The Formulation and Optimization Algorithm for Mission Scheduling Problem of Vehicles

ZHENG YANXING  
BeiJing Institute of System Engineering  
China

DENG BO  
BeiJing Institute of System Engineering  
China

Abstract: The mission scheduling problem of Vehicles is formulated as a special multi-objective combinatorial optimization problem. The model takes the characteristics of problem into account, such as time window constraints and number of available vehicles, and is more suitable for applications compared with the previous formulations. Then a multi-objective evolutionary algorithm MSLUEA is proposed to solve the problem. Firstly, an effective heuristic construction algorithm is put forward to generate initial feasible solutions for the evolution process. The initial population comprises feasible solutions and random solutions to keep algorithm MSLUEA from sticking in a local optimum and taking a long time to converge as well. Then algorithm MSLUEA uses Pareto dominance based tournament selection with elitism mechanism to generate parent population which prevents bias to any objective. Problem specific evolutionary operators, including novel crossover operator and mutation operators are designed to ensure the feasibility of the children. Simulation results show that MSLUEA algorithm can solve the problem efficiently and has good convergence performance and multi-objective optimization effort.

Key–Words: mission scheduling, multi-objective optimization, Pareto optimality

1 Introduction

The mission scheduling problem of Vehicles can be described as that, several kinds of vehicles with different capabilities are asked to visit multiple sites. Mission plans should be quickly made for vehicles so as to meet the demands and deal with the real-world constraints such as time windows, number of available vehicles, and capabilities of the vehicles (i.e. maximum permitted travel time for vehicles etc.).

Some solutions have been proposed for the problem[1][2][3]. Although those previous studies are proven to be efficient to some extent, the mission demands are not considered thoroughly. In both [1][2][3] researches, the limited number of vehicles, which is an important constraint because only a limited number of vehicles are available in the real world, was ignored. Considering a limited number of vehicles, not all sites will be visited in some cases. Hence one of the objectives for the problem should be maximizing the number of visited sites with a limited number of vehicles. Thus the problem is more complicated to solve.

The main contributions of the paper are: (1) We present a mathematical model formulating mission scheduling problem with a limited number of vehicles as a multi-objective optimization problem MSPLU. The model is different from previous ones in that it takes more constraints into account which are usually common in real cases. (2) A multi-objective evolutionary algorithm MSLUEA is proposed to deal with MSPLU. MSLUEA algorithm introduces the concept of Pareto optimality and elitism mechanism, which prevents bias to any objective and drives the algorithm converging to the approximate Pareto set of the problem.

The rest of the paper is organized as follows. Section 2 presents the mathematical model of MSPLU. The proposed MSLUEA is described in detail in section 3, including the program flowchart, chromosome representation, initial population creation, Pareto dominance based selection, and evolutionary operators etc.. Section 4 shows the simulation results and section 5 concludes the paper.

2 Problem Formulation

The sites set is denoted by $T_0 = \{1, 2, ..., N_T\}$, where $N_T$ is the number of sites. $R_i$ is the demand of site $i$. Each site $i$ has a time window $[e_i, l_i]$, where $e_i$ is the earliest time allowed for visiting it and $l_i$ is the latest time allowed. $T_i$ is the time to begin visit site $i$. An vehicle may arrive before $e_i$, which incurs the waiting time $w_i$ until visit is possible. However, no vehicle may arrive past $l_i$. The set $T = \{0, 1, 2, ..., N_T\}$ denotes the extended sites set, where 0 indicates the base that vehicles depart from and return to. $V =$
\{V^1, V^2, ..., V^{NV}\} denotes vehicles set with different capabilities, where NV is the kind of vehicles, \(V^k\) denotes the \(k\)-th kind of vehicles set and \(v^k_q\) is the \(q\)-th vehicle in \(V^k\). The maximum travel time permitted for vehicles in \(V^k\) is \(TL_k\), and the mission requirement is \(r_k\). \(N^k_v\) is the number of vehicles available in \(V^k\). \(s^k_{ij}\) denotes the mission duration that an vehicle in \(V^k\) operation on site \(i\). The routes set between site pairs is denoted by \(A = \{(i, j) | i, j \in T, i \neq j\}\). Each route \((i, j) \in A\) is associated with a distance \(d_{ij}\) denoting the length of \((i, j)\) and a travel time \(t^k_{ij}\) denoting the travel time for vehicle in \(V^k\) between sites \(i\) and \(j\).

