doi:10.21311/002.31.03

A two-stage priority-based MOGA for materials allocation and loading problem in large-scale emergencies

Yuanwen Chen*, Dali Jiang*, Xiaobo Wu, Zuofei Tan, Yu Jiang

Institute of Modern Logistics, Logistical Engineering University, Chongqing 401311, China

Abstract

Reasonable allocation and reliable delivery of rescue materials are of vital importance in large-scale emergency relief operations. Aiming at lower the average completion time, lower overall risk and higher utilization of transportation resources, this paper sets up a multi-objective model based on the problem of direct distribution of various relief materials from multiple supply points to multiple demand points in the affected areas considering quantity constraints of materials and available vehicles in each supply points, as well as considering capacity and volume constraints of each kinds of vehicles. A two-stage priority based MOGA algorithm has been designed to solve the problem. The experiment shows that this algorithm is fast and reliable, and has high practicability.

Keywords: large-scale emergency relief; multi-objective; allocation and delivery; two-stage priority based MOGA algorithm

1. INTRODUCTION

Various kinds of large-scale emergencies, such as, natural disasters, public health events, and social security events, have caused great losses to people’s life and their property safety. Efficient rescue operations is the most effective way to save lives and reduce losses after the disasters, among them the key task is to prepare emergency relief materials at the fastest speed and distribute them reasonably to distress areas.

In recent years, emergency logistics theory has been put forward and obtained plenteous achievements (Ou et al., 2004; Manopiniwes and Irohara, 2014). The major research content of accident rescue is how to timely master and comprehensively utilize the environment information to make decision on how to distribute various relief materials from multiple material storage and collection points to handout and sub-handout points in affected areas in an efficient and reliable way. Gen et al. (2006) sets up a two-stage model considering the allocation of materials and opening of distribution centres, a priority based GA has been designed to solve the problem. Özdamar and Demir (2012) establish a joint distribution mathematical model with relief materials first and the wounded people second, which aims at minimizing the transportation time and maximizing the utilization of resources. In (El-Sherbiny and Alhamali, 2013), the study introduces a hybrid PSO algorithm with artificial immune learning for solving allocation problem considering distributing one type of material from multiple SPs (supply points) to multiple DPs (demand points) by one kind of vehicle. In this algorithm, decoding procedure and allocation procedure are used and a spanning tree that used with genetic algorithms by which a feasible solution can be found for each generated particle. Lin et al. (2011) propose an emergency supply multi-objective model considers multi-items, multi-vehicles, multi-periods, soft time windows, and a split and prioritized delivery strategy scenario. Chang and other researchers (Chang et al., 2014) propose a multi-objective greedy-search-based GA to solve the problem of distributing materials (one type) to various distress areas aiming at minimizing unsatisfied demand, delivering time, and transportation costs. Yi and Özdamar (2007) sets up a location-distribution model.
considering dispatching commodities to affected areas and transferring wounded people out to hospitals. A pseudo-polynomial algorithm was designed to solve the mixed integer multi-commodity network flow model.

There has been certain achievements in distribution or allocation of relief supplies, but it should be noted that there is always not a small gap between the existing research and actual situations because of the complicated and changeable environment. With urgent and enormous requirements of relief materials after large-scale emergencies, materials should be delivered directly from SPs to DPs considering many environment factors such as traffic condition and supply-demand relations. In this paper, a new allocation and loading mathematical model has been set up and a two-stage priority based MOGA algorithm has been designed to solve the problem which is described in the following sections.

2. PROBLEM DESCRIPTION AND PRESUMPTION

2.1 Problem description

Under the large-scale emergency circumstance, the process of materials allocation and loading is as following: According to the types and quantity of material and capacity of each types of vehicles available in SPs (including materials storage and collection points), on the basis of the estimation of the road condition (including traveling time and risk of road damage) between all SPs and DPs (DP including material handout points/sub-handout points), as well as demand urgency of each type of materials for each DP, the disaster relief organizations formulate a detailed allocation and delivering plan including each used vehicle loaded with how many and what type of materials from which SP to which DP. The objectives are minimum average completion time, minimum overall risk, and maximum average utilization of capacity and volume for each vehicle.

2.2 Problem presumption

This paper proposes the following presumptions: 1) There is no material dispatching among SPs and DPs themselves respectively; 2) All the information is available in real time conditions. The traveling time and risk of road damage is estimable; 3) No account is taken into the time of packaging, loading and unloading; 4) While loading the materials, it should be neglected that the load capacity cannot be fully utilized due to the shapes of the packages.

3. MODELING

3.1 Illustration of symbols

(1) Sets:

\[ SP = \{ i | i = 1, 2, \ldots, N_i \} : \text{set of SPs}; \]

\[ DP = \{ j | j = 1, 2, \ldots, N_j \} : \text{set of DPs}; \]

\[ MS = \{ m | m = 1, 2, \ldots, N_m \} : \text{set of categories of materials}; \]

\[ VS = \{ v | v = 1, 2, \ldots, N_v \} : \text{set of categories of vehicles}; \]

\[ HS = \{ h | h = 1, 2, \ldots, q_v \} : \text{set of vehicles type } v \text{ in } i^{th} \text{ SP}; \]
(2) Parameters:

\[ d_{jm} / p_{jm} : \text{demand amount and urgency of materials type } m \text{ at } j^{th} \text{DP}; \]
\[ st_{im} : \text{material of type } m \text{ available at } i^{th} \text{SP}; \]
\[ load_{iv} / v_{i} : \text{the maximum capacity/volume of } v^{th} \text{vehicles}; \]
\[ qv_{iv} : \text{number of } v^{th} \text{vehicles available at } i^{th} \text{SP}; \]
\[ m_{im} / v_{im} : \text{unit weight/volume of materials } m; \]
\[ t_{ij} / t_{ij} : \text{time required and risk of no delivery to travel from } i^{th} \text{SP to } j^{th} \text{DP}. \]

(3) Decision variables

\[ x_{ijmvh} : \text{positive integer number of materials } m \text{ delivered from } i^{th} \text{SP to } j^{th} \text{DP by } h^{th} \text{vehicles}; \]
\[ s_{ijvh} : \text{binary number of vehicle from } i^{th} \text{SP to } j^{th} \text{DP by } h^{th} \text{vehicles}. \]

3.2 Modeling

Objectives:

\[
\text{max } F_1 = \min \left( \sum_{j \in DP} \sum_{m \in MS} d_{jm} * x_{ijmvh} + \sum_{j \in DP} \sum_{m \in MS} d_{jm} * p_{jm} \right),
\]
\[
\text{min } F_2 = \sum_{i \in SP} \sum_{j \in DP} \sum_{m \in MS} \sum_{v \in HS} \sum_{h \in HS} x_{ijmvh} * t_{ij}.
\]
\[
\text{min } F_3 = \sum_{i \in SP} \sum_{j \in DP} \sum_{m \in MS} \sum_{v \in HS} \sum_{h \in HS} x_{ijmvh} * m_{im} * v_{im} * v_{ms} * h_{sv} * h_{sh}.
\]

Subject to:

\[
\sum_{i \in SP} \sum_{j \in DP} \sum_{m \in MS} x_{ijmvh} = d_{jm} \quad \forall j \in DP, m \in MS
\]
\[
\sum_{j \in DP} \sum_{m \in MS} x_{ijmvh} \leq st_{im} \quad \forall i \in SP, m \in MS
\]
\[
\sum_{m \in MS} \sum_{h \in HS} x_{ijmvh} \leq load_{iv} \quad \forall i \in SP, j \in DP, v \in MV
\]
\[
\sum_{m \in MS} \sum_{h \in HS} x_{ijmvh} \leq v_{iv} \quad \forall i \in SP, j \in DP, v \in MV
\]
\[
S_{ijvh} = \begin{cases} 
1 & \sum_{h \in \text{HS}} x_{ijvh} \neq 0 \\
0 & \text{else}
\end{cases}
\] (8)

\[
\sum_{j \in \text{DP}} \sum_{h \in \text{HS}} S_{ijvh} \leq q_v \quad \forall i \in \text{SP}, v \in \text{VS}
\] (9)

Eq. (1) to Eq. (3) are three objective functions representing the maximum average capacity and volume utilization, minimum average delivery completion time and minimum risk of no delivery. Eq. (4) and Eq. (5) mean that the actual amount of any kind of delivered material should be equal to its demand amount and less than its actual storage amount. Eq. (6) and Eq. (7) respectively mean that the total weight and volume of all the materials delivered from one SP are less than the total carrying capacity and volume of all the vehicles. In Eq. (8), \(S_{ijvh}\) is an intermediate variable used to count the total loaded vehicles bound from SP to a DP. Eq. (9) denotes that the total number of any kind of vehicle departing from any SP is no more than the total available number of this vehicle.

