Bi-Objective Optimization Genetic Algorithm for Locating Concrete Mixing Plants

Shujin Ye$^1$, Han Huang$^2$,$^*$, Changjian Xu$^2$, Liang Lv$^2$, and Yihui Liang$^2$

$^1$Hong Kong Baptist University, Hong Kong, China
$^2$South China University of Technology, Guangzhou 510006, Guangdong, China

The location of the concrete mixing plants (LCMP) is a kind of combinatorial optimization problem. The reasonable locations will significantly improve the quality and reduce costs of the construction companies. LCMP has two objectives. One is to optimize the number of mixing plants, and the other is to optimize the distance between the mixing plants and the construction areas. In this paper, a location model based on concrete mixing plants has been built. A hybrid algorithm combining genetic algorithm (GA) and local search strategy has been applied to solving the proposed problem. We designed the implementation steps of the hybrid algorithm according to the objectives and constrains of LCMP, including the coding of the solution, the cross over operation, the mutation operation and the local search strategy. The simulation experiment shows that our proposed algorithm achieves better solution than greedy algorithm and GA.

Keywords: Location of the Concrete Mixing Plants (LCMP), Genetic Algorithm, Local Search.

1. INTRODUCTION

Location of the concrete mixing plants (LCMP) plays a significant role in distribution, which helps construction companies to control their cost and quality. The locations of mixing plants will influence not only the quality of the concrete but also the costs of constructions. In many underdeveloped countries such as China, the output of the existing mixing plants could not meet the rapidly growing need. For example, the concrete requirement of Nanning (a city in China) will rise to 14 million m$^3$/year, which requires the supplement to reach at least 19.5 million m$^3$/year in 2015. However, the fact that the output of mixing plants just provide 13.8 million m$^3$/year supplement shows that the supplying amount is far less than the requirement. To achieