to be optimized simultaneously. NSGA-II was adopted to tackle the problem. Jain and Sharma (2012) emphasized the QoS MOMRP where cost and available bandwidth were objectives while delay and delay jitter were constraints. A NSGA-II base on tree structured encoding was developed to solve the problem.

SPEA and its improved version SPEA2 have also been used to solve MOMRPs. Crichigno and Barán (2004) applied SPEA to study the MOMRP for traffic engineering, where four objectives were to be minimized simultaneously, including the link utilization, total cost, maximum and mean of the transmission delay. Bueno and Oliveira (2010) also studied the same MOMRP above via SPEA2, where two problem specific heuristics were incorporated to improve the performance of SPEA2.

The following introduces MOEAs base on other techniques including ant colony optimization (ACO), particle swarm optimization (PSO), artificial immune system (AIS), and simulated annealing (SA). ACO was adopted for addressing the MOMRPs, where the cost, average and maximum delay were three objectives to be minimized (Pinto *et al*,

2005; Pinto and Barán, 2006). Báez and Pinto (2009) studied a MOMRP with wavelength assignment in optical networks. Four parameters are chosen as objectives according to the nature of the problem concerned, including the hop count, number of splitting power light, number of splitter node and balancing for a number of multicast demands. PSO was used to address the problem. Another PSO based MOEA was proposed for QoS MOMRP in wireless mesh networks (Zhuo *et al*, 2010). Wang *et al* (2006) proposed a MOEA based on immune optimization for solving a QoS MOMRP with three objectives, the transmission delay, cost of multicast tree and bandwidth utilization. Specifically, the principle of AIS was adopted in the design the MOEA. Xu and Qu (2010) presented a MOEA hybridizing SA and problem specific neighbourhood structures for tackling the MOMRP, where cost, delay and link utilization are conflicting objectives. Recently, a hybrid MOEA combining SA and genetic local search has been developed and used to solve MOMRP with many objectives (Xu *et al*, 2013).

On the other hand, it is noted that some state-of-the-art MOEAs, such as MOEA/D in Section 2.3.2, have not been used in the field of MOMRPs. However, these MOEAs may also be used for addressing MOMRPs due to their excellent performance in solving multi-objective combinatorial optimization problems (Li and Landa-Silva, 2011; Konstantinidis and Yang, 2012).

According to the review above, there are a number of MOEAs that may be suitable for studying the CDBO problem, such as NSGA-II, SPEA2, ACO, PSO, AIS, MOEA/D. In this chapter, NSGA-II is selected to be adapted for the CDBO problem as explained below. There have been several attempts using NSGA-II for MOMRPs, especially for a variant of the NCRM problem (Kim *et al*, 2007c). The success of NSGA-II in solving this variant and the similarity between it and the CDBO problem encourage the adaptation of NSGA-II for the CDBO problem. Admittedly, NSGA-II is not viewed as state-of-the-art MOEA nowadays and its performance may be beaten by more advanced MOEAs such as MOEA/Ds introduced in Section 2.3.2. However, it is still applicable to investigate the features of the new problem from the point of view of MOEA. So, we adapt NSGA-II for the CDBO problem and expect this piece of initial work could advocate more research efforts spent in developing fitter MOEAs for the CDBO problem.

6.4 An Overview of the Original NSGA-II

Let **X** be an individual (solution) and $f_i(\mathbf{X})$, $i = 1,\ldots,M$, be the value of the i-th objective of **X**, where M is the number of objectives. Let N denote the population size. Let $P(t)$ and $Q(t)$ be the parent population (where elite solutions are kept) and the offspring population at generation t, respectively. The pseudo code of NSGA-II is shown in Figure 6.1. More details can be found in Deb *et al* (2002).

1. **Initialization**
2. Set $t = 0$ and $Q(t) = \emptyset$;
3. Randomly create $P(t)$;
4. **repeat**
5. Set $t = t + 1$;
6. Set $S(t) = P(t) \cup Q(t)$;
7. Sort individuals in $S(t)$ by using two attributes: (1) the nondomination rank and (2) the crowding distance;
8. Set $P(t) = \emptyset$ and put the first N individuals into $P(t)$;
9. Select N individuals from $P(t)$ to fill the mating pool by using tournament selection with crowded-comparison operator;
10. Implement crossover and mutation to the individuals in the mating pool and store the offspring population in $Q(t)$;
11. **until** *the termination condition is met*
12. Output the nondominated solutions in $P(t)$ and $Q(t)$;

Figure 6. 1 The pseudo-code of NSGA-II (Deb *et al*, 2002)

In the evolution loop, the parent population $P(t)$ and the offspring population $Q(t)$ are combined together so that the elite solutions (individuals) already obtained cannot be lost. In step 7, all individuals in the combined population $S(t)$ are evaluated (e.g. objective values $f_i(\mathbf{X})$, $i = 1,\ldots, M$, for individual **X** are calculated) and sorted into different nondomination levels according to the individual dominance scheme (Deb *et al*, 2002). Individuals that are not dominated by any other individual are assigned rank 1. Those dominated by only the individuals of rank 1 are assigned rank 2, and this recursive process iterates until all individuals in $S(t)$ have a rank number. The nondominated

sorting approach in NSGA-II has a computational complexity of O(MN^2), where M and N are the number of objectives and population size, respectively.

When comparing two individuals in parent selection, the crowded comparison operator is adopted to guide the search towards the PF (Deb *et al*, 2002). Given two individuals, the one with lower rank value is selected. If the two have the same rank, the crowding distance is used to estimate the density of individuals surrounding them. And the one with larger crowding distance is selected. Step 10 applies crossover (with a crossover probability p_c) and mutation (with a mutation probability p_m) to the individuals in the mating pool to create the offspring population $Q(t)$. The final output is the set of all nondominated individuals in $P(t)$ and $Q(t)$.

6.5 Adaptation of NSGA-II

This section introduces how NSGA-II is adapted for the CDBO problem, including the chromosome representation, objective evaluation, and two problem specific schemes (namely initialization and niching in objective space).

6.5.1 Chromosome Representation and Objective Evaluation

6.5.1.1 Chromosome Representation

Like the delay constrained NCRM problem in Chapter 5 and the coding and link usage trade-off problem (Kim *et al*, 2007c), the CDBO problem is another variant of the original NCRM problem. These problems are based on the same underlying network model introduced in Section 3.1. So, the encoding approaches used in the original NCRM problem and its variants can also be used to represent solutions for the CDBO problem. These encoding approaches include BLS encoding in Section 3.5.1, BTS encoding in Section 3.5.2, and the path-oriented encoding in Section 4.3.1. Among them, BLS encoding has been applied in the original NCRM problem (Kim *et al*, 2007a, 2007b; Xing and Qu, 2011a, 2012), the delay constrained NCRM problem (Xing and Qu, 2011b) and the coding and link usage trade-off problem (Kim *et al*, 2007c). Hence, BLS encoding can be considered in the CDBO problem.

6.5.1.2 Objective Evaluation

To evaluate a given individual **X** in all objectives, we first check if **X** is feasible. The feasibility checking procedure here is the same as that for the original NCRM problem in Section 3.5.3. The following summarizes the procedure. Each individual corresponds to a unique secondary graph G_S. We apply a classical max-flow algorithm (Goldberg, 1985) to compute the max-flow between the source s and an arbitrary receiver $t_k \in T$ in G_S. As mentioned in Section 3.1, each link in G has a unit capacity. The max-flow between s and t_k is equal to the number of link-disjoint paths between s and t_k found by the max-flow algorithm. If all d max-flows are at least R, where d is the number of receivers, rate R is achievable and the individual **X** is feasible. Otherwise, **X** is infeasible.

In the CDBO problem, there are two objectives for minimization, i.e. the total cost C_{total} and the maximum transmission delay D_{max}. For each infeasible individual **X**, we set a sufficiently large value to $C_{total}(\mathbf{X})$ and $D_{max}(\mathbf{X})$, respectively (in this thesis, the value is 10,000). If **X** is feasible, we first find a corresponding NCM subgraph $G_{s \to T}$ and then calculate $C_{total}(\mathbf{X})$ and $D_{max}(\mathbf{X})$. For each receiver $t_k \in T$, we select R paths from the obtained link-disjoint paths from s to t_k (if the max-flow is R then we select all the link-disjoint paths) and thus obtain $R \cdot d$ paths, i.e. $P_i(s, t_k)$, $i = 1,\ldots,R$, $k = 1,\ldots,d$. We map all the selected paths to G_S and obtain a feasible $G_{s \to T}$ based on which the two objectives $C_{total}(\mathbf{X})$ and $D_{max}(\mathbf{X})$ can be easily calculated.

6.5.2 The Initialization Scheme

As discussed in Section 4.1 and 4.2, BLS encoding corresponds to a search space with large number of infeasible solutions when solving the NCRM problem. Solutions closer to the all-one solution are more likely to be feasible since more auxiliary links are active and able to forward the incoming data. To ensure that the search begins with at least one feasible solution, an all-one solution is inserted into the initial population (see Section 4.1 and 4.2 for details).

The feasibility checking criteria for the CDBO problem is the same as the one for the NCRM problem, as described in Section 6.5.1. Hence, when BLS encoding is adopted for the CDBO problem, infeasible solutions account for a high proportion of the search space. Feasible solutions are not evenly distributed over the search space. Instead, they

usually have short (Hamming) distances to the all-one solution. Experimental results can be found in Section 6.6.3. So, it is necessary to include an all-one individual in the initialization in order to guide the search towards feasible regions in the search space. However, inserting an all-one individual may not be sufficient to create a good initial population for the MOP concerned here. With limited number of feasible individuals (e.g. the all-one individual may be the only feasible individual), the population may still not evolve efficiently towards the PF. Hence, we propose a new initialization scheme that not only generates a considerable amount of feasible individuals but also make sure these individuals are diversified. Let $\mathbf{X} = \{x_1, x_2, \ldots, x_L\}$ be an arbitrary individual, where x_i, i = 1, 2, …, L, is the i-th bit of $\mathbf{X}$. Let N be the population size. Let set_{Ini} be the set of generated initial solutions. Figure 6.2 shows the procedure of the new initialization scheme.

1. Set $set_{Ini} = \emptyset$ and add an all-one individual to set_{Ini}
2. **while** $|set_{Ini}| < N$ **do**
3. Randomly select an individual $\mathbf{X}$ from set_{Ini};
4. Randomly generate an integer i in the interval [1, L];
5. Mutate $\mathbf{X}$ by flipping the i-th bit of $\mathbf{X}$ to generate $\mathbf{X}_{new}$;
6. Evaluate the new individual $\mathbf{X}_{new}$;
7. **if** $\mathbf{X}_{new}$ is feasible **then**
8. Add $\mathbf{X}_{new}$ to set_{Ini};
9. **else**
10. Discard $\mathbf{X}_{new}$;
11. Output set_{Ini} as the initial population $P(0)$

Figure 6. 2 The procedure of the proposed initialization scheme (Xing and Qu, 2013)

The new initial scheme enables the proposed NSGA-II to start with a set of feasible individuals, which helps it to obtain nondominated solutions that are as close to the true PF as possible (see Section 6.6.3 for details).

6.5.3 The Individual Delegate Scheme

As introduced in Subsection 3.2.4, each individual $\mathbf{X}$ corresponds to a unique secondary graph G_S. If a valid NCM subgraph is obtained, $\mathbf{X}$ is seen as a feasible

individual; otherwise, **X** is infeasible. For a feasible **X**, to determine its NCM subgraph, the max-flow from the source to each receiver is computed from the corresponding G_S. Clearly, the max-flow to each receiver may remain the same if G_S is slightly changed, e.g. the links and nodes occupied by the max-flow are not removed from G_S. If none of the max-flows is affected, the corresponding NCM subgraph remains unchanged. According to this feature, one may imagine that different secondary graphs may result into the same NCM subgraph, which has also been reflected by Kim *et al* (2007c). In other words, it is possible that similar individuals may have identical objective values (C_{total} and D_{max}).

In the study of the CDBO problem, we notice that NSGA-II is not good at producing feasible individuals. This is because (1) infeasible solutions constitute the majority in the search space, and (2) the major evolutionary force, namely crossover, is not good at locating feasible regions in the search space and hence the offsprings may be infeasible (also see discussion in Section 4.1.2). According to the analysis above, a few feasible individuals are generated at each generation. Hence, duplicates and similar variants of the nondominated individuals will spread in the population because of the elitism strategy and the selection pressure. Note that nondomianted individuals and their similar variants may have identical objective values, as explained in the first paragraph of this section. There is a diversity-preservation mechanism (DPM) in the original NSGA-II (Deb *et al*, 2002). This mechanism selects fitter individuals according to two attributes, the nondomination rank and the crowding distance. Individuals with a lower rank level and a higher crowding distance are more likely to enter the mating pool (see Section 6.4). This was reported to diversify the search towards the whole PF (Deb *et al*, 2002). However, this DPM may not well handle the spread of duplicates and similar variants of the nondominated individuals for the CDBO problem, due to the insufficient supply of feasible individuals during the evolution. Duplicates and similar variants of the nondominated individuals gradually constitute the majority of the population, which deteriorates the population diversity in objective space. This is due to that identical or similar parents are likely to produce identical or similar offspring after recombination. And similar individuals may correspond to a single point in objective space. This may deteriorate the performance of NSGA-II in finding a diversified PF. The search may not be well guided towards the entire PF, e.g. some parts of the PF may not be covered.

Hence, when addressing the CDBO problem, we need to diversify the population not only in objective space (like the original the DPM in NSGA-II) but also in decision space (for producing significant and promising individuals).

We consider the crowding issue in both decision and objective spaces and develop an individual delegate scheme (IDS) to reduce the number of individuals with identical objective values in the population. Only significant individuals are preserved in order to maintain the diversity. This scheme can be seen as a complement of the original DPM in NSGA-II.

This scheme is inserted between step 6 and step 7 in Figure 6.1. Assume population $S(t)$ has $2N$ individuals (half from $P(t)$ and half from $Q(t)$). There are two types of individuals in $S(t)$: (1) those with unique C_{total} and/or D_{max}, and (2) those sharing identical C_{total} and D_{max} with others (duplicates and similar individuals). In objective space, the former type individuals are situated in less crowed areas while the latter type individuals are densely situated. For the latter type of individuals (e.g. k_1 individuals in $S(t)$), we divide them into k_2 groups of individuals, where in each group individuals have identical C_{total} and D_{max}. For each group, the individual delegate scheme reserves one individual of the group and deletes all others from the group, thus leaving $2N - k_1 + k_2$ individuals remaining in $S(t)$. It is noted that the population size is not fixed in IDS. For any individual $\mathbf{X}$ in $S(t)$, there is no other individual which has the same objective values with $\mathbf{X}$. Hence, the search is diversified towards the whole PF. The following shows the rule of choosing a delegate from each group.

For an arbitrary group (out of the k_2 groups), its individuals correspond to a single point in objective space. However, the size of the group may not be reduced by the DPM in NSGA-II, due to the insufficient supply of feasible individuals at each generation. And this does not help a diversified search. So, IDS is adopted to select a delegate for each of the k_2 groups and remove all others from the k_2 groups. Hamming distance is widely used to measure the difference between two individuals in terms of their genes. We use average Hamming distance (AHD) to find the delegate of each group. For an arbitrary group, the AHD of individual $\mathbf{X}_i$ is defined in formula (6.5), where $\mathbf{X}_j$ is an arbitrary individual in $S(t)$, $j = 1,\ldots,2N$ and $j \neq i$.

$$AHD(\mathrm{X}_i) = \frac{1}{2N-1}\sum_{\forall j \neq i}\left|\mathrm{XOR}(\mathrm{X}_i, \mathrm{X}_j)\right| \tag{6.5}$$

A larger AHD indicates $\mathbf{X}_i$ is situated in a less crowded region in decision space. In the IDS, for each group, we select the individual with the largest AHD as the delegate of that group to maintain the diversity in decision space. The IDS ensures that, in the parent population $P(t)$, any two individuals are different in both decision space and objective space. In this way, duplicates and similar variants of nondominated individuals will not dominate the population and more space can be left for other significant individuals. This helps to diversify the search towards unexplored parts of the PF and hence should lead to a better coverage of the PF.

The IDS is a problem specific scheme developed to adapt MOEAs based on BLS encoding and Pareto dominance for the CDBO problem. Hence, it could be incorporated into other Pareto-dominance based MOEAs for the above MOP. For example, the IDS can be incorporated into the maintenance and update of the external population in SPEA2 to diversify the search. The effectiveness of the IDS is evaluated in Section 6.6.4.

6.5.4 The Procedure of the Adapted NSGA-II

The adapted NSGA-II is characterized by two adjustments when applied to solve the CDBO problem. We hereafter call it as NSGA-II for network coding based multicast, denoted by NSGAII-4N. Figure 6.3 shows the detailed procedure of NSGAII-4N.

Instead of randomly generating the initial population, the proposed initialization scheme is used to produce N promising individuals. The main loop is similar to the procedure of the original NSGA-II (shown in Figure 6.1) except for the use of the individual delegate scheme (IDS) (step 8 in Figure 6.3). After applying the IDS, the size of $S(t)$ may become smaller than N due to the deletion of individuals. So we replace step 8 in Figure 6.1 with steps 10-11 to determine $P(t)$. It is noted that the IDS is based on the crowding distance in both decision and objective spaces. Step 9 is the original DPM in NSGA-II which is based on nondomination rank and crowding distance in objective space (see Section 6.4 for details). The IDS can be seen as a complement of the DPM in original NSGA-II. The termination condition is the search runs, either (1) a predefined

number of generations, or (2) a predefined amount of computational time.

1. Initialization
2. Set $t = 0$;
3. Set the offspring population $Q(t) = Ø$;
4. Create the parent population $P(t)$ by using the new initialization scheme; (see Section 6.5.2)
5. repeat
6. Set $t = t + 1$;
7. Set $S(t) = P(t) \cup Q(t)$;
8. Delete the individuals in $S(t)$, chosen by the individual delegate scheme; (see Section 6.5.3)
9. Sort the remaining individuals in $S(t)$ by using:
(1) nondomination rank and (2) crowding distance;
10. **if** $|S(t)| \leq N$ **then** Set $P(t) = S(t)$;
11. **else** Set $P(t) = Ø$ and put the first N individuals of $S(t)$ into $P(t)$;
12. Select N individuals from $P(t)$ by using the tournament selection with crowded-comparison operator;
13. Recombine the N individuals and produce offspring population $Q(t)$;
14. **until** *the termination condition is met*
15. Output the nondominated solutions in $P(t)$ and $Q(t)$;

Figure 6. 3 The procedure of NSGAII-4N. (Xing and Qu, 2013)

6.6 Performance Evaluation

First, we introduce the experimental datasets and performance metrics for evaluating NSGAII-4N. Then, we study the effectiveness of the two schemes that adapt NSGA-II for the CDBO problem, namely the initialization scheme and the individual delegate scheme, respectively. Finally, we verify the performance of NSGAII-4N.

6.6.1 Experimental Datasets

As stated in Section 6.2, the CDBO problem is a new MOP formulated in the thesis. There is no benchmark dataset available to study this MOP. The CDBO problem is a variant of the delay constrained NCRM problem. The two problems are based on the

same network model and application (i.e. NCM). The only difference between them is in that the CDBO problem considers two objectives (total cost and delay) simultaneously while the delay constrained NCRM problem only considers a single objective (coding cost) with delay constraint. Hence, the benchmark datasets for the delay constrained NCRM problem can be adapted for the CDBO problem.

In the datasets for the delay constrained NCRM problem, each link is associated with a propagation delay and each coding operation is associated with a processing delay in order to calculate the transmission delay of each path from source s to receiver $t_k \in T$ in the constructed NCM subgraph $G_{s\to T}$. The coding cost is measured by the number of coding links in the NCM subgraph (see Section 5.2 and 5.5.1 for details).

When designing the datasets for the CDBO problem, the following ingredients are added. Each link $e \in G$ is associated with a link cost $c_{link}(e)$ and a propagation delay. The link cost and propagation delay are uniformly distributed in the range [5, 15] and [2*ms*, 10*ms*], respectively, where the distribution of propagation delay is the same as in delay constrained NCRM problem. In addition, the coding cost $c_{code}(v_c)$ for coding node v_c is in proportion to the number of data streams which are involved in the coding operation in v_c. This makes sense because the more data streams involved, the more public buffering and computational resources consumed, and hence the larger cost. We assume the cost per data stream for coding is fixed at 4. For example, if there are two data streams involved in one coding operation, the coding cost is 8. The total cost is the summation of coding cost and link cost incurred by the NCM subgraph; the maximum transmission delay is the maximum path delay in the NCM subgraph.

The experimental networks and instance parameters are given in Table 6.1. Detailed instance description can be found at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm. We consider ten instances, two on fixed networks and eight on randomly generated networks. The two fixed networks are the 7-copy and 15-copy networks. The eight random networks (Rnd-i, i = 1,…,8) are directed networks with 20 to 50 nodes. All experiments were run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM. The results are achieved by running each algorithm 20 times (unless stated otherwise).

Table 6. 1 Experimental Networks and Instance Parameters

Network	nodes	links	sinks	rate	L
7-copy	57	84	8	2	80
15-copy	121	180	16	2	176
Rnd-1	20	37	5	3	43
Rnd-2	20	39	5	3	50
Rnd-3	30	60	6	3	86
Rnd-4	30	69	6	3	112
Rnd-5	40	78	9	3	106
Rnd-6	40	85	9	4	64
Rnd-7	50	101	8	3	145
Rnd-8	50	118	10	4	189

Note: L: the encoding length of individuals

6.6.2 Performance Measures

To evaluate the performance of a MOEA from various aspects, we use the following three performance metrics which have been widely recognized in the studies of multiobjective optimization problems (Tan *et al*, 2006; Zhang and Li, 2007; Li and Zhang, 2009):

- Inverted generational distance (IGD): Let PF_{ref} be a reference set of nondominated solutions of the true PF and PF_{known} be the set of nondominated solutions obtained by an algorithm. Solutions in PF_{ref} are expected to be uniformly distributed in the objective space along the true PF. IGD is defined as (6.6).

$$IGD = \frac{1}{|PF_{ref}|} \sum_{v \in PF_{ref}} d(v, PF_{known}) \tag{6.6}$$

  where $d(v, PF_{known})$ is the Euclidean distance (in the objective domain) between solution v in PF_{ref} and its nearest solution in PF_{known}. This metric measures both the diversity and the convergence of an obtained nondominated solution set. A lower IGD indicates a better overall performance of an algorithm.
- Generational distance (GD): GD measures the average distance from the obtained nondominated solution set PF_{known} to the reference set PF_{ref}, defined as (6.7).

$$GD = \left(\frac{1}{|\ PF_{known}\ |} \sum_{v \in PF_{known}} d(v, PF_{ref}) \right)^{1/2} \tag{6.7}$$

where $d(v, PF_{ref})$ is the Euclidean distance (in the objective domain) between solution v in PF_{known} and its nearest solution in PF_{ref}. A smaller GD indicates the obtained PF is closer to the true PF.

- Maximum spread (MS): this metric reflects how well the true PF is covered by the nondominated solutions in PF_{known} through the hyperboxes formed by the extreme function values observed in PF_{ref} and PF_{known}, as shown in (6.8).

$$MS = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left(\frac{\min(f_i^{\max}, F_i^{\max}) - \max(f_i^{\min}, F_i^{\min})}{F_i^{\max} - F_i^{\min}} \right)^2} \tag{6.8}$$

where M is the number of objectives; $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum values of the i-th objective in PF_{known}, respectively; and $F_i^{\max}$ and $F_i^{\min}$ are the maximum and minimum values of the i-th objective in PF_{ref}, respectively. A larger MS shows the obtained PF has a better spread.

Note that we may not know the true PF for the CDBO problem. In this case, to determine a reference set PF_{ref}, we combine the best-so-far solutions obtained by all algorithms in all runs and select the nondominated solutions as the reference set. This has been widely adopted in evaluating multiobjective algorithms in the literature.

6.6.3 The Effectiveness of The Initialization Scheme

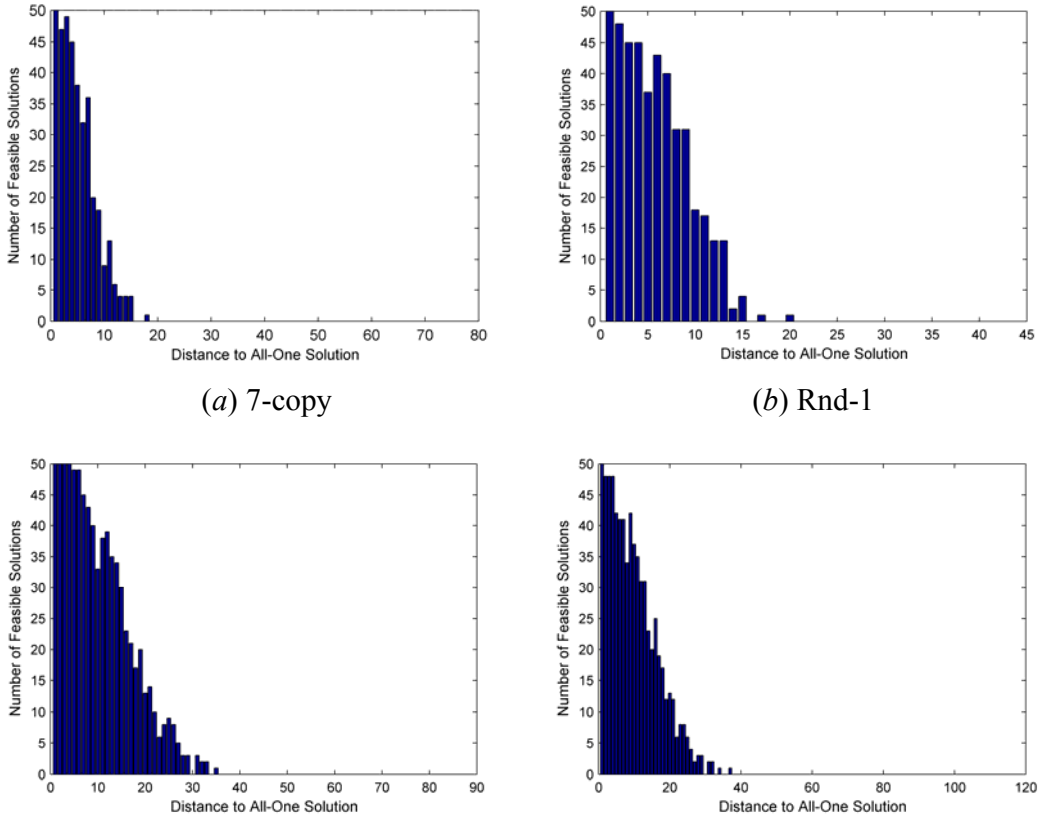
The proposed initialization scheme is based on the use of an all-one individual. First, we investigate the distribution of feasible solutions in relation to the all-one solution. We then evaluate the new initialization scheme against existing initialization schemes.

6.6.3.1 The Distribution of Feasible Solutions over the Search Space

In this experiment, we generate a set of neighbourhood solutions with different hamming distance to the all-one solution and study the distribution of those feasible neighbours to the all-one solution. Assume a solution is represented as L binary bits,

where L is an integer. The all-one solution thus has a maximum distance of L to the all-zero solution.

For each hamming distance of i ($i = 1,\ldots, L-1$) to the all-one solution, we obtain 50 neighbours which represent solutions moving away from the all-one solution. We record the number of feasible solutions in the 50 neighbours and plot the number against i on four selected instances, as shown in Figure 6.4.



(*a*) 7-copy (*b*) Rnd-1

(*c*) Rnd-3 (*d*) Rnd-5

Figure 6. 4 Number of feasible solutions vs. distance to all-one solution.

The results are interesting. Feasible solutions are not evenly distributed over the search space. Those around the all-one solution are more likely to be feasible. That is, the region where the all-one solution resides contains more feasible solutions. Hence inserting an all-one solution, to some extent, helps to guide the search towards promising

region. This is also the reason that we generate a set of variants with small distance to the all-one solution as the initial population.

6.6.3.2 The Effectiveness of the Initialization Scheme

To evaluate the proposed initialization scheme, we compare the performance of three initialization schemes, as listed below.

- Ini-1: randomly generating the initial population. This helps to show what will happen if all-one solution is not included in the initial population.
- Ini-2: randomly generating the initial population with an all-one vector inserted. This shows the effectiveness of the all-one solution when compared with Ini-1. It is noted that Ini-2 is the only existing initialization scheme for the NCRM problem and the delay constrained NCRM problem.
- Ini-3: the proposed initialization scheme (see Subsection 6.2.2).

As aforementioned, the CDBO problem is highly constrained. An initial population with more feasible solutions is more likely to lead to better optimization results. The following experiment focuses on the impact of different initialization schemes on the performance of NSGA-II. We hereafter set initial population size as 40, offspring population size $N = 40$, the crossover probability $p_c = 0.9$, and the mutation probability $p_m = 1/L$, where L is the encoding length. NSGA-II terminates after 200 generations.

Table 6. 2 The Number of Runs That Output Feasible Solutions. (The Best Results are in Bold)

Network	Ini-1	Ini-2	Ini-3
7-copy	0	**20**	**20**
15-copy	0	**20**	**20**
Rnd-1	20	**20**	**20**
Rnd-2	6	**20**	**20**
Rnd-3	14	**20**	**20**
Rnd-4	18	**20**	**20**
Rnd-5	0	**20**	**20**
Rnd-6	5	**20**	**20**
Rnd-7	1	**20**	**20**
Rnd-8	0	**20**	**20**

First, for each scheme, we count the number of runs which output feasible solutions after the evolution, as shown in Table 6.2. We see that both Ini-2 and Ini-3 perform much

better than Ini-1 (compare column *Ini-1*, *Ini-2*, *Ini-3* of the same row). This observation illustrates that Ini-1 is not an appropriate initialization scheme for the CDBO problem. This is because NSGA-II with Ini-1 cannot guarantee that a feasible solution is found at the end of the evolution. We therefore consider Ini-2 and Ini-3 as possible initialization schemes and compare them in terms of their effect on the performance of NSGA-II.

Table 6. 3 The Impact of Ini-2 and Ini-3 on the Performance of NSGA-II. X and Y in X(Y) Denotes the Average and Standard Deviation, Respectively, from 20 Runs. (Best Results are in Bold)

	IGD		GD		MS	
Networks	Ini-2	Ini-3	Ini-2	Ini-3	Ini-2	Ini-3
7-copy	5.42(1.778)	**4.04**(1.559)	5.09(3.654)	**3.03**(2.339)	0.77(0.149)	**0.80**(0.130)
15-copy	14.16(4.161)	**13.15**(3.998)	8.25(3.975)	**7.34**(3.696)	0.42(0.163)	**0.46**(0.161)
Rnd-1	4.27(2.541)	**2.69**(2.502)	4.34(3.059)	**3.86**(3.005)	0.62(0.290)	**0.77**(0.219)
Rnd-2	1.37(2.520)	**0.37**(0.676)	1.16(3.419)	**0.60**(1.250)	0.74(0.498)	**0.89**(0.107)
Rnd-3	8.52(4.458)	**6.95**(2.205)	15.51(9.436)	**14.34**(6.820)	0.71(0.219)	**0.73**(0.115)
Rnd-4	25.69(5.490)	**21.50**(5.138)	19.17(9.543)	**17.24**(11.56)	0.21(0.169)	**0.25**(0.151)
Rnd-5	38.81(13.44)	**33.91**(10.08)	25.44(10.93)	**18.50**(9.001)	0.28(0.144)	**0.30**(0.230)
Rnd-6	8.32(4.848)	**5.55**(4.650)	**0.00**(0.000)	**0.00**(0.000)	0.10(0.307)	**0.40**(0.502)
Rnd-7	16.91(14.34)	**16.20**(10.77)	18.18(19.90)	**16.96**(17.13)	**0.41**(0.380)	0.39(0.346)
Rnd-8	29.73(15.24)	**27.87**(14.61)	21.37(17.72)	**20.20**(16.14)	0.39(0.220)	**0.49**(0.185)

We collect the mean and standard deviation of IGD, GD and MS obtained by NSGA-II with Ini-2 and with Ini-3, respectively, as seen in Table 6.3. As mentioned in Subsection 6.6.2, IGD can reflect both convergence and diversity of a MOEA, showing its overall performance; GD shows how far the obtained PF (PF_{known}) is away from the true PF (PF_{ref}) and MS represents how well PF_{ref} is covered by PF_{known}. A smaller IGD indicates a better overall performance; a smaller GD means the PF_{known} is closer to the true PF; a larger MS implies a better coverage of the PF_{known}.

Regarding the IGD, when adopted in NSGA-II, Ini-3 leads to a better overall performance for all instances compared with Ini-2, namely Ini-3 always lead to a smaller mean and standard deviation of IGD (see columns *Ini-2*, *Ini-3* of IGD). Similarly, for GD, Ini-3 outperforms on 9 instances (except Rnd-6), meaning that PF_{known} obtained by NSGA-II with Ini-3 is usually closer to PF_{ref} than that of Ini-2 (see columns *Ini-2*, *Ini-3* of GD). For MS, Ini-3 outperforms Ini-2 on 9 instances, implying PF_{known} obtained by NSGA-II with Ini-3 has a better spread in most of the instances (see columns of MS).

According to the experiments above, we find that Ini-3 is better than Ini-2, when used in NSGA-II. On the one hand, Ini-2 ensures that at least one feasible individual is in the population. Due to the elitism strategy used in NSGA-II, feasible individuals will remain unless more promising individuals appear. Hence, Ini-2 guides the search to explore at least one feasible region in the search space. However, all-one individual may not be enough to diversify the search towards the whole PF. Some of the unexplored parts may be missed during the search and the true PF may not be well covered. On the other hand, Ini-3, the proposed initialization scheme, can provide a diversified initial population than Ini-2. This scheme is based on that feasible individuals are not evenly distributed over the search space; those closer to the all-one individual are more likely to be feasible (which has been justified in Section 6.6.3.1). With multiple and significant individuals provided by Ini-3, a diversified initial population is more likely to lead to better results.

6.6.4 The Effectiveness of The Individual Delegate Scheme (IDS)

In Section 6.5.3, we develop a diversity preservation mechanism (DPM) in both decision and objective spaces, namely the individual delegate scheme (IDS). The IDS is designed based on the feature of the CDBO problem, where only a few feasible individuals are generated at each generation and similar individuals may have identical objective values. To examine the effectiveness of IDS, we compare the performance of the following four variants of NSGA-II:

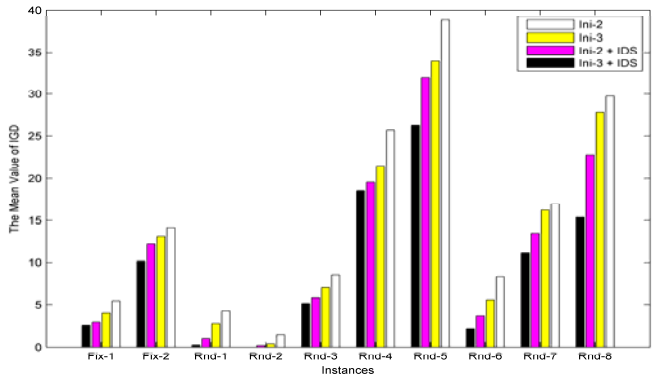
- Ini-2: NSGA-II using Ini-2 as its initialization scheme;
- Ini-3: NSGA-II using Ini-3 as its initialization scheme;
- Ini-2 + IDS: NSGA-II using Ini-2 and IDS;
- Ini-3 + IDS: NSGA-II using Ini-3 and IDS;

Since the performance of the first two algorithms is given in Table 6.3, we provide the statistics of IGD, GD and MS of the latter two variants in Table 6.4. To assist reading, we provide an intuitive view of the performance of the four variants of NSGA-II, by ploting the mean value of IGD, GD and MS from Table 6.3 and 6.4, as shown in Figure 6.5.

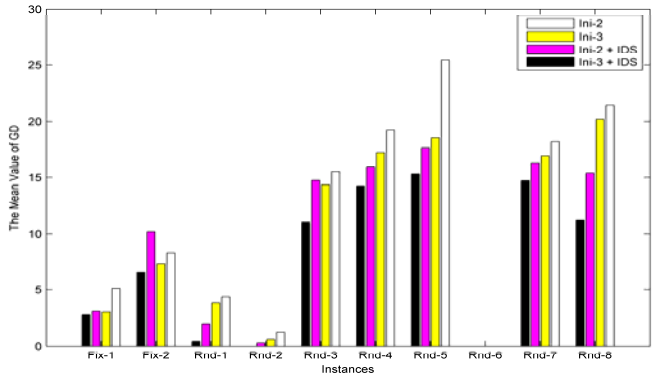
Table 6. 4 The Impact of Ini-2+IDS and Ini-3+IDS on the Performance of NSGA-II. (Best Results are in Bold)

Networks	IGD		GD		MS	
	Ini-2 + IDS	Ini-3 + IDS	Ini-2 + IDS	Ini-3 + IDS	Ini-2 + IDS	Ini-3 + IDS
7-copy	2.86(1.192)	**2.50**(0.765)	3.10(2.195)	**2.80**(2.024)	0.84(0.105)	**0.86**(0.078)
15-copy	12.17(3.538)	**10.14**(3.060)	10.18(7.627)	**6.62**(3.621)	0.53(0.149)	**0.57**(0.132)
Rnd-1	1.00(1.561)	**0.24**(0.746)	1.89(2.895)	**0.42**(1.293)	0.93(0.119)	**0.98**(0.044)
Rnd-2	0.20(0.251)	**0.00**(0.000)	0.28(0.355)	**0.00**(0.000)	0.91(0.105)	**1.00**(0.000)
Rnd-3	5.75(2.204)	**5.13**(2.083)	14.74(7.860)	**10.98**(7.316)	0.78(0.181)	**0.82**(0.107)
Rnd-4	19.55(5.199)	**18.57**(4.737)	15.92(7.676)	**14.25**(5.615)	0.22(0.247)	**0.27**(0.218)
Rnd-5	31.98(13.44)	**26.35**(12.39)	17.64(9.913)	**15.29**(8.239)	0.27(0.180)	**0.32**(0.178)
Rnd-6	3.70(4.650)	**2.09**(3.746)	**0.00**(0.000)	**0.00**(0.000)	0.60(0.502)	**0.80**(0.410)
Rnd-7	13.49(9.120)	**11.10**(5.583)	16.34(16.09)	**14.71**(14.32)	0.47(0.362)	**0.53**(0.312)
Rnd-8	22.85(17.22)	**15.35**(9.359)	15.36(12.44)	**11.15**(8.876)	0.50(0.139)	**0.50**(0.103)

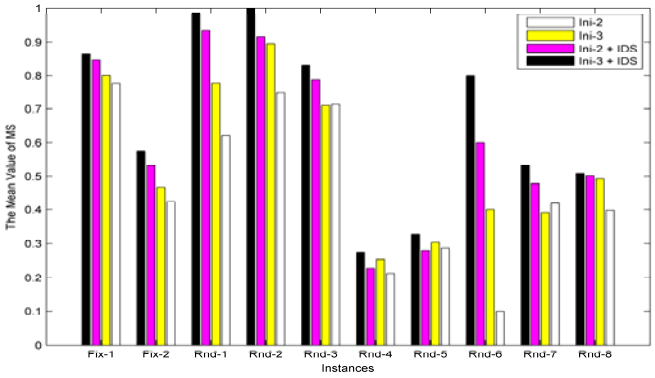
If comparing column *Ini-2* of IGD in Table 6.3 and column *Ini-2 + IDS* of IGD in Table 6.4, we find that Ini-2 + IDS performs better than Ini-2 in all instances regarding IGD. This can also be observed from Figure 6.5(a), where a smaller IGD indicates a better overall performance. If comparing column Ini-2 of GD in Table 6.3 and column Ini-2 + IDS of IGD in Table 6.4, it is seen that the latter outperforms the former in eight instances (except 15-copy and Rnd-6). This is supported by Figure 6.5(b), where a smaller GD implies a closer distance from the approximation to the true PF. If comparing comparing column *Ini-2* of MS in Table 6.3 and column *Ini-2 + IDS* of MS in Table 6.4, we observe that the IDS helps to achieve a better coverage of the true PF in nine instances (except Rnd-5). It is also observed by comparing Ini-2 and Ini-2+IDS in Figure 6.5(c), where a larger MS means a better coverage. On the other hand, if comparing column *Ini-3* of IGD, GD and MS in Table 6.3 and column *Ini-3 + IDS* of IGD, GD and MS in Table 6.4, respectively, we find that Ini-3 + IDS is better than Ini-3 in most of the instances in terms of the IGD, GD and MS. This can also be observed by comparing Ini-3 and Ini-3+IDS in each subfigure of Figure 6.5.



(*a*) The mean value of IGD.



(*b*) The mean value of GD.



(*c*) The mean value of MS.

Figure 6. 5 Comparisons of the average performance of different algorithms.

The above observations demonstrate that the IDS can improve the performance of NSGA-II when addressing the CDBO problem. The following explains why. As discussed in Section 6.5.3, the DPM in NSGA-II (which considers nondominance rank and crowding in objective space only) cannot handle the spread of duplicates and similar variants of the nondominated individuals, because of the lack of feasible and promising individuals generated by evolutionary operations during the evolution. So, duplicates and similar variants of the nondominated individuals gradually dominate the population and may cause the search getting stuck at heavily crowded regions in objective space, which may prevent NSGA-II from exploring other promising regions along the PF. On the other hand, the IDS aims to reduce the number of individuals crowded at a single point in objective space and leave more space to allow other significant individuals to collaborate and diversify the search towards unexplored parts of the PF. This scheme is a complement of the original DPM in NSGA-II. Therefore, with the IDS incorporated, NSGA-II exhibits its ability to diversify the search towards different regions of the PF and hence gains better optimization results.

If comparing Ini-2 + IDS and Ini-3 + IDS, we find that the latter outperforms the former in terms of IGD, GD and MS in all instances. This can be observed by comparing columns *Ini-2 + IDS*, *Ini-3 + IDS* of IGD, GD and MS in Table 6.4, respectively. The comparison demonstrates that incorporating the new initialization scheme and the IDS helps to adapt NSGA-II for the CDBO problem. The new initialization scheme provides a diversified population and the IDS maintains a diversified search during the evolution. Hence, with the two schemes, NSGA-II is more likely to approximate the PF and hence more powerful to solve the CDBO problem.

6.6.5 The Performance Evaluation Overview

In order to evaluate the performance of NSGAII-4N in solving the CDBO problem, we compare it with four MOEAs in the literature. The four existing MOEAs are listed below.

- NSGA-II: the original NSGA-II introduced in Section 6.1 (Deb *et al*, 2002).

- NSGA-II-2: a modified NSGA-II with problem specific selection (Kim *et al*, 2007c). NSGA-II-2 is proposed to address the issue of the tradeoff between coding cost and link cost in NCM.
- SPEA2: the improved strength Pareto evolutionary algorithm, one of the widely applied and recognized MOEAs (Zitzler *et al*, 2001).
- MOPBIL: multiobjective PBIL (Kim *et al*, 2009b). We include this algorithm for comparison because: (1) PBIL shows excellent performance when minimizing the NCRM and delay constrained NCRM problems (Xing and Qu, 2011a, 2011b); and (2) MOPBIL outperforms NSGA-II when applied to solve multiobjective robot system optimization problems (Kim *et al*, 2009b).

Experiments have been carried out on ten instances (see Table 6.1). The parameter settings of the five MOEAs are shown in Table 6.5. To make fair comparisons, each algorithm terminates at the same amount of computational time, i.e. *c*-time. For each instance, we fix its *c*-time in proportion to the number of nodes in the decomposed graph, namely the problem size of the instance. For example, we set *c*-time = 117 sec. for the 7-copy instance and *c*-time = 253 sec. for the 15-copy instance (see Table 6.1 for details). All simulations were run on a Windows XP computer with Intel(R) Core(TM)2 Duo CPU E8400 3.0GHz, 2G RAM. It is noted that all MOEAs share the same objective functions. NSGA-II-2 and NSGAII-4N are two variants of NSGA-II, sharing the basic evolutionary structure with NSGA-II and incorporating problem specific schemes (the evolutionary structure can be found in Section 6.4). NSGA-II, NSGA-II-2, NSGAII-4N and SPEA2 share the same evolutionary operators, including crossover and mutation.

Table 6. 5 Parameter Settings of the Five MOEAs

NSGA-II	NSGA-II-2	SPEA2	MOPBIL	NSGAII-4N
$N = 40$; $p_c = 0.9$; $p_m = 1/L$; $t_{size} = 2$;	$N = 40$; $p_c = 0.9$; $p_m = 1/L$; $t_{size} = 2$;	$N = 40$; $N_{arc} = 40$; $p_c = 0.9$; $p_m = 1/L$; $t_{size} = 2$;	$N = 40$; $N_{arc} = 10$; $n_{PV} = 5$; $p_m = 0.02$; $\alpha = 0.15$; $\varepsilon = 0.05$;	$N = 40$; $p_c = 0.9$; $p_m = 1/L$; $t_{size} = 2$;

Note: N: population size; N_{arc}: archive size; p_c: crossover probability; p_m: mutation probability; L: individual length; t_{size}: tournament size; n_{PV}: number of probability vectors; α: learning rate; ε: amount of shift in the mutation.

The statistics of IGD, GD and MS of the five algorithms are shown in Table 6.6. First of all, NSGA-II and SPEA2 cannot achieve decent IGD, GD and MS values in most of the instances (see rows *NSGA-II*, *SPEA2* of the same instance). On the one hand, the CDBO problem may be relatively hard so NSGA-II and SPEA2 cannot perform well. On the other hand, NSGA-II and SPEA2 are baseline MOEAs and they may not effectively approximate the PF. When comparing rows *NSGA-II* and *SPEA2*, we see that NSGA-II always obtains smaller IGD and GD. It indicates NSGA-II is more suited for addressing the CDBO problem.

If comparing row *NSGAII-4N* with the other rows, we find that in most of the instances, NSGAII-4N obtains the smallest mean IGD, the smallest mean GD, and the largest mean MS, demonstrating the best performance among the five algorithms. This is because NSGAII-4N incorporates two problem specific schemes for better adaptation, namely the new initialization scheme and the individual delegate scheme (IDS). The new initialization provides a population of promising individuals which helps to initiate a diversified search at the beginning of the evolution. The IDS reduces the number of individuals crowded at a single point in objective space and accepts other significant individuals to the population. This scheme helps to diversify the search towards different areas of the PF during the evolution. Hence, with the two schemes incorporated, NSGAII-4N obtains the best performance.

As aforementioned, IGD reflects the overall performance of an algorithm, measuring its convergence and diversity features simultaneously. To further support our observation that NSGAII-4N outperforms the others, we compare the IGD values of the five algorithms by using Student's *t*-test. The statistical results obtained by a two-tailed *t*-test with 38 degrees of freedom at a 0.05 level of significance are given in Table 6.7. The result of Alg-1↔Alg-2 is shown as "+", "–", or "~" when Alg-1 is significantly better than, significantly worse than, or statistically equivalent to Alg-2, respectively. Table 6.7 shows that NSGAII-4N outperforms the other algorithms in most of the instances except the Rnd-4 and Rnd-8 networks. For the Rnd-4 network, the performance of NSGAII-4N is equivalent to those of NSGA-II-2 and MOPBIL, while for the Rnd-8 network NSGAII-4N and NSGA-II-2 are statistically the same. We therefore conclude the proposed NSGA-II has the best optimization performance over all instances concerned.

Table 6. 6 Performance Comparisons of the Five Algorithms in Terms of IGD, GD and MS (The Best Mean Values Are in Bold).