The decision variable of MSPLU is

\[
x^k_{pij} = \begin{cases} 
1 & \text{if route } (i, j) \text{ is used by vehicle } v^k_p \\
0 & \text{otherwise} 
\end{cases} 
\]  

(1)

The MSPLU model can be mathematically formulated as follows:

\[
\text{(MSPLU)} \quad \min \bar{f} = (f_1, f_2) 
\]  

(2)

\[
f_1 = N_T - \sum_{j \in T} \sum_{V^k \in V} \sum_{v^k_p \in V^k} x^k_{pij} 
\]  

(3)

\[
f_2 = \sum_{V^k \in V} \sum_{v^k_p \in V^k} \sum_{(i,j) \in A} f^k_{ij} x^k_{pij} 
\]  

(4)

where \(f^k_{ij} = \eta d_{ij} + \zeta (T_i + s^k_i + t^k_{ij} + w_j)\) is the cost that an vehicle in \(V^k\) visit site \(j\) following visiting on site \(i\), \(\eta\) and \(\zeta\) are weights.

Subject to:

\[
\sum_{j \in T_0} x^k_{pij} = 1, \sum_{j \in T_0} x^k_{pij} = 1, \forall v^k_p \in V^k, V^k \in V 
\]  

(5)

\[
\sum_{v^k_p \in V^k} \sum_{(i,j) \in A} x^k_{pij} \leq N^k_v, \forall V^k \in V 
\]  

(6)

\[
\sum_{V^k \in V} \sum_{v^k_p \in V^k} x^k_{pij} \leq 1, \forall j \in T_0 
\]  

(7)

\[
e_i \leq T_i \leq l_i \forall i \in T_0 
\]  

(8)

\[
I f \quad x^k_{pij} = 1, \ then \ T_i + s^k_i + t^k_{ij} + w_j = T_j 
\]  

(9)

\[
w_j = \max(0, e_j - (T_i + s^k_i + t^k_{ij})) 
\]  

(10)

\[
\sum_{(i,j) \in A} x^k_{pij} (t^k_{ij} + s^k_j + w_j) \leq TL_k \forall v^k_p \in V^k, V^k \in V 
\]  

(11)

\[
I f \quad x^k_{pij} = 1, \ then \ r_k \leq R_j 
\]  

(12)

MSPLU has two objectives to minimize. Objective \(f_1\) means to minimize the number of sites not visited. Objective \(f_2\) means to minimize the total vehicles travel distances and the consumed time for finishing the mission. \(f_1\) is called the number of unvisited sites(NUS) and \(f_2\) the mission cost(MC) in the rest of the paper. Eq.(5) specifies that each vehicle should begin at the base and return to the base finally. Eq.(6) constrains that the number of vehicles employed should not exceed the number of vehicles available. Eq.(7) ensures that each site is visited by at most one vehicle. The time feasibility constraints are defined in equations (8)-(11). Inequality (8) imposes the time window constraints. Waiting time is the amount of time that an vehicle has to wait if it arrives at a site location before the earliest time for that site. Inequality (11) states that the total travel time of an vehicle can not exceed the maximum permitted travel time of it. Inequality (12) restricts that the vehicle visiting on a site should satisfy the demand of the site.

The proposed MSPLU model has the advantage over the previous formulations of the problem, such as TSP in [1][2] and MILP in [3]. First, the mission scheduling problem of vehicles is formulated from the viewpoint of multi-objective optimization. While in the previous formulations of the problem, only one objective is considered. For example, Ryan[1] considered only the number of visited sites while Ousingsawat[3] tried to minimized only the total travel time. In fact, when we plan the missions, we not only want to visit sites as more as possible but also want to finish the mission at cost as low as possible. Therefore, we formulate the problem as a multi-objective optimization problem(MOP), which represents the problem more properly than the single objective optimization formulations. Second, the MSPLU model considers more mission characteristics, such as the time window constraints and the limited number of vehicles. The proposed MSPLU model is more suitable for real-world applications.

MSPLU can be viewed as a generalization of the TSP and therefore is NP-Hard. On the other hand, MSPLU is a typical MOP. The optimal solutions to MOP are non-dominated solutions known as Pareto optimal ones. For the given MSPLU problem, two objective vectors \(\vec{u} = (u_1, u_2)\) and \(\vec{v} = (v_1, v_2)\), \(\vec{u}\) is said to dominate \(\vec{v}\) if and only if \(\forall i \in \{1, 2\}: u_i \leq v_i \land \exists j \in \{1, 2\}: u_j < v_j\), \(\vec{u}\) is denoted as \(\vec{u} < \vec{v}\). A solution \(\vec{x}\) is said to dominate another solution \(\vec{y}\) iff \(f(\vec{x}) < f(\vec{y})\). \(\vec{x}_0\) is Pareto optimal if there is no other solution in the search space that dominates \(\vec{x}_0\). The set of all Pareto optimal solutions is called as Pareto optimal set.
3 Multiobjective Evolutionary Algorithm for MSPLU: MSLUEA

Multi-objective Evolutionary Algorithm (MOEA) has been increasingly investigated to solve NP-Hard MOP and has been proven to be a general, robust and powerful search mechanism [4][5][6]. This section provides a MOEA based algorithm MSLUEA to find the approximate Pareto optimal set of MSPLU.

3.1 The flowchart of MSLUEA

Fig.1 illustrates the flowchart of MSLUEA, which provides an overall view of the algorithm. Each component such as chromosome representation, initial population creation, evolutionary operators and elitism strategy, will be described in detail in the following sections. The initial population is composed of feasible solutions and random solutions. The vector objective of each individual is computed according to Eq.(2)-(4). An outside population archive is updated, which is introduced to store the non-dominated individuals generated in the evolution process. Then Pareto optimality based selection process is carried out to generate the parent individuals. Evolutionary operators including problem specific crossover and mutation operators will be applied to generate a new feasible population. Then the new population repeats the evolutionary process until the stopping criterion is met. Finally, the individuals in the archive are output as the solutions of the problem.

3.2 Chromosome representation

To facilitate chromosome representation in MSLUEA, we categorize the sites according to the mission demands of the sites and the capabilities of the vehicles. First, the vehicles set is sorted according to the requirements, i.e., for $V = \{V^1, V^2, ..., V^{Nv}\}, r_1 < r_2 < \ldots < r_{Nv}$ holds. All sites in $T_0$ are clustered according to their demands. For site $t \in T_0$, if $r_j < R_t < r_{j+1}$, then $t \in C_j$, which means that vehicles in $V^1, V^2, ..., V^j$ can satisfy the demands of site $t$, while vehicles in $V^{j+1}, ..., V^{Nv}$ can not.

In our approach, a chromosome is given by an integer string. A chromosome $S = \{s_1, s_2, ..., s_{Nv}, RP\}$ consists of $N_v$ sequences and a rejection pool (RP) which is a data structure to store the unvisited sites. Sequence $s_i$ is the visiting sites sequence for vehicles in $V^i$. Each sequence $s_i$ comprises several subsequences and is represented as $s_i = \{s_{i1}, ..., s_{in}\}$, where subsequence $s_{ip}$ is the sites sequence of vehicle $V^i$.

According to the site categorizes, all sites in $C_j$ could appear in one of the sequences in $\{s_i|i \leq j\}$ and could not be in any of the sequences in $\{s_i|i > j\}$. Generally, a solution of MSPLU contains some sites not visited because only a limited number of vehicles are available. Therefore, RP is introduced. Obviously, the size of RP is equal to the NUS defined in Eq.(3). The most benefit of this chromosome representation is that the solution that the chromosome encodes can satisfy the constraint(12).

3.3 Creation of initial population

In MSLUEA, we generate a mix population comprising feasible solutions and random solutions. The reason for constructing this mix population is that, a population of entirely feasible members gives up the opportunity of exploring the whole regions, and a completely random population may be largely infeasible taking a long time to converge.

To generate initial feasible solutions, an algorithm, Construction-Initial-Feasible-Solution (CIFS), is presented. The detailed steps of CIFS are shown in Fig.2.

$C_{Sk}$ in step(1) is a data structure used to store sites that will be visited by vehicles in $V^k$. A site $t$ in $C_k$ is added to $C_{Si}$ randomly chosen from $\{C_{S1}, ..., C_{Sk}\}$, which ensures that the vehicle which will visit $t$ can satisfy the mission demand of $t$. Step(6) describe how to insert all sites in $C_{Sk}$ to sequence $s_k$ which is initially null. The cost function for inserting a site $t$ as the first site to a new subsequence $s'_k$ is as follows:

$$Cost1(t) = -\alpha t_{0k} + \beta l_t \quad (13)$$
Figure 2: Construction-Initial-Feasible-Solution

The site to be selected as the first site in a new subsequence $s_k^r$ is the one $t^* = \arg \min_t C\text{ost}1(t)$. Eq.(13) implies that the desired site is the one that is far from the base and has an early time window. The weights were derived empirically and were set to $\alpha = 0.7$ and $\beta = 0.3$.

Once the first site $t^*$ is selected for the current subsequence $s_k^r$, the algorithm selects from the rest sites the site $t$ which minimizes the total insertion cost between every two adjacent sites $p$ and $q$ in $s_k^r$ without violating the time window and maximum permitted travel time constraints. The cost for inserting site $t$ between $p$ and $q$ in $s_k^r$ is:

$$Cost2(t, p, q) = \omega_1 D + \omega_2 W + \omega_3 O + \omega_4 T$$

(14)

Where

$$D = \sum_{(i,j) \in \mathcal{E}_k^r} d_{ij}$$

is the total distance travelled by vehicle $v_k^r$, $W = \sum_{(i,j) \in \mathcal{E}_k^r} \left(T_i + s_k^r + t^r_{ij} + w_j \right)$ is the total consumed time of vehicle $v_k^r$, $O = \sum_{(i,j) \in \mathcal{E}_k^r} \max\{0, (T_i + s_k^r + t^r_{ij} - l_j)\}$ is the penalty for tardiness.

$T = \max\{0, W - TL_k\}$ is the penalty for exceeding the maximum travel time.

$\tilde{s}_k^r$ is the subsequence after $r$ is inserted between $p$ and $q$. The cost function includes the components weight by $\omega_1$ for distance, $\omega_2$ for consumed time, and penalty weighting factors $\omega_3$ for tardiness, $\omega_4$ for travel time in excess of the maximum time for vehicles. The weights are set as $\omega_1 = 0.1, \omega_2 = 0.1, \omega_3 = 10$, and $\omega_4 = 10$, which are biased towards finding a feasible solution in comparison to reducing the total distance and route time.

The site $t^*$ with the least cost is inserted to the position between $(p^r,q^r)$ in $s_k^r$ which produces subsequence $s_k^r$. Then the feasibility of $s_k^r$ is checked. If $s_k^r$ is feasible, it will replace $s_k^r$ and next site is found to insert the best position until no more site can be inserted to current subsequence. Otherwise, a new subsequence is created and above steps are repeated until all sites in $C_{sk}$ are inserted to sequence $s_k$.

Until all the sites are inserted, we get a solution which provides an upper bound of the number of vehicles employed to visit all the sites. In the worst case, the number is equal to the number of sites. Considering the limited number of available vehicles, in each sequence $s_k$, only the $N_{V_k}^k$ subsequences with the largest numbers of sites are kept. All other subsequences are removed from the solutions and the sites in these subsequences are copied to the RP. Due to the randomicity in step(3), repeated running of the algorithm will generate different feasible solutions.

3.4 Pareto optimality based selection with elitism

In contrast to single-objective optimization problem, in which objective function and fitness function are often identical, both fitness assignment and selection must consider several objectives in MOP. To find the approximate Pareto set of the problem and avoid bias to any objective, we introduce a Pareto dominance based tournament selection strategy combined with diversity preservation to generate a new population.

Two individuals are randomly selected from the population. Firstly, the feasibilities of the two individuals are checked according to the constraints defined in Eqs.(8-11). If the two individuals are all infeasible solutions, one is randomly chosen to be the parent. If only one of the two individuals is a feasible solution, the feasible one is chosen to be the parent. If the two individuals are all feasible solutions, the vector objectives of the individuals are computed according to Eqs.(2-4). Based on the vector objectives, the non-dominated individuals will be selected as the parents. If none of the two individuals dominates the other, we choose the one according to diversity preser-
vation by estimating the density of the individuals. In our approach, we use crowding distance defined by Deb[7] as the density estimation. The crowding distance is the average distance of the two points on either side of a particular point along each of the objectives, which indicates the density of solutions surrounding the point in the population. According to the definition, bigger crowding distance means less density. Then, for the two feasible solutions which do not dominate each other, the one with bigger crowding distance is chosen to be the parent. The process is repeated until the parent individuals fill the mating pool. In our approach, the size of mating pool is equal to the population size.

