An Iterative Local Search Approach based on Fitness Landscapes Analysis for the Delay-constrained Multicast Routing Problem

Ying Xu[1] · Rong Qu[2]

[1]College of Information Science and Engineering, Hunan University
Changsha, 410082, CHINA
hnxy@hnu.edu.cn

[2] The Automated Scheduling, Optimisation and Planning (ASAP) Group
School of Computer Science, The University of Nottingham
Nottingham, NG8 1BB, UK
rxq@cs.nott.ac.uk

Abstract. This paper presents the first fitness landscape analysis on the delay-constrained least-cost multicast routing problem (DCLC-MRP), a well-known NP-hard problem. Although the problem has attracted an increasing research attention over the past decade in telecommunications and operational research, little research has been conducted to analyze the features of its underlying landscape. Two of the most commonly used landscape analysis techniques, the fitness distance correlation analysis and the autocorrelation analysis, have been applied to analyze the global and local landscape features for DCLC-MRPs. A large amount of simulation experiments on a set of problem instances generated based on the benchmark Steiner tree problems in the OR-library reveals that the landscape of the DCLC-MRP is highly instance dependent with different landscape features. Different delay bounds also affect the distribution of solutions in the search space. The autocorrelation analysis on the benchmark instances of different sizes and delay bounds shows the impact of different local search heuristics and neighborhood structures on the fitness distance landscapes of the DCLC-MRP. The delay bound constraint in the DCLC-MRP has shown a great influence on the underlying landscape characteristic of the problem. Based on the fitness landscape analysis, an iterative local search (ILS) approach is proposed in this paper for the first time to tackle the DCLC-MRP. Computational results demonstrate the effectiveness of the proposed ILS algorithm for the problem in comparison with other algorithms in the literature.

Keywords: landscape analysis; multicast routing; Iterative local search algorithms.

1. Introduction

Multicast routing becomes an important communication technique due to the increasing development of numerous multimedia applications (e.g. distance education, E-commerce and video-conferencing) where information needs to be sent to multiple end-users at the same time in the underlying computer networks. Most of these real time multimedia applications require a certain level of Quality of Service (QoS) including the bandwidth, cost, delay, delay variation, lost ratio and hop counts, etc. The multicast routing problem (MRP) can be defined as to construct a multicast tree in computer networks to simultaneously transfer the information from a source node to a group of destination nodes satisfying the required QoS constraints.

In this work we consider the delay-constrained least-cost MRP (DCLC-MRP) in computer communication networks. The problem concerns two of the most common and important QoS requirements, namely the cost and delay of the multicast tree. The cost of a multicast tree is the total cost occurred from occupying all the links in the tree. The end-to-end delay is defined as the total delay of transferring information via the links along the path from the source to a destination. The delay of a multicast tree is the maximum end-to-end delay among all the end-to-end delays to each destination. The DCLC-MRP is to construct a multicast tree which connects the source and all the destinations with the minimum tree cost while satisfying the delay constraint, i.e., the maximum end-to-end delay is within the required delay bound.

The MRP can be reduced to the Minimum Steiner Tree Problem in Graphs [1], a well-known NP-hard combinatorial optimization problem [2]. The widely studied DCLC-MRPs we concern are also a well-known NP-hard problem [3], and can be defined as the Delay-Constrained Steiner Tree problem. Due to the complexity and challenges of multicast routing with different QoS requirements in real world applications, a large amount of research has been carried out for solving various QoS based MRPs since the 1990s. Since the first algorithm for the DCLC-MRP, i.e. the Kompella-Pasquale-Polyzos (KPP) heuristic [4], has been developed in the early 1990's, various heuristic and meta-heuristic algorithms have been proposed for solving a wide range of MRPs with different constraints. The majority of state-of-art algorithms for the DCLC-MRP are based on meta-heuristics such as simulated annealing [5, 6], tabu search [7, 8], genetic algorithms [9, 10], greedy random adaptive search procedure (GRASP) [11, 12], variable neighborhood search [13], and Scatter Search [14] etc. The rich literature of optimization algorithms for various MRPs can be found in several surveys [15-19].

Although a range of search algorithms has been investigated for the DCLC-MRP over the past decade, to the best of our knowledge, no research effort has been made to study the underlying landscape features of the highly constrained and complex problem. Most of the work in the literature mainly focuses on proposing specific heuristics or meta-heuristic algorithms for solving the particular problem concerned. The performance of the heuristic or meta-heuristic algorithm is usually determined by comparing it with other algorithms. The quality of the solutions found by the algorithm is used to reflect the effectiveness, and the computational time is usually used as a measure of the efficiency of the algorithm. Instead of demonstrating the application of specific heuristics or meta-heuristics for the particular combinatorial optimization problem being concerned, some recent research in operational research focuses on analyzing the fitness landscape of the problem. The theoretical analysis of the fitness landscape has shown to be useful for observing the behavior of search algorithms and thus valuable for designing efficient algorithms with better performance [20].

As far as we know, there are only three related work [10, 21, 22] on the landscape analysis for a different type of MRP, namely the group MRP, where a sequence of multicast requests is scheduled. In [21], a logarithmic simulated annealing (LSA) is used to perform detailed landscape analysis on three instances. Their work focuses on the approximation of optimal parameter settings. The difference between two parameters: the estimation of the maximum value of the minimum escape height from the local optima, and the maximum increase value of the objective function between two successive improvements of the best objective function value found during the local search time, are observed in LSA by a number of experiments. The results show that the difference depends mainly on the cost function and the capacity constraint, while only slightly on the particular network structure. Zahrani et al. have further extended their work in [10], by introducing a LSA based genetic local search (GLS) algorithm for the landscape analysis on the group multicast routing problem. The GLS algorithm applies the partial mixed crossover (PMX) operation to pairs of individuals. As the pre-processing step in the GLS, the LSA based landscape analysis aims to estimate the depth of the deepest local minima. Experimental results on two benchmark instances show that the proposed LSA-based GLS together with the PMX operation outperforms two variants of GLS algorithms with either LSA or PMX only. The same GLA algorithm with a LSA as the pre-processing step is applied by Zahrani et al. in [10] on more instances for the group multicast routing problem with the capacity constraint. Similarly, the LSA is also used to estimate the value of maximum depth of local optima in the landscape. The same conclusion that the GLS combined with LSA and PMX outperforms the algorithms using either LSA or PMX is obtained as that drawn by Zahrani et al. [22].

Before properly choosing a specific heuristic or meta-heuristic for a hard optimisation problem, the theoretical landscape analysis of the problem is very useful for observing the behaviour of search algorithms and thus can help predicting their performance. The lack of the theoretical analysis on the landscapes of the DCLC-MRP motivates the first fitness landscape analysis for the problem in this paper. Our aim is to obtain a better understanding of landscape properties for the DCLC-MRP with the single multicast request and to provide the theoretical basis before designing effective search algorithms for the problem. Firstly, we present the fitness distance analysis and the autocorrelation analysis of the insight landscape features on some benchmark instances of the DCLC-MRP. Then, based on the large amount of theoretical landscape analyses, iterative local search (ILS) [23, 24] has been chosen in this paper as the appropriate meta-heuristic for the DCLC-MRP based on the observations of the existence of many local optimal solutions in the search space. ILS is a simple and powerful meta-heuristic which has been successfully applied to a wide variety of optimization problems [23-29]. ILS has two basic operators, namely the local search procedure and the perturbation operator for generating new solutions. Whenever the local search is trapped in a local optimum, a perturbation operator is applied to the local optimum to generate a new starting point in a neighboring area of the local optimum. ILS iteratively applies the local search to the new starting points generated by perturbing the current search point, leading to a randomized walk of local optima in the search space. This search procedure of ILS is very suitable for tackling the DCLC-MRP since many local optima are distributed over the search space.

The paper is organized as follows. Section 2 introduces the concept of the fitness landscape and describes the methods of analyzing fitness distance correlation and autocorrelation. Section 3 presents the formal definition of the DCLC-MRP. In Section 4, the fitness distance correlation and autocorrelation analysis have been carried out on a set of DCLC-MRPs based on the extended benchmark Steiner tree instances (Steinb) in the OR-library[30]. Section 5 presents the proposed iterative local search algorithms and shows the effectiveness of the proposed algorithms by a series of experiments. Finally, Section 6 concludes the paper and suggests future work.

2. Fitness Landscape Analysis

The concept of fitness landscapes [20] in biological evolutionary optimization algorithms has been introduced to combinatorial optimization to measure the problem landscape, and shown to be a very useful tool for understanding and predicting the behavior and performance of algorithms. Based on a defined solution representation and fitness function, the landscape of an optimization problem can be defined as consists of points of different height (solutions of different fitness function value) in the multi-dimensional hyper-space. A heuristic algorithm can be seen as a search procedure to traverse through the landscape in order to find the highest peak (or the lowest point) for the maximization (or minimization) problem.

A fitness landscape of a given combinatorial optimization problem can be defined as $L(X, f, d)$, where X represents a set of points (solutions) in the landscape, f: $X \mapsto R$ is a fitness function which associates a real-valued fitness to each of the points in X, and d is a distance metric which defines the spatial structure of the landscape. Based on this, a fitness landscape can be interpreted as a graph $G_L = (V, E)$ with a set of vertices $V = X$ and a set of links $E = \{(x, y) \in X \times X \mid d(x, y) = d_{min}\}$, where d_{min} denotes the minimum distance between two points in the search space by using the distance metric d for a given optimization problem. Another property of the landscape is the diameter $diamG_L$, which is defined as the maximum distance between two points in the search space.

2.1 The Fitness Distance Correlation Analysis

The fitness distance correlation (*fdc*) coefficient proposed by Jones and Forrest [31] is the most commonly used measure to estimate the global feature of the fitness landscape. It has been widely used as a measure to predict the problem difficulty for search algorithms. Given a set of points $X=\{x_1, x_2, \ldots, x_m\}$ and their fitness values, the *fdc* coefficient ρ is defined as follows:

$$\rho(f, d_{opt}) = \frac{Cov(f, d_{opt})}{\sigma(f)\sigma(d_{opt})} \quad (1)$$

where $Cov(\cdot, \cdot)$ denotes the covariance of two variables; $\sigma(\cdot)$ denotes the standard deviation; f and d_{opt} represents the fitness of a sampled solution x and the distance between x and the global optimum (or the best-known solution) in the search space.

$$\overline{f} = \frac{1}{|X|}\sum_{x \in X} f(x) \quad (2)$$

where $\overline{f}$ is the average fitness value of the points in X, $|X|=m$.

$$\sigma(f) = \frac{1}{|X|}\sum_{x \in X} (f(x) - \overline{f}) \quad (3)$$

$$\sigma(d_{opt}) = \frac{1}{|X|}\sum_{x \in X} (d_{opt} - \overline{d_{opt}}) \quad (4)$$

where $\sigma(\cdot)$ denotes the standard deviation,

$$Cov(f, d_{opt}) = \frac{1}{|X|}\sum_{x \in X} ((f(x) - \overline{f})(d_{opt} - \overline{d_{opt}})) \quad (5)$$

For problems where the global optimal solution is not known, the best-known solution is usually used in the literature as an estimation of the global optimum in many studies. We denote x_{opt} in this study as the global optimal solution or the best-known solution in the search space. For a set of arbitrarily sampled points from the search space, *fdc* determines how closely their fitness and distance to x_{opt} are correlated by examining the statistic correlation between their f and d_{opt} using (1). If the fitness of the sampled solutions increases when a search moves towards x_{opt} (the distance becomes smaller), then the search is expected to be easy, as it is properly guided by the correlation along a "path" of solutions with increasing fitness to x_{opt}. A value of ρ = -1.0 (or ρ = 1.0) for a maximization (or minimization) problem indicates a perfect correlation between the fitness and the distance of a solution to x_{opt}, and thus predicts an easy problem which can be easily solved using any search algorithms. On the contrary, a value of ρ = 0 means that no correlation exists between the fitness and the distance to x_{opt}, and thus the underlying problem is hard to solve.