	7-copy (*c*-time = 117 sec)			15-copy (*c*-time = 253 sec)		
Algorithms	IGD	GD	MS	IGD	GD	MS
NSGA-II	5.875(3.014)	4.156(3.692)	0.727(0.182)	14.985(3.327)	9.984(3.463)	0.376(0.139)
NSGA-II-2	5.510(2.245)	4.539(2.302)	0.751(0.151)	15.315(3.269)	9.239(3.213)	0.379(0.193)
SPEA2	8.572(13.15)	6.262(4.008)	0.667(0.333)	15.129(4.744)	9.112(3.496)	0.451(0.240)
MOPBIL	5.693(1.989)	4.407(3.352)	0.717(0.141)	16.850(3.621)	11.329(4.586)	0.342(0.159)
NSGAII-4N	**1.564**(0.866)	**1.923**(1.499)	**0.948**(0.080)	**11.357**(4.595)	**7.793**(5.642)	**0.523**(0.177)
	Rnd-1 (*c*-time = 54 sec)			Rnd-2 (*c*-time = 65 sec)		
Algorithms	IGD	GD	MS	IGD	GD	MS
NSGA-II	4.761(2.861)	6.232(3.358)	0.669(0.225)	1.167(1.164)	0.318 (0.360)	0.555(0.426)
NSGA-II-2	4.169(3.042)	4.776(2.985)	0.700(0.274)	0.609(0.490)	0.671(0.158)	0.751(0.176)
SPEA2	10.577(3.95)	6.949(4.563)	0.351(0.396)	2.541(2.649)	2.000(3.482)	0.678(0.828)
MOPBIL	6.533(3.148)	4.072(3.374)	0.434(0.316)	1.840(1.345)	0.816(1.743)	0.448(0.420)
NSGAII-4N	**0.680**(1.491)	**0.763**(1.885)	**0.938**(0.144)	**0.025**(0.111)	**0.035**(0.158)	**0.989**(0.046)
	Rnd-3 (*c*-time = 94 sec)			Rnd-4 (*c*-time = 113 sec)		
Algorithms	IGD	GD	MS	IGD	GD	MS
NSGA-II	8.292(4.647)	16.151(8.204)	0.770(0.143)	20.972(5.822)	14.120(5.933)	0.196(0.144)
NSGA-II-2	9.283(6.608)	16.475(8.841)	0.799(0.289)	18.807(5.341)	15.662(9.026)	0.229(0.178)
SPEA2	16.191(9.803)	23.159(11.25)	**1.048**(0.576)	30.188(9.033)	19.761(9.515)	0.292(0.277)
MOPBIL	12.017(7.849)	16.000(13.03)	0.820(0.395)	19.705(8.447)	10.809(7.705)	0.218(0.228)
NSGAII-4N	**4.419**(1.912)	**10.270**(7.036)	0.873(0.135)	**12.966**(8.466)	**9.700**(6.010)	**0.493**(0.325)
	Rnd-5 (*c*-time = 124 sec)			Rnd-6 (*c*-time = 91 sec)		
Algorithms	IGD	GD	MS	IGD	GD	MS
NSGA-II	33.651(12.89)	21.245(10.73)	0.308(0.179)	8.328(2.848)	**0.000**(0.000)	0.100(0.307)
NSGA-II-2	34.660(13.30)	22.360(15.15)	0.223(0.160)	8.790(2.069)	**0.000**(0.000)	0.050(0.223)
SPEA2	47.996(17.01)	24.578(10.94)	0.255(0.192)	9.595(1.528)	0.450(2.012)	0.045(0.203)
MOPBIL	37.806(11.02)	20.669(11.10)	0.238(0.163)	8.778(2.066)	0.450(2.012)	0.086(0.270)
NSGAII-4N	**24.247**(9.210)	**13.564**(7.793)	**0.419**(0.216)	**4.164**(4.723)	**0.000**(0.000)	**0.550**(0.510)
	Rnd-7 (*c*-time = 178 sec)			Rnd-8 (*c*-time = 194 sec)		
Algorithms	IGD	GD	MS	IGD	GD	MS
NSGA-II	15.123(10.72)	19.718(25.22)	0.360(0.284)	28.044(16.48)	19.250(11.47)	0.460(0.159)
NSGA-II-2	16.471(11.26)	16.483(13.72)	0.401(0.370)	23.029(17.74)	14.940(10.73)	0.496(0.139)
SPEA2	25.307(17.40)	24.619(22.61)	**0.732**(0.695)	52.952(29.05)	30.391(22.17)	0.380(0.295)
MOPBIL	18.608(15.62)	14.829(14.24)	0.563(0.376)	44.184(23.11)	28.274(20.99)	0.437(0.236)
NSGAII-4N	**9.272**(5.572)	**12.015**(10.99)	0.584(0.279)	**19.230**(9.354)	**9.648**(3.631)	**0.566**(0.262)

Table 6. 7 The *t*-Test Results of Different Algorithms on 10 Instances.

Algorithm-1 ↔ Algorithm-2	7-copy	15-copy	Rnd-1	Rnd-2	Rnd-3
NSGAII-4N ↔ NSGA-II	+	+	+	+	+
NSGAII-4N ↔ NSGA-II-2	+	+	+	+	+
NSGAII-4N ↔ SPEA2	+	+	+	+	+
NSGAII-4N ↔ MOPBIL	+	+	+	+	+
Algorithm-1 ↔ Algorithm-2	Rnd-4	Rnd-5	Rnd-6	Rnd-7	Rnd-8
NSGAII-4N ↔ NSGA-II	+	+	+	+	+
NSGAII-4N ↔ NSGA-II-2	~	+	+	+	~
NSGAII-4N ↔ SPEA2	+	+	+	+	+
NSGAII-4N ↔ MOPBIL	~	+	+	+	+

6.7 Summary

Service providers and network users have conflicting interests, one for cheap cost and the other for high quality. The trade-off between the two sides needs to be studied so that the decision maker (or network administrator) can provide compromised solutions. This chapter formulates a bi-objective optimization problem to investigate the conflicting interests of service providers and network users, namely the cost-delay bi-objective optimization (CDBO) problem. The CDBO problem can be seen as an extended version of the delay constrained NCRM problem in Chapter 5 and another variant of the NCRM problem in Chapter 3. The datasets for the delay constrained NCRM problem are adapted for the CDBO problem, which has been publicly available.

We adapt NSGA-II for the CDBO problem by developing two problem specific schemes, namely a new initialization scheme and an individual delegate scheme (IDS). The first scheme is developed based on the feature of the BLS encoding, where solutions closer to the all-one solution are more likely to be feasible solutions. This scheme can provide a population of diversified and feasible solutions, which helps to diversify the search at the beginning of the evolution. As insufficient feasible individuals are produced during the evolution, duplicates and similar variants of the nondominated individuals gradually dominate the population, which could harm the population diversity and avoid the search approximating upexplored parts of the PF. To overcome this problem, the IDS is proposed which removes those individuals crowded at a single point in objective space from the population and leaves more space to other significant individuals, which helps to diversify the search during the evolution. This scheme is a complement of the original diversity preservation mechanism in NSGA-II. With the two schemes incorporated, the adapted NSGA-II is reported to perform better than a number of MOEAs when tackling the CDBO problem.

This work has been published at Information Sciences.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

As a breakthrough concept emerged at the beginning of the new century, network coding is leading the evolution of routing and switching technologies in communication networks. An ideal application of network coding is supporting real-time and high-speed multicast data transmission. However, there are a number of barriers to be overcome, before the deployment of efficient network coding based multicast (NCM). Among them, to find appropriate NCM routing solutions are of practical importance. However, routing problems for NCM have received only a few concerns in the literature (Xing and Qu, 2012). The objective of the thesis is to use population-based evolutionary algorithms (EAs) to study the routing problems for NCM. The main findings and achievements are summarized below.

7.1.1 Three EAs for the Network Coding Resource Minimization (NCRM) Problem

The thesis started with the investigation into the NCRM problem. This is a NP-hard combinatorial optimization problem (COP) (Kim *et al*, 2007a). Prior to the research in my thesis, there had been a number of attempts for addressing the NCRM problem by using greedy search (Fragouli and Soljanin, 2006; Langberg *et al*, 2006) and EAs. EAs have shown better performance than greedy search (Kim *et al*, 2006). However, there are some limitations in the existing EAs when used to solve the NCRM problem. We find that there is still large gap between the best results achieved by the existing EAs and the global optima (Xing and Qu, 2011a, 2012). The existing EAs include GAs and QEA (Kim *et al*, 2006, 2007a, 2007b; Xing *et al*, 2010), which are all based on BLS encoding. In the study of the existing GAs and QEA, we have the following findings. We find the GAs (Kim *et al*, 2007a, 2008b) cannot effectively explore feasible regions in the search

space. This is because there are too many infeasible solutions in the search space and the crossover operation cannot effectively locate feasible solutions. We also find that QEA (Xing *et al*, 2010) are sensitive to instances since its performance is overly dependant on a key parameter, RAD. The above gaps motivate us to develop appropriate EAs for the NCRM problem. In the thesis, we propose three tailored EAs, including PBIL, cGA and pEA (see Chapter 4).

We first adapt PBIL for the NCRM problem. We choose PBIL because it is fit for BLS encoding. The gradually changing probability vector (PV) reflects the continuity in the search and takes advantage of the promising solutions previously found to guide the search to locate and explore promising regions in the search space, which is a significant advantage when used in solving the highly-constrained problem concerned. We also notice that EDAs having similar PV update schemes with PBIL may also be suited to BLS encoding. During the evolution, PV gradually converges to an explicit solution and its uncertainty degree decreases. We find that the search may fail to escape some local optimum if the degree of the uncertainty of PV is lower enough. According to this feature, we for the first time develop an entropy-based restart scheme for PBIL. This scheme uses the average entropy to represent the degree of uncertainty. In the thesis, the threshold value of the average entropy is calculated based on empirical experiments and used as the criterion to restart the search. The threshold value affects the performance of PBIL and should be set properly. If the value is extremely small, the search may waste time on exhaustively exploiting a very small region; if it is extremely large, the search is forced to restart without carrying out effective local exploitation. The restart scheme has shown to enhance the global exploration ability of PBIL. The adapted PBIL performs better than existing EAs in terms of the best results obtained (Xing and Qu, 2011a). However, this PBIL still suffers high computational cost. Motivated by this, we decided to develop another EA which requires less computational cost.

We then investigate the adaptation of cGA for the NCRM problem. We choose cGA because: (1) its PV update process also reflects the continuity in the search, which helps to track promising regions; (2) cGA was reported to be very fast (Callagher *et al*, 2004). We develop three enhancement schemes to adapt cGA for the problem concerned, i.e. the use of all-one solution, a PV restart scheme and a problem specific local search operator.

According to the feature of BLS encoding, solutions closer to the all-one solution are more likely to be feasible. The first scheme sets the initial elite solution as an all-one solution, which helps the PV to locate feasible regions in the search space. The second scheme is a PV restart scheme and designed based on the features of BLS encoding and the structure of cGA. In this restart scheme, the key parameter is the predefined number of consecutive generations (g_c). We notice that the PV restart scheme helps to stop ineffective evolution and increase the global exploration ability of cGA. We also notice that the setting of g_c is more tolerant than that of the parameter in the restart scheme in PBIL. The third scheme is a local search operator based on graph theory. To the best of my knowledge, this is the very first problem specific local search operator incorporated into evolutionary process for the NCRM problem. This operator has shown to greatly improve the best results obtained by cGA. The adapted cGA gains better performance than PBIL in terms of the best results found and the computational time.

The two pieces of work above are based on the application of BLS encoding, the typical encoding approach used in solving the NCRM problem. We find two weaknesses of BLS encoding and its variant (BTS encoding), namely high proportion of infeasible solutions in the search space and high computational cost in the fitness evaluation, which may deteriorate the performance of EAs. Therefore, we investigate an alternative to BLS and BTS encodings, namely the path-oriented encoding. We adapt the new encoding for the NCRM problem and develop three basic components to build up the evolutionary framework (called pEA), including the initialization, crossover, and mutation. The new encoding and its corresponding fitness evaluation do not suffer the weaknesses above. It is noted that the mutation operator is greedy-based and only accepts the solution if it is better. In addition, we discuss the setting of mutation probability in pEA. We argue that a constant value is not suited to the situation in pEA since the basic units (BUs) change with the number of receivers (d). A constant value may cause dramatically difference in the mutation operations incurred during the evolution. Hence, the mutation probability is set to $1/d$ to ensure a stablized performance of mutation for different NCM sessions. We also develop a local search operator which is based on the path-oriented encoding. This operator shows strong local exploitation ability. pEA with this operator incorporated is

shown to outperform the proposed PBIL and cGA and other existing EAs regarding the overall optimization performance.

7.1.2 Formulation of the Delay Constrained NCRM Problem and the Adaptation of PBIL

NCM has promising potentials to support real-time multimedia applications. These applications usually require stringent quality-of-service (QoS) guarantees, in particular, the transmission delay (i.e. end-to-end delay) (Aurrecoechea *et al*, 1998; Chalmers and Sloman, 1999; Stregel and Manimaran, 2002). To make NCM applicable to support those real-time multimedia applications, one need to consider the QoS issues in routing (Walsh and Weber, 2008). This importance is not considered in the NCRM problem. Therefore, we extend the original NCRM problem by adding the maximum transmission delay as a constraint and for the first time formulate the delay constrained NCRM problem (Xing and Qu, 2011b).

The delay constrained NCRM problem is a variant of the original NCRM problem. They are based on the same underlying network model. The only difference between them is in that the former considers the maximum transmission delay (also called end-to-end delay) of NCM as a constraint. It is noted that the transmission delay of a path can be estimated in advance since the propogation time of each link along the path can be estimated based on the distance and the propogation medium. In the new problem, we provide a model to calculate the transmission delay in each path in the NCM. We adapt the datasets for the NCRM problem for the delay constrained NCRM problem by associating each link in the network with a propogation delay and each coding operation with a data processing delay, respectively. The datasets are available at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm. Compared with the NCRM problem, the new one with delay constraint also reflects the user experience which is necessary when using NCM to support real-time multimedia applications.

We select three EAs to study the delay constrained NCRM problem, including GAs based on BLS/BTS encoding and PBIL in Section 4.1. Only minor changes are made to the fitness evaluation to handle the delay constraint. A feasible solution in the NCRM problem can become infeasible if it cannot meet the delay constraint. We notice that a

severer delay bound leads to an increase in the difficulty for problem solving. This is because a severer delay constraint turns a proportion of feasible solutions to infeasible ones, which reduces the number of feasible solutions in the search space. And the reduction contributes to a harder problem. We find that PBIL is better than GAs and has a more stablized performance when the delay constraint is changed. This is because PBIL is more suited to BLS encoding and relatively easily locate promising regions. On the other hand, we also notice that the restart scheme used in PBIL in Section 4.1 has some weaknesses when used in solving the delay constrained NCRM problem, namely that the threshold value of the average entropy is calculated based on empirical experiments for the NCRM problem (but not for the new problem) and that the search restarts from scratch without the use of the promising solutions found (which would lose necessary guidance towards promising regions). We hence propose a substitute for the restart scheme to adapt PBIL for the delay constrained problem. The substitute consists of a new PV update scheme and a PV mutation. The PV update scheme is deliberately designed for the highly constrained problem here, which takes advantage of the historically best solutions. The combination of the two has shown to improve the global exploration ability and avoid local optima simultaneously. PBIL with the combination outperforms PBIL with restart scheme in terms of the best results obtained.

7.1.3 Formulation of the Cost Delay Bi-Objective Optimization (CDBO) Problem and the Adaptation of NSGA-II

Naturally, the interests of service providers and those of network users conflict with each other. Service providers look for cheap routing solutions that may sacrifice service quality for end users; while network users desire for good user experience that may be quite expensive (Pu *et al*, 2009). It is worth studying how to well compromise between the two sides (Xu, 2011). NCM is a young communication technique and potential suitable for supporting one-to-many data delivery (Li *et al*, 2003). When considering the applications of NCM, one should be concerned with not only the cost of routing solutions but also user experience (Wang *et al*, 2006). It is important to investigate the trade-off between the two sides in the context of NCM.