4. THE PROPOSED ALGORITHM

The model above is a nonlinear mixed integer programming with three conflicting objectives for which it is generally hard to get an exact solution. As one sort of the evolutionary algorithms, the multi-objective genetic algorithm (MOGA) is sensitive to a large increase in the volume of the search space which has been receiving great attention and successfully applied for combinatorial optimization problems (Zang et al., 2010). Based on the predecessor’s research about MOGA and materials allocation algorithm (Gen et al., 2006; Deb et al., 2002), this article proposes a two-stage priority-based MOGA algorithm to solve the multi-objective material allocation and loading problem. The detailed process of the algorithm and key operators is described in this section, and the flow chart is shown in Figure 1.

![Figure 1. Flowchart of the two-stage priority-based MOGA](image-url)
4.1 Initialization

Materials allocation and loading plans are organized using input data from disaster relief organization and command department. It should be noted that \( \sum_j d_{jm} \) may be not equal to \( \sum_j d_{jm} \), so it is necessary to add a virtual DP and set as follows:

\[
p_{i(N_j+1)m} = 0, \quad d_{i(N_j+1)m} = \sum_i s_{im} - \sum_j d_{jm} \quad \forall m \in MS
\]

\[
r_{i(N_j+1)} = t_{i(N_j+1)} = \text{BEN}(\text{Big Enough Number}) \quad \forall i \in SP
\]

Coding is a key step in the design of GA. For this problem, a sequential coding chromosome stands for priorities of DPs and SPs can obtain direct distribution tree and its length equals to \( N_I + N_J + 1 \). For example, with 3 SPs and 5 DPs a chromosome ‘7 9 8 3 6 4 1 5 2’ represents the 7th point (4th DP) has the top priority, the 9th point (6th DP the virtual DP) has the second priority...and the 2nd point (2nd SP) has the least priority to satisfy or be satisfied. (The meaning of encoding and detailed decoding will be described in Section 4.2).

4.2 Two-stage priority based decoding (TSPBD)

The essence of the TSPBD is to work out the detailed allocation and delivering plan including each used vehicle should be loaded with how many and what type of materials from which SP to which DP. The algorithm consists of two stages.

4.2.1 Stage1: material allocation algorithm based on priority

In this stage, a material allocation algorithm based on maximum satisfaction and non-dominated sort is put forward aiming at minimizing the deliver time \( (F_2 \text{ in Eq.}(2)) \) and risk \( (F_3 \text{ in Eq.}(3)) \) by which can develop a detailed allocation plan for each SP. Pseudo code of the algorithm is shown in Figure 2. In step3, SeqD and SeqS (pseudo code is shown in Figure 4, Fast Non-dominated Sort refers to (Deb et al., 2002) are optimal candidate order matrix for each DP and SP, by which for a given point it can obtain its corresponding optimal DP/SP selection. For example, there is 3 candidate SPs \( (t_1 = 0.2, \quad r_1 = 0.3; \quad t_2 = 1.2; \quad r_2 = 0.4; \quad t_3 = 0.1; \quad r_3 = 0.2) \) for the 5th DP, then SeqD can be equal to \( (3 1 2) \). Because the third has the least risk and time cost while the second has the highest. Take the Case1 as an example, step4 in Figure 2 ensures the maximum satisfaction for every categories of materials and updates the supply/demand information; in step5, it inspects the chosen DP whether has been satisfied, if it is, choose the next feasible DP in SeqD; if the chosen SP and DP has asymmetric supply and demand relations, then get the next DP; step6 check whether all the materials in the chosen SP has been delivered, if it is, get the point with lower priority in the chromosome \( ch(c) \).
Algorithm 1 The first stage of TSPBD algorithm

Input: \( s_{i,n}, d_{i,n}, SeqS, SeqD, ch \quad \forall i \in SP, j \in DP, m \in MS \)

Output: \( x_{ijm} \) quantity of \( m_{ai} \) commodities transported from \( i^{th} \) SP to \( j^{th} \) DP

Begin
Step1: \( x_{ijm} = 0, s_{i,m} = st_{i,m}, d_{j,m} = d_{j,m} \) \( c=1 \) \%initialize
While \( c \leq N_j + N_i + 1 \)
Step2: \( l = ch(c) \) \%;select a SP or DP
Case1: if \( l \leq N_i \) then \( l' = l, ch(l') = 0 \); \%select a SP
\( id = 1; \) \%id is an indicator for the sequence of SP candidates
Step3: while \( ch(SeqD(l',id)+N_j) = 0; \)
\( id = id + 1; \) \( i' = SeqD(l',id) \); \%select a next feasible SP
Step4: \( x_{i'j'} = \min(s_{i,m}, d_{j,m}); d_{j,m} = d_{j,m} - x_{i'j'} \); \( st_{i,m} = st_{i,m} - x_{i'j'} \) \( \forall m \in MS \); 
Step5: if \( d_{j,m} = 0 \) then \( ch(j'+N_j) = 0; id = id + 1; \) go toStep3
Step6: if \( d_{j,m} \neq 0 \) \& \( st_{i,m} \neq 0 \) then \( id = id + 1; \) go toStep3
Step7: if \( st_{i,m} = 0 \) then \( ch(l') = 0, c = c+1; \) go toStep 2 
else: go toStep3
Case2: if \( l > N_i \) then \( l' = l, ch(l'+N_j) = 0 \); \%select a SP
\( id = 1; \)
Step3: while \( ch(SeqS(l',id)) = 0; \)
\( id = id + 1; \) \( i' = SeqS(l',id) \); \%select a next feasible SP
Step4: \( x_{i'j'} = \min(s_{i,m}, d_{j,m}); d_{j,m} = d_{j,m} - x_{i'j'} \); \( st_{i,m} = st_{i,m} - x_{i'j'} \) \( \forall m \in MS \); 
Step5: if \( st_{i,m} = 0 \) then \( ch(l') = 0, id = id + 1; \) go toStep3
Step6: if \( d_{j,m} \neq 0 \) \& \( st_{i,m} \neq 0 \) then \( id = id + 1); \) go toStep3
Step7: if \( d_{j,m} = 0 \) then \( ch(j'+N_j) = 0, c = c+1; \) go toStep 2 
else: go toStep3
End(while \( ch=0) \)

Figure 2. Pseudo code of the first stage of TSPBD algorithm

4.2.2 Stage2: maximum load utilization oriented material loading algorithm

The main target of this stage is to obtain the detailed loading plan in which each vehicle in each SP can be endowed with loading information what materials it should load and which DP the vehicle should transport to. This paper designs a maximum load utilization oriented material loading algorithm to solve the above problem, and its pseudo code is shown in

Figure 3. It should be noted that \( x_{ijm} \) (including \( x_{ijm, typo} \) and \( x_{ijm, ms} \)) is another form of \( x_{ijm} \) and \( s_{ijm} \) representing the \( c^{th} \) vehicle from \( i^{th} \) SP using \( x_{ijm, typo} \) vehicle loading \( x_{ijm, ms} \) materials to \( j^{th} \) DP. \( SeqV \) (Sequence of all the vehicles for each SP) is generated by their capacity and volume in descending order by the way of fast non-dominated sort (Deb et al., 2002). Step7 to Step9 is the key progress which can make better use of capacity and volume of each vehicle. Considering the length of the article, the detailed process or pseudo code of the Add materials, Replace the material and Replace the vehicle will not be described in this paper.
Algorithm 2 The second stage of TSPBD algorithm

Input: $x_{ijm}, q_{ij}, load_i, v, m, v_m, v_i \forall i \in SP, j \in DP, m \in MS, v \in VS$

Output: $x_{ij}$: Quantity of materials $m$ transported from $i$th SP to $j$th DP by $c$th vehicle (including $x_{ij}^{type}$ and $x_{ij}^{ms}$)

$c$: Total number of vehicles been loaded (used) or mark of infeasible solution

Begin
For each $i \in SP$
Step 1: $SeqV \leftarrow $ Fast Non-dominated Sort ($q_{ij}, load_i, v$) ; %generate the vehicle sequence

For each $v \in SP$
While $x_{ijm} \neq 0 \forall m \in MS$
Step 3: $Mstay \leftarrow $ Arrange the material sequence randomly
Step 4: $c = c + 1$; %Arrange the vehicle $Vstay(1)$ to load
Step 5: if $c > \sum q_{ij}$ then $c = -1$ and go to End %marked as infeasible solution
Step 6: $[x_{ij}^{type}, x_{ij}^{ms}] \leftarrow $ Loading Process; %Load the materials into vehicle $Vstay(1)$ to its load and capacity limit as $Mstay$’s order;
Update the loaded vehicles set and material set $Vstay, Mstay, Mload$
Step 7: Add materials the vehicle can load as added from $Mstay$ set out of order;
Update $Mstay$ set and $Mload$ set
Step 8: Replace the material with larger mass or volume;
Update $Mstay$ set and $Mload$ set
Step 9: Replace the vehicle with smaller volume or capacity;
Update $x_{ij}^{type}$
Step 10: if $Mstay \neq []$ then Go To Step 4
End

Figure 3. Pseudo code of the second stage of TSPBD algorithm

Algorithm 3 Optimal candidate order for each point

Input: $t_i, t_j \forall i \in SP, j \in DP$

Output: $SeqS, SeqD$

Begin
$c = 1; SeqS = 0; SeqD = 0$ %initialize
while $c \leq N_i + N_j + 1$
   if $c \leq N_i$ then $i_j = t_{ij}, r_j = r_{ij} \forall j \in DP$
   $SeqS_c \leftarrow $ Fast Non-dominated Sort($i, r$) \forall $j \in DP$
   else $i_j = t_{ij}, r_j = risk_i \forall i \in SP$
   $SeqD_c \leftarrow $ Fast Non-dominated Sort($i, r$) \forall $i \in SP$
   $c = c + 1$
End

Figure 4. Pseudo code of optimal candidate order for each point
4.3 Crossover operator

Figure 5 is a sample graph of crossover operator between two selected parents. In the process of step 2, the offspring become illegal because some genes appear twice in one chromosome (as underlined marked in step 3). It can be inferred that the chromosomes ch1 can be mended by replacing illegal genes in n2 with illegal genes in m2, so as ch2. In this study, to keep their priority better, the replacing genes has the same relative order as their first substring (m1 or n1).

\[
\begin{align*}
\text{Step 1:} & \quad \begin{array}{c}
\text{ch}_1 \quad \text{Select a cut-point (p=3): Get substrings m}_1, m_2, n_1, n_2 \\
\text{ch}_2
\end{array} \\
\text{Step 2:} & \quad \begin{array}{c}
\text{ch}_1 \quad \text{Exchange substrings between parents, then mark the illegal genes which appear twice in the same temp_offspring and put them in array c1 and c2 according to their order in m}_1 \text{ or } n_1. \\
\text{ch}_2
\end{array} \\
\text{Step 3:} & \quad \begin{array}{c}
\text{ch}_1 \quad \text{Replace the illegal genes by c1 or c2 sequentially} \\
\text{ch}_2
\end{array} \\
\text{Step 4:} & \quad \begin{array}{c}
\text{ch}_1 \quad \text{Crossover finished} \\
\text{ch}_2
\end{array}
\end{align*}
\]

Figure 5. Crossover operator in genetic algorithm implementation process

5. CASE CALCULATION AND ANALYSIS

5.1 Case Introduction

In this case, presuming after a large-scale emergency, 15DPs \(N_J = 15\) need disaster relief materials at the same time (demand amount \(d_{jm}\) and urgency \(p_{jm}\) of materials in each DP shows in Table 1). And now 7 SPs \(N_I = 7\) in total have enough relief materials (The layout of SPs and DPs is shown in Figure 7. The amount of storage and number of available vehicles shows in Table 3). Table 2 providestime required and risk of no delivery from each SP to each DP. Table 4 is the capacity and volume of all the vehicles, in addition, the unit weight and volume of each type of materials is also provided.
### Table 1