The *fdc* analysis has been conducted for various combinatorial optimization problems, including the travelling salesman problem [32], the flow-shop scheduling problem [33], the graph matching problem [34], the graph bipartitioning problem [35] and the timetabling problem [36], etc. A summary of landscape metrics and related issues on evolutionary algorithms can be found in [37].

2.2 The Autocorrelation Analysis

An important property of landscape is its ruggedness, which significantly influences the performance of search algorithms. In general, the more rugged a landscape is, the harder the problem can be solved by heuristic search algorithms. To measure the ruggedness of a fitness landscape, Weinberger [38] suggests a statistical method to analyze the correlation structure based on the autocorrelation method. The idea is to generate a random walk of a sufficiently large number of steps and calculate the correlation of neighboring points in the random walk. The fitness of each solution encountered during the random walk is recorded to obtain a time series of fitness values. The autocorrelation of fitness values obtained from the random walk of *W* steps in the search space can be empirically estimated by *r*(*i*) as follows:

$$r(i) = \frac{\sum_{t=1}^{W-i}(f_t - \bar{f})(f_{t+i} - \bar{f})}{\sum_{t=1}^{W}(f_t - \bar{f})} \quad (6)$$

where $\bar{f}$ is the mean fitness of the *W* points sampled, and *i* is the time lag or distance between two points along the random walk.

Based on this autocorrelation function, the correlation length *l* can be defined as follows [48]:

$$\ell = -\frac{1}{\ln(|r(1)|)} \quad (7)$$

where *r*(1) ≠ 0 is defined according to Eq. (6). The correlation length *l* directly reflects the ruggedness of a landscape, a smaller value of *l* indicates a more rugged landscape. A landscape is said to be smooth if there is a high correlation between neighboring points. The correlation length *l* is typically related to the size of the problem instance [39], so the relative correlation length to the landscape diameter *n* is often reported, i.e. *l* / *n*. The closer the value of *l* / *n* to 1, the smoother the landscape is presented by a search method. The autocorrelation analysis has been carried out in the literature for measuring the landscapes of different problems [40, 41, 36, 38].

3. Problem Definition of the DCLC MRP

A computer network can be modeled as a connected and directed graph *G* = (*V*, *E*) with |*V*| = *n* nodes and |*E*| = *l* links. Each link $e = (i, j) \in E$ which connects two adjacent vertices *i* and *j* is associated with a cost *c*(*e*) and a delay *d*(*e*). The cost *c*(*e*) is a measure of the utilization of the network resources, for example the bandwidth, along the link. The delay *d*(*e*) is the transferring delay of sending messages via the link. We assume the network is asymmetric, i.e. for the two links *e* = (*i*, *j*) and *e'* = (*j*, *i*), it is possible that *c*(*e*) ≠ *c*(*e'*) and/or *d*(*e*) ≠ *d*(*e'*). In a MRP, the information is sent simultaneously from a source node $s \in V$ to a given set of destination nodes $r_i \in D$, $D \subseteq V \setminus \{s\}$. The set of destination nodes *D* is called the multicast group, and |*D*| is defined as the group size of the MRP.

A solution for a MRP is a multicast tree $T(s, D) \subseteq E$ which roots at the source node *s* and connects all destination nodes $r_i \in D$. Extra nodes in $V \setminus (\{s\} \cup D)$ may be added to the multicast tree, called the Steiner nodes. A path in *T*(*s*, *D*) is a set of links along the path from the source node *s* to a destination node r_i, denoted by $P(s, r_i) \subseteq T(s, D)$. The end-to-end delay from *s* to each destination r_i is the sum of the delays of all links in $P(s, r_i)$, denoted by $Delay(r_i)$, and defined as follows:

$$Delay(r_i) = \sum_{e \in P(s, r_i)} d(e) \quad (8)$$

The delay of the multicast tree, denoted by *Delay*(*T*), is the maximum delay among all $Delay(r_i)$, defined as follows:

$$Delay(T) = \max\{\, Delay(r_i) \mid \forall r_i \in D \,\} \quad (9)$$

The total cost of the multicast tree, denoted by *Cost*(*T*), is defined as the sum of the costs of all links in the tree as follows:

$$Cost(T) = \sum_{e \in T} c(e) \tag{10}$$

In MRP, the delay bound is an upper bound of the end-to-end delay for the path from the source to each destination, i.e. $Delay(r_i)$. In reality, different applications may have different upper bound δ_i for each destination $r_i \in D$. In this paper and the majority of the literature, the same delay bound is applied to all destinations, denoted by $\Delta = \delta_i$ for all $r_i \in D$.

Given the above definitions, the DCLC-MRP can be formally defined as follows:

The Delay-Constrained Least Cost Multicast Routing Problem **(DCLC-MRP)**: Given a network topology G, a source node s, a set of destination nodes $r_i \in D$, a link cost function $c(\cdot)$, a link delay function $d(\cdot)$, and a delay bound Δ, the objective of the DCLC-MRP is to construct a multicast tree $T(s, D)$ such that the delay bound is satisfied, and the tree cost is minimized, defined as follows:

$$\min\{\ Cost(T) \mid P(s, r_i) \subseteq T(s, D), Delay(r_i) \leq \Delta, \forall\, r_i \in D\ \} \tag{11}$$

4. Landscape Analysis of the DCLC Multicast Routing Problem

4.1 Problem Instances and Solution Representation

There are no benchmark problem instances for DCLC-MRPs in the literature. In this study we generate problem instances of the DCLC-MRP by randomly setting delays of links in a set of Steiner tree benchmark instances in the OR-library. Table 1 presents the characteristics of the 18 Steinb instances, i.e. *B01-B18*, being concerned in this work. It is a set of Steiner tree problems of different sizes from SteinLib, a library of test instances for the Minimum Steiner Tree Problem in Graphs [42]. The global optimal solutions of these Steiner tree instances have already been obtained [30].

Table 1. Characteristics of the Steinb instances from the OR-library. $|V|$, $|E|$ and $|D|$ stand for the number of nodes, links and destinations, respectively. OPT: the cost of the optimal solution for each instance given by the OR-library.

Instances	$\|V\|$	$\|E\|$	$\|D\|$	OPT	Instances	$\|V\|$	$\|E\|$	$\|D\|$	OPT
B01	50	63	9	82	***B10***	75	150	13	86
B02	50	63	13	83	***B11***	75	150	19	88
B03	50	63	25	138	***B12***	75	150	38	174
B04	50	100	9	59	***B13***	100	125	17	165
B05	50	100	13	61	***B14***	100	125	25	235
B06	50	100	25	122	***B15***	100	125	50	318
B07	75	94	13	111	***B16***	100	200	17	127
B08	75	94	19	104	***B17***	100	200	25	131
B09	75	94	38	220	***B18***	100	200	50	218

To carry out the analysis on the properties of the underlying landscape for the DCLC-MRP, we use a multicast routing simulator based on Salama's generator [18] to run all experiments. The simulator can generate network topologies by using a graph generation method [43], where the distance $l(u, v)$ between pairs of nodes (u, v) is determined by the Euclidean metric. In our simulations, the capacity of each link is set to a large enough value thus restrictions on link capacity can be neglected in DCLC-MRPs. The link delay $d(e)$ in the simulator is defined as the propagation delay on the link (queuing and transmission delays are negligible), the link cost $c(e)$ is assigned according to those set in SteinLib in the OR Library.

For the DCLC-MRP, the delay bound plays an important role in obtaining feasible solutions in search algorithms. The smaller the delay bound, the tighter the problem is constrained. To observe the effect of different delay bounds on the fitness landscape of the DCLC-MRP, we have set three different delay bounds in the Steinb instances, namely $\Delta_1 = \infty$, $\Delta_2 = 1.1\times Delay(T_{OPT})$, and $\Delta_3 = 0.9\times Delay(T_{OPT})$, described as follows:

- The first delay bound $\Delta_1 = \infty$ is set as a very large positive number in our experiments. The DCLC-MRP is then actually relaxed to the unconstrained Steiner tree problem, since the delays of the links do not introduce any restriction in constructing the Steiner tree. The global optimal solutions of this set of unconstrained MRP, i.e. Steiner tree problems, have already been obtained in the OR-library, shown in Table 1.
- The second delay bound $\Delta_2 = 1.1\times Delay(T_{OPT})$ is set as greater than the delay in the optimal solution in Table 1, $Delay(T_{OPT})$, where T_{OPT} denotes the multicast tree of the optimal solution for each unconstrained Steiner tree instance. Therefore, we know that the optimal solution to the unconstrained Steiner tree problem is also the optimal solution to the DCLC-MRP.
- The third delay bound $\Delta_3 = 0.9\times Delay(T_{OPT})$ is set as lower than that of the optimal solutions for the unconstrained Steiner tree problem. The optimal solution is thus not known to any of the instances with this tighter delay bound. We therefore employ the best-known solutions obtained in the literature in our landscape analysis.

The landscape of these DCLC-MRP instances with different delay bounds are investigated in our work. More detailed information of all the problem instances tested and some example solutions obtained by the algorithms are publicly available at http://www.cs.nott.ac.uk/~rxq/benchmarks.htm.

In this first study of landscape analysis on the DCLC-MRP, we employ the mostly used binary vector as the solution representation. A binary vector of ordered *n* bits presents the solution (the multicast tree) for a DCLC-MRP with $|V| = n$ nodes, where each bit corresponds to one node in the network. All possible solutions are thus encoded as $X = \{0, 1\}^n$ of a fixed length *n*. This generates a hyper-space of $(n + 1)$ dimensions, where *n* is the size of the problem and the $(n + 1)$-th dimension represent the height (fitness) of the point (solution). Each bit in the vector takes a value of 1 if the corresponding node is in the multicast tree, 0 otherwise. As the source and all destination nodes must be included in a multicast tree, the value of their corresponding bits in the binary vector (of a feasible solution) should always be 1. This simple binary vector has been widely used in the literature and shows to be simple yet effective for representing a multicast tree [44, 11].

Based on the binary solution representation, Hamming distance has been used to calculate the distance between solutions. The minimum distance d_{min} between two DCLC-MRP solutions is 1 (one bit with a different value between two vectors), and the maximum distance $diamG_L$ between two solutions, also known as the diameter of the landscape, is $n - |D| - 1$, where the $|D|$ is the number of destination nodes in the multicast group. The range of distances thus will be within $[1, n - |D| - 1]$.

In this work, we perform our landscape analysis on three heuristic methods and two neighborhood operators using the mostly used two statistical methods in the literature of landscape analysis, namely the fitness distance correlation and the autocorrelation, to measure properties of the fitness landscape for DCLC-MRPs. The *fdc* of local optima to x_{opt} and autocorrelation of random walks is carried out to measure the global and local structure of the landscape of DCLC-MRPs with different delay constraints. The effects of different neighborhood operators and different heuristic algorithms upon the landscape of DCLC-MRPs with different delay constraints are further analyzed with regard to problem difficulty. All simulations have been run on a Windows XP computer of PVI 3.4GHZ with 1G RAM. For each instance 30 runs have been carried out to observe the average performance of different algorithms with different neighborhoods in the fitness landscape analysis.

4.2 Landscape Analysis of the DCLC Multicast Routing Problem

To sample a set of local optimal solutions which are evenly distributed in the search space to x_{opt}, a fixed number (10 in our experiments) of feasible solutions have been randomly generated for each distance of i bits away from x_{opt}, $i = 1, \ldots, n - |D| - 1$. This set of random solutions is then used as the starting points of a local search method to generate a set of local optima. As the local search starting from some random solutions may lead to the same local optima, up to $10\times(n - |D| - 1)$ number of local optima may be generated by the local search heuristics.

Different local search methods may be applied to generate different sets of local optima. In our basic *fdc* analysis, based on the binary solution representation defined above, a simple greedy local search heuristic is used to produce the local optima in a non-deterministic manner. Details of the local search heuristic are shown in Figure 1. A node-based neighborhood operator is employed to flip a randomly selected bit (excluding the bits of the source node and the destination nodes) in the binary vector of the current tree. This operator thus either removes an existing node from or adds a new node to the multicast tree. The modified Prim's spanning tree algorithm is then used to create a new minimum spanning tree based on the newly produced set of nodes in the binary vector. The new solution with a smaller tree cost or a lower delay than the current best solution will be accepted as the new current best solution during the search. This procedure is repeated until all possible bits have been flipped and the best solution found during the procedure is accepted as the local optimum. For a multicast tree in the network of n bits with a source node and $|D|$ destination nodes in the multicast group, the number of possible neighboring solutions of the current multicast tree is bounded by $n - |D| - 1$. This node-based neighborhood operator has also been applied in [44, 11].

```
LS(G = (V, E), s, D, Δ, T0)
{  // s: the source node; D: the destination nodes set, i.e. the multicast group; Δ ≥ 0: the delay bound;
   // T0: the random initial solution, represented by a binary vector corresponding to n nodes, n = |V|;
   Tbest = T0;
   k  = 1; kmax = n - |D| - 1;
   Mark all bits in T0 as unvisited;
   while k ≤ kmax do {
         Randomly select an unvisited bit x in T0; // s and nodes in D are excluded
         Flip the value of bit x to generate a new binary vector Tx;
         Generate a minimum spanning tree T' of the given nodes in Tx by using the Prim's algorithm
         if(((Cost(T') < Cost(Tbest)) and (Delay(T') ≤ Δ)) or
             ((Cost(T') = Cost(Tbest)) and (Delay(T') < Delay(Tbest)))
         then Tbest = T';
         mark bit x as visited;
         k++;
   end of while loop }
   return Tbest;
}
```

Figure 1. Pseudo code of the local search heuristic for producing the local optima.

It is known that higher values of *fdc* correlation ρ indicate that the fitness value and distance of solutions to x_{opt} in the search space are correlated and thus the problem presents to be easier for search algorithms. As suggested in [31], problem difficulty has been classified based on *fdc* coefficients, where a value of $\rho \geq 0.15$ for minimization problems indicates a straightforward and easy problem for search algorithms. For a value of $\rho \leq -0.15$, the distance to x_{opt} increases with decreasing fitness, indicating that the problem being concerned is deceptive and misleading. For problems with $-0.15 < \rho < 0.15$, the landscape presents no correlation, thus indicating that the problem being concerned are difficult to solve.

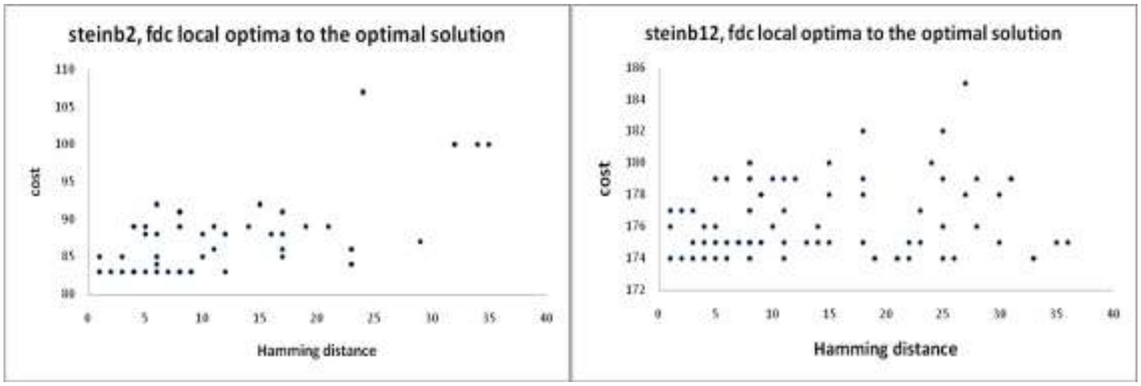
The average cost and the *fdc* coefficient ρ for each instance of different delay constraints are reported in Table 2. For the problem with $\Delta_1 = \infty$, four instances (*B03*, *B09*, *B15* and *B18*) show to be difficult (where $-0.15 < \rho < 0.15$). One instance (*B01*) presents to be a misleading problem, and others are relatively easy to solve, i.e. the fitness and distance to x_{opt} are correlated. For the DCLC-MRP with a slightly tighter delay bound $\Delta_2 = 1.1 \times Delay(T_{OPT})$, two instances (*B03* and *B09*) remain to be difficult, while all other instances turn to be relatively easier (where $\rho \geq 0.15$). For the DCLC-MRP with the tightest delay bound $\Delta_3 = 0.9 \times Delay(T_{OPT})$, two instances (*B09* and *B14*) are difficult problems, three instances (*B03*, *B13* and *B15*) are misleading, while the others are easy problems.

Table 2. The *fdc* of the DCLC-MRP instances with different delay bounds. Δ: the delay bound for the corresponding problem instance; σ: the standard deviation of the costs of the multicast tree; ρ: the *fdc* coefficient.

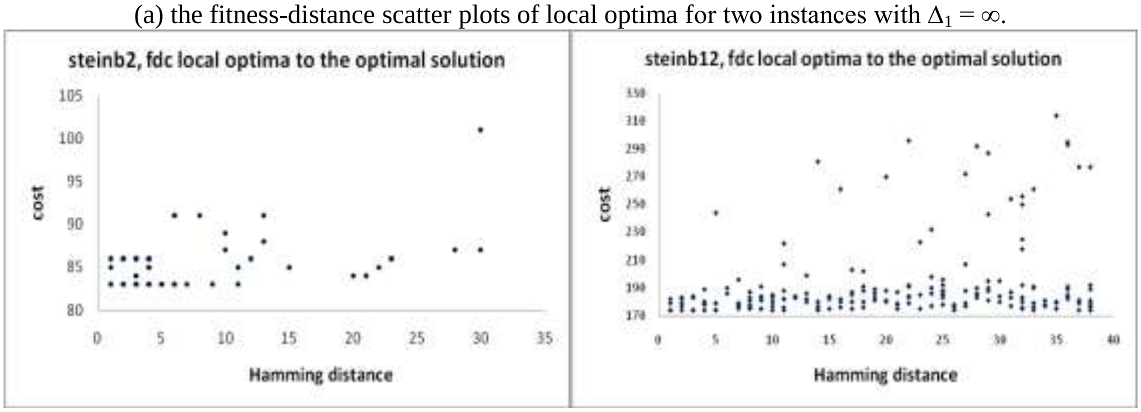
Problem instances	OPT	$\Delta_1 = \infty$			$\Delta_2 = 1.1 \times Delay(T_{OPT})$				$\Delta_3 = 0.9 \times Delay(T_{OPT})$			
		cost	σ	ρ	Δ	**cost**	σ	ρ	Δ	**cost**	σ	ρ
B01	82	82.35	0.79	-0.22	145	82.1	0.33	0.80	118	84	3	0.61
B02	83	86.09	4.09	0.62	228	84.94	3. 04	0.52	187	90.01	3. 44	0.38
B03	138	139.48	2.64	-0.04	248	139.5	2.58	-0.03	203	144.08	3.52	-0.19
B04	59	64.84	6.12	0.53	173	70.15	10.89	0.59	142	72.46	10.09	0.59
B05	61	62.70	2.31	0.31	125	64.32	5.28	0.50	102	66.25	7. 81	0.29
B06	122	125.32	2.62	0.23	281	125.76	3.11	0.37	199	134.01	10.76	0.21
B07	111	111.73	2.29	0.70	212	111.70	2.86	0.72	173	112.96	2.71	0.42
B08	104	104.54	1.94	0. 46	209	105.07	5.34	0.64	171	112.64	10.04	0.23
B09	220	220.58	1.57	0.11	280	220.38	0.45	-0.01	229	223.44	3.84	-0.11
B10	86	89.15	5. 03	0.43	262	90.58	5.52	0.56	215	91.43	5.72	0.39
B11	88	93.08	3.51	0.54	235	95.55	9.07	0.45	180	103.05	14.62	0.54
B12	174	182.86	2.22	0.33	225	189.96	25.86	0.37	184	203.44	37.42	0.47
B13	165	165.72	3.77	0.81	190	165.99	2.67	0.56	139	169.06	0.43	-0.19
B14	235	236.86	2.14	0.20	221	235.97	3.40	0.74	180	240.72	1.91	0.09
B15	318	318.55	1.07	-0.05	308	318.50	1. 15	0.69	194	339.51	9.89	-0.34
B16	127	132.26	7.15	0.50	291	132.25	6.26	0. 54	238	136.87	6.89	0.33
B17	131	136.03	4.46	0.58	219	137.69	6. 89	0.32	180	143.13	16.34	0.24
B18	218	219.55	2.22	0.13	425	219.67	1.92	0.27	348	221.47	2.83	0.21
Avg	140.11	142.87	3.00	0.34	/	143.89	5.70	0.47	/	149.36	8.75	0.23

From the experimental results, we can see that the fitness landscape for the DCLC-MRP is instance dependant with different level of difficulties for the search algorithm. In addition, different delay bounds significantly affect the landscapes of the same network topology. However, for the problem with delay bound Δ_2, although all instances except two instances (*B03* and *B09*) show to be easy to solve ($\rho \geq 0.15$), it does not necessarily mean those instances are easier for search algorithms compared with the same instances with $\Delta_1 = \infty$, as the delay bound restricts the search within disconnected areas due to the added constraint. This can be seen in Table 2, that for some instances with the delay bound Δ_2, the cost of the best multicast tree is higher than the problem of the same topology with no delay bound, i.e. $\Delta_1 = \infty$. Generally, the smaller the delay bounds, the tighter the constraints to the problem. This is shown by the worst average tree cost 149.36 over the 18 instances obtained for the problem with delay bound Δ_3, compared to 143.89 for the problem with delay bound Δ_2 and 142.87 for the problem with delay bound Δ_1.

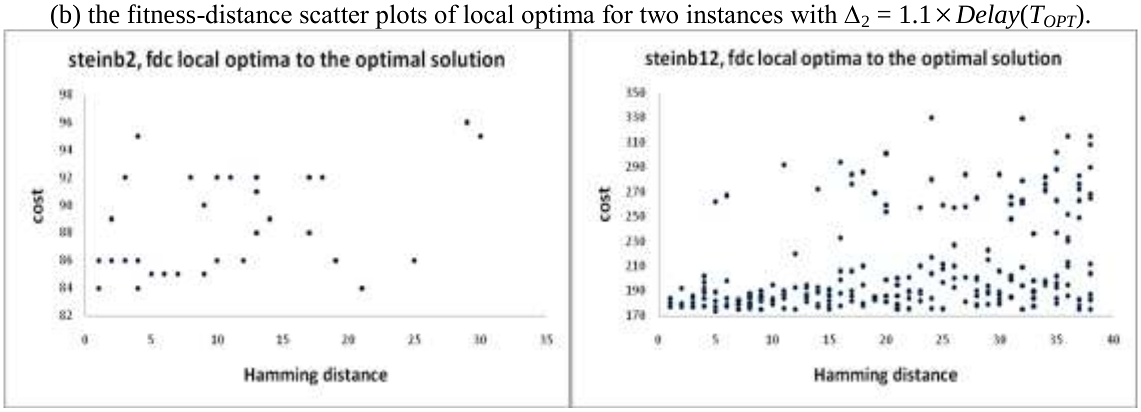
In addition to the *fdc* coefficient, a fitness-distance scatter plot can provide more insightful information of the landscape. Figure 2 presents the example fitness-distance scatter plots of two instances, clearly demonstrating the changes of the fitness distance landscape for problem instances with different delay bounds. For instance *B02*, the correlation between the fitness and the distance decreases from 0.71 for Δ_1, to 0.51 for Δ_2 and 0.4 for Δ_3, meaning the tighter the delay bound, the lower the ρ value, thus the higher level of difficulty for instance *B02*. For instance *B12*, all scatter plots show positive correlations with a ρ value larger than 0.15. However, with delay bound Δ_2 and Δ_3, we can see many shallow valleys (local optima) are crowed near x_{opt} for *B12*. This makes the search to be easily stuck at local optima. The introduction of the delay bounds also results in a different distribution of solutions in the search space.



(a) the fitness-distance scatter plots of local optima for two instances with $\Delta_1 = \infty$.



(b) the fitness-distance scatter plots of local optima for two instances with $\Delta_2 = 1.1 \times Delay(T_{OPT})$.



(c) the fitness-distance scatter plots of local optima for two instances with $\Delta_3 = 0.9 \times Delay(T_{OPT})$.

Figure 2. Fitness-distance scatter plots on selected instances with different delay bounds.

4.3 Impact of Different Local Search Methods on the Fitness Distance Correlation

Based on the above basic *fdc* analysis, we further analyze the impact of different local search methods on the fitness landscape. In addition to the above mentioned simple greedy local search in Figure 1 (hereinafter named LS), we re-implement three other local search heuristics, namely BSMA [45], TS-CST [44] and VNDMR [13] in the literature, for solving the DCLC-MRP with different delay constraints. We give the brief description of the three local search heuristics as follows:

1) The BSMA heuristic

A well known deterministic algorithm for the DCLC-MRP is BSMA (Bounded Shortest Multicast Algorithm) developed in mid 1990s [45]. Due to its good performance on tree cost, it has been seen as the best algorithm in early research on MRPs. In BSMA, a delay-bounded path switching operator is devised to replace a *superpath* in the multicast tree by a new alternative path, hopefully resulting in a new delay-bounded tree with a lower tree cost. The *superpath* is defined as the longest simple path between two end nodes, where all the internal Steiner nodes, except the two end nodes of the path, connect exactly only two other nodes in the tree. At each step, a chosen *superpath* is removed, leading to two sub-trees. The Dijkstra's shortest path algorithm is then consecutively called to find a new delay-bounded shortest path that connects the two sub-trees with a reduced tree cost. The BSMA heuristic iteratively refines the tree to lower costs until it cannot be further reduced.

2) The TS-CST heuristic

The TS-CST algorithm in [44] is developed based on a tabu search, where a binary vector solution representation has been used (and adapted in our study here, see Section 4.2 above). A solution (a multicast tree) is represented by a binary set of $|V \setminus (\{s\} \cup D)|$ bits, each corresponding to a node in $V \setminus (\{s\} \cup D)$. A value of 0 for the corresponding bit in the binary set represents that the node is included in the multicast tree, 1 otherwise. Neighborhood of a solution includes all the solutions which are exactly one bit different in the binary set from the chosen solution. In other words, the neighboring solutions are all those multicast trees generated by adding or removing exactly one node in $V \setminus (\{s\} \cup D)$ in the incumbent solution. The Prim's algorithm is applied to generate a new delay-constrained spanning tree on the given set of nodes. The best new neighboring solution is chosen as the current solution in the next iteration, i.e. a move in TS-CST. To prevent the heuristic from oscillating between neighboring solutions, a tabu list of length one is updated to remember the corresponding bit in the last performed move. The process stops after a pre-defined number of iterations without improvements, set to 2 in [44].

3) The VNDMR algorithm

A variable neighborhood descent search algorithm (VNDMR) has been investigated in our previous work in [13] for the DCLC-MRP. Three neighborhood structures, one is node-based and the other two are link-based neighborhoods, have been designed concerning the structure of the multicast network. The node-based neighborhood is similar to the neighborhood designed in the TS-CST algorithm. In generates a neighboring tree by removing or adding a Steiner node from or to the current multicast tree, and obtaining a minimum spanning tree which spans the nodes using the Prim's algorithm. Once a better multicast tree is found, the current solution is updated. This procedure is repeated until no improvement can be achieved for more than 3 iterations. The two link-based neighborhoods are designed based on a path replacement operator which is similar to the path switching operator in BSMA [45]. The path replacing operator replaces a *superpath* by a new delay-bounded path of a lower cost using the *k*-shortest path algorithm [46]. Both of the two link-based neighborhoods iteratively refine the multicast tree to lower costs until no better tree can be found. All three neighborhoods have been designed to reduce the tree cost while satisfying the delay bound constraint in the problem.

We present the *fdc* of local optima to x_{opt} on the landscape of different local search methods for four instances (*B02*, *B10*, *B12* and *B18*) of different sizes and delay bounds, results are shown in Table 3. The initial random solutions are generated by using the same method described above to sample random solutions of even distributions in the search space.

Table 3. The *fdc* of the landscape from four local search heuristics on four instances of different sizes and delay bounds. σ: the standard deviation of the costs of the multicast tree; ρ: the *fdc* coefficient. Best solutions are in bold.

Delay bounds	Problem instances	LS			BSMA			TS-CST			VNDMR		
		cost	σ	ρ	cost	σ	ρ	cost	σ	ρ	cost	σ	ρ
Δ_1	***B02***	86.09	4.09	0.62	85.58	3.51	0.63	86.49	5.33	0.68	**83.13**	0.48	0.66
	B10	89.15	5. 03	0.43	94.15	6.23	0.66	87.51	4.21	0.47	**86.54**	1.34	0.65
	B12	182.86	2.22	0.33	186.66	5.63	0.12	174.11	0.67	0.18	**174.01**	0.1	0.09
	B18	219.55	2.22	0.16	244.21	8.13	0.21	**218.31**	0.62	0.51	218.42	0.67	0.48
Δ_2	***B02***	84.94	3. 04	0.52	85.64	4.15	0.64	85.47	4.75	0.57	**83.27**	1.03	0.51
	B10	90.58	5.52	0.56	94.71	6.73	0.62	87.55	4.10	0.53	**86.46**	1.33	0.42
	B12	189.96	25.86	0.37	188.85	5.37	0.23	**175.32**	0.35	0.34	175.67	2.12	0.35
	B18	219.67	1.92	0.27	243.02	7.79	0.05	218.32	0.71	0.52	**218.31**	0.66	0.51
Δ_3	***B02***	90.01	3. 44	0.38	86.89	3.94	0.45	87.05	3.56	0.54	**83.76**	1.83	0.61
	B10	91.43	5.72	0.39	93.07	5.24	0. 42	89.04	3.36	0.50	**87.15**	1.26	0.53
	B12	203.44	37.42	0.47	189.46	5.32	0.08	**174.21**	0. 84	0.41	177.67	0.94	-0.38
	B18	221.47	2.83	0.21	243.76	7.12	0.20	**218.19**	0.47	0.43	218.80	0.66	0.43

In Table 3, VNDMR has the best overall performance, finding the best average tree costs on 8 out of 12 instances. The TS-CST algorithm is the second best approach, obtaining the lowest tree costs for 4 out of 12 instances. Although not necessarily the case in all instances, the best solutions obtained by the two algorithms are in general associated with higher *fdc* values for problems with different delay bounds. In some cases (*B12*), landscapes of VNDMR and TS-CST are less correlated compared to LS or BSMA. This is probably due to that the landscape near the global optimum is more rugged with many local optima of the instance, as shown in Figure 2. For the simple LS, although it can obtain higher ρ on some instances, most of the solutions found are worse compared with the other algorithms mainly due to the greedy acceptance criterion used in the search. The fitness landscape of different local search methods is highly dependent on the neighborhood operators and search strategies.

The *fdc* analysis indicates that the binary solution representation is a simple yet effective method to encode the complex structure of the multicast tree. Based on this solution representation, neighborhood operators (the node or link based operators) and search strategies (the tabu list) designed for DCLC-MRPs show to effectively guide search algorithms towards better solutions in the landscape. It can also be seen that VNDMR is also highly stable on obtaining good solutions for all instances, obtaining the lowest standard deviation of tree cost (1.04) compared with 2.56 for TS-CST, 5.76 for BSMA, and 9.75 for the simple LS. This indicates that, by employing multiple neighborhoods, flexible search is able to traverse in different regions of search space and effectively escape from local optima, and thus is superior to standard local search methods in solving the DCLC-MRP with a complex and disconnected search space.

4.4 Autocorrelation Analysis for the DCLC Multicast Routing Problem

The *fdc* gives a good indication of the global feature of the landscape. However, features of the local regions in the search space (such as those many local optima near x_{opt} in Figure 2) are not easily observed with the *fdc* analysis. Another important measure of the landscape is its ruggedness in local regions. We perform the autocorrelation analysis on the benchmark DCLC-MRP instances by conducting a random walk of W = 1000 steps on each instance. The random walk starts from a random multicast tree which is constructed by starting from the source node and randomly adding the next node until all destination nodes are connected.

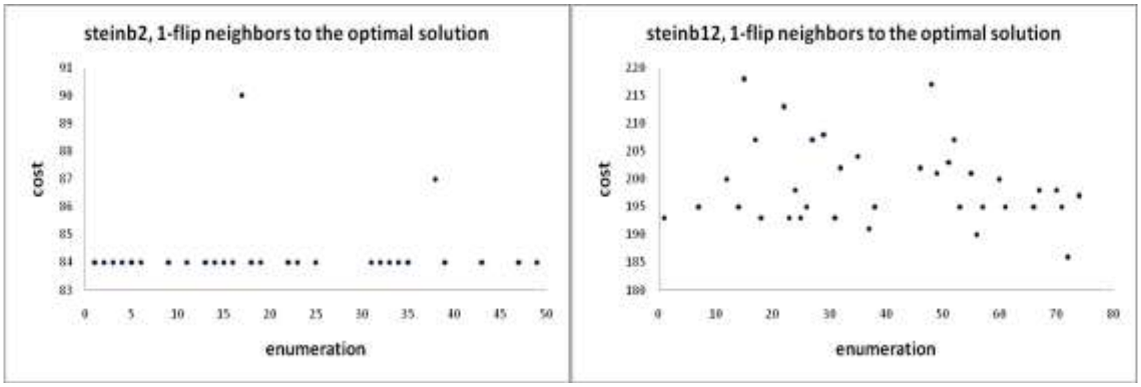
Based on the binary solution representation, the random walk is conducted by using a pure random one-flip neighborhood to randomly flip a bit corresponding to a Steiner node in the binary vector of the current solution. A random tree is then generated by randomly connecting a given node (with a value of 1 at the corresponding bit in the binary vector) until all destination nodes are connected. This procedure is repeated until a feasible tree that satisfies the delay constraint is generated.

The correlation of two neighboring points in the search space provides a good indication of the ruggedness of the local regions of the fitness landscape. It is known that the higher the correlation length l (or the closer the relative correlation length to the diameter of the landscape l / n to 1), the smoother the landscape and hence the easier the search of an algorithm based on the neighborhood employed. Table 4 presents the results of the autocorrelation analysis by using the pure random operator to generate random walks for the DCLC-MRP instances with the same three different delay bounds as used in the *fdc* analysis. We can see that although the correlation length is also problem instance dependent, the landscape of all the instances with different delay bounds are highly rugged. The delay bounds also significantly affect the ruggedness of the landscapes on problem instances, introducing higher as well as lower ruggedness to some instances.

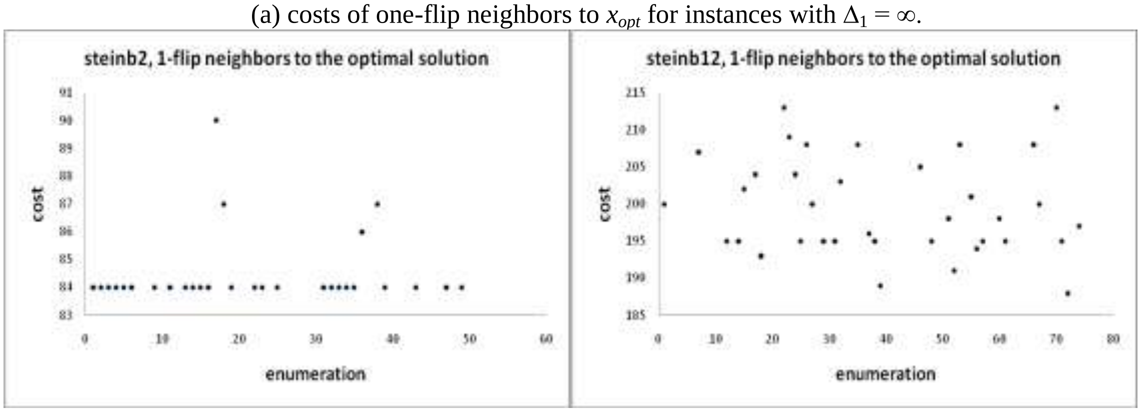
Table 4. The autocorrelation analysis on the DCLC-MRP instances with different delay bounds. l / n: the correlation length l to the diameter n of the landscape. The highest autocorrelation is in bold.

Problem instance	n	Δ_1		Δ_2		Δ_3	
		l	l / n	l	l / n	l	l / n
B01	50	0.61	0.012	0.61	0.012	3.58	**0.072**
B02	50	0.35	0.007	0.19	0.004	0.63	0.013
B03	50	0.26	**0.005**	0.21	0.004	0.18	0.004
B04	50	0.54	**0.011**	0.57	**0.011**	0.57	**0.011**
B05	50	0.26	0.005	0.43	0.009	0.88	**0.018**
B06	50	0.4	0.008	0.46	**0.009**	0.4	0.008
B07	75	0.22	0.003	0.22	0.003	0.32	**0.004**
B08	75	0.34	0.005	0.36	0.005	0.56	**0.007**
B09	75	0.31	**0.004**	0.21	0.003	0.33	**0.004**
B10	75	0.44	**0.006**	0.41	0.005	0.42	**0.006**
B11	75	0.37	**0.005**	0.28	0.004	0.35	**0.005**
B12	75	0.34	**0.005**	0.33	0.004	0.28	0.004
B13	100	0.41	0.004	1.46	**0.015**	0.33	0.003
B14	100	0.36	0.004	0.41	0.004	0.46	**0.005**
B15	100	0.27	0.003	0.4	0.004	0.45	**0.005**
B16	100	0.46	**0.005**	0.27	0.003	0.37	0.004
B17	100	0.41	**0.004**	0.29	0.003	0.31	0.003
B18	100	0.26	**0.003**	0.29	**0.003**	0.27	**0.003**

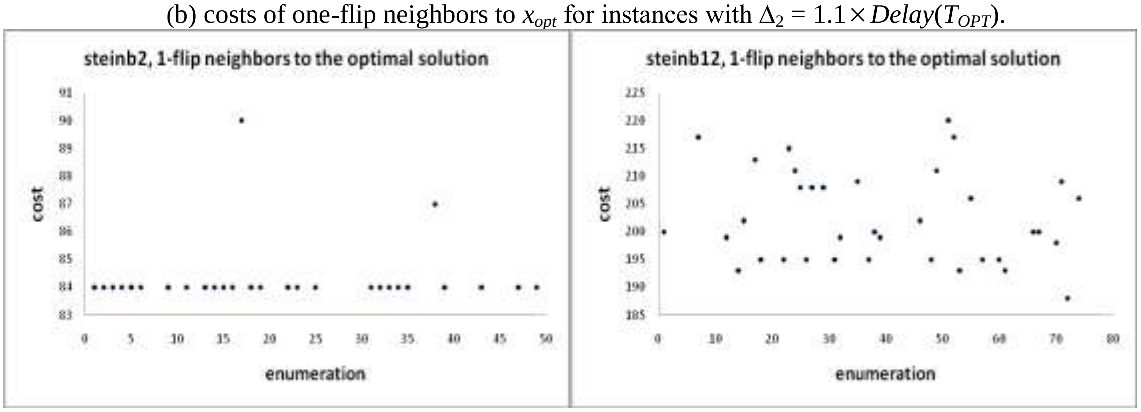
To visually observe the local regions of DCLC-MRP landscapes, we plot the neighbors of x_{opt} of the DCLC-MRP instances by using the same one-flip operator in the random walk. Figure 3 shows the example plots of the cost of these neighbors in the search space. Infeasible neighbors are not shown and an empty space appears at the corresponding positions. The plots again illustrate that the ruggedness of landscapes is highly dependent on the specific instances. It is obvious that the landscape of *B12* with different delay bounds is more rugged than that of *B02,* as small differences in the solution make a huge difference in the cost of the solutions for *B12*. The landscape of *B02* is smoother than *B12* as most of the one-flip neighbors have the same cost located in a plateau.



(a) costs of one-flip neighbors to x_{opt} for instances with $\Delta_1 = \infty$.



(b) costs of one-flip neighbors to x_{opt} for instances with $\Delta_2 = 1.1 \times Delay(T_{OPT})$.



(c) costs of one-flip neighbors to x_{opt} for instances with $\Delta_3 = 0.9 \times Delay(T_{OPT})$.

Figure 3. Scatter plots of costs of the one-flip neighbors to x_{opt}.

4.5 Autocorrelation Analysis of Different Neighborhoods

In addition to the pure random neighborhood used above, we carry out the autocorrelation analysis to analyze the ruggedness of the landscape from different neighborhoods in local search algorithms for the DCLC-MRP. Two more neighborhoods, namely the node-based neighborhood and the link-based neighborhood used in our previous VNDMR algorithm, are evaluated.

1) The Link-based Neighborhood

The link-based neighborhood operates on links in the multicast network. It replaces a randomly selected *superpath* in the current tree with a feasible alternative path (which satisfies the delay bound) generated by using the *k*-shortest path algorithm.

2) The Node-based Neighborhood

The node-based neighborhood operates on nodes in the multicast network. At each step, it randomly flips a bit of a Steiner node in the binary vector of the current tree and uses the Prim's algorithm to generate a minimum spanning tree on the given nodes. This operator repeats until a new feasible tree is generated.

For a fair comparison, the same maximum number of iterations (here set as 5) has been carried out in the local search with different neighborhoods for solving four instances of different sizes (*B02*, *B10*, *B12* and *B18*). Table 5 presents the results of the autocorrelation analysis on the four instances with different delay bounds.

In Table 5, the link-based neighborhood and node-based neighborhood lead to a smoother landscape, where longer correlation lengths have been obtained compared with the pure random neighborhood. This means that neighborhoods which are designed with regard to network structures are more effective than the pure random neighborhood for the DCLC-MRP. It is also interesting to see that, for larger problem instances (with 75 and 100 nodes), the landscape defined by the link-based neighborhood operator is smoother than that of node-based neighborhood operator. This demands that effective local search algorithms should be defined for solving DCLC-MRPs with different characteristics. Employing more than one neighborhood in the search algorithm provides one such solution. This is consistent with the better performance we observed in our previous work on the variable neighborhood descent search algorithm for the DCLC-MRP [13], and the simulated annealing based evolutionary algorithm employing multiple neighborhoods for multi-objective MRPs [47]. Both algorithms obtained better solutions compared with existing heuristics and algorithms employing single neighborhood operators.

Table 5. The autocorrelation analysis of three neighborhood operators for four instances with different delay bounds. *pure*: the pure random one-flip operator; *link*: the link based operator; *node*: the node based operator; *l* / *n*: the correlation length *l* to the diameter *n* of the landscape. The highest autocorrelation is in bold.

No.	*n*	Metrics	Δ_1			Δ_2			Δ_3		
			pure	*link*	*node*	*pure*	*link*	*node*	*pure*	*link*	*node*
B02	**50**	***l***	0.35	1.19	**2.02**	0.19	1.06	**1.65**	0.63	1.31	**3.7**
		l* / *n	0.007	0.024	**0.040**	0.004	0.021	**0.033**	0.013	0.026	**0.074**
B10	**75**	***l***	0.44	1.66	**2.53**	0.41	1.54	**2.16**	0.42	1.71	**1.57**
		l* / *n	0.006	0.022	**0.034**	0.005	0.021	**0.029**	0.006	0.023	**0.021**
B12	**75**	***l***	0.34	**1.98**	0.52	0.33	**2.04**	0.54	0.28	**1.88**	0.81
		l* / *n	0.005	**0.026**	0.007	0.004	**0.027**	0.007	0.004	**0.025**	0.011
B18	**100**	***l***	0.26	**2.91**	0.62	0.29	**2.73**	0.6	0.27	**2.74**	1.1
		l* / *n	0.003	**0.029**	0.006	0.003	**0.027**	0.006	0.003	**0.027**	0.011

5. An Iterative Local Search Approach for the DCLC-MRP

In this section, we investigate a new Iterative Local Search (ILS) approach for solving the DCLC-MRP based on the above fitness landscape analysis. We choose the ILS meta-heuristic for the DCLC-MRP in this work due to the following reasons.

- Firstly, the fitness-distance scatter plots of local optima on some example instances of the DCLC-MRP in Figure 2 show that there are many local optima in the search space and the distribution of these local optima is instance dependent and affected by the delay constraint. The scatter plots of one-flip neighbors to the optimal solutions in Figure 3 also show that the slight change in the solution presentation results into many local optima distributed in the search space.

- Secondly, the fitness-distance correlation analysis in Section 4 reveals that VNDMR is the most suitable local search method for the DCLC-MRP in comparison with the other three algorithms including LS, BSMA and TS-CST. The experimental results of the auto-correlation analysis demonstrate that both the node-based neighborhood and the link-based neighborhood are more effective than the pure random neighborhood. These fitness landscape analysis results are consistent with the better performance obtained by VNDMR for the DCLC-MRP in [13].
- Thirdly, ILS has some desirable features such as the simplicity, robustness and high effectiveness when applied to a wide range of optimisation problems [24-29]. The basic idea of ILS is to iteratively search better solutions by moving from one local optimal solution to another in the search space. Aiming at a better solution and escaping from the local optimum, a perturbation operator is applied to the current solution and a new starting point is then generated which may guide the search to a promising area in the search space. The basic idea of ILS lays on carrying out a series of random walk within the search space of the local optima with respect to the local search algorithm. The search procedure of ILS is thus very suitable for exploring the search space of the DCLC-MRP with many local optima. Our idea is thus to further extend VNDMR by applying it as the local search method in the new ILS algorithm. In order to intelligently guide the search to a promising area in the search space, the value of the maximum depth to escape from the current local optimum in the landscape is estimated in the perturbation procedure of the proposed ILS algorithm.

5.1 Iterative Local Search

A typical ILS meta-heuristic includes four components: Initial Solution Generation, Local Search Procedure, Perturbation, and Acceptance Criterion. As shown in Figure 4, the basic ILS works as follows: Step (1) constructs an initial solution s_0, Step (2) applies a local search method to produce a solution s^*. Perturbation in step (3) mutates the current local optimal solution s^* and generates an intermediate solution s'. Then LocalSearch is applied to s' and produces a new local optimal solution s'' in step (4). If s'' wins s^* in AcceptanceCriterion of step (5), then s'' will replace s^* and become the current solution; otherwise, the previous optimal solution s^* is maintained. Steps (3)-(5) proceed iteratively after a number of iterations until the termination condition is met, and finally the best solution found is output as the final solution.

```
(1) Generate Initial Solution s0;
(2) s* = LocalSearch (s0)
Repeat
    (3) s' = Perturbation (s*)
    (4) s" = LocalSearch (s')
    (5) s* = AcceptanceCriterion (s*, s")
Until (Termination condition is met)
```

Figure 4. The pseudo code of the basic ILS.

5.2 The Proposed ILS Algorithm for the DCLC-MRP

As stated in [28], the success of ILS is directly associated with the choice of the local search procedure, the perturbation procedure, and the acceptance criterion. We describe our proposed ILS algorithm in the following subsections.

5.2.1 Generate Initial Solution

The initial solution is randomly generated by using the same method of generating the starting point of the series of random walk in the auto-correlation analysis in Section 4.4. A random initial multicast tree is constructed by starting from the source node and randomly adding the next node until all destination nodes are mounted on the tree. The solution, i.e. a multicast tree, is also represented by the binary vector described in Section 4.1.

5.2.2 Perturbation Operator

To avoid the search travelling into local optima, a perturbation operator is applied to generate new starting point for the local search procedure. The pure random one-flip neighborhood operation of the auto-correlation analysis in Section 4.4 is applied as the perturbation in the proposed ILS algorithm. During the pure random one-flip perturbation, a bit corresponding to a Steiner node in the binary vector of the current solution is flipped. Then, a random multicast tree which satisfies the delay constraint is generated by spanning all the nodes with a value of 1 at the corresponding bit in the binary vector.

An estimation of the maximum depth, denoted by d, to escape from the current local optimum in the landscape is thus estimated as the hamming distance between the worst solution and best solution obtained at each iteration of the ILS algorithm. The perturbation operation repeats d times to escape from the current local optimum, potentially leading the search to a promising area of search space. If $d > 1$, the perturbation is called the guided-perturbation, if $d = 1$, the perturbation is a random process without guidance.

5.2.3 Local Search Procedure

Due to the effectiveness of VNDMR demonstrated in the landscape analysis of both the fitness distance analysis and auto-correlation analysis in Section 4, we apply VNDMR as the local search procedure in our proposed ILS algorithm. Different from the traditional single neighborhood based local search algorithm, VNDMR can effectively explore different areas of search space defined by variable neighborhood structures (the node-based or link-based neighborhood), thus is able to escape from local optima and find better solutions. Experimental results of the above auto-correlation analysis in Table 5 show that different neighborhood operators within VNDMR can complement each other for solving different DCLC-MRP instances with different characteristics.

In order to compare the effect of different local search procedures to the performance of ILS for the DCLC-MRP, we also implement TS-CST as the local search procedure in the proposed ILS algorithm. The detailed descriptions of these two local search algorithms are described in Section 4.3.

5.2.4 Acceptance Criterion

The acceptance criterion decides which of the locally optimal solutions, $s^*$and s'' in Figure 4, obtained from previous steps is to be selected as the starting point for the next perturbation step. In the implementation of our proposed ILS algorithm, the better solution with respect to the fitness values f, i.e. the cost and the delay of the multicast tree, will be accepted as shown in Eq. (12).

$$\text{AcceptanceCriterion}(s^*, s'') = \begin{cases} s^* & f(s^*) < f(s'') \\ s'' & otherwise \end{cases} \qquad (12)$$

5.3 Experiments on the Proposed ILS Algorithm

In order to test the impact of the local search method and the effect the proposed new guided-perturbation operator as described in Section 5.2.2., four variants of the ILS algorithm have been implemented in our experiments as follows:

a) ILS-VND-g

The first variant of the ILS algorithm applies VNDMR as the local search method and uses the guided-perturbation operator, i.e. $d > 1$.

b) ILS-VND

The second variant of the ILS algorithm adopts VNDMR as the local search method, but the perturbation is not guided, i.e. $d = 1$.

c) ILS-TS-g

In the third variant of the algorithm, TS-CST is applied as the local search method and the guided-perturbation is used.

d) ILS-TS

The fourth variant applies TS-CST as the local search method and the perturbation operator without guidance.

5.3.1 Performance of variants of the ILS algorithm

In the first group of experiments, the performance of the four variants of ILS algorithms is compared on four problem instances (*B02*, *B10*, *B12* and *B18*) of different sizes and delay bounds (i.e., $\Delta_1 = \infty$, $\Delta_2 = 1.1 \times Delay(T_{OPT})$, and $\Delta_3 = 0.9 \times Delay(T_{OPT})$, see Section 4.1).

Table 6. Experiment results on four instances of minimum Steiner tree problem with different sizes and delay bounds. (**avg.**/**best**/***σ***: the average, best tree cost and the standard deviation. The best solutions are in bold.)

Delay bounds	**Problem instances**	**ILS-VND-g**			**ILS-VND**			**ILS-TS-g**			**ILS-TS**		
		avg.	**best**	***σ***	**avg.**	**best**	***σ***	**avg.**	**best**	***σ***	**avg.**	**best**	***σ***
Δ_1	***B02***	**83**	83	0	**83**	83	0	90.9	90	0.32	94.3	90	7.6
	B10	**86**	86	0	**87**	86	2.11	95.6	91	2.37	114.2	101	7.05
	B12	**174**	174	0	174	174	0	220.6	215	3.86	236.5	227	6.28
	B18	**218**	218	0	218.3	218	0.67	264.3	261	2.63	282.8	274	11.75
Δ_2	***B02***	**83**	83	0	**83**	83	0	91	91	0	95.3	91	9.07
	B10	**86**	86	0	86.5	86	1.58	94.5	91	2.22	107	102	4.27
	B12	**174**	174	0	174.8	174	1.69	223	220	2.31	236	227	7.06
	B18	218.3	218	0.48	**218.1**	218	0.32	265.5	259	4	282.2	271	9.2
Δ_3	***B02***	**85.2**	84	1.03	**86**	86	0	91	91	0	91	91	0
	B10	**88**	88	0	88.3	88	0.95	96	91	5.64	115.2	107	5.37
	B12	**177.2**	177	0	177.4	177	0.84	223.2	210	6.63	238.4	233	4.81
	B18	**219**	219	0	219.1	219	0.32	264.7	258	3.77	281.5	261	12.46

For a fair comparison, the same amount of time (60 seconds) is set for each variant of ILS. Table 6 presents the average tree cost, the best tree cost and the standard deviation obtained from 30 runs on each instance. From the table, we can see that ILS-VND-g performs the best among the four variants of algorithms, finding 11 best results out of the 12 tests. On the one hand, both ILS-VND-g and ILS-VND perform much better than the other two variants with TS-CST as the local search method. This is due to the fact that VNDMR has better performance than TS-CST as shown in Table 3 of the *fdc* landscape analysis on the same four instances. It demonstrates that the local search method with better performance improves the search results of the proposed ILS. On the other hand, the proposed ILS algorithms with the guided-perturbation (ILS-VND-g and ILS-TS-g) obtain better search results than those of the other variants of algorithms without the guided-perturbation (ILS-VND, ILS-TS), respectively. It means that the new guided-perturbation can improve the performance of the proposed ILS algorithm by estimating the distance of escaping from the current local optimum.

5.3.2 Comparisons on Steinb Instances with Different Delay Bounds

In this group of experiments, we compare ILS-VND-g with three other existing algorithms with good performance in the literature, including SSPR-VND [14], GRASP-VND[12], and GRASP-CST[11], on the 18 benchmark instances (Steinb) from the OR-library with three different delay bounds ($\Delta_1 = \infty$, $\Delta_2 = 1.1 \times Delay(T_{OPT})$, and $\Delta_3 = 0.9 \times Delay(T_{OPT})$). SSPR-VND is a hybrid scatter search with path relinking algorithm for the DCLC-MRP, where VNDMR is applied as the local search method in it. Two GRASP algorithms, GRASP-VND and GRASP-CST, have been proposed for the DCLC-MRP. The difference between them is that the local search phase in GRASP-VND is VNDMR, while GRASP-CST applies the modified tabu search heuristic [44] as the local search phase. Both GRASP algorithms have shown to be effective approaches for solving the DCLC-MRP. Experimental results in [14] demonstrate that SSPR-VND obtains the best performance so far in comparison with other algorithms in the literature. In order to test the performance of our proposed ILS-VND-g, we compare these existing four algorithms by setting the same computing time (60 seconds) in each run and running each algorithm 30 times on each instance.

Table 7. Experimental results for the Steinb instances with the delay bound $\Delta_1 = \infty$. (**avg.**/**best**/$\boldsymbol{\sigma}$: the average, the best and the standard deviation of the tree cost. The values marked with '*' denote the optimal solutions and the best results are in bold.)

No.	ILS-VND-g			SS-VND			GRASP-VND			GRASP-CST		
	Avg.	**Best**	σ	**Avg.**	**Best**	σ	**Avg.**	**Best**	σ	**Avg.**	**Best**	σ
B01	**82***	82	0	**82***	82	0	**82***	82	0	**82***	82	0
B02	**83***	83	0	**83***	83	0	**83***	83	0	**83***	83	0
B03	**138***	138	0	**138***	138	0	**138***	138	0	**138***	138	0
B04	**59***	59	0	**59***	59	0	**59***	59	0	**59***	59	0
B05	**61***	61	0	**61***	61	0	**61***	61	0	**61***	61	0
B06	**122***	122	0	**122***	122	0	**122***	122	0	**122***	122	0
B07	**111***	111	0	**111***	111	0	**111***	111	0	**111***	111	0
B08	**104***	104	0	**104***	104	0	**104***	104	0	**104***	104	0
B09	**220***	220	0	**220***	220	0	**220***	220	0	**220***	220	0
B10	**86***	86	0	**86***	86	0	**86***	86	0	**86***	86	0
B11	**88***	88	0	**88***	88	0	**88***	88	0	**88***	88	0
B12	**174***	174	0	**174***	174	0	**174***	174	0	**174***	174	0
B13	**165***	165	0	168.1	165	1.92	167.3	165	2.39	165.4	165	1.09
B14	**235***	235	0	235.3	235	0.47	235.1	235	0.22	**235***	235	0
B15	319.6	318	1.17	**318***	318	0	319.5	318	0.89	319.8	318	0
B16	128.5	127	1.58	**127***	127	0	**127***	127	0	**127***	127	0
B17	**131***	131	0	**131***	131	0	131.2	131	0.67	**131***	131	0
B18	**218***	218	0	**218***	218	0	218.2	218	0.41	**218***	218	0

Firstly, we compare the average tree cost, the best tree cost and the standard deviation obtained by the four algorithms in Table 7. For this set of instances with the looser delay constraint, ILS-VND-g, SSPR-VND and GRASP-CST have the similar performance, obtaining best solutions for 15, 16 and 15 out of 18 instances in terms of the average tree cost, respectively, while GRASP-VND only finds 13 best solutions. ILS-VND-g performs better than SSPR-VND on two instances (B13 and B14), which shows that the single-population based ILS-VND-g has competitive performance compared with as that of population-based SSPR-VND meta-heuristic.

In the second set of experiments, for each instance, the delay bound is set to a slightly tighter value. Computational results in Table 8 show that ILS-VND-g has the similar overall performance as the other three algorithms, since all algorithms obtain the best solutions for 15 out of the 18 instances with respect to the average tree cost. The better results found by ILS-VND-g on two instances (B13 and B14) than those of SSPR-VND demonstrate again that the proposed ILS approach is competitive compared with the hybrid scatter search algorithm. In addition, the results obtained by ILS-VND-g is more stable than those of SSPR-VND, since ILS-VND-g has a smaller average standard deviation (0.254) over the 18 instances compared with that of SSPR-VND (0.469).

Table 8. Experimental results for the Steinb instances with the delay bound Δ_2 =1.1×$Delay(T_{OPT})$. (**avg.**/**best**/$\boldsymbol{\sigma}$: the average, best tree cost and the standard deviation. The values marked with '*' denote the optimal solutions and the best results are in bold.)

No.	Δ	ILS-VND-g			SS-VND			GRASP-VND			GRASP-CST		
		Avg.	**Best**	σ	**Avg.**	**Best**	σ	**Avg.**	**Best**	σ	**Avg.**	**Best**	σ
B01	145	**82***	82	0	**82***	82	0	**82***	82	0	**82***	82	0
B02	228	**83***	83	0	**83***	83	0	**83***	83	0	**83***	83	0
B03	248	**138***	138	0	**138***	138	0	**138***	138	0	**138***	138	0
B04	173	**59***	59	0	**59***	59	0	**59***	59	0	**59***	59	0
B05	125	**61***	61	0	**61***	61	0	**61***	61	0	**61***	61	0
B06	281	**122***	122	0	**122***	122	0	**122***	122	0	**122***	122	0
B07	212	**111***	111	0	**111***	111	0	**111***	111	0	**111***	111	0
B08	209	**104***	104	0	**104***	104	0	**104***	104	0	**104***	104	0
B09	280	**220***	220	0	**220***	220	0	**220***	220	0	**220***	220	0
B10	262	**86***	86	0	**86***	86	0	**86***	86	0	**86***	86	0
B11	235	**88***	88	0	**88***	88	0	**88***	88	0	**88***	88	0
B12	225	**174***	174	0	**174***	174	0	**174***	174	0	**174***	174	0
B13	190	**165***	165	0	168.1	165	1.67	167.6	165	2.39	**165***	165	0
B14	221	**235***	235	0	236.6	235	4.61	**235***	235	0	**235***	235	0
B15	308	319	318	1.33	**318.6**	318	0.92	319.6	318	0.82	319.6	318	0.82
B16	291	127.8	127	1.03	**127***	127	0	**127***	127	0	**127***	127	0
B17	219	131.7	131	2.21	131.7	131	1.24	131.9	131	0.88	**131.5**	131	0.51
B18	425	**218***	218	0	**218***	218	0	**218***	218	0	218.2	218	0.37

In the third set of experiments, we set the delay bound to a smaller value 0.9×$Delay(T_{OPT})$. The optimal solutions are thus not known to any of the cases. Due to the tighter delay constraint, the problem becomes much harder to solve. For some instances, some algorithms cannot even obtain feasible solutions, as presented in Table 9. The table shows that ILS-VND-g, GRASP-VND and GRASP-CST have the similar performance by finding 10, 10 and 9 best solutions out of the 18 instances with respect to the average tree costs. SSPR-VND performs the best among the four algorithms by finding 12 best results. It means that the population-based SSPR-VND approach is more flexible when exploring the search spaces of the Steinb instances with this tighter delay bound. Although ILS-VND-g is local search based, it still shows good performance by obtaining the same results as those of SSPR-VND on 7 instances and even performs better than SSPR-VND on two instances (B10 and B17) upon this set of DCLC-MRP.

6. Conclusions

In this paper, we analyze the fitness landscape of the delay-constrained least cost multicast routing problem (DCLC-MRP). The problem can be defined as the Delay-Constrained Steiner tree problem and has been proved to be NP-hard. Due to the lack of theoretical study on the underlying features of the DCLC-MRP landscape, we conduct the first fitness landscape analysis on the problem by applying two widely used landscape analysis measures: the fitness distance correlation and the autocorrelation analysis. A large amount of simulations have been carried out on a set of DCLC-MRPs which have been generated by adding different delay bounds to the benchmark Steiner tree instances in the OR-library.

The landscape analysis reveals that the landscape of the DCLC-MRP is instance dependent. The delay constraint of the DCLC-MRP has a great influence on the underlying landscape characteristics of the problem. Based on these observations, we may conclude that DCLC-MRP problem is a highly complex problem, thus tailor designed algorithm may not work well on a different variant of the problem, or even a different problem instance. Adaptive and robust search methodologies are needed to obtain reliable and good results across problems/instances in DCLC-MRP.

Table 9. Experiment results for the DCLC multicast routing algorithm with $\Delta_2 = 0.9 \times Delay(T_{OPT})$. (**avg.**/**best**/***σ***: the average, best tree cost and the standard deviation. The values marked with '*' denote the optimal solutions and the best results are in bold.)

No.	Δ	ILS-VND-g			SS-VND			GRASP-VND			GRASP-CST		
		Mean	**Best**	*σ*	**Mean**	**Best**	*σ*	**Mean**	**Best**	*σ*	**Mean**	**Best**	*σ*
B01	118	**83**	83	0	**83**	83	0	**83**	83	0	**83**	83	0
B02	187	85.2	84	1.03	**84**	84	0	**84**	84	0	**84**	84	0
B03	203	141	141	0	**140.8**	139	0.75	/	/	/	/	/	/
B04	142	**62**	62	0	**62**	62	0	**62**	62	0	**62**	62	0
B05	102	62.2	62	0.92	**62**	62	0	**62**	62	0	**62**	62	0
B06	199	125	125	0	125	125	0	**124.6**	124	0.93	125	125	0
B07	173	/	/	/	**112**	112	0	/	/	/	/	/	/
B08	171	**107**	107	0	**107**	107	0	**107**	107	0	**107**	107	0
B09	229	**221**	221	0	**221**	221	0	**221**	221	0	**221**	221	0
B10	215	**87.6**	87	0.84	87.9	87	0.11	88	88	0	88	88	0
B11	180	**89**	89	0	**89**	89	0	**89**	89	0	**89**	89	0
B12	184	**177**	177	0	**177**	177	0	**177**	177	0.89	178.5	177	4.24
B13	139	169.7	169	3.47	**169**	169	0	169.3	168	0.83	172	168	2.44
B14	180	**237**	237	0	/	/	/	**237**	237	0	238.3	236	2.37
B15	194	345	337	4.47	332.1	328	5.22	323.7	322	1.55	**321.3**	319	1.31
B16	238	132.8	132	1.03	131.4	129	1.08	130.7	129	1.53	**129.3**	129	0.92
B17	180	**133.5**	133	0.71	134	134	0	134.5	134	0.61	134.3	134	0.44
B18	348	**219**	219	0	**219**	219	0	219.1	219	0.22	219.2	219	0.37

In addition, we have investigated the impact of different local search heuristics on the fitness distance landscape and compared the effectiveness of different neighborhoods in the autocorrelation analysis for solving the problem. Both the node-based neighborhood and the link-based neighborhood have shown to be effective and suitable for solving the DCLC-MRP. The fitness landscape analysis techniques discussed in this paper have shown to be useful for analyzing the underlying properties of the DCLC-MRP and predicting the effectiveness of heuristic algorithms and neighborhood operators.

Based on the analysis of the fitness landscapes of the DCLC-MRP, an Iterative Local Search (ILS) approach has been investigated in this paper for the first time for DCLC-MRP. Due to the existence of many local optima, in order to intelligently explore the search space of the DCLC-MRP, a new guided-perturbation is proposed to guide the search to potentially better solutions. Experiments on the Steiner tree instances demonstrate that the guided-perturbation contributes to a better performance of the proposed ILS algorithm. In addition, the local search method also affects the performance of the proposed ILS algorithm. A large amount of experiments on the Steiner tree instances reveals that this simple yet effective iterative local search approach has a competitive performance in comparison with other existing best algorithms in the literature for solving the DCLC-MRP.

Our work in this paper can be seen as a case study of theoretical analysis on the landscapes, aiming at a better understanding of features of the landscapes of the DCLC-MRP before designing an effective search algorithm for the problem. The work can be further extended in different ways. For example, the fitness landscape analysis methods can be used to analyze the DCLC-MRP of larger size or various multicast routing problems with additional real life constraints, e.g., the bandwidth and delay-variation, to motivate the development of advanced meta-heuristic algorithms for various MRPs of different features.

Other more advanced landscape analysis techniques in the literature can be applied to the DCLC-MRP as well as other MRPs. In addition, the performance of the proposed ILS algorithm can be further improved by designing different perturbation operations or being hybridized with other meta-heuristics.

Acknowledgement

This research has been supported by Hunan University, China, and the School of Computer Science at The University of Nottingham, UK.

References

1. Cheng X., Du D.Z. Steiner Trees in Industry. Kluwer Academic Publishers, Dordrecht, Netherlands (2001)
2. Garey M.R., Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, NewYork (1979)
3. Guo L., Matta I. QDMR: An efficient QoS dependent multicast routing algorithm. In: Proceedings of the 5th IEEE RealTime Technology and Applications Symposium, pp. 213-222 (1999)
4. Kompella V.P., Pasquale J.C., Polyzos G.C., Multicast routing for multimedia communication, IEEE/ACM Transactions on Networking, 1: 286-292 (1993)
5. Wang X.L., Jiang Z. QoS multicast routing based on simulated annealing algorithm. In: Proceedings international and applications, pp. 511-516 (2004)
6. Zhang K., Wang H., Liu F.Y. Distributed multicast routing for delay variation-bounded Steiner tree using simulated annealing. Computer Communications, 28: 1356-1370 (2005)
7. Ghaboosi N., Haghighat A.T. A tabu search based algorithm for multicast routing with QoS constraints. In: 9th International Conference on Information Technology, pp. 18-21 (2006)
8. Yang C.B., Wen U.P. Applying tabu search to backup path planning for multicast networks. Computers & Operations Research, 32: 2875-2889 (2005)
9. Hamdan M., El-Hawary M.E. Multicast routing with delay and delay variation constraints using genetic algorithm. Canadian Conference on Electrical and Computer Engineering, pp. 2363-2366 (2004)
10. Zahrani M.S., Loomes M.J., Malcolm J.A., Dayem Ullah A.Z.M., Steinhofel K., Albrecht A.A. Genetic local search for multicast routing with pre-processing by logarithmic simulated annealing. Computer and Operations Research, 35: 2049-2070 (2008)
11. Skorin-Kapov N., Kos M. A GRASP heuristic for the delay-constrained MRP. Telecommunication Systems, 32(1): 55-69 (2006)
12. Xu Y., Qu R. A GRASP approach for the delay-constrained Multicast routing problem. In: Proceedings of the 4th Multidisplinary International Scheduling Conference. pp. 93-104 (2009)
13. Qu R., Xu Y., Kendall G. A variable neighborhood descent search algorithm for delay-constrained least-cost multicast routing. Lecture Notes in Computer Science volume 5851, pp. 15-29. (2009)
14. Xu Y., Qu R. A hybrid scatter search meta-heuristic for delay-constrained multicast routing problems. Applied intelligence, doi: 10.1007/s10489-010-0256-x (2010)
15. Diot C., Dabbous W., Crowcroft J. Multipoint communication: a survey of protocols, functions, and mechanisms. IEEE Journal on Selected Areas in Communications, 15: 277-290. (1997)
16. Masip-Bruin X., Yannuzzi M., Domingo-Pascual J., Fonte A., Curado M., Monteiro E., Kuipers F., Van Mieghem P., Avallone S., Ventre G., Aranda-Gutierrez P., Hollick M., Steinmetz R., Iannone L., Salamatian K. Research challenges in QoS routing. Computer Communications, 29: 563-581. (2006)
17. Oliveira C.A.S., Pardalos P.M. A survey of combinatorial optimization problems in multicast routing. Computers & Operations Research, 32(8): 1953-1981. (2005)
18. Salama H.F., Reeves D.S., Viniotis Y. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. IEEE Journal on Selected Areas in Communications, 15: 332-345. (1997)

19. Yeo C.K., Lee B.S., Er M.H. A survey of application level multicast techniques. Computer Communications, 27: 1547-1568. (2004)
20. Wright S. The roles of the mutation, inbreeding, crossbreeding, and selection in evolution. In Proceedings of the Sixth Congress on Genetics, 1: 365. (1932)
21. Zahrani M.S., Loomes M.J., Malcolm J.A., Albrecht A.A. Landscape analysis for multicast routing. Computer Communications, 30: 101-116. (2006)
22. Zahrani M.S., Loomes M.L., Malcolm J.A. LSA-based landscape analysis for multicast routing. In: Proceedings of AI-2006, the 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence. pp.187-200. Springer-Verlag New York. (2006)
23. Katayama K., Narihisa H. Iterated local search approach using genetic transformation to the traveling salesman problem. In: Proceedings of GECCO'99.1: 321-328. (1999)
24. Stützle T. Iterated local search for the quadratic assignment problem. European Journal of Operational Research, 174: 1519-1539. (2006)
25. Ruiz R., Stützle T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research, 177: 2033-2049. (2007)
26. Wang C., Li X., Wang Q. Iterative Local Search Algorithm for No-Wait Flowshop Scheduling Problems to Minimize Makespan. In: Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design. pp. 908-912. (2008)
27. Tounsi M., Ouis S. An Iterative local-search framework for solving constraint satisfaction problem. Applied Soft Computing 8: 1530-1535. (2008)
28. Brito J., Ochi L., Montenegro F., Maculan N. An iterative local search applied to the optimal stratification problem. International Transactions in Operational Research. 17: 753-764. (2010)
29. Rajalakshmi K., Kumar P., Bindu H.M. Hybridizing Iterative Local Search Algorithm for Assigning Cells to Switch in Cellular Mobile Network. International Journal of Soft Computing. 5(1): 7-12. (2010)
30. Beasley J.E. OR-Library: distributing test problems by electronic mail. Journal of the Operational Research Society, 41(11): 1069-1072. (1990)
31. Jones T., Forrest S. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Proceedings of the 6th International Conference on Genetic Algorithms, pp. 184-192. Morgan Kaufmann. (1995)
32. Boese K. Cost versus distance in the traveling salesman problem, Tech. Rep. TR-950018, UCLA CS Department. (1995)
33. Reeves C. R., Landscapes, operators and heuristic search. Annals of Operations Research, 86(0): 473-490 (1998)
34. Stadler P.F. Correlation in landscapes of combinatorial optimization problems. Europhysics Letter, 20: 479-482. (1992)
35. Merz P., Freisleben B. Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In Proceedings of the 5th International Conference on Parallel Problem Solving from Nature - PPSN V, pp. 765-74, Springer. (1998)
36. Ochoa G., Qu R., Burke E.K. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems, The Genetic and Evolutionary Computation Conference (GECCO'09), pp. 341-348. (2009)
37. Kallel L., Naudts B., Reeves C.R. Properties of fitness functions and search landscapes. In: Theoretical Aspects of Evolutionary Computing, pp. 175-206. (2001)
38. Weinberger E.D. Correlated and uncorrelated fitness landscapes and how to tell the difference, Biological Cybernetics, 63: 325-336. (1990)
39. Stadler P.F. Landscape and their correlation functions. Journal of Mathematical Chemistry, 20: 1-45. (1996)
40. Hordijk W. A measure of landscapes. Evolutionary Computation, 4(4): 335-360. (1997)
41. Manderick B., de Weger M., Spiessens P. The genetic algorithm and the structure of the fitness landscape. Proceedings of the 4th International Conference on Genetic Algorithms. Morgan Kaufmann, pp. 143-150. (1991)
42. The OR Library. http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html
43. Waxman B.M. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 6: 1617-1622. (1988)

44. Skorin-Kapov N., Kos M. The application of Steiner trees to delay constrained multicast routing: a tabu search approach. In: Proceedings of the 7th international Conference on Telecommunications, pp. 443-448. (2003)
45. Zhu Q., Parsa M., Garcia-Luna-Aceves J. J. A source-based algorithm for delay-constrained minimum-cost multicasting. In: Proceedings of the 14th Annual Joint Conference of the IEEE Computer and Communication (INFOCOM'95), pp. 377-385. IEEE Computer Society Press, Boston, Massachusetts. (1995)
46. Betsekas D., Gallager R. Data networks (2nd edition). Englewood Cliffs, HJ: Prentice-Hall. (1992)
47. Xu Y., Qu R. Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighborhoods. Journal of Operational Research Society, 62: 313-325 (2010).
48. Stadler P.F. Towards a theory of landscapes. In: Complex Systems and Binary Networks, Lecture Notes in Physics volume 461, pp, 77-163, Berlin, New York. Springer Verlag. (1995)