For the above reasons, we extend the delay constrained NCRM problem by taking into account the conflicting interests of service providers and network users and, for the first time, formulate the CDBO problem where total cost and delay are two objectives for minimization (Xing and Qu, 2013). This is a multiobjective optimization problem (MOP). The total cost is the summation of coding and link costs incurred in the NCM. We select total cost as an objective because the sum is more accurate than coding/link cost to estimate the network resource used in the NCM and it can reflect the interests of service providers (Xing and Qu, 2013). We select maximum transmission delay as the other objective because it is one of the most important QoS parameters and reflects the interests of network users (Zhang *et al*, 2009). The CDBO problem reflects the interests of both sides; the resulting trade-off can be used by the decision maker to compromise between the two sides (Xing and Qu, 2013). We believe it provides an early step research towards wider range of intelligent resource optimization in NCM.

The CDBO problem is a variant of the delay constrained NCRM problem (also a variant of the NCRM problem). Their basis is the same underlying network model and the graph decomposion method. In order to obtain the datasets for the new problem, we extend the datasets for the delay constrained NCRM problem by associating each link in the network with a link cost and each coding operation with a coding cost. The datasets have been made publicly available at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm.

NSGA-II is selected to initiate the investigation into the CDBO problem. Two problem specific schemes are developed to adapt NSGA-II for the MOP, including an initialization scheme and an individual delegate scheme (IDS). The initializaiton scheme is designed based on the feature of BLS encoding and aims to generate an initial population of diversified and promising individuals. It helps to initiate a diversified search for NSGA-II. We find the original diversity preservation mechanism (DPM) in the NSGA-II cannot well handle the spread of duplicates and similar variants of the nondominated individuals in the population, which prevents the search from effectively approximating the unexplored parts on Pareto-optimal front (PF). The IDS complements the original DPM in NSGA-II and diversify the population regarding both decision and objective spaces. It is shown that the two adaptation schemes help NSGA-II achieve better results in terms of the obtained approximation of PF.

7.2 Future Work

NCM routing problems are challenging yet practically important to the deployment of NCM. The more the practical factors considered, the fitter the problems modelled with respect to the real world applications. On the other hand, sophisticated algorithms need to be developed to effectively address the related problems. Therefore, further investigation can be dedicated to the following problems and issues.

7.2.1 More Practical Problem Formulation

The three routing problems in the thesis are based on a simple network model in the context of NCM. There are some limitations in the problem formulations above. First, the problems above are modelled as static optimization problems which only mimic the environment of a snapshot of a communication network at some time point. The status of the network never changes over time. This may not mimic the dynamic feature of real communications networks, where the states of links/nodes may change over time (Xu, 2011). The members (including the source and the receivers) in the multicast group do not change. There is no one leaving the group and no extra receiver joining the group, which is also unrealistic (Zhao *et al*, 2009). In the future work, the problems above can be extended considering the dynamic and stochastic environments in real world networks, e.g. assuming the changes incur in the network topology and the multicast group (Zhao *et al*, 2009).

On the other hand, the network model in the thesis is simple and does not include the heterogeneous feature of the real networks, where a large scale network consists of a set of sub-networks which may be based on different topology structures, operating systems and protocols. For example, some sub-networks may have ring topology and others may have clustering structures. Heterogeneous feature is an important feature in large scale communications networks (Benslimane, 2007; Harte, 2008). As NCM has promising potentials to support data delivery in large scale networks, it would be more practical to incorporate the heterogeneous feature (such as different topological structures) into the problem formulations above.

7.2.2 Fitter Algorithmic Design

In Chapter 6, NSGA-II is used to study the CDBO problem. This algorithm does not show very promising results when addressing the MOP above since it is only a baseline multiobjective EA (MOEA) and not quite suited to combinatorial optimization problems (COPs) (Zhou *et al*, 2011). Nowadays, MOEAs based on decomposition (MOEA/Ds) are considered as state-of-the-art and have been successfully applied to a number of COPs such as multi-objective knapsack problem, multiobjective travelling salesman problem, and deployment in sensor networks (Li and Landa-Silva, 2008, 2011; Konstantinidis *et al*, 2010; Cheng *et al*, 2011; Zhou et al, 2011; Shim *et al*, 2012). In the future, efforts can be made in adapting MOEA/D for the CDBO problem. On the other hand, problem-specific local search operators have reflected significant advantage in enhancing the exploitation ability of EAs, e.g. cGA in Section 4.2 and pEA in Section 4.3. Also, incorporating local search into the evolutionary framework is encouraged in the field of EC so as to take advantage of the two sides (Moscato and Cotta, 2007; Krasnogor, 2009; Whitley *et al*, 2010; Zhou *et al*, 2011). It would be interesting to develop appropriate local search operators (based on domain knowledge) and incorporate them into the evolutionary framework of EAs.

References

Abdullah A H, Enayatifar R, and Lee M (2012). A hybrid genetic algorithm and chaotic function model for image encryption. *AEU-International Journal of Electronics and Communications*, **66**(10): 806-816.

Adra S F, Hamody A I, Griffin I, and Fleming P J (2005). A hybrid multi-objective evolutionary algorithm using an inverse neural network for aircraft control system design. In: *Proc. 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, pp. 1-8.

Ahlswede R, Cai N, Li S Y R and Yeung R W (2000). Network information flow. *IEEE Transactions on Information Theory*, **46**(4): 1204-1216.

Ahn C W and Ramakrishna R S (2003). Elitism-based compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, **7**(4): 367-385.

Alsheddy (2011). Empowerment scheduling: a multi-objective optimization approach using guided local search. PhD Thesis, School of Computer Science and Electronic Engineering, University of Essex.

Anh C W, Kim H T, Kim Y, and An J (2009). A hybrid evolutionary algorithm for multiobjective optimization. In: *Proc. Fourth International Conference on Bio-inspired Computing*, pp. 1-5.

Ahn C W, Kim E, Kim H T, Lim D H, and An J (2010). A hybrid multiobjective evolutionary algorithm: striking a balance with local search. *Mathematical and Computer Modelling* **52**: 2048-2059.

Ahn C W (2011). Fast and adaptive evolutionary algorithm for minimum-cost multicast with network coding. *Electronics Letters*, **47**(12): 700-701.

Aickelin U, Burke E K, and Li J (2007). An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*, **58**(12): 1574-1585.

Aickelin U and Li J (2007). An estimation of distribution algorithm for nurse scheduling. *Annals of Operations Research*, **155**(1): 289-309.

Alexandrova-Kabadjova B, Tsang E, and Krause A (2011). Profit-maximizing strategies for an artificial payment card market, is learning possible? *Journal of Intelligent Learning Systems and Applications* **3**(2): 70-81.

Amir H S, Babak H K, Pedro M C, and Mohammad R A (2010). QoSNC: a novel approach to QoS-based network coding for fixed networks. *Journal of Communications and Networks*, **12**(1): 86-94.

Amuso V J and Enslin J (2007). The strength Pareto evolutionary algorithm 2 (SPEA2) applied to simultaneous multi-mission waveform design. In: *Proc. 2007 International Waveform Diversity and Design Conference*, pp. 407-417.

Anshelevich E, Caskurlu B, Kar K, and Zhang H (2011). Capacity allocation games for network-coded multicast streaming. In: *Proc. 2011 International Conference on Game Theory for Networks (GameNets2011)*.

Aporntewan C and Chongstitvatana P (2001). A hardware implementation of the compact genetic algorithm. In *Proc. of 2001 IEEE Congress Evolutionary Computation*, pp. 624-629.

Armananzas R, Inza I, Santana R, Saeys Y, Flores J L, Lozano J A, Peer Y V, Blanco R, Robles V, Bielza C, and Larranaga P (2008). A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, **1**:1-12. Doi:10.1186/1756-0381-1-6.

Aurrecoechea C, Campbell A T, and Hauw L (1998). A survey of QoS architectures. *Multimedia Systems*, **6**: 138-151.

Bäck T, Hoffmeister F, and Schwefel H P (1991). A survey of evolution strategies. In: *Proc. 4th International Conference on Genetic Algorithms*, pp. 2-9.

Bäck T, Fogel D B, Michalewicz Z (1997). *Handbook of evolutionary computation*, Oxford University Press.

Bader J and Zitzler E (2008). HypE: an algorithm for fast hypervolume-based many objective optimization. Technical Report TIK 286, Computer Engineering and Networks Laboratory, ETH Zurich.

Bader J and Zitzler E (2010). Robustness in hypervolume-based multiobjective search. Technical Report TIK 317, Computer Engineering and Networks Laboratory, ETH Zurich.

Báez A and Pinto D (2009). Multicast routing and wavelength assignment in optical networks with particle swarm optimization. In: *IV WARSO*, pp. 1142-1151.

Bagad V S and Dhotre I A (2010). *Data communication and networking*, Technical Publications Pune.

Ballardie A, Crowcroft J, Francis P (1993). Core based trees (CBT) – an architecture for scalable interdomain multicast routing. In: *Proceedings of ACM SIGCOMM*, pp.85-95.

Baluja S (1994). Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University.

Bashir S, Naeem M, Khan A A, and Shah S I (2010). An application of univariate marginal distribution algorithm in MIMO communication systems. *International Journal of Communication Systems*, **23**(1): 109-124.

Basseur M and Zitzler E (2006). Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research* **2**(3): 255-272.

Benslimane A (2007). *Multimedia multicast on the internet*, ISTE.

Benyahia I (2012). A survey of ant colony optimization algorithms for telecommunication networks. *International Journal of Applied Metaheuristic Computing*. DOI: 10.4018/jamc.2012040102.

Berretta R, Cotta C, and Neri F (2012). Memetic algorithms in Bioinformatics. In: *Neri, Cotta and Moscato (eds.) Handbook of Memetic Algorithms, Sudies in Computational Intelligence* 379, pp. 261-271, Springer.

Bhadra S and Shakkottai S (2006). Looking at large networks: coding vs. queueing. In *Proc. of IEEE INFOCOM*, Barcelona, Spain.

Bhadra S, Shakkottai S, and Gupta P (2006). Min-cost selfish multicast with network coding. *IEEE Transactions on Information Theory*, **52**(11): 5077-5087.

Bhattad K, Ratnakar N, Koetter R, and Narayanan K R (2005). Minimal network coding for multicast. In *Proc. of International symposium on Information Theory (ISIT2005)*, Adelaide pp. 1730-1734.

Blake I F, Gao X H, Mullin R C, Vanstone S A and Yaghoobian T (1993). Applications of finite fields. *Kluwer Academic Publishers*.

Blum C and Roli A (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, **35**(3): 268-308.

Boonma P and Suzuki J (2011). PIBEA: prospect indicator based evolutionary algorithm for multiobjective optimization problems. In: *Proc. 2011 IEEE Congress on Evolutionary Computation (CE 2011)*, New Orleans, USA.

Borella M S (2000). Source models of network game traffic. *Computer Communications*, **23**: 403-410.

Branke J, Kauβler T, Schmidth C, and Schmeck H (2000). A multi-population approach to dynamic optimization problems. In *Proc. of 4th International Conference on Adaptive Computing in Design and Manufacturing*, pp. 299-308.

Branke J and Deb K (2004). Integrating user preferences into evolutionary multiobjective optimization. Technical Report KanGAL, Indian Institute of Technology.

Brockhoff D and Zitzler E (2007). Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *Proc IEEE Congress on Evolutionary Computation (CEC2007)*, pp. 2086-2093.

Bueno M L P and Oliveira G M B (2010). Pareto-based optimization of multicast flows with QoS and traffic engineering requirements. In: *9th IEEE International Symposium on Network Computing and Applications (NCA)*, pp. 257-260.

Buriol L S, Resende M G C, Ribeiro C C, and Thorup M (2005). A hybrid genetic algorithm for the weight setting problem in OSPF-IS-IS routing. *Networks*, **46**(1): 36-56.

Burke E, Elliman D, and Weare R (1995). The automated timetabling of university exams using a hybrid genetic algorithm. In: *Proc. The AISB Workshop on Evolutionary Computing.*

Burke E K and Newall J P (1999). A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, **3**(1): 63-74.

Cai B, Wang S, and Hu H (2011). An effective hybrid genetic algorithm for job shop scheduling problem. *World Academy of Science, Engineering and Technology*, **58**: 42-48.

Cai N and Yeung R W (2002). Secure network coding. In *Proc. of the IEEE International Symposium on Information Theory (ISIT'02)*, June 2002.

Cai N and Yeung R W (2011). Secure network coding on a wiretap network. *IEEE Transactions on Information Theory*, **57**(1): 424-435.

Castro-Gutierrez J, Landa-Silva D, and Moreno Pérez J (2011). Nature of real-world multi-objective vehicle routing with evolutionary algorithms. In *Proc. the IEEE International Conference on Systems, Man and Cybernetics (SMC2011)*, Alaska, USA, pp. 257-264.

Carvalho R, Saldanha R R, Gomes B N, Lisboa A C, Martins A X (2012). A multi-objective evolutionary algorithm based on decomposition for optimal design of Yagi-Uda antennas. *IEEE Transactions on Magnetics* **48**(2): 803-806.

Chakraborty B and Chaudhuri P (2003). On the use of genetic algorithm with elitism in robust and nonparametric multivariate analysis. *Austrian Journal of Statistics*, **32**: 13-27.

Chalmers D and Sloman M (1999). A survey of quality of service in mobile computing environments. *IEEE Communications Surveys*, **2**(2): 2-10.

Chang P C, Chen S H, and Liu C H (2007). Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems. *Expert Systems with Applications*, **33**: 762-771.

Chang P C, Chen S S, and Zhang Q (2008). MOEA/D for flowshop scheduling problems. In *Proc 2008 IEEE Congress on Evolutionary Computation (CEC 2008)*, pp. 1433-1438.

Chen J, Lin Q, and Ji Z (2010). A hybrid immune multiobjective optimization algorithm. *European Journal of Operational Research* **204**: 294-302.

Cheng H and Yang S (2010a). Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. *Engineering Applications of Artificial Intelligence*, **23**(5): 806-819.

Cheng H and Yang S (2010b). Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile Ad Hoc networks. In: Chio et al (eds.) *EvoApplications* 2010, LNCS 6024, pp. 562-571.

Cheng J, Zhang G, Li Z, and Li Y (2011). Multi-objective ant colony optimization based on decomposition for bi-objective traveling salesman problems. *Soft Computing*, 16(4): 597-614.

Chou P A and Wu Y (2007). Network coding for the internet and wireless networks. *IEEE Signal Processing Magazine*, **24**(5): 77-85.

Chu X and Jiang Y (2010). Random linear network coding for peer-to-peer applications. *IEEE Network*, **24**(4): 35-39.

Cochran J K, Horng S M, and Fowler J W (2003). A multi-population genetic algorithm to solve multi-objecitve scheduling problems for parallel machines. *Computers and Operations Research*, **30**(7): 1087-1102.

Coelho L D S and Alotto P (2008). Multiobjective electromagnetic optimization based on a nondominated sorting genetic approach with a chaotic crossover operator. *IEEE Transactions on Magnetics*, **44**: 1078-1081.

Coelho R F (2013). Co-evolutionary optimization for multi-objective design under uncertainty. *Journal of Mechanical Design* **135**: 1-8.

Coello Coello C A and Sierra M R (2003). A coevolutionary multi-objective evolutionary algorithm. In: *Proc. 2003 Congress on Evolutionary Computation (CEC'03)*, pp. 482-489.

Coello Coello C A, Pulido G T, and Montes E M (2005). Current and future research trends in evolutionary multiobjective optimization. In: Graña et al (eds.) *Information Processing with Evolutionary Algorithms: From Industrial Applications to Academic Speculations*, Springer, pp. 213-231.

González O, León C, Miranda G, Rodríguez, Segura C, Estadística D, and Computación I O Y (2007). A parallel skeleton for the strength Pareto evolution algorithm 2. In: *Proc. 15th EUROMICRO International Conference on Parallel, Distributed and Network-based Processing (PDP'07)*, pp. 434-441.

Côté P, Wong T, and Sabourin R (2004). Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. In *Proc. 5th International Conference for the Practice and Theory of Automated Timetabling (PATAT'2004)*, pp. 151-168.

Crichigno J and Barán B (2004). Multiobjective multicast routing algorithm for traffic engineering. In: *Proc. 13th International Conference on Computer Communications and Networks (ICCCN2004)*, pp. 301-306.

Cui T and Ho T (2007). Minimum cost integral network coding. In: *Proc. 2007 IEEE International Symposium on Information Theory (ISIT2007)*, pp.2736-2740.

Cutello C, Narzisi G, and Nicosia G (2005). A class of Pareto archived evolution strategy algorithms using immune inspired operators for ab-initio protein structure prediction. In: *Proc. 3rd European Conference on Application of Evolutionary Computing (EC'05)*, pp. 54-63.

De Bonet J S, Isbell C L, and Viola P (1997). MIMIC: finding optima by estimating probability densities. In: Mozer et al (eds.) *Advances in Neural Information Processing Systems*, **9**: 424-430.

De Jong K A (1975). An analysis of the behaviour of a class of genetic adaptive systems. Doctoral dissertation, University of Michigan.

De Jong K (2005). Genetic algorithms: a 30 year perspective. In: Booker et al (eds.) *Perspectives on Adaptation in Natural and Artificial Systems*, Oxford University Press.

Deb K and Chaudhuri S (2005). I-EMO: an interactive evolutionary multiobjective optimization tool. In *Proc. 1st International Conference on Pattern Recognition and Machine Intelligence (PReMI2005)*, pp. 690-695.

Deb K, Pratap A, Agarwal S, and Meyarivan T (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**(2): 182-197.

Deb K and Sunder J, Rao N U B, and Chaudhuri S (2006). Reference point based multiobjective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, **2**(3): 273-286.

Deb K (2008). Introduction to evolutionary multiobjective optimization. In: Branke et al (ed.) *Multiobjective Optimization*, Springer.

Deb K and Datta R (2011). Hybrid evolutionary multi-objective optimization of machining parameters. KanGAL Technical Report, Indian Institute of Technology.

Deering S (1989). Host extensions for IP multicasting. RFC 1112.

Deering S, Estrin D L, Farinacci D, Jacobson V, Liu C G, and Wei L (1996). The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, **4**(2): 153-162.

Dijkstra E W (1959). A note on two problems in connection with graphs. *Numerical Mathematics*, **1**: 267-271.

Dimakis A G, Godfrey P B, Wainwright M J, and Ramchandran K (2007). Network coding for distributed storage systems. In *Proc. of IEEE INFOCOM*.

Dorigo M, Maniezzo V, and Colorni A (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, **26**(1): 29-41.

Dorigo M and Blum C (2005). Ant colony optimization theory: a survey. *Theoretical Computer Science*, **344**(2-3): 243-278.

Eiben A E, Hinterding R, and Michalewicz Z (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, **3**: 124-141.

Ei-Mihoub T, Hopgood A A, Nolle L, and Battersby A (2006). Hybrid genetic algorithms-a review. *Engineering Letters*, **13**: 124-137.

Elahi A (2001). *Network communications technology*, New York: Delmar.

Elias P, Feinstein A, and Shannon C E (1956). A note on the maximum flow through a network. *IRE Transactions on Information Theory*, **2**: 117-119.

Feltl H and Raidl G R (2004). An improved hybrid genetic algorithm for the generalized assignment problem. In: *Proc. 2004 ACM Symposium on Applied Computing*, pp. 990-995.

Fister I and Brest J (2011). Using differential evolution for the graph coloring. In: *Proc. 2011 IEEE Symposium on Differential Evolution (SDE 2011)*, pp. 1-7.

Fragouli C and Soljanin E (2006). Information flow decomposition for network coding. *IEEE Transactions on Information Theory*, **52**(3): 829-848.

Fragouli C, Boudec J Y L, Widmer J (2006). Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, **36**(1): 63-68.

Fragouli C and Soljanin E (2007). Network coding fundamentals. *Foundations and Trends ® in Networking*, now Publishers Inc., Hanover, USA, **2**(1): 1-133.

Friedrick T, Kroeger T, and Neumann F (2011). Weighted preferences in evolutionary multi-objective optimization. In *Proc. 24th Australasian Joint Conference on Artificial Intelligence (AI 2011)*, LNCS 7106, pp. 291-300.

Friedrick T, Kroeger T, and Neumann F (2013). Weighted preferences in evolutionary multi-objective optimization. *International Journal of Machine Learning and Cybernetics* **4**(2): 139-148.

Fonseca C M and Fleming P J (1993). Genetic algorithm for multiobjective optimization: formulation, discussion and generalization. In: *Proc. Fifth International Conference on Genetic Algorithms*, pp. 416-423.

Ford L R and Fulkerson D R (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, **8**: 399-404.

Gaddam N, Gathala S A, Lastine D, and Somani A (2008). Energy minimization through network coding for lifetime constrained wireless networks. In *Proc. of ICCCN*, St Thomas, USA.

Gallagher J C, Vigraham S, and Kramer G (2004). A family of compact genetic algorithms for intrinsic evolvable hardware. *IEEE Transactions on Evolutionary Computation*, **8**(2): 111-126.

Ghosh A and Mühlenbein H (2004). Univariate marginal distribution algorithms for non-stationary optimization problems. *International Journal of Knowledge-based and Intelligent Engineering Systems*, **8**(3): 129-138.

Glover F and Kochenberger G (2003). *Handbook of Metaheuristics*, Kluwer.

Goh C K and Tan K C (2009). A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, **13**(1): 103-127.

Goldberg A V (1985). A new max-flow algorithm. MIT Technical Report MIT/LCS/TM-291, Laboratory for Computer Science, MIT.

Goldberg D E (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, Massachusetts.

Gonzalez C, Lozano J A, and Larranaga P (2000). Analyzing the population based incremental learning algorithm by means of discrete dynamical systems. *Complex Systems*, **12**(4): 465-479.

Goralski W (2009). *The illustrated network*, Morgan Kaufmann Publishers.

Gosling T, Jin N, and Tsang E (2005). Population based incremental learning with guided mutation versus genetic algorithms: iterated prisoners dilemma. In: *Proc. 2005 IEEE Congress on Evolutionary Computation (CEC2005)*, pp. 958-965.

Greenwood G, Hu X, and Ambrosio J (1996). Fitness functions for multiple objective optimization problems: combining preferences with Pareto rankings. In: *Foundations of Genetic Algorithms*, pp. 437-455.

Grefenstette J J (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16: 122-128.

Grefenstette J J (1992). Genetic algorithm for changing environments. In: *Proc. of 2nd International Conference on Parallel Problem Solving From Nature*, pp. 137-144.

Guo Q, Hierons R M, Harman M, and Derderian K (2005). Constructing Multiple Unique Input/Output Sequences Using Metaheuristic Optimisation Techniques. *IEE Proceedings - Software*, **152**(3): 127-140.

Guo Q, Luo M, Li L, and Yang Y(2010). Secure network coding against wiretapping and Byzantine attacks. *EURASIP Journal on Wireless Communications and Networking*, Doi:10.1155/2010/216524.

Jaszkiewicz A (2002). On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment. *IEEE Transactions on Evolutionary Computation* **6**(4): 402-412.

Jemai J, Zekri M, and Mellouli K (2012). An NSGA-II algorithm for the green vehicle routing problem. In: *Proc. EvoCOP 2012*, LNCS 7245, pp. 37-48.

Ji Y, and Xing H (2011). A memory-storable quantum-inspired evolutionary algorithm for network coding resource minimization. In: Kita E. (Ed.), *Evolutionary Algorithm*, InTech, pp.363-380.

Halsall F (1992). *Data communications, computer networks and open systems*, Wokingham: Addison-Wesley.

Han K H and Kim J H (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, **6**(6): 580-593.

Han K H and Kim J H (2004). Quantum-inspired evolutionary algorithms with a new termination criterion, *H* gate, and two-phase scheme. *IEEE Transactions on Evolutionary Computation*, **8**(2): 156-169.

Han D, Hu G, and Cai L (2008). Multiobjective optimal secure routing algorithm using NSGA-II. In: *Proc. CIS 2008*, pp. 1343-1347.

Hao J K (2011). Memetic algorithms in discrete optimization. In: Neri, Cotta and Moscato (eds.) *Handbook of Memetic Algorithms, Studies in Computational Intelligence* 379, pp. 73-94.

Harik G (1999). Linkage learning via probabilistic modelling in the ECGA. IlliGAL Report no. 99010, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

Harik G R, Lobo F G, and Goldberg D E (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, **3**(4): 287-297.

Harte L J (2008). *Introduction to data multicasting*, Althos.

Hashemi M and Meybodi M R (2011). Univariate marginal distribution algorithm in combination with extremal optimization (EO, GEO). In: *Proc. 18th International Conference on Neural Information Processing (ICONIP'11)*, pp.220-227.

Hauschild M and Pelikan M (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, **1**: 111-128.

Heris S M K and Khaloozadeh H (2011). Open- and closed- loop multiobjective optimal strategies for HIV therapy using NSGA-II. *IEEE Transactions on Biomedical Engineering*, **58**: 1678-1685.

Hidalgo J I, Baraglia R, Perego R, Lanchares J, and Tirado F (2001). A parallel compact genetic algorithm for multi-FPGA partitioning. In: *Proc. of the 9th Workshop on Parallel and Distributed Processing*.

Holland J H (1975). Adaptation in Natural and Artificial Systems. The University of Michigan Press.

Hong T P, Lin W Y, Liu S M, and Lin J H (2007). Dynamically adjusting migration rates for multi-population genetic algorithms. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **11**(4): pp. 1399-1417.

Hou I H, Tsai Y E, Abdelzaher T F, and Gupta I (2008). AdapCode: adaptive network coding for code updates in wireless sensor networks. In *Proc. of IEEE INFOCOM,* Phoenix, Aariz, USA.

Huang C H, Lee J D, and Huang J H (2005). Registration of CT image and facial surface data using adaptive genetic algorithm. *Biomedical Engineering: Applications, Basis and Communications*, **17**(4): 201-206.

Huang M, Liu P, and Liang X (2010). An improved multi-population genetic algorithm for job shop scheduling problem. In: *Proc. 2010 IEEE International Conference on Progress in Informatics and Computing (PIC2010)*, pp. 272-275.

Ihara S (1993). *Information theory for continuous systems*. World Scientific.

Inglis S, Ault G W, Alarcon-Rodriguez A, and Galloway S J (2010). Multi-objective network planning tool for networks containing high penetrations of DER. In: *Proc. 45th International Universities Power Engineering Conference*, pp. 1-6.

Ishibuchi H and Murata T (1996). Multi-objective genetic local search algorithm. In: *Proc. 1996 IEEE International Conference on Evolutionary Computation (CEC 1996)*, pp. 119-124.

Ishibuchi H and Murata T (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, **28**(3): 392-403.

Ishibuchi H, Hitotsuyanagi Y, Tsukamoto N, and Nojima Y (2009). Use of biased neighbourhood structures in multiobjective memetic algorithms. *Soft Computing*, **13**(8-9): 795-810.

Ishibuchi H, Sakane Y, Tsukamoto N, and Nojima Y (2010). Simultaneous use of different scalarizing functions in MOEA/D. In Conference on Genetic and Evolutionary Computation (GECCO2010), pp. 519-526.

Jain S and Sharma J D (2012). Tree structured encoding based multi-objective multicast routing algorithm. *International Journal of Physical Sciences* **7**(10): 1622-1632.

Jin Y, Okabe T, and Sendhoff B (2001). Adapting weighted aggregation for multi-objective evolution strategies. In: *1st International Conference on Evolutionary Multi-criterion Optimization*, LNCS 1993, pp. 96-110.

Jones T, Forrest S (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithm. In: *Proc. 6th International Conference on Genetic Algorithms*.

Kamal A E (2006). 1+N protection in optical mesh networks using network coding on p-cycles. In *Proc. IEEE Globecom,* San Francisco, California, USA.

Kamal A E (2007a). 1+N protection against multiple faults in mesh networks. In *Proc. IEEE International Conference on Communications (ICC),* Glasgow, Scotland, UK.

Kamal A E (2007b). GMPLS-based hybrid 1+N link protection over p-cycles: design and performance. In *Proc. IEEE Globecom,* Washington, DC, USA.

Kamal A E (2008). A generalized strategy for 1+N protection. In *Proc. of IEEE International Conference on Communications (ICC),* Beijing, China.

Kamal A E (2010). 1+N network protection for mesh networks: network coding-based protection using p-cycles. *IEEE/ACM Transactions on Networking*, **18**(1): 67-80.

Kamal A E, Ramamoorthy A, Long L, and Li S (2011). Overlay protection against link failures using network coding. *IEEE/ACM Transactions on Networking*, **19**(4): 1071-1084.

Kannan S, Baskar S, McCalley J D, and Murugan P (2009). Application of NSGA-II algorithm to generation expansion planning. *IEEE Transactions on Power Systems*, **24**: 454-461.

Kao Y C, Lee C N, Wu P J, and Kao H H (2012). A network coding equivalent content distribution scheme for efficient peer-to-peer interactive VoD streaming. *IEEE Transactions on Parallel and Distributed Systems*, **23**(6): 985-994.

Kasireddy B, Wang Y, and Liu J (2011). Energy conservation using network coding in grid wireless sensor networks. In: *Proc. 3rd International Workshop on Intelligent Systems and Applications (ISA)*, pp. 1-4.

Katti S, Rahul H, Hu W, Katabi D, Medard M, and Crowcroft (2006). XORs in the air: practical wireless network coding. In *Proc. of SIGCOMM'06*, Pisa, Italy.

Kennedy J and Eberhart R (1995). Particle swarm optimization. In *Proc. IEEE International Conference on Neural Networks*, pp. 1942-1948.

Kern M (2006). Parameter adaptation in heuristic search – a population-based approach. PhD Thesis, University of Essex.

Khaleghi M, Mohammadian M, Khorasaninejad M, and Farsangi M M (2009). Design of two power system stabilizers co-ordinately based on strength Pareto evolutionary algorithm. In: *Proc. 4th IEEE Conference on Industrial Electronics and Applications (ICIEA 2009)*, pp. 1527-1532.

Khezri S, Meybodi M R, and Osmani A (2011). Fuzzy adaptive PBIL based sensor placement in wireless sensor networks. In: *Proc. 2011 International Symposium on Computer Networks and Distributed Systems (CNDS 2011)*, pp. 216-221.

Khreishah A, Wang C, Shroff N B (2009). Cross-layer optimization for wireless multihop networks with pairwise intersession network coding. *IEEE Journal on Selected Areas in Communications*, **27**(5): 606-621.

Kim M, Ahn C W, Médard M, Effros M (2006). On minimizing network coding resources: An evolutionary approach. *Second Workshop on Network Coding, Theory, and Applications (NetCod2006)*, Boston.

Kim M, Médard M, Aggarwal V, Reilly V O, Kim W, Ahn C W, and Effros M (2007a). Evolutionary approaches to minimizing network coding resources. In *Proc. of 26th IEEE International Conference on Computer Communications (INFOCOM2007)*, Anchorage. pp.1991-1999.

Kim M, Aggarwal V, Reilly V O, Médard M, and Kim W (2007b). Genetic representations for evolutionary optimization of network coding. In *Proc. of 4th European Workshop on the Application of Nature-Inspired Techniques to Telecommunication Networks and Other Connected Systems (EvoCOMNET2007)*, Valencia.

Kim M, Médard M, Aggarwal V, and Reilly V O (2007c). On the coding-link cost tradeoff in multicast network coding. In *Proc. of MILCOM*.

Kim M, Médard M, Aggarwal V, and Reilly V O (2009a). Network coding and its implications on optical networking. In *Proc. of OFC 2009*.

Kim J H, Kim Y H, Choi S H, and Park I W (2009b). Evolutionary multi-objective optimization in robot soccer system for education. *IEEE Computational Intelligence Magazine*, **4**: 31-41.

Kishor A, Yadav S P, and Kumar S (2007). Application of a multiobjective genetic algorithm to solve reliability optimization problem. In *Proc. of 2007 International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 07)*, Sivakasi, pp. 458-462.

