ISSN : 0974 - 7435 Volume 10 Issue 9

2014

BioTechnology
An Indian Journal

FULL PAPER
BTAIJ, 10(9), 2014 [3856-3867]

Emergency supplies scheduling model for single demand point and its constrained multi-objective particle swarm optimization algorithm

Yong Lin, Dali Jiang, Li Zhang, Yisheng Wang
Institute of Modern Logistics, Logistical Engineering University, Chongqing 401311, (CHINA)
E-mail : lycq04183@163.com

ABSTRACT

Aiming at the shortest emergency rescue completion time and the maximum mean full-load ratio of transportation, a newly multi-objective model of emergency supplies scheduling for single demand point is proposed. Through the analysis of the diversity, the finiteness, the maximum load and the maximum capacity of transportation, the suggested model is more comprehensive and realistic. In order to solve this model, a constrained multi-objective particle swarm optimization algorithm fused with multiple constraint handling techniques (CMOPSO-MCHT) is presented, in which integral iteration, non-negative solution space limitation and hyper-plane constraints are constructed to update the velocity of each particle, a dynamic threshold constraint dominance rule is put forward to update the individual best location of each particle and a method of objective function modification is applied to update the global best location of the swarm. The results of numerical experiment show that the set of Pareto optimal solutions obtained by CMOPSO-MCHT has a much better convergence and spread.

KEYWORDS

Single demand point; Emergency supplies; Scheduling; Constrained multi-objective optimization; Particle swarm optimization.

INTRODUCTION

According to the reviews provided by refs.[1], many emergency supplies scheduling models for single demand point have been proposed, and also can be specifically divided into two classes in accordance with the type of emergency supplies. On the one hand, some previous studies only focused on one type of emergency supplies. Han et al.[2] studied a multi-objective optimization model to optimize the timing, economy and reliability objective by use of the fuzzy optimization theory, and then adopted ideal point method to solve the proposed model. Liu et al.[3] considered the transport-power as a new constraint condition, and established an emergency resource dispatching model with two alternative objective functions, one that minimizes starting time for rescue activity and another that minimizes the number of supply points. On the other hand, another stream of research devoted to multi-type of emergency supplies. Zhang et al.[4] addressed an emergency supplies scheduling model in which the objective is weighted by lost and cost, and then proposed an adaptively mutate genetic algorithm to generate solutions.

Although emergency supplies scheduling for single demand point is clearly receiving increased attention in the literature, the existing models exhibit two drawbacks. First, most of the existing models didn't take into account the constraints of transport-power. Even if Liu et al.[3] considered the constraints of transport-power, they don't pay attention to the diversity and the finiteness of transportation. Second, although many objective functions were considered in the existing models, existing research didn't take the full-load ratio of transportation as an optimization objective.

To overcome above drawbacks, this paper takes the diversity, the finiteness, the maximum load and the maximum capacity of transportation into consideration, and then develops a newly constrained multi-objective nonlinear integer programming model of emergency supplies scheduling for single demand point.

Note that the model of this paper is an important part of constrained multi-objective optimization problem (CMOP). Even though many evolutionary algorithms and constraint handling techniques which can be applied to solve CMOP have been proposed in recent years[5], the existing algorithms are inapplicable for our model. Two reasons are the following. First, our model requires that solutions should be non-negative integer, however the available algorithms are put forward on the basis of condition that solutions is not only real number but also bounded[6]. Second, our model has equality constraints, but the available algorithms can't deal with equality constraints very well[7]. Furthermore, there exists no benchmark function previously used in literatures involving equality constraints for constrained multi-objective optimization algorithms[8].

The remainder of this paper is organized as follows. Section 2 gives a description of the emergency supplies scheduling problem for single demand point. Notations and mathematical formulation of this model are presented in Section 3. The CMOPSO-MCHT algorithm is discussed in Section 4. An illustrative example is provided in Section 5. Finally, conclusion and consideration for future work are presented in Section 6.

PROBLEM DESCRIPTION

The problem of emergency supplies scheduling for single demand point can be described as follows. There are multiple supply points and only one demand point in the two-level supply-demand network. Many types of emergency supplies are stored and many types of transportation are possessed at each supply point. At a certain time, when some types of emergency supplies are applied for, crisis manager considers both the loading time and transportation time to determine the quantities of emergency supplies transported by each type of transportation at each supply point. Crisis manager hopes to fulfill the requirements of the demand point as far as possible, meanwhile, to minimize emergency rescue completion time and maximize mean full-load ratio of transportation. Several assumptions are given as follows. 1) All reserves of emergency supplies at supply points are assumed to be sufficient to meet the demand. 2) Only one trip between supply points and demand point is permitted, transportations among supply points are not considered. 3) Only loading time and transportation time between supply points and demand point are taken into account, regardless of the unloading time at demand point.

MODELING

Notations

$SP = \{i \mid i = 1,2,\cdots,I\}$: set of supply point numbers, and I is the total number of supply point; $SA = \{m \mid m = 1,2,\cdots,M\}$: set of emergency supplies types, and M is the total number of emergency supplies types; $SV = \{k \mid k = 1,2,\cdots,K\}$: set of transportation types, and K is the total number of transportation types.

S_i: supply point i; O: demand point; b_m: amount of the emergency supplies m demanded at O; $\sup_{im}$: amount of emergency supplies type m reserved at S_i; t_{ik}: estimated time required for transporting emergency supplies from S_i to O by transportation type k; av_{ik}: amount of transportation type k possessed at S_i; q_m: unit weight of emergency supplies m; v_m: unit volume of emergency supplies m; t_m: unit loading time of emergency supplies m; Q_k: maximum load of transportation type k; V_k: maximum capacity of transportation type k;

x_{imk}: decision variable, amount of emergency supplies m transported from S_i to O by transportation type k; δ_{ik}: equal to one if any type of emergency supplies is transported by transportation type k from S_i to O, and 0 otherwise; η_{ik}: variable synthesized by x_{imk}, amount of transportation type k actually used at S_i.

Formulation

According to above preliminaries, the emergency supplies scheduling model for single demand point can therefore be stated as follows:

$$\min f_1 = \max_{\substack{\forall i \in SP \\ \forall k \in SV}} \left(\sum_{m \in SA} t_m x_{imk} + \delta_{ik} t_{ik} \right) \tag{1}$$

$$\max f_2 = \frac{\sum_{i \in SP} \sum_{k \in SV} \max\left(\frac{\sum_{m \in SA} x_{imk} q_m}{\eta_{ik} Q_k}, \frac{\sum_{m \in SA} x_{imk} v_m}{\eta_{ik} V_k} \right)}{\sum_{i \in SP} \sum_{k \in SV} \delta_{ik}} \tag{2}$$

$$\eta_{ik} = \max\left\{ \left\lceil \frac{\sum_{m \in SA} x_{imk} q_m}{Q_k} \right\rceil, \left\lceil \frac{\sum_{m \in SA} x_{imk} v_m}{V_k} \right\rceil \right\} \tag{3}$$

Subject to:

$$\sum_{i \in SP} \sum_{k \in SV} x_{imk} = b_m \tag{4}$$

$$\sum_{k \in SV} x_{imk} \le sup_{im} \tag{5}$$

$$\eta_{ik} \le av_{ik} \tag{6}$$

$$x_{imk} \ge 0 \text{ and} \tag{7}$$

$$\delta_{ik} = \begin{cases} 0 & \sum_{m \in SA} x_{imk} = 0 \\ 1 & \sum_{m \in SA} x_{imk} \ne 0 \end{cases} \text{ integer} \tag{8}$$

There are two objective functions in this model. The objective function (1) aims at the shortest emergency rescue completion time, and the objective function (2) maximizes the mean full-load ratio of transportation. Eq. (3) calculates the amount of transportation type k actually used at S_i. $\lceil\ \rceil$ is a rounding mode to round towards positive infinity. The constraints (4) ensures that demand point O receives the requested amount of each emergency supplies m, which are equality constraints. The constraints (5) guarantee that the total amount of a given emergency supplies m delivered from S_i does not exceed its reserves. The constraints (6) indicate that the amount of transportation type k actually used at S_i should be less than or equal to the amount of transportation type k possessed at S_i. The constraints (7) require that the decision variables must be non-negative integer.

THE PROPOSED ALGORITHM

To solve the above constrained multi-objective optimization model, particle swarm optimization (PSO) is adopted for the basic evolutionary framework of our proposed algorithm. Moreover, three strategies are presented to deal with different constraints appeared in this model. First, integral iteration, non-negative solution space limitation and hyper-plane constraints are put forward to update the velocity of each particle, which meets the requirements of non-negative integer constraints and equality constraints. Second, a dynamic threshold constraint dominance rule is proposed to update the personal best location of each particle. Finally, a method of objective function modification[9] is adopted for updating the global best location of the swarm. The last two strategies serve as inequality constraints, and maintain the diversity of Pareto optimal solutions by use of infeasible solutions.

Individual encoding

$X = (X_1, X_2, \cdots, X_N)$ denotes particle swarm. $X_q = (X_{q1}, X_{q2}, \cdots, X_{qD})$ denotes the location of particle q. where D is the total dimension of particle, $d = 1,2,\cdots,D$. N is the swarm size, $q = 1,2,\cdots,N$. X_q is encoded in non-negative integer according to the sequence of supply point, and each supply point can be expressed by Eq. (9). The relationship between d and i,m,k is given by Eq. (10).

$$\underbrace{\overbrace{x_{i11}, x_{i21}, \cdots, x_{iM1}}^{\text{transportation type 1}}, \cdots, \overbrace{x_{i1K}, x_{i2K}, \cdots, x_{iMK}}^{\text{transportation type K}}}_{S_i} \tag{9}$$

$$\mathbf{d = (i-1)\cdot M\cdot K + (k-1)\cdot M + m} \quad (10)$$

Strategies for updating the velocity of each particle

A standard PSO algorithm can be described as Eq. (11) and Eq. (12).

$$\begin{aligned}\mathbf{V_{qd}^{s+1}} &= \mathbf{w^s \cdot V_{qd}^s + c_1^s r_1 (XP_{qd}^s - X_{qd}^s)} \\ &\quad \mathbf{+ c_2^s r_2 (XG_{qd}^s - X_{qd}^s)}\end{aligned} \quad (11)$$

$$\mathbf{X_{qd}^{s+1} = X_{qd}^s + V_{qd}^{s+1}} \quad (12)$$

Where s is iteration index; c_1^s and c_2^s are cognitive and social parameters respectively, dynamically update by Eq. (13) in our proposed algorithm, $c_{\max}$ and $c_{\min}$ are initial maximum and minimum value respectively; w^s is the inertia weight, dynamically updates by Eq. (14) in our proposed algorithm; T is the maximal iteration time, $s = 1,2,\cdots,T$; r_1 and r_2 are two random real numbers uniformly distributed in the range $[0,1]$; V_q^s and XP_q^s denote the velocity and the personal best location of particle q in the s^{th} iteration respectively. XG_q^s represents the global best location of the swarm in the s^{th} iteration.

$$\mathbf{c_1^s = c_2^s = c_{max} - \frac{c_{max} - c_{min}}{T} \cdot s} \quad (13)$$

$$\mathbf{w^s = 0.5 + \frac{1}{2[\ln(s) + 1]}} \quad (14)$$

(1) Integral iteration

According to r_1 and r_2 appeared in Eq. (11), PSO is mainly dedicated to continuous problems. In order to ensure that each candidate solution encoded in X_q is an integer, the velocity of each particle should be updated to an integer uniformly distributed in the range $[V_{lower}, V_{upper}]$. V_{lower} and V_{upper} are given in Eq. (15) and Eq. (16) respectively.

$$\begin{aligned}\mathbf{V_{lower}} &= \mathbf{round(w^s \cdot V_{qd}^s)} \\ &\quad \mathbf{+ \min(0, round(c_1^s (XP_{qd}^s - X_{qd}^s)))} \\ &\quad \mathbf{+ \min(0, round(c_2^s (XG_{qd}^s - X_{qd}^s)))}\end{aligned} \quad (15)$$

$$\begin{aligned}\mathbf{V_{upper}} &= \mathbf{round(w^s \cdot V_{qd}^s)} \\ &+ \mathbf{max(0, round(c_1^s(XP_{qd}^s - X_{qd}^s)))} \\ &+ \mathbf{max(0, round(c_2^s(XG_{qd}^s - X_{qd}^s)))}\end{aligned} \tag{16}$$

(2) Non-negative solution space limitation

According to Eq. (12), the location of particle X_q^{s+1} can't be fixed in non-negative solution space, so the lower and upper bounds for updating velocity must be adjusted. Detailed adjustment strategies are listed in TABLE 1. C denotes the conditions of adjustment, and S denotes the operations of adjustment.

TABLE 1 : Adjustment strategies for upper and lower bounds of the velocity updating interval

C	$(V_{upper} + X_{qd}^s) \le 0$	$(V_{lower} + X_{qd}^s) < 0$ **&&** $(V_{upper} + X_{qd}^s) > 0$
S	$V_{lower} = V_{upper} = -X_{qd}^s$	$V_{lower} = -X_{qd}^s$

(3) Hyper-plane constraints

Equality constraints (4) can be expressed as Eq. (17) by the location of a particle.

$$\sum_{\mathbf{d \in H}} \mathbf{X_{qd}^s = b_m} \tag{17}$$

$$\mathbf{H = \{d \mid d = (i-1) \cdot M \cdot K + (k-1) \cdot M + m\}} \tag{18}$$

According to Eq. (12), Eq. (17) and Eq. (19), in order to ensure that the updated location X_{qd}^{s+1} satisfies the equality constraints, V_{qd}^{s+1} must be content with the hyper-plane constraints given by Eq. (20).

$$\sum_{\mathbf{d \in H}} \mathbf{X_{qd}^{s+1}} = \sum_{\mathbf{d \in H}} \mathbf{X_{qd}^{s}} + \sum_{\mathbf{d \in H}} \mathbf{V_{qd}^{s+1}} \tag{19}$$

$$\sum_{\mathbf{d \in H}} \mathbf{V_{qd}^{s+1} = 0} \tag{20}$$

Strategies for updating the personal best location of each particle

When PSO is applied to solve CMOP, constraint dominance rule must be used to update the personal best location of each particle. Though a widely used rule has been proposed[10], this rule may give rise to evolutionary stagnation in the iterative process. Actually, the updated location X_q^{s+1} is probably infeasible because of the inequality constraints (5) and (6), as result, XP_q^s can't be updated by use of this rule. To make full use of infeasible solutions, a dynamic threshold constraint dominance rule is presented.

Dynamic threshold β^s is formally defined as follows.

$$\beta^s = \begin{cases} \beta_0 \cdot \cos(\frac{5\pi}{7T} \cdot s) & 1 \le s \le 0.7T \\ 0 & 0.7T < s \le T \end{cases} \tag{21}$$

Where β_0 is the initial constraint violation tolerance.

According to the definition of constraint violation[8], the constraint violations of X_q^{s+1} and XP_q^s can be denoted as $Q(X_q^{s+1})$ and $Q(XP_q^s)$ respectively. In addition, F_q^{s+1} and FP_q^s are denoted as the objective function values of X_q^{s+1} and XP_q^s respectively. If Eq. (22) holds, the personal best location of particle q is updated as $XP_q^{s+1} = X_q^{s+1}$, otherwise, $XP_q^{s+1} = XP_q^s$. Eq. (22) indicates that X_q^{s+1} dominates XP_q^s in dynamic threshold constraint dominance rule.

$$X_q^{s+1} \succ_\beta CD\ XP_q^s \Leftrightarrow \begin{cases} F_q^{s+1} \succ FP_q^s & Q(X_q^{s+1}),\ Q(XP_q^s) < \beta^s \\ F_q^{s+1} \succ FP_q^s & Q(X_q^{s+1}) = Q(XP_q^s) = \beta^s \\ Q(X_q^{s+1}) < Q(XP_q^s) & \text{otherwise} \end{cases} \tag{22}$$

Strategies for updating the global best location of the swam

In our proposed algorithm, all Pareto optimal solutions searched in iterations are preserved in an external repository, which is denoted as GF. The global best location of the swarm is selected randomly in GF when the velocity of each particle will be updated. GF is updated via the following three steps.

1) Unite GF with the updated particle swarm X^{s+1} to be a population denoted as Pop, and set $GF = \varnothing$.

2) Modify the objective function value of each particle in Pop by use of the constraint handling technique proposed in refs.[9].

3) Select the particle that can't be dominated by other particles in Pop into GF according to modified objective function values.

Before modifying the objective function values, each objective function value of each particle must be normalized. In detail, the objective function value of the particle q in Pop is normalized as follows.

$$\tilde{f}_r(X_q) = \frac{f_r(X_q) - f_r^{min}}{f_r^{max} - f_r^{min}} \tag{23}$$

Where $\tilde{f}_r(X_q)$ is the normalized r^{th}-objective value of particle X_q; $f_r(X_q)$ is the primary objective value of particle X_q; $f_r^{\min}$ and $f_r^{\max}$ are the minimum and maximum values of each objective function in Pop respectively; r is the number of objective function, $r = 1,2$ in our algorithm.

After normalization, each objective function value of each particle in Pop is modified in accordance with the following three conditions.

1) If all particles in Pop are feasible solutions, the objective function values of all particle need not to be modified.

2) If all particles in Pop are infeasible solutions, each objective function value of each particle is modified to its constraint violation.

3) If feasible and infeasible solutions exist in Pop simultaneously, each objective function value of particle X_q is modified by Eq. (24).

$$\mathbf{f_r'(X_q)} = \sqrt{\tilde{\mathbf{f}}_\mathbf{r}(\mathbf{X_q})^2 + \mathbf{Q(X_q)}^2} + (1-\alpha)\cdot \mathbf{Q(X_q)} + \alpha \cdot \tilde{\mathbf{f}}_\mathbf{r}(\mathbf{X_q}) \qquad \mathbf{(24)}$$

Where $f_r'(X_q)$ is the modified r^{th}-objective value of particle X_q; $Q(X_q)$ is the constraint violation of particle X_q; α is the ratio of the number of feasible solutions in Pop to the swarm size of Pop.

Implementation steps

Step1: Initialize parameters N, T, $c_{\max}$, $c_{\min}$ and β_0. Generate an initial particle swarm X^1 of N particles in feasible region (Section 4.1). Set $s=1$, $V_q^s=0$ and $XP_q^s = X_q^s$ for $q=1,2,\cdots,N$. Calculate the objective function values and the constraint violations of all particles. Select the Pareto optimal solutions from X^1 to external repository F. Set $GF = F$.

Step2: update w^s, c_1^s and c_2^s. Carry out the following operations for X_q^s according to the order $q=1,2,\cdots,N$.

1) Select a particle from GF as the global best location of X_q^s.

2) Compute V_q^{s+1} and X_q^{s+1} (Section 4.2). Calculate the objective function values and the constraint violations of X_q^{s+1}.

Step3: Unite X^{s+1} to F. Update F according to dominance relation. Update GF (Section 4.4).

Step4: Update personal best location of X_q^s according to the order $q=1,2,\cdots,N$ (Section 4.3).

Step5: If $s<T$, set $s=s+1$, return Step2. Otherwise, end the algorithm and output F.

NUMERICAL EXPERIMENT

Experimental settings

An experiment is used to test the validity of this model and algorithm mentioned above, which assumes that there are four supply points ($S_1 \sim S_4$), two types of transportation (truck and helicopter), four types of emergency supplies (tent, quilt, clothes and food) demanded at O. Parameters required in this model are given as follows.

As mentioned above (Section 1), the existing algorithms are inapplicable for this model, so there are no available comparative algorithms. However, to validate the performance of the proposed algorithm (hereinafter referred to as CMOPSO-MCHT), this paper embeds Section 4.2 into two outstanding algorithms to form two comparative algorithms. The first selected algorithm is the modified constrained multi-objective particle swarm optimization algorithm[11] (hereinafter referred to as M-CMOPSO). The second algorithm is the barebones multi-objective particle swarm optimizer[12] (hereinafter referred to as BB-MOPSO).

$$sup_{im} = \begin{bmatrix} 274 & 349 & 268 & 170 \\ 254 & 200 & 300 & 210 \\ 136 & 237 & 198 & 432 \\ 197 & 202 & 430 & 169 \end{bmatrix},$$

$$d_m = [549 \quad 430 \quad 566 \quad 658],$$

$$t_{ik} = \begin{bmatrix} 1.6 & 0.7 \\ 1.8 & 0.9 \\ 2.0 & 1.1 \\ 2.0 & 1.4 \end{bmatrix},$$

$$av_{ik} = \begin{bmatrix} 8 & 0 \\ 9 & 2 \\ 5 & 0 \\ 2 & 3 \end{bmatrix},$$

$$q_m = [30 \quad 6 \quad 5 \quad 10],$$

$$v_m = [1.1 \quad 0.15 \quad 0.3 \quad 0.5],$$

$$t_m = \begin{bmatrix} \frac{1}{300} & \frac{1}{300} & \frac{1}{600} & \frac{1}{600} \end{bmatrix},$$

$$Q_k = [5000 \quad 8000], V_k = [40 \quad 600].$$

Public parameters for three algorithms: $N = 200$, $T = 1000$, $c_{\max} = 1.5$, $c_{\min} = 0.5$. Private parameter for CMOPSO-MCHT: $\beta_0 = 0.5$. No private parameters for M-CMOPSO. Private parameters for BB-MOPSO: $N_1 = 100$, $N_2 = 100$.

Performance metrics

Two performance metrics are selected to evaluate the performance of three algorithms. One is the two sets coverage (TSC), Another is the spacing (SP).

•TSC: Suppose A and B be two approximations to the Pareto front of a CMOP, $TSC(A,B)$ gives the percentage of the solutions in B that are dominated by at least one solution in A, i.e.,

$$\mathbf{TSC(A,B)} = \frac{\left|\{\mu \in \mathbf{B} \mid \exists \upsilon \in \mathbf{A} : \upsilon \succ \mu\}\right|}{|\mathbf{B}|} \quad \mathbf{(25)}$$

It may be noted that it is not necessary that $TSC(A,B) = 1 - TSC(B,A)$. Moreover, if $TSC(A,B) = 1$, then all solutions in B are dominated by some solutions in A, and if $TSC(A,B) = 0$, then no solution in B is dominated by a solution in A. Generally, TSC measures the extent of convergence to a known set of Pareto optimal solutions.

•SP is a value measuring the spread (distribution) of obtained solutions. The smaller the SP is, the more equidistant the solutions spaces. The formula is presented as follows.

$$\mathbf{SP}=\sqrt{\frac{1}{h-1}\sum_{g=1}^{h}(\bar{c}-c_g)^2} \quad \bar{c}=\frac{1}{h}\sum_{g=1}^{h}c_g$$
$$c_g=\min\left\{\sum_{r=1}^{p}\left|f_r^g(x)-f_r^e(x)\right|\right\} \tag{26}$$
$$e=1,2,\cdots h, e\neq g$$

Where h is the total number of Pareto optimal solutions; p is the number of objective functions; c_g is the distance of neighboring solutions.

Results analysis

Figure 1 shows the typical Pareto front (PF) obtained by above three algorithms. From Figure 1, although three algorithms all can achieve some Pareto optimal solutions in total iterations, the performance of the three algorithms is greatly different.

•By comparing three curves of PF in Figure 1, we can conclude that the Pareto optimal solutions obtained by CMOPSO-MCHT can absolutely dominate those obtained by the other two algorithms. In another word, the extent of convergence for CMOPSO-MCHT is much better.

•By observing the distribution of Pareto optimal solutions obtained by three algorithms (three independent graph in Figure 1), we can know that the Pareto optimal solutions obtained by CMOPSO-MCHT and BB-MOPSO distributes more uniformly, those acquired by M-CMOPSO has the worst distribution.

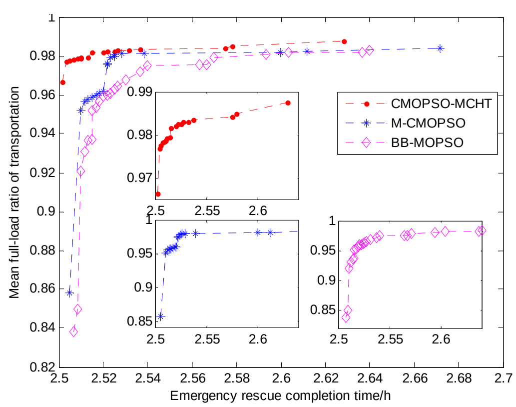

Figure 1 : Typical Pareto front obtained by three algorithms

TABLE 2 presents the statistics of performance metrics of the final solutions obtained by each algorithm based on 50 independent runs. These statistics include mean and standard deviation values of TSC and SP. From the perspective of TSC, this table reveals that average 72% solutions obtained by M-CMOPSO are dominated by solutions achieved by CMOPSO-MCHT in each solving process. Moreover, this data of BB-MOPSO is as high as 87%. On the contrary, few solutions obtained by CMOPSO-MCHT are dominated by those obtained by M-CMOPSO and BB-MOPSO. These statistical results also indicate that CMOPSO-MCHT has a much better convergence. From the perspective of SP, the smallest mean and standard deviation value of SP for CMOPSO-MCHT illustrate that CMOPSO-MCHT has a much better spread and stability.