Elitism mechanism is introduced to avoid losing good solutions during the optimization process due to random effects. We adopt archiving strategies in MSLUEA. A secondary population archive is maintained during the evolutionary process, to which non-dominated solutions in the population are copied at each generation. An individual in the current population can be copied to the archive if and only if that it is non-dominated in the current population and is not dominated by any individuals in the archive.

### 3.5 Pareto optimality based selection with elitism

#### 3.5.1 Sequence exchange crossover

According to the special chromosome representation described above, a novel sequence exchange crossover operator is introduced. Given two parents, $P_1 = \{s_{11}, ..., s_{1Nv}, RP_1\}$ and $P_2 = \{s_{21}, ..., s_{2Nv}, RP_2\}$, a random integer $pc$ between 1 and $Nv$ is generated. The sequence $s_{1pc}$ and $s_{2pc}$ are exchanged, which results in $C_1 = \{s_{11}, ..., s_{1pc}, ..., s_{1Nv}, RP_1\}$ and $C_2 = \{s_{21}, ..., s_{2pc}, ..., s_{2Nv}, RP_2\}$. Then the repeated sites in other subsequences and in the RP are deleted. The next step is to locate the best possible locations for the missing sites that do not exit in the chromosome. For a missing site $t$ belonging to $C_{pt}$ will be inserted into chromosome $C = \{s_1, ..., s_{Nv}, RP\}$, $s(t) = \{s_i \in C | j \leq pt\}$ is the set of all the subsequences that site $t$ can be inserted into. The best subsequence that $t$ will be inserted to is

$$s^* = \arg \min_{s_i \in s(t)} \min_{s_{ik} \in s_i, (p,q) \in s_k} \text{Cost2}(t, p, q)$$ (15)

If the subsequence generated by inserting $t$ to $s^*$ is infeasible and no more vehicles in $V^k$ is available, $t$ is added to RP.

#### 3.5.2 Mutation operators

Mutation aids evolutionary algorithm to break away from fixation at any given point in the search space. For MSLUEA, since the constraints can easily be violated, it may be better to introduce the smaller destruction on the good schemas. Three different problem specific mutation operators are put forward:

- **Relocate to RP**: Selecting a site in the sequences randomly and adding it into the RP.
- **Relocate from RP**: Selecting a site in the RP randomly and inserting it to the sequences.
- **Exchange with RP**: Selecting a site in the sequences randomly and adding it into the RP. Selecting a site in the RP randomly and inserting it to the sequences.

For an individual, the above three mutation operators are applied with mutation probabilities $p_{mut}$, $p_{mf}$ and $p_{me}$ respectively. The three different mutation operators ensure that the mutated individuals will still satisfy all the constraints defined in the problem formulation.

### 4 Simulation results

Simulation experiments are carried out to verify the performance of MSLUEA. In our experiments, 3 different kinds of vehicles with different capabilities are employed to visiting 100 sites. The demands of the 3 kinds vehicles are $r_1 = 1$, $r_2 = 5$, $r_3 = 10$ respectively. The locations and time windows of 100 sites are the same with those of the customers in the Solomon’s Vehicle Routing Problems with Time Window instances[8]. The mission demands of the sites are generated randomly between 1 and 50, and visiting duration time are generated randomly between 10 and 100 time units.

The parameters of MSLUEA are set as

- population size: 100
- generation span: 200
- crossover rate: $p_C = 0.80$
- mutation rate: $p_{mut} = 0.10$, $p_{mf} = 0.10$, $p_{me} = 0.10$

#### 4.1 Good multi-objective optimization effort

The convergence trend is a useful indication to validate the performance of any optimization algorithm.
MSPLU has two objectives, MC and NUS. Both objectives are chosen as measures to show the convergence trend of MSLUEA. We show how minimization of both objectives occurs throughout the generations. To eliminate the random influence, we run MSLUEA 10 times and present the average results. Fig.3(a) shows the reducing of NUS over generations and Fig.3(b) shows the reducing of MC throughout every generation. The declines of the two objectives are faster at the earlier generations as compared to later.