Knowles J D and Corne D W (1999). The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimization. In: Proc. *1999 Congress on Evolutionary Computation (CEC 1999)*, pp. 98-105.

Knowles J D and Corne D W (2000a). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, **8**(2): 149-172.

Knowles J and Corne D (2000b). M-PAES: a memetic algorithm for multiobjective optimization. In: *Proc. IEEE Congress on Evolutionary Computation (CEC2000)*, pp. 325-332.

Koetter R and Médard M (2003). An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, **11**(5): 782-795.

Kokosiński Z, Kwarciany K, and Kolodziej M (2005). Efficient graph coloring with parallel genetic algorithms. *Computing and Informatics*, **24**: 1001-1024.

Konstantinidis A, Charalambous C, Zhou A, and Zhang Q (2010). Multi-objective mobile agent-based sensor network routing using MOEA/D. In *Proc 2010 IEEE Congress on Evolutionary Computation (CEC 2010)*, pp. 1-8.

Konstantinidis A and Yang K (2012). Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D. *Applied Soft Computing*, **12**(7): 1847-1864.

Krasnogor N (2009). Memetic algorithms. In: Bäck and Rozenberg (eds.) *Handbook of Natural Computation*, Springer.

Kursawe F (1991). A variant of evolution strategies for vector optimization. In: Schwefel and Männer (eds.) *Parallel Problem Solving from Nature*, pp. 193-197.

Lakshman T V, Mishra P P, and Ramakrishnan K K (1999). Transporting compressed video over ATM networks with explicit-rate feedback control. *IEEE/ACM Transactions on Networking*, **7**(5): 710-723.

Langberg M, Sprintson A, and Bruck J (2006). The encoding complexity of network coding. *IEEE Transactions on Information Theory*, **52**(6): 2386-2397.

Lara A, Sanchez G, Coello Coello C A, and Schutze O (2010). HCS: a new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, **14**(1): 112-132.

Larranaga P and Lozano J A. (Eds.) (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*, Kluwer Academic Publishers, Boston.

Lau T W, Chung C Y, Wong K P, Chung T S and Ho S L (2009). Quantum-inspired evolutionary algorithm approach for unit commitment. *IEEE Transactions on Power Systems*, **24**(3): 1503-1512.

Law N L and Szeto K Y (2007). Adaptive genetic algorithm with mutation and crossover matrices. In: *Proc. IJCAI*, pp. 2330-2333.

Li R, Etemaadi R, Emmerich M T M, and Chaudron M R V (2011). An evolutionary multiobjective optimization approach to component-based software architecture design. In: *2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, pp. 432-439.

Li B and Niu D (2011). Random network coding in peer-to-peer networks: from theory to practice. *Proceedings of the IEEE*, **99**(3):513-523.

Li B and Wang L (2007). A Hybrid Quantum-Inspired Genetic Algorithm for multiobjective flow shop scheduling. *IEEE Transactions on System, Man, And Cybernetics-Part B: Cybernetics*, **37**(3): 576-591.

Li H and Landa-Silva D (2008). Evolutionary multiobjective simulated annealing with adaptive and competitive search direction. In *Proc. 2008 IEEE Congress on Evolutionary Computation (CEC 2008)*, pp. 3310-3317.

Li H, Landa-Silva D, and Gandibleux X (2010). Evolutionary multiobjective optimization algorithms with probabilistic representation based on pheromone trails. In *Proc. 2010 IEEE Congress on Evolutionary Computation (CEC 2010)*, pp. 2307-2314.

Li H and Landa-Silva D (2011). An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolutionary Computation*, **19**(4): 561-595.

Li H, Zhang Q, Tsang E P K, and Ford J (2004). Hybrid estimation of distribution algorithm for multiobjective knapsack problem. In *Proc. the 4th European Conference on Evolutionary Computation in Combinatorial Optimization*, Coimbra, Portugal.

Li H and Zhang Q (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, **13**(2): 284-302.

Li J and Jiang Y (2009). Estimation of distribution algorithms for job schedule problem, In *Proc. of Second International Conference on Information and Computing Science (ICIC2009)*, Manchester, UK.

Li S Y R, Yeung R W, and Cai N (2003). Linear network coding. *IEEE Transactions on Information Theory*, **49**(2): 371-381.

Li S Y R and Yeung R W (2006). On convolutional network coding. In: *Proceeding of 2006 IEEE International Symposium on Information Theory (ISIT2006)*, Seattle, pp.1743-1747.

Li Z, Li B, Jiang D, and Lau L C (2005). On achieving optimal throughput with network coding. In: *Proc. IEEE INFOCOM 2005*.

Li Z, Li B, and Lau L C (2006). On achieving maximum multicast throughput in undirected networks. *IEEE Transactions on Information Theory*, **52**(6): 2467-2485.

Li Z, Li B, and Lau L C (2009). A constant bound on throughput improvement of multicast network coding in undirected networks. *IEEE Transactions on Information Theory*, **55**(3): 1016-1026.

Lin S F, Chang J W, and Hsu Y C (2010). A self-organization mining based hybrid evolution learning for TSK-type fuzzy model design. *Applied Intelligence*, **36**(2): 454-471.

Liu X and Liu Y (2009). Adaptive genetic algorithm based on population diversity. In: *Proc. 2009 International Forum on Information Technology and Applications*, pp. 510-512.

Liu Bo, Aliakbarian H, Radiom S, Vandenbosch G A E, and Gielen G (2012). Efficient multi-objective synthesis for microwave components based on computational intelligence techniques. In *Proc 49th Annual Design Automation Conference (DAC'12)*, pp. 542-548.

Lobo F G and Lima C F (2005). A review of adaptive population sizing schemes in genetic algorithms. In: *Proc. GECCO'05*, pp. 228-234.

Lohn J, Kraus W, and Haith G (2002). Comparing a coevolutionary genetic algorithm for multiobjective optimization. In: *Proc. IEEE Congress on Evolutionary Computation (CEC2002)*, pp. 1157-1162.

Lun D S, Médard M, Ho T, and Koetter R (2004). Network coding with a cost criterion. In: P*roc. 2004 International Symposium on Information Theory and its Applications (ISITA2004)*.

Lun D S, Ratnakar N, Koetter R, Médard M, Ahmed E, and Lee H (2005). Achieving minimum-cost multicast: a decentralized approach based on network coding. In: *Proc. IEEE INFOCOM2005*, pp. 1607-1617.

Lun D S, Ratnakar N, Médard M, Koetter R, Karger D R, Ho T, Ahmed E, and Zhao F (2006). Minimum-cost multicast over coded packet networks. *IEEE Transactions on Information Theory*, **52**(6): 2608-2623.

Luong H N, Nguyen H T T and Ahn C W (2012). Entropy-based efficiency enhancement techniques for evolutionary algorithms. *Information Sciences*, **188**: 100-120.

Ma G, Xu Y, Lin M, and Xuan Y (2007). A content distribution system based on sparse linear network coding. In *Proc. of NetCod07*.

Majumdar J and Bhunia A K (2007). Elitist genetic algorithm for assignment problem with imprecise goal. *European Journal of Operational Research*, **177**(2): 684-692.

Marczewski E and Steinhaus H (1958). On a certain distance of sets and the corresponding distance of functions. *Colloquium Mathematicum* 6.

Marler R T and Arora J S (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, **26**: 369-395.

Marsili Libelli S and Alba P (2000). Adaptive mutation in genetic algorithms. *Soft Computing*, **4**: 76-80.

Mashohor S, Evans J R, and Arslan T (2005). Elitist selection schemes for genetic algorithm based printed circuit board inspection system. In: *Proc. 2005 IEEE Congress on Evolutionary Computation (CEC2005)*, pp.974-978.

Mei Y, Tang K, and Yao X (2011). Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. *IEEE Transactions on Evolutionary Computation*, **15**(2): 151-165.

Merz P and Freisleben B (2001). Memetic algorithms for the travelling salesman problem. *Complex Systems*, **13**(4): 297-345.

Michalewicz Z, Fogel D B (2004). *How To Solve It: Modern Heuristics*, Springer.

Miettinen K (1999). Nonlinear multiobjective optimization. Kluwer Academic Publishers.

Miller C K (1998). *Multicast networking and applications*, Pearson Education.

Mills P, Tsang E, and Ford J (2003). Applying an extended guided local search on the quadratic assignment problem. *Annals of Operations Research* **118**: 121-135.

Mitchell M (1996). *An introduction to genetic algorithms*. MIT Press.

Mitchell M and Taylor C E (1999). Evolutionary computation: an overview. *Annual Review of Ecology and Systematics*, **30**: 593-616.

Montoya J, Donoso Y, Montoya E, and Echeverri D (2008). Multiobjective model for multicast overlay networks over IP/MPLS using MOEA. In: *Proc. International Conference on Optical Network Design and Modeling (ONDM 2008)*, pp. 1-6.

Moscato P (1989). On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, no.826, California Institute of Technology.

Moscato P, Cotta C, and Mendes A (2004). Memetic algorithms. In: Onwubolu and Babu (eds.) *New Optimization Techniques in Engineering*, Springer-Verlag, pp. 53-85.

Moscato P and Cotta C (2007). Memetic algorithms. In: González (ed.) *Handbook of Approximation Algorithms and Metaheuristics*, Chapman & Hall / CRC Press.

Moscato P and Cotta C (2010). A Modern Introduction to Memetic Algorithms. In: Gendreau and Potvin (eds.) *Handbook of Metaheuristics* (2nd edition), *International Series in Operations Research and Management Science* 146, pp. 141-183.

Moy J (1994). MOSPF: analysis and experience. RFC 1585.

Mühlenbein H and Paaß G (1996). From recombination of genes to the estimation of distributions I. Binary parameters. In: *Proc. Parallel Problem Solving From Nature (PPSN IV)*, LNCS 1141, pp. 178-187.

Mühlenbein H, Mahnig T, Rodriguez A O (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, **5**: 215-247.

Nebro A J and Durillo J J (2010). A study of the parallelization of the multi-objective metaheuristic MOEA/D. In *Proc 4th International Conference on Learning and Intelligent Optimization (LION4)*, pp. 303-317.

Nissen V and Gold S (2008). Survivable network design with an evolution strategy. *Studies in Computational Intelligence*, **92**: 263-283.

Noguchi T, Matsuda T, and Yamamoto M (2003). Performance evaluation of new multicast architecture with network coding. *IEICE Transactions on Communications*, **E86-B**(6): 1788-1795.

Oduguwa V, Tiwari A, and Roy R (2005). Evolutionary computing in manufacturing industry: an overview of recent applications. *Applied Soft Computing*, **5**: 281-299.

Oh I S, Lee J S, and Moon B R (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(11): 1424-1437.

Pal S K, Bandyopadhyay S, and Ray S S (2006). Evolutionary computation in bioinformatics: a review. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, **36**(5): 601-615.

Pang H, Hu K and Hong Z (2006). Adaptive PBIL algorithm and its application to solve scheduling problems. In *Proc. ISIC 2006*, pp.784-789.

Pang A C, Wang S M and Tsao S L (2005). QoS multicasting over mobile networks. In: *Proc. 2005 IEEE GLOBECOM*.

Pau G, Vieron J, and Pesquet-Popescu, B (2005). Video coding with flexible MCTF structures for low end-to-end delay. In: *Proc. 2005 IEEE International Image Processing (ICIP2005)*.

Pedro L R and Takahashi R H C (2013). Decision-maker preference modeling in interactive multiobjective optimization. In *EMO 2013*, LNCS 7811, pp. 811-824.

Pelikan M and Mühlenbein H (1999). The bivariate marginal distribution algorithm. In: Roy, Furuhashi and Chawdhry (eds.) *Advances in Soft Computing — Engineering Design and Manufacturing*, pp. 521-535.

Pelikan M, Goldberg D E, and Cantú-Paz E (2000). Hieratchical problem solving by the Bayesian optimization algorithms. IlliGAL Report no. 2000002, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

Pelikan M, Goldberg D E, and Lobo F G (2002). A survey of optimisation by building and using probabilistic models. *Computational Optimization and Applications*, **21**(1): 5-20.

Phan D H, Suzuki J, and Hayashi I (2011). BIBEA: boosted indicator based evolutionary algorithm for multiobjective optimization. In: *Proc. 15th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES2011)*, Yokosuka, Japan.

Pinto D, Barán B, and Fabregat R (2005). Multi-objective multicast routing based on ant colony optimization. In: *Proc. 2005 Conference on Artificial Intelligence Research and Development*, pp. 363-370.

Pinto D and Barán B (2006). Multiobjective multicast routing with ant colony optimization. In: *Proc. 5th IFIP International Conference on Network Control and Engineering for QoS, Security and Mobility*.

Platel M D, Schliebs S, and Kasabov N (2008). Quantum-inspired evolutionary algorithm: a multimodel EDA. *IEEE Transaction on Evolutionary Computation*, **13**(6): 1218-1232.

Poli R (2008). Analysis of the publications on the applications of particle swarm optimization. *Journal of Artificial Evolution and Applications*, 1-10.

Prado R S, Silva R C P, Guimarães F G, and Neto O M (2010). Using differential evolution for combinatorial optimization: a general approach. In: *Proc. 2010 IEEE International Conference on Systems Man and Cybernetics (SMC 2010)*, pp. 11-18.

Preux P and Talbi E G (1999). Towards hybrid evolutionary algorithms. *International Transactions in Operational Research*, **6**: 557-570.

Pu W, Luo C, Wu F, and Chen C W (2009). QoS-driven network coded wireless multicast. *IEEE Transactions on Wireless Communications*, **8**(11): 5662-5670.

Puchinger J and Raidl G R (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification. In: *Proc. the First International Work-Conference on the Interplay Between Natural and Artificial Computation*, LNCS, Springer.

Qian B, Wang L, Huang D, and Wang X (2009). Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution. *Soft Computing*, **13**(8-9): 847-869.

Qu R, Burke E K, McCollum B, Merlot L T G and Lee S Y (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, **12**(1): 55-89.

Rami S Y, Cheng W, and Tao S (2008). Cost minimization for multi-source multi-sinks network coding. In *Proc. of ICYCS*, Washington, DC, USA.

Reeves C R (1995). *Modern heuristic techniques for combinatorial problems*. Advanced Topics in Computer Science, McGraw-Hill.

Reeves C R (2003). Genetic algorithms. In: Glover and Kochenberger (Eds) *Handbook of Metaheuristics*. pp.55-82. Kluwer Academic Publishers.

Ribeiro C C C, Martins S L, and Rosseti I C M (2007). Metaheuristics for optimisation problems in computer communications. *Computer Communications*, **30**(4): 656-669.

Rodríguez Villalobos C A and Coello Coello C A (2012). A new multi-objective evolutionary algorithm based on a performance assessment indicator. In *Proc. Genetic and Evolutionary Computation Conference (GECCO'12)*, Philadelphia, Pennsylvania, USA.

Sakawa M and Kato K (2002). An interactive fuzzy satisficing method for general multiobjective 0-1 programming problems through genetic algorithms with double strings based on a reference solution. *Fuzzy Sets and Systems* **125**(3): 289-300.

Sanchis J, Martínez M A, and Ferragud X B (2008). Integrated multiobjective optimization and a priori preferences using genetic algorithms. *Information Sciences*, **178**(4): 931-951.

Santana R (2011). Estimation of distribution algorithms: from available implementations to potential developments. In: *Proc. GECCO'11*, pp. 679-686.

Santana-Quintero L V, Hernández-Díaz, Molina J, Coello Coello C A, and Caballero R (2010). A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems. *Computers & Operations Research* **37**(3): 470-480.

Schaffer J D (1985). Multiple objective optimization with vector evaluated genetic algorithms. In: *Proc. International Conference on Genetic Algorithms and Their Applications*, pp. 93-100.

Schaffer J D, Caruana R A, Eshelman L J, and Das R (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In: *Proc. 3th International Conference on Genetic Algorithms and Their Applications*, San Mateo, California, pp.51-60.

Schutze O, Esquivel X, Lara A, and Coello Coello C A (2012). Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* **16**(4): 504-522.

Sengupta S, Rayanchu S, and Banerjee S (2007). An analysis of wireless network coding for unicast sessions: the case for coding-aware routing. In *Proc. of IEEE INFOCOM*.

Serrano V, Alvarado M, and Coello Coello C A (2007). Optimization to manage supply chain disruptions using the NSGA-II. In *Proceedings of IFSA 2*, pp. 476-485.

Sheetekela S P and Folly K A (2010). Power system controller design: a comparison between breeder genetic algorithm and population based incremental learning. In: *Proc. 2010 International Joint Conference on Neural Networks (IJCNN 2010)*, pp. 1-8.

Shi Y J, Liu Z C, and Ma S (2010). An improved evolutionary strategy for constrained circle packing problem. In: Huang et al (eds.) *ICIC 2010*, LNCS 6215, pp. 86-93.

Shim V A, Tan K C, and Cheong C Y (2012). A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **42**(5): 682-691.

Shwe H Y and Adachi F (2011). Power efficient adaptive network coding in wireless sensor networks. In: *Proc. 2011 IEEE International Conference on Communications (ICC)*, pp. 1-5.

Silva R R, Lopes H S, and Erig Lima C R (2008). A compact genetic algorithm with elitism and mutation applied to image recognition. In *Proc. of the 4th International Conference on Intelligent Computing (ICIC'08)*, pp. 1109-1116.

Silverman B W (1986). *Density estimation for statistics and data analysis*, London: Chapman and Hall.

Singh V K and Sharma V (2012). Elitist genetic algorithm based energy efficient routing scheme for wireless sensor networks. *International Journal of Advanced Smart Sensor Network Systems*, **2**(2): 15-21.

Smith R E, Forrest S, and Perelson A S (1993). Searching for diverse, cooperative population with genetic algorithms. *Evolutionary Computation*, **1**: 127-149.

Srinivas N and Deb K (1995). Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation*, **2**(3): 221-248.

Srinivas M and Patnaik L M (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, **24**(4): 656-667.

Storn R and Price K (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, **11**: 341-359.

Striegel A and Manimaran G (2002). A survey of QoS multicasting issues. *IEEE Communications Magazine*, **40**(6): 82-87.

Surry P D and Radcliffe N J (1996). Inoculation to initialize evolutionary search. In: Fogarty (ed.) *Evolutionary Computing: AISB Workship*, LNCS, 1143: 269-285.

Szeto K Y and Lo M H (2004). An application of adaptive genetic algorithm in financial knapsack problem. In: *Innovations in Applied Artificial Intelligence*, LNCS, 3029: 1220-1228.

Tai K, Wang N, and Yang Y (2008). Hybrid GA multiobjective optimization for the design of compliant micro-actuators. In: *Proc. 2008 IEEE International Conference on Systems, Man and Cybernetics*, pp. 559-564.

Tan K C, Yang Y J, and Goh C K (2006). A distributed cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, **10**(5): 527-549.

Tan Y Y, Jiao Y C, Li H, and Wang X K (2012). MOEA/D-SQA: a multi-objective memetic algorithm based on decomposition. *Engineering Optimization* **44**(9): 1095-1115.

Tang Z, Zhu Y, Wei G, and Zhu J (2008). An elitist selection adaptive genetic algorithm for resource allocation in multiuser packet-based OFDM systems. *Journal of Communications*, **3**(3): 27-32.

Tang L and Wang X (2013). A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* **17**(1): 20-46.

Thaler D, Estrin D, Meyer D (1998). Border gateway multicast protocol (BGMP): protocol Specification. RFC 2362.

Thiele L, Miettinen K, Korhonen P J, and Luque J M (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, **17**(3): 411-436.

Thierens D (1998). Selection schemes, elitist recombination and selection intensity. In: P*roc. 7th International Conference on Genetic Algorithms*, pp. 152-159.

Thomos N and Frossard P (2009). Network coding and media streaming. *Journal of Communications*, **4**(9): 628-639.

Tinos R and Yang S (2007). A self-organizing random immigrants genetic algorithm for dynamic optimization problems. *Genetic Programming and Evolvable Machines*, **8**(3): 255-286.

Tsang E and Voudouris C (1997). Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. *Operations Research Letters* **20**(3): 119-127.

Vasile M and Zuiani F (2010). A hybrid multiobjective optimization algorithm applied to space trajectory optimization. In: *Proc. 2010 IEEE Congress on Evolutionary Computation (CEC 2010)*, pp. 1-8.

Vellev S (2008). An adaptive genetic algorithm with dynamic population size for optimizing join queries. In: *Advanced Research in Artificial Intelligence*, pp. 82-88.

Ventresca M and Tizhoosh H R (2008). A diversity maintaining population-based incremental learning algorithm. *Information Sciences*, **178**(21): 4038-4056.

Voudouris C and Tsang E (1999). Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research* **113**(2): 469-499.

Voudouris C and Tsang E (2003). Guided local search. In: Glover (ed.) Handbook of Metaheuristics, Kluwer, pp. 185-218.

Voudouris C, Tsang E, and Alsheddy (2010). Guided local search. In: Gendreau and Potvin (eds.) Handbook of Metaheuristics, Springer, pp. 321-361.

Wagner T and Trautmann H (2010). Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation* **14**(5): 688-701.

Waitzman D, Partridge C, Deering S (1988). Distance vector multicast routing protocol. RFC 1075.

Walsh J M and Weber S (2008). A concatenated network coding scheme for multimedia transmission. In: *2008 Workshop on Network Coding, Theory and Applications (NetCod2008)*.

Wang J Q, Qin J, and Kang L S (2006). A new QoS multicast routing model and its immune optimization algorithm. In: *Ma et al (Eds) Ubituitous Intelligence and Computing*, LNCS 4159, pp. 369-378.

Wang L, Tang F, and Wu H (2005). Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. *Applied Mathematics and Computation*, **171**(2): 1141-1156.

Wang Y, Cai Z, Guo G, and Zhou Z (2007). Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B-Cybernetics* **37**(3): 560-575.

Wang L and Li L (2010). An effective hybrid quantum-inspired evolutionary algorithm for parameter estimation of chaotic systems. *Expert Systems with Applications*, **37**(2): 1279-1285.

Wang L, Yang Y, and Zhao W (2012). Network coding-based multipath routing for energy efficiency in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*. Doi:10.1186/1687-1499-2012-115.

Wang M and Li B (2007a). Lava: a reality check of network coding in peer-to-peer live streaming. In *Proc. of IEEE INFOCOM*.

Wang M and Li B (2007b). R2: Random push with random network coding in live peer-to-peer streaming. *IEEE Journal on Selected Areas in Communications,* **25**(9): 1655-1666.

Wang S, Gao X, and Zhuo L (2009a). Survey of network coding and its benefits in energy saving over wireless sensor networks. In *Proc. of International Conference on Information, Communications and Signal Processing (ICICS 2009)*.

Wang N, Li X, and Liu L (2009b). An adaptive multiobjective image watermarking scheme for QIM using NSGA-II. In: *Proc. of 2009 International Conference on Computational Intelligence and Security (CIS 2009)*, Beijing, pp. 273-277.

Wang X, Wang J, and Xu Y (2010). Data dissemination in wireless sensor networks with network coding. *EURASIP Journal on Wireless Communications and Networking*. Doi:10.1155/2010/465915.

Whitley D (1995). Modeling hybrid genetic algorithms. In: Winter (eds.) *Genetic Algorithms in Engineering and Computer Sciences*, Wiley, pp. 191-201.

Whitley D, Rana S, and Heckendorn R B (1998). The island model genetic algorithm: on separability, population size and convergence. *Journal of Computing and Information Technology*, **7**: 33-47.

Whitley D, Hains D, and Howe A (2010). A hybrid genetic algorithm for the travelling salesman problem using generalized partition crossover. In: *Proc. 11th International Conference on Parallel Problem Solving from Nature*, pp.566-575.

Widmer J, Fragouli C, and LeBoudec J Y (2005). Energy efficient broadcasting in wireless ad hoc networks. In *Proc. of First Workshop on Network Coding (NetCod05)*.

Wu Y, Chou P, Zhang Q, Jain K, Zhu W, and Kung S Y (2005a). Network planning in wireless ad hoc networks: a cross-layer approach. *IEEE Journal on Selected Areas in Communications*, **23**(1): 136-150.

Wu Y, Chou P A, and Kung S Y (2005b). Minimum-energy multicast in mobile Ad Hoc networks using network coding. *IEEE Transactions on Communications*, **53**(11): 1906-1918.

Wu Y, Wang Y and Liu X (2010). Multi-population based univariate marginal distribution algorithm for dynamic optimization problems. *Journal of Intelligent & Robotic Systems*, **59**(2): 127-144.

Xing H, Liu X, Jin X, Bai L, and Ji Y (2009a). A multi-granularity evolution based quantum genetic algorithm for QoS multicast routing problem in WDM networks. *Computer Communications*, **32**(2): 386-393.

Xing H, Ji Y, Lin Bai, Liu X, Qu Z, and Wang X (2009b). An adaptive-evolution-based quantum-inspired evolutionary algorithm for QoS multicasting in IP/DWDM networks. *Computer Communications*, **32**(6): 1086-1094.

Xing H, Ji Y, Bai L, and Sun Y (2010). An improved quantum-inspired evolutionary algorithm for coding resource optimization based network coding multicast scheme. *International Journal of Electronics and Communications* (AEU), **64**(12): 1105-1113.

Xing H and Qu R (2011a). A population based incremental learning for network coding resources minimization. *IEEE Communications Letters*, **15**(7): 698-700.

Xing H and Qu R (2011b). A population based incremental learning for delay constrained network coding resource minimization. In *Proc. of EvoApplications 2011*.

Xing H and Qu R (2012). A compact genetic algorithm for the network coding based resource minimization problem. *Applied Intelligence*, **36**(4): 809-823.

Xu Y (2011). Metaheuristic approaches for QoS multicast routing problems. Ph.D. Thesis, School of Computer Science, The University of Nottingham.

Xu Y and Qu R (2010). Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighborhoods. *Journal of Operational Research Society* **62**: 313-325.

Xu Y and Qu R (2012). An iterative local search approach based on fitness landscapes analysis for the delay-constrained multicast routing problem. *Computer Communications*, **35**: 352-365.

Xu Y, Qu R, and Li R (2013). A simulated annealing based genetic local search algorithm for multi-objective multicast routing problems. *Annals of Operations Research*. doi: 10.1007/s10479-013-1322-7.

Xu J, Tan S, and Panda S K (2011). Optimization of economic load dispatch for a microgrid using evolutionary computation. In: *Proc. IECON 2011*, pp. 3192-3197.

Xuan Y and Lea C T (2011). Network-coding multicast networks with QoS guarantees. *IEEE/ACM Transactions on Networking*, **19**(1): 265-274.

Yang S (2005). Memory-enhanced univariate marginal distribution algorithms for dynamic optimization problems. In: *Proc. 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, pp.2560-2567.

Yang S (2008). Genetic algorithms with memory- and elitism-based immigrants in dynamic environments. *Evolutionary Computation*, **16**(3): 385-416.

Yang S, Cheng H, and Wang F (2010). Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, **40**(1): 52-63.

Yang S and Yao X (2005). Experimental study on population-based incremental learning algorithm for dynamic optimization problems. *Soft Computing*, **9**(6): 815-834.

Yang S and Yao X (2008). Population-based incremental learning with associate memory for dynamic environments. *IEEE Transactions on Evolutionary Computation*, **12**(5): 542-561.

Yang S Y, Ho S L, Ni G Z, Machado J M, and Wong K F (2007). A new implementation of population based incremental learning method for optimizations in electromagnetics. *IEEE Transactions on Magnetics*, **43**(4): 1601-1604.

Yeow W L, Hoang A T, and Tham C K (2009a). Minimizing delay for multicast-streaming in wireless networks with network coding. In: *Proc. 2009 IEEE INFOCOM*.

Yeow W L, Hoang A T, and Tham C K (2009b). On average packet delay bounds and loss rates of network-coded multicasts over wireless downlinks. In: *Proc. 2009 International Conference on Communications (ICC'09)*, pp.4880-4885.

Zapotecas Martínez S and Coello Coello C A (2011). A multi-objective particle swarm optimizer based on decomposition. In *Proc. 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'2011)*, pp.69-76.

Zeng D, Guo S, Leung V, and Hu J (2011). The exploration of network coding in IEEE 802.15.4 networks. *International Journal of Digital Multimedia Broadcasting*. Doi:10.1155/2011/310647.

Zhan Z H, Li J, Cao J, Zhang J, Chung S H, and Shi Y H (2013). Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems. *IEEE Transactions on Cybernetics* **43**(2): 445-463.

Zhang Z and Chen S (2008). Dynamic constrained multiobjective optimization immune algorithm and its application. In: *Proc. 7th World Congress on Intelligence Control and Automation (WCICA2008)*, pp. 8727-8732.

Zhang Q and Li H (2007). MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, **11**(6): 712-731.

Zhang Q, Sun J, Xiao G, and Tsang E (2007). Evolutionary algorithms refining a heuristic: a hybrid method for shared path protections in WDM networks under SRLG constraints. *IEEE Transactions on Systems, Man and Cybernetics (SCM-B)* **37**(1): 51-61.

Zhang Q, Li H, Maringer D, and Tsang E (2010a). MOEA/D with NBI-style Tchebycheff approach for portfolio management. In *Proc 2010 IEEE Congress on Evolutionary Computation (CEC 2010)*, pp. 1-8.

Zhang Q, Liu W, and Li H (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. Technical Report CES-491, The School of Computer Science and Electronic Engineering, University of Essex.

Zhang Q, Liu W, Tsang E, and Virginas B (2010b). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, **14**(3): 456-474.

Zhang Y and Li X (2011). Estimation of distribution algorithm for permutation flow shops with total flowtime minimization. *Computers and Industrial Engineering*, **60**(4): 706-718.

Zhang G, Wang S and Li Y (2010c). A self-adaptive genetic algorithm based on the principle of searching for things. *Journal of Computers*, **5**(4): 646-653.

Zhang G, Zhang G, Cheng S (2011). LANC: Locality-aware network coding for better P2P traffic localization. *Computer Networks*, **55**(6): 1242-1256.

Zhang H, Zhou J, Chen Z, and Li J (2009). Minimizing delay for video conference with network coding. In *Proc. of ACM SIGCOMM 2009*.

Zhang P, Jiang Y, Lin C, Fan Y, and Shen X (2010d). P-coding: secure network coding against eavesdropping attacks. In: *Proc*. *IEEE INFOCOM 2010*.

Zhao F, Médard M, Lun D S, and Ozdaglar A (2009). Minimum cost subgraph algorithms for static and dynamic multicasts with network coding. In: Tarokh V (ed.) *New Directions in Wireless Communications Research*, Springer, pp.317-349.

Zhao S Z, Suganthan P N, and Zhang Q (2012). MOEA/D with an ensemble of neighbourhood sizes. *IEEE Transactions on Evolutionary Computation* **16**(3): 442-446.

Zhou A, Qu B Y, Li H, Zhao S Z, Suganthan P N, and Zhang Q (2011). Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm and Evolutionary Computation*, **1**(1): 32-49.

Zhu H, Jiao L and Pan J (2006). Multi-population genetic algorithm for feature selection. In: Jiao et al (eds) *ICNC*, LNCS 4222, pp. 480-487.

Zhuo J, Chen Y Z, and Chen Y P (2010). A novel multi-objective optimization algorithm based on discrete PSO for QoS multicast routing in wireless mesh networks. In: *Proc. International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, pp. 1-4.

Zitzler E and Künzli S (2004). Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature (PPSN VIII)*, LNCS 3242, pp. 832-842.

Zitzler E, Laumanns M, Thiele L (2001). SPEA2: improving the strength Pareto evolutionary algorithm. TIK-Report 103, Computer Engineering and Networks Laboratory (TIK), ETH.

Zitzler E, Laumanns M, Bleuler S (2004). A tutorial on evolutionary multiobjective optimization. In: Gandibleux et al (eds.) *Metaheuristics for Multiobjective Optimization*, **535**:3-37.

Zitzler E and Thiele L (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, **3**(4): 257-271.