Demand amount $d_{jm}$ and urgency $p_{jm}$ of materials in each DP

<table>
<thead>
<tr>
<th>$m$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>$d$</td>
<td>$p$</td>
<td>$d$</td>
<td>$p$</td>
<td>$d$</td>
<td>$p$</td>
<td>$d$</td>
<td>$p$</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>2.5</td>
<td>17</td>
<td>1.9</td>
<td>17</td>
<td>1.2</td>
<td>0.9</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1.6</td>
<td>8</td>
<td>2.1</td>
<td>8</td>
<td>1.9</td>
<td>0</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>2.1</td>
<td>0</td>
<td>1.8</td>
<td>6</td>
<td>2.4</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>2.5</td>
<td>0</td>
<td>2.2</td>
<td>0</td>
<td>1.6</td>
<td>7</td>
<td>2.4</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>2.5</td>
<td>30</td>
<td>1.6</td>
<td>17</td>
<td>1.5</td>
<td>13</td>
<td>2.1</td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>1.8</td>
<td>0</td>
<td>2.1</td>
<td>11</td>
<td>1</td>
<td>13</td>
<td>1.8</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>2.5</td>
<td>9</td>
<td>1.5</td>
<td>13</td>
<td>2.4</td>
<td>10</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>2.1</td>
<td>2</td>
<td>2.1</td>
<td>17</td>
<td>1.3</td>
<td>11</td>
<td>2.1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1.8</td>
<td>19</td>
<td>2.5</td>
<td>0</td>
<td>2.4</td>
<td>43</td>
<td>1.6</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>2.1</td>
<td>0</td>
<td>2.5</td>
<td>11</td>
<td>1</td>
<td>35</td>
<td>2.1</td>
</tr>
<tr>
<td>11</td>
<td>27</td>
<td>1.9</td>
<td>43</td>
<td>2.4</td>
<td>27</td>
<td>2.2</td>
<td>29</td>
<td>2.2</td>
</tr>
<tr>
<td>12</td>
<td>35</td>
<td>1.3</td>
<td>10</td>
<td>2.4</td>
<td>25</td>
<td>2.2</td>
<td>11</td>
<td>1.8</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
<td>1.6</td>
<td>0</td>
<td>1.5</td>
<td>0</td>
<td>1.2</td>
<td>13</td>
<td>1.5</td>
</tr>
<tr>
<td>14</td>
<td>52</td>
<td>1.9</td>
<td>18</td>
<td>2.5</td>
<td>0</td>
<td>2.1</td>
<td>0</td>
<td>1.2</td>
</tr>
<tr>
<td>15</td>
<td>47</td>
<td>2.2</td>
<td>29</td>
<td>2.4</td>
<td>0</td>
<td>2.2</td>
<td>69</td>
<td>2.1</td>
</tr>
</tbody>
</table>

### Table 2

Time required and risk of no delivery from each SP to each DP

<table>
<thead>
<tr>
<th>$i$</th>
<th>$j$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
<th>$t$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9.4</td>
<td>3.8</td>
<td>8.6</td>
<td>14.1</td>
<td>10.2</td>
<td>13.2</td>
<td>13.5</td>
<td>8.2</td>
<td>3.1</td>
<td>2.2</td>
<td>9</td>
<td>15.1</td>
<td>5.8</td>
<td>9.8</td>
<td>9.8</td>
<td>15.7</td>
</tr>
<tr>
<td>2</td>
<td>5.3</td>
<td>0.9</td>
<td>3.2</td>
<td>3.5</td>
<td>10.6</td>
<td>7.2</td>
<td>7.9</td>
<td>10.1</td>
<td>5.5</td>
<td>2.6</td>
<td>7.4</td>
<td>6.2</td>
<td>9.7</td>
<td>1.9</td>
<td>5</td>
<td>10.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10.3</td>
<td>0.4</td>
<td>3.2</td>
<td>3.5</td>
<td>10.6</td>
<td>7.2</td>
<td>7.9</td>
<td>10.1</td>
<td>5.5</td>
<td>2.6</td>
<td>7.4</td>
<td>6.2</td>
<td>9.7</td>
<td>1.9</td>
<td>5</td>
<td>10.5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>10.3</td>
<td>7.3</td>
<td>8.1</td>
<td>7.8</td>
<td>3.9</td>
<td>10.9</td>
<td>15.1</td>
<td>2.1</td>
<td>4.1</td>
<td>7.2</td>
<td>11.4</td>
<td>11.5</td>
<td>7.1</td>
<td>5.6</td>
<td>14.1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12.3</td>
<td>0.9</td>
<td>7.3</td>
<td>8.1</td>
<td>7.8</td>
<td>3.9</td>
<td>10.9</td>
<td>15.1</td>
<td>2.1</td>
<td>4.1</td>
<td>7.2</td>
<td>11.4</td>
<td>11.5</td>
<td>7.1</td>
<td>5.6</td>
<td>14.1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2.1</td>
<td>12.3</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
<td>9.8</td>
</tr>
<tr>
<td>7</td>
<td>0.2</td>
<td>8.4</td>
<td>8.1</td>
<td>6.5</td>
<td>17.7</td>
<td>15.2</td>
<td>8.9</td>
<td>1.9</td>
<td>14</td>
<td>10.8</td>
<td>14</td>
<td>2.8</td>
<td>11.8</td>
<td>7.1</td>
<td>11.5</td>
<td>8.4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.5</td>
<td>10.3</td>
<td>7.3</td>
<td>8.1</td>
<td>7.8</td>
<td>3.9</td>
<td>10.9</td>
<td>15.1</td>
<td>2.1</td>
<td>4.1</td>
<td>7.2</td>
<td>11.4</td>
<td>11.5</td>
<td>7.1</td>
<td>5.6</td>
<td>14.1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>7.6</td>
<td>0.5</td>
<td>0.8</td>
<td>0.9</td>
<td>0.5</td>
<td>0.7</td>
<td>0.2</td>
<td>0.1</td>
<td>0.5</td>
<td>0.8</td>
<td>0.1</td>
<td>0.7</td>
<td>0.8</td>
<td>0.5</td>
<td>0.8</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>10</td>
<td>8.8</td>
<td>7.6</td>
<td>11</td>
<td>6.8</td>
<td>8.7</td>
<td>8.4</td>
<td>1.9</td>
<td>10.4</td>
<td>8.7</td>
<td>10.7</td>
<td>15.6</td>
<td>9.9</td>
<td>1.4</td>
<td>8.7</td>
<td>4.8</td>
<td>4.9</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.3</td>
<td>7.6</td>
<td>11</td>
<td>6.8</td>
<td>8.7</td>
<td>8.4</td>
<td>1.9</td>
<td>10.4</td>
<td>8.7</td>
<td>10.7</td>
<td>15.6</td>
<td>9.9</td>
<td>1.4</td>
<td>8.7</td>
<td>4.8</td>
<td>4.9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11.2</td>
<td>0.3</td>
<td>0.7</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
<td>0.8</td>
<td>0.1</td>
<td>0.5</td>
<td>0.1</td>
<td>0.1</td>
<td>0.9</td>
<td>0.6</td>
<td>0.5</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>13</td>
<td>0.5</td>
<td>11.2</td>
<td>9.7</td>
<td>8.8</td>
<td>10.1</td>
<td>5.2</td>
<td>13.4</td>
<td>11</td>
<td>14</td>
<td>18.9</td>
<td>13.5</td>
<td>2.3</td>
<td>12.3</td>
<td>7.5</td>
<td>6.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.5</td>
<td>11.2</td>
<td>9.7</td>
<td>8.8</td>
<td>10.1</td>
<td>5.2</td>
<td>13.4</td>
<td>11</td>
<td>14</td>
<td>18.9</td>
<td>13.5</td>
<td>2.3</td>
<td>12.3</td>
<td>7.5</td>
<td>6.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.5</td>
<td>11.2</td>
<td>9.7</td>
<td>8.8</td>
<td>10.1</td>
<td>5.2</td>
<td>13.4</td>
<td>11</td>
<td>14</td>
<td>18.9</td>
<td>13.5</td>
<td>2.3</td>
<td>12.3</td>
<td>7.5</td>
<td>6.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3 Storagematerials st and available vehicles qvateach SP

<table>
<thead>
<tr>
<th>i</th>
<th>st</th>
<th>gq</th>
<th>st</th>
<th>gq</th>
<th>st</th>
<th>gq</th>
<th>st</th>
<th>gq</th>
<th>st</th>
<th>gq</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>51</td>
<td>4</td>
<td>28</td>
<td>6</td>
<td>20</td>
<td>4</td>
<td>30</td>
<td>0</td>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>5</td>
<td>30</td>
<td>7</td>
<td>89</td>
<td>3</td>
<td>94</td>
<td>5</td>
<td>97</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>98</td>
<td>6</td>
<td>78</td>
<td>8</td>
<td>90</td>
<td>5</td>
<td>82</td>
<td>0</td>
<td>80</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>70</td>
<td>2</td>
<td>40</td>
<td>6</td>
<td>27</td>
<td>4</td>
<td>38</td>
<td>6</td>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>63</td>
<td>2</td>
<td>36</td>
<td>0</td>
<td>23</td>
<td>4</td>
<td>50</td>
<td>0</td>
<td>38</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>86</td>
<td>3</td>
<td>90</td>
<td>5</td>
<td>60</td>
<td>7</td>
<td>72</td>
<td>6</td>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>43</td>
<td>5</td>
<td>29</td>
<td>2</td>
<td>38</td>
<td>0</td>
<td>20</td>
<td>4</td>
<td>38</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4 Capacity and volume of vehicles & unit weight and volume of materials

<table>
<thead>
<tr>
<th>Vehicle Type</th>
<th>Capacity (Kg)</th>
<th>Volume (m³)</th>
<th>Materials Type</th>
<th>Unit Weight (Kg)</th>
<th>Unit Volume (m³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7000</td>
<td>42</td>
<td>1</td>
<td>90</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>8000</td>
<td>40</td>
<td>2</td>
<td>230</td>
<td>1.35</td>
</tr>
<tr>
<td>3</td>
<td>4000</td>
<td>25</td>
<td>3</td>
<td>130</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>6000</td>
<td>28</td>
<td>4</td>
<td>120</td>
<td>1.3</td>
</tr>
<tr>
<td>5</td>
<td>5000</td>
<td>22</td>
<td>5</td>
<td>160</td>
<td>1.1</td>
</tr>
<tr>
<td>6</td>
<td>7000</td>
<td>38</td>
<td>6</td>
<td>100</td>
<td>1.2</td>
</tr>
<tr>
<td>7</td>
<td>3000</td>
<td>18</td>
<td>7</td>
<td>220</td>
<td>0.9</td>
</tr>
<tr>
<td>8</td>
<td>3400</td>
<td>20</td>
<td>8</td>
<td>280</td>
<td>1.4</td>
</tr>
</tbody>
</table>

5.2 Result and Analysis

To verify the efficacy of the proposed two-stage priority-based MOGA algorithm, we run it with the parameters provided in Section 5.1 by MATLAB 2015b on a PC installed with WIN7 (64bit), 4-core processor (2.33GHz), and a memory with 4GB. Except for the key operators of the algorithm presented in Section 4, other parameters are listed in Table 5.

Table 5 Parameters used in two-stage priority based MOGA

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>100</td>
<td>Selection</td>
<td>Binary tournament selection</td>
</tr>
<tr>
<td>Generation</td>
<td>90</td>
<td>Mutation</td>
<td>Swap 2 genes of the chromosome randomly</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.25</td>
<td>Ranking</td>
<td>Fast Non-dominated Sorting(Deb et al., 2002)</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.7</td>
<td>Fitness function</td>
<td>$f_1=1/F_i, f_2=1/F_i, f_3=1/F_i$</td>
</tr>
</tbody>
</table>

After 20 runs of the algorithm, average total time consumed to generate the result is 615.8 seconds, and consumption time of loading algorithm (described in Section 4.2.2) accounted for 87% because of the complexity and the high number to be called of the this operator. Figure 6 is the Pareto front graph of three generations. It can be found that the algorithm has high evolution efficiency, and maintains good population diversity. As the iteration proceeds, the Pareto front is converging to the optimal value rapidly.
One of the Pareto front individuals has been chosen as output (with the objectives are $F_1 = 0.76$, $F_2 = 4.28$, and $F_3 = 792.91$) and Figure 7 shows its materials allocation relationships between SPs and DPs (The numbers in this figure is the serial numbers of SPs or DPs, and location of these points represents their geographical position and required traveling time to some extent). By analysing this allocation relationship diagram and the risk of no delivery (in Table 2), it can be found that the nearest SP is not always their best choice for materials provisioning, because risk is another important consideration except the required traveling time in the allocation process.

The 64 detailed materials allocation and loading plans with 64 vehicles in each SP is listed in Table 6 which is also the output variables $x_{ijmvh}$ of this case (considering the length of this paper, it only lists part of the plan).
**Table 6** Detailed materials allocation and loading plan (partial)

<table>
<thead>
<tr>
<th>No.</th>
<th>i</th>
<th>j</th>
<th>m</th>
<th>v</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>64</td>
<td>7</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 7.** Diagram of allocation relationship between SPs and DPs

**6. CONCLUSION**

To improve the efficiency of relief materials supply and utilization of relief resources in large-scale relief operations, this paper sets up a multi-objective model to solve the allocation and loading problem of multiple demand points, multiple supply points, multiple categories of materials and multiple types of vehicles. A two-stage priority based MOGA algorithm has been designed and is proven to have a fast speed, reliable result and high practicability. Moreover, this model and its solution algorithm can provide guidance for other material distribution problems. Deficiently, the loading algorithm takes a large proportion of the program running time. In further study, self-adaptive call of the loading algorithm would be put out.

**7. REFERENCES**


genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.