Fuzzy Greedy Search and Job-Shop Problem

Kaveh Sheibani

Faculty of Engineering & Computing, Coventry University,
Priory Street, Coventry CV1 5FB, UK
k.sheibani@coventry.ac.uk

Abstract

This paper describes a new metaheuristic for combinatorial optimisation problems with specific reference to the job-shop scheduling problem (JSP). A fuzzy greedy search algorithm (FGSA) which is a combination of a genetic algorithm (GA) and a greedy randomised adaptive search procedure (GRASP) is considered to solve the problem. The effectiveness and efficiency of the proposed hybrid method will be investigated through the experimental results on standard benchmark problems.

Introduction

The development of a job-shop system has been one of the major topics of manufacturing as well as service industries research (Baker, 1974, Pinedo, 2002). In general form a job-shop scheduling problem (JSP) can be stated as follows. There are *n* jobs $J = \{1,2,\ldots,n\}$ that are available at time zero to be processed by *m* machines $M = \{1,2,\ldots,m\}$ on which each job visits each machine at most once in its own predetermined route. The usual objective is to minimise the completion time of the last job to leave the system, named the *makespan* ($C_{\max}$). A feasible schedule can be formed from a permutation of jobs in *J* on each of the machines in *M*. In this case, clearly there are $(n_1)!(n_2)!\cdots(n_m)!$ possible solutions, where n_i is the number of operations to be performed on machine i. The makespan for the JSP can be computed using a disjunctive graph (Pinedo, 2002). The JSP with the makespan minimisation criterion is denoted by $Jm \mid\mid C_{\max}$ in the literature. The two-machine problem $J2 \mid\mid C_{\max}$ is the only case which is known to be polynomial solvable through *Johnson's rule* (Johnson, 1954). The general case belongs to the class of combinatorial optimisation problems known to be NP-hard (Brucker, 2001, Pinedo, 2002). Hence, approximate methods are generally considered to be the only practical way to solve most real-life JSPs with reasonable sizes.

The fuzzy greedy evaluation concept in the field of combinatorial optimisation problems was introduced by Sheibani (2005) in his doctoral thesis at the London Metropolitan University. The initial applications of this idea in development of approximate methods for the travelling salesman problem (TSP) and the flow-shop scheduling problem (FSP) performed very well. For a particular case, his developed fuzzy greedy heuristic (FGH) for the FSP significantly improved the well-known NEH heuristic (Nawaz et al., 1983) which has dominated the field for many years. The basic idea of the fuzzy greedy search algorithm (FGSA), which is an extension of the fuzzy greedy evaluation concept, was proposed in (Sheibani, 2005). The methodology can be viewed as a combination of a genetic algorithm (GA) (Holland, 1975) and greedy randomised adaptive search procedure (GRASP) (Feo and Resende, 1995).

Fuzzy Greedy Evaluation

The fuzzy greedy evaluation concept in the field of combinatorial optimisation problems was introduced in (Sheibani, 2005). In greedy methods, we deal with the most significant question of what is the best choice at a particular stage in the algorithm. The greedy algorithm always makes the choice that looks best at the moment without any consideration of previous choices or future consequences. Hence, we will only know that the choices made are absolutely the best ones possible if and when the algorithm yields an optimal solution. Otherwise the 'best' choice may end up being the worst one! In such cases, the greedy algorithm may produce the unique worst possible solution; see for example the survey papers by (Gutin and Yeo, 2002, Gutin et al., 2002, Bang-Jensen et al., 2004).

The theory of fuzzy sets allows us to represent vague concepts that are expressed in natural language, such as the kind of choice made in the greedy methods.

Consider combinatorial optimisation problems, each of which is associated with a discrete solution space *X*, a feasible space *S* with property $S \subseteq X$ which is defined by the problem constraints, and an objective function $f : X \mapsto \Re$. In the case of minimisation, the aim is to find a feasible solution $\mathbf{x}^* \in S$ such that $f(\mathbf{x}^*) \le f(\mathbf{x})$, $\forall \mathbf{x} \in S$, where, $\mathbf{x} = (x_1,\ldots,x_n)$ is a vector of decision variables (solution). A cost function $c(x)$, $\forall x \in \{x_1,\ldots,x_n\}$ is defined for each specific problem. We will refer to function $c(x)$ as a *greedy evaluation function* which denotes a *degree of priority* for incorporating the corresponding element *x* into the solution under construction without causing infeasibility. We now describe an alternative approach. We will treat the set *X* as a fuzzy set with a well-defined membership function $\mu(x)$, the form of which is given by equation 1.

$$\mu(x) = \frac{1}{1 + \lambda^2 \rho \left(\left(\frac{1-\lambda}{\lambda} \right) x - \theta \right)^2} \qquad \textbf{(1)}$$

The variable $x \in X$ corresponds to one of the variables in the definition of our combinatorial optimisation problem; here we consider $x \in \Re$. The parameter θ is a basic measure for evaluating the priority to be assigned to variable x; here we require $\theta \in \Re$. The parameter λ is a tuning parameter that is chosen by experimentation such that $0 \le \lambda < 1$. This parameter will be seen to play an important role in the developed algorithm that we will consider later. The parameter $\rho > 0$ is effectively a shape parameter, so that as the value of ρ increases, the graph of $\mu(x)$ becomes narrower. The proposed evaluation function $\mu : X \mapsto (0,1]$ has the following properties: $\mu(\lambda\theta/(1-\lambda)) = 1$ and $\mu(x) < 1$ for all $x \neq \lambda\theta/(1-\lambda)$.

Equation 1 is a modification of the general formulas of the families of fuzzy membership functions described in (Klir and Yuan, 1995). We will refer to this function as a *fuzzy greedy evaluation function,* and it will replace the role of the greedy evaluation function in determining the degree of priority assigned to an element *x*.

Fuzzy Greedy Search Algorithm

The fuzzy greedy search algorithm (FGSA) is an extension of the fuzzy greedy evaluation concept, in the form of a metaheuristic (Sheibani, 2005). The proposed method is a population-based and iterative procedure. The process starts off with an initial population. This can be generated using the construction phase of the GRASP (Feo and Resende, 1995). The algorithm works on the set of individuals (called population) which is divided into two different sub-sets generated in different ways. One of the sets is generated through a recombination operator and also with a selection scheme. This is a standard evolutionary approach. The other set is built employing a construction procedure. It is similar to the GRASP construction phase except that the procedure adopts a generalised version of the fuzzy greedy evaluation function instead of the classical greedy evaluation function. The general structure of this construction procedure is given by a pseudo-code in figure 2.

CONSTRUCTION PROCEDURE

```
construction (seed,λ,θ)
    BEGIN
        Solution = Ø;
        μx(x,λ,θ) = membership_of_candidates (x);
    WHILE (solution is not complete) DO
        BEGIN
            UCE = update_candidates (elements , Solution);
            RCL = build_restricted_candidate_list (UCE);
            s = select_an_element_at_random (RCL);
            Solution = Solution ∪ {s};
            μx(x,λ,θ) = re-evaluate_ memberships (x);
        END
    RETURN Solution;
    END
```

Figure 2. A general procedure for the construction phase of the proposed metaheuristic.

The proposed metaheuristic uses an adaptation strategy in the sense that it attempts to adapt its knowledge from the best solution obtained in the previous iterations. This is achieved by the updating of parameter θ at each iteration. The algorithm is also based on the concept of machine learning systems. The hope is that it improves the quality of the solutions over many iterations with each iteration having better solutions than its predecessor. The structure of a generic algorithm of the proposed metaheuristic is given by a pseudo-code in figure 3.

FUZZY GREEDY SEARCH ALGORITHM

```
BEGIN
    t = 0;
    P(t) = Ø;
    WHILE (P(t) is not complete) DO
        P(t) = construction (seed,λ,θ);
    WHILE (not termination condition) DO
        BEGIN
            t = t + 1;
            P(t) = recombine (P(t - 1));
            evaluate (P(t));
            update (λ,θ);
            WHILE (P_C(t) is not complete) DO
                P_C(t) = construction (seed,λ,θ);
            P_R(t) = select (P(t));
            P(t) = P_C(t) ∪ P_R(t);
        END
END
```

Figure 3. Pseudo-code for the proposed metaheuristic. The variable $P(t)$ represents a set of population members at iteration t. The variables $P_C(t)$ and $P_R(t)$ represent constructed and reproduced sub-populations respectively at iteration t.

The proposed metaheuristic improves its relative performance on a given problem by the exploration of promising areas in the search space over time.

Fuzzy Greedy Construction

In the proposed metaheuristic we build a feasible solution, which is represented as a string of integers (successive jobs on each of the machines i.e. operations). The proposed method uses a list of candidate operations, in which the operation with the highest priority (i.e. maximum value of $\mu(x)$) is incorporated into the partial solution under construction without causing infeasibility. For the JSP, selection of the next operation for incorporating into the partial solution under construction is determined by the evaluation of all candidate operations according to a re-interpretation of the standard fuzzy greedy evaluation function $\mu(x)$ in equation 1 above. Here, x is a generic variable. We will define $x_{i,j}$ corresponding to job j on machine i. The parameter θ is a basic measure that is used for evaluating x. The parameter λ is a greedy tuning parameter that is chosen by experimentation to take values between $0 \le \lambda < 1$. For the work discussed in this paper, varying the parameter ρ has no significant effect on the metaheuristic. Its value will be set to $\rho = 1$. For the problem considered we will introduce x and θ as follows.

First we give indices to x with $x_{i,j}$ corresponding to job j on machine i. This takes the value of the processing time of job j on machine i ($p_{i,j}$). A sequence of the jobs on machine i can be represented as a permutation $\boldsymbol{\pi}_i = (\pi_i(1), \pi_i(2), \ldots, \pi_i(n))$, where we interpret $\pi_i(k)$ to be the job in position k corresponding to operation $x_{i,\pi_i(k)}$. We define τ_i in equation 2 as the sum of the processing times of jobs on machine i.

$$\tau_i = \sum_{j=1}^{n} p_{i,j} \qquad \textbf{(2)}$$

We represent θ in equation 3 as the mean of the sums of the processing times of jobs on the machines.

$$\theta = 1/n \sum_{i=1}^{m} \tau_i \qquad \textbf{(3)}$$

In the developed model, the predetermined route (i.e. a sequence of the machines) corresponding to job j can be represented as a permutation $\boldsymbol{\pi}_j = (\pi_j(1), \pi_j(2), \ldots, \pi_j(m))$, where we interpret $\pi_j(k)$ to be the machine in position k. We define operation of job j on its predetermined route $\boldsymbol{\pi}_j$ on machine $\pi_j(k)$ as $x_{\pi_j(k),j}$ in which operation $x_{\pi_j(k+1),j}$ can only start after the completion of $x_{\pi_j(k),j}$. Let $O = \{x_{i,j} \mid \forall i, j\}$ be the set of operations, and consider O_c to be a set of all candidate operations can be scheduled. We denote the value of the fuzzy greedy function for candidate operation $x_{i,j} \in O_c$ by $\mu(x_{i,j})$. The *fuzzy greedy* choice is to next schedule operation $x_{i,j} = \arg\max\{\mu(x_{i,j}) \mid x_{i,j} \in O_c\}$. It is also easy to observe that tuning λ in the fuzzy greedy evaluation function μ to a small enough value (e.g. zero), or setting the measure parameter θ to zero, the choice becomes *purely greedy* (i.e. the minimum value of $x_{i,j} \in O_c$). The selected operation is scheduled in the next available feasible time slot on the sequence $\boldsymbol{\pi}_i$ under construction.

Fuzzy Greedy Search Operator

The use of search techniques on a solution space are central to the design of metaheuristics. Indeed, adopting a robust search technique significantly improves the overall performance. We applied a modification of the fuzzy greedy search operator (FGSX) proposed in (Sheibani, 2005) for a special case of the JSP when each job has an identical route, known as the flow-shop scheduling problem (FSP). The mechanism of FGSX may be a bit more complex than classical crossover, but the primary idea is simple.

In this case, a sequence (solution) can be represented as a permutation $\boldsymbol{\pi} = (\pi(1), \pi(2), \ldots, \pi(n))$, where we interpret $\pi(k)$ to be the job in position k. We give index to x in equation 1 with x_j corresponding to job j and define x_j in equation 4 as the sum of processing times of job j on the m machines.

$$x_j = \sum_{i=1}^{m} p_{i,j} \qquad \textbf{(4)}$$

Here, the parameter θ is corresponding to an objective function e.g. the mean completion time for the current best schedule obtained so far. This parameter will be seen to play an adaptive role, in that good choices made at previous stages (giving rise to the best solution so far) will also influence future choices.

The operating principle of FGSX is shown schematically in figure 4. Let P_1 and P_2 be two randomly selected chromosomes from the previous generation. Each is a sequence of 9 arranged jobs, numbered 1 to 9, which represents the order of the jobs on the machines. First, we arbitrarily select a job, say 5, as the starting point in the offspring O_1. Then, we duplicate all jobs in the selected parent chromosome which have not been incorporated in the offspring O_1 – between two cut points marked by '|', as shown under the heading "Duplication" in figure 4. This guarantees that the next two possible candidate jobs have not already been incorporated in the offspring under construction (Qu and Sun, 1999). The duplication stage is optional and so we may utilise another way. The next job in the offspring is determined by the fuzzy greedy evaluation function using equation 1 – as shown under the heading "Selection" in figure 4. Assume that $\mu(x_6)$ is greater then $\mu(x_4)$, indicating that the choice of job 5 is more *suitable* than 4 at the moment, so we should select job 6 as the second job in the offspring O_1. Thus, the first two jobs in the offspring O_1 are 5 and 6 (the symbol 'x' means 'not yet determined'). The process is continued until a completely new offspring is formed.

Parents

P_1: (1 2 3 4 5 6 7 8 9)

P_2: (4 3 2 1 9 8 7 6 5)

Duplication

1 2 3 4 **5** | **6** 7 8 9 1 2 3 4 | 5 6 7 8 9

4 3 2 1 9 8 7 6 **5** | **4** 3 2 1 9 8 7 6 | 5

Selection

Assuming that $\mu(x_4) < \mu(x_6)$,

therefore, we select 6 for the offspring O_1

Offspring

O_1: (**5 6** x x x x x x x)

Duplication

6 | **7** 8 9 1 2 3 4 | 6 7 8 9 1 2 3 4

4 3 2 1 9 8 7 **6** | 4 3 2 1 9 8 7 | 6

Selection

Assuming that $\mu(x_7) < \mu(x_4)$,

therefore, we select 4 for the offspring O_1

Offspring

O_1: (**5 6 4** x x x x x x)

Figure 4. Proposed fuzzy greedy search operator (FGSX).

It is important to note that the FGSX can be adaptive in the sense that it attempts to learn from the best solution obtained in the previous generation. As mentioned above, this is achieved by the parameter θ in equation 1, which is updated at each generation.

Concluding Remarks

In this paper, the potential application of the proposed new metaheuristic FGSA for the job-shop scheduling problem has been investigated. We explained some of the most significant parts of the proposed method such as the construction and recombination phases. It should be noted that the fitness function calculates the value corresponding to the objective function (e.g. minimising the mean completion time), in that individuals with higher fitness values have a higher probability to be chosen as a member of the population of the next generation. For the selection scheme, individuals in the current iteration are chosen from the population and allowed to reproduce. This can be proportional to the individual fitness, as in the classical roulette wheel selection. An alternative method or a mixed strategy can be adopted as a selection scheme. We need some other parameters such as the size of the population as well as its division into the reproduced and the constructed sub-populations that should be determined experimentally. Indeed, a termination criterion is also necessary. We believe that the FGSA has some potential to deal with a wide range of optimisation problems.

Acknowledgments

This research is supported by the EPSRC (grant no. EP/C54952X/1). This manuscript is based on part of the author's PhD thesis. The author would like to thank, Mr JB Atkinson, for his guidance and supervision throughout the author's doctoral research programme at the STORM research centre of London Metropolitan University.

References

Baker, K. R. (1974): *Introduction to Sequencing and Scheduling*. John Wiley & Sons, New York

Bang-Jensen, J., Gutin, G., Yeo, A. (2004): When the Greedy Algorithm Fails. *Discrete Optimization*, Vol. 1. 121-127

Brucker, P. (2001): *Scheduling Algorithms*. 3rd edn. Springer-Verlag, Berlin, Heidelberg

Feo, T. A., Resende, M. G. C. (1995): Greedy Randomized Adaptive Search Proce-dures. *Journal of Global Optimization*, Vol. 6. 109-133

Gutin, G., Yeo, A. (2002): Anti-matroids. *Operations Research Letters*, Vol. 30. 97-99

Gutin, G., Yeo, A., Zverovich, A. (2002): Traveling Salesman Should Not Be Greedy: Domination Analysis of Greedy-type Heuristics for the TSP. *Discrete Applied Mathematics*, Vol. 117. 81-86

Holland, J. H. (1975): *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor, Michigan

Johnson, S. M. (1954): Optimal Two-and-Three-Stage Production Schedules with Set-up Times Included. *Naval Research Logistics*, Vol. 1. 61-68

Klir, G. J., Yuan, B. (1995): *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Englewood Cliffs, NJ

Nawaz, M., Enscore Jr, E. E., Ham, I. (1983): A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem. *Omega the International Journal of Management Science*, Vol. 11. 91-95

Pinedo, M. (2002): *Scheduling: Theory, Algorithms, and Systems*. 2nd edn. Prentice Hall, Upper Saddle River, NJ

Qu, L., Sun, R. (1999): A Synergetic Approach to Genetic Algorithms for Solving Travelling salesman Problem. *Information Sciences*, Vol. 117. 267-283

Sheibani, K. (2005): *Fuzzy Greedy Evaluation in Search, Optimisation, and Learning*. PhD thesis, London Metropolitan University, London, UK

Slowinski, R. (Ed.) (1998): Fuzzy Sets in Decision Analysis, Operations Research and Statistics, Series: *The Handbooks of Fuzzy Sets*. Springer-Verlag, Berlin