TABLE 2 : Statistical results of the performance metrics found by three algorithms based on 50 independent runs

	TSC			*SP*
	CMOPSO-MCHT	**M-CMOPSO**	**BB-MOPSO**	
CMOPSO-MCHT	—	0.72±0.29	0.87±0.18	0.013±0.009
M-CMOPSO	0.13±0.23	—	0.59±0.30	0.022±0.014
BB-MOPSO	0.055±0.14	0.28±0.28	—	0.017±0.013

CONCLUSIONS

In this paper, a constrained multi-objective nonlinear integer programming model is proposed to achieve the scheduling of emergency supplies for single demand point. The distinguishing feature of this model is to consider the loading time of emergency supplies, different types of transportation and to encompass mean full-load ratio of transportation as an objective. In addition, a constrained multi-objective particle swarm optimization algorithm fused with multiple constraint handling techniques (CMOPSO-MCHT) is developed. The performance of this algorithm is analyzed and its efficiency is investigated. We find that, in general, CMOPSO-MCHT can not only resolve this model effectively, but also provide much better Pareto optimal solutions.

We suggest two directions for future work. The first is to put forward a much better algorithm to solve this model, and test our model and algorithm on much more experiments. The second is to introduce the robust approach to investigate the uncertainty including uncertain demand and uncertain transportation time, and then develop more realistic scheduling model.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (NO. 71101152), and the Natural Science Foundation Project of CQ (NO. cstc2011bb0045). This work is also supported in part by the Natural Science Foundation Project of CQ (NO. cstc2010bb9200, NO. cstc2011jjA00008).

REFERENCES

[1] A.M.Caunhye, X.F.Nie, S.Pokharel; Optimization models in emergency logistics: A literature review, Socio-Economic Planning Sciences, **46(1)**, 4-13 **(2012)**.

[2] Han Jingti, Chi Weidie, Han Xiaomei; Study of emergency materials scheduling problem based on ELS3, Journal of System Simulation, **21(18)**, 5828-5830, 5835 **(2009)**.

[3] Liu Mu, Qian Xinming, Guo Chen et al.; Dispatching model and algorithm for emergency resource of oil & gas long-distance pipeline under transport-power constraint condition, Transactions of Beijing Institute of Technology, **31(8)**, 996-1000 **(2011)**.

[4] Zhang Liming, Lin Yuhua, Yang Guofeng et al.; Emergency resources scheduling based on adaptively mutate genetic algorithm, Computers in Human Behavior, **27(5)**, 1493-1498 **(2011)**.

[5] E.M.Montes, C.A.C.Coello; Constraint-handling in nature-inspired numerical optimization: Past, present and future, Swarm and Evolutionary Computation, **1(4)**, 173-194 **(2011)**.

[6] M.A.Jan, R.A.Khanum; A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D, Applied Soft Computing, **13(1)**, 128-148 **(2013)**.

[7] Xu Bin, Qi Rongbin, Qian Feng; Constrained multi-objective optimization with hybrid differential evolution and alpha constrained domination technique, Control Theory & Applications, **29(3)**, 353-360 **(2012)**.

[8] Shang Ronghua, Jiao Licheng, Hu Chaoxu; Modified immune clonal constrained multi-objective optimization algorithm, Journal of Software, **23(7)**, 1773-1786 **(2012)**.

[9] Y.G.Woldesenbet, G.G.Yen, B.G.Tessema; Constraint handling in multi-objective evolutionary optimization, IEEE Transactions on Evolutionary Computation, **13(3)**, 514-525 **(2009)**.

[10] K.Deb, A.Pratap, S.Agarwal et al.; A fast and elitist multi-objective genetic algorithm NSGA-II,IEEE Transactions on Evolutionary Computation, **6(2)**, 182-197 **(2002)**.

[11] L.D.Li, Yu Xinghuo, Li Xiaodong et al.; A modified PSO algorithm for constrained multi-objective optimization, in third International Conference on Network and System Security, Gold Coast,Australia, October, **19-21**, 462-467 **(2009)**.

[12] Zhang Yong, Gong Dunwei, Ren Yongqiang et al.; Barebones multi-objective particle swarm optimizer for constrained optimization problems, Acta Electronica Sinica, **39(6)**, 1436-1440 **(2011)**.