![Figure 3: NUS and MC over generation](image)

The major character of MSLUEA is that it tries to find the approximate Pareto set of the problem instead of a single optimal solution. To verify the multi-objective optimization effort of our algorithm, the elitist solutions in the archive at several generations are shown in Fig.4, where Initial stands for the non-dominated solutions in the initial population, Int1 and Int2 stands for the non-dominated solutions in the archive during some two generations, and Final stands for the non-dominated solutions in the archive after evolution. From the results, we can see that MSLUEA moves toward Pareto optimal front through evolutions.

![Figure 4: Plot of elitist points](image)

Although it is difficult to prove that we have found the optimal solution, it is reasonable to believe that MSLUEA is able to optimize the two objectives of MSPLU concurrently and effectively.

### 4.2 The effectiveness of mixed initial population

In MSLUEA, the initial population comprises feasible solutions and random generated solutions. We conduct experiments to validate the effectiveness of this mixed initial population. The Pareto set generated with initial population comprising all random generated individuals is denoted as R, and with initial population comprising all feasible individuals generated by CIFS is denoted as F, and with mixed initial population is denoted as M. We evaluate the performance of these three Pareto sets based on the C-measure and D-measure proposed by Zitzler[9]. C(A,B) represents the coverage of two sets A and B, defined as

\[
C(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \preceq b\}|}{|B|}
\]

(16)

\(C(A, B) = 1\) means that all vectors in B are weakly dominated by A. The opposite, \(C(A, B) = 0\), represents the situation when none of the points in B are weakly dominated by A. The comparison results of the three Pareto sets are shown in Table 1. From the results in Table 1, we can see that \(C(M, R) > C(R, M)\), \(C(M, F) > C(F, M)\), and \(C(F, R) > C(R, F)\), which means that in the case of C-measure, Pareto set M is better than R and F, and F is better than R.

**Table 1: C-measure between R, M and F**

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>0.7619</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0.8571</td>
<td>0.4231</td>
<td>1.000</td>
</tr>
</tbody>
</table>

D(A,B) is the relative coverage difference of two sets A and B, defined as

\[
D(A, B) = \delta(A + B) - \delta(B)
\]

(17)

where, function \(\delta(B)\) denotes the relative size in the objective space which dominated by Pareto set B (relative here means that the objective space is normalized). \(D(A, B)\) gives the relative size of the space weakly dominated by A but not weakly dominated by B. \(D(A, B) > D(B, A)\) means that A is better...
Table 2: D-measure between R, M and F

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>R</td>
<td>M</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
<td>3.6079e-004</td>
<td>8.3612e-005</td>
</tr>
<tr>
<td>M</td>
<td>0.0258</td>
<td>0</td>
<td>0.0160</td>
</tr>
<tr>
<td>F</td>
<td>0.0156</td>
<td>0.0061</td>
<td>0</td>
</tr>
</tbody>
</table>

than B in the case of D-measure. The results of the three Pareto sets are shown in table2. From the results in table2, we can see that \( D(M, R) > D(R, M) \), \( D(M, F) > D(F, M) \), and \( D(F, R) > D(R, F) \), which means that in the case of D-measure, Pareto set M is better than R and F, and F is better than R.

Above comparison results prove that the mixed initial population outperforms the random generated initial population and the initial population comprising all feasible solutions.

5 Conclusions

The modelling and solving problem is solved for mission scheduling problem with a limited number of vehicles. The proposed MSPLU model formulates the problem from the viewpoint of multi-objective optimization, and takes more specifics of the problem into account compared with previous studies, which makes it more suitable for real-world applications. MSPLU is a typical NP-Hard MOP and the solutions to it are Pareto optimal. The proposed algorithm MSLUEA tries to find the approximate Pareto optimal set of MSPLU. The integer string chromosome representation is designed which ensures that the solution can satisfy the demands of the sites. MSLUEA uses an effective construction heuristic algorithm CIFS to generate initial feasible solutions. The initial population comprises feasible solutions and random solutions to keep algorithm MSLUEA from sticking in a local optimum and taking a long time to converge as well. Then Pareto optimality based selection mechanism combined with diversity preservation is introduced to generate parent population, which prevents bias to any objective and drives the algorithm converging to the approximate Pareto set of the problem. Problem specific evolutionary operators, including a novel sequence exchange crossover operator and three different mutation operators are presented to ensure the feasibilities of the offsprings. The simulation results show that MSLUEA performs very well in solving MSPLU, and has good multi-objective optimization effort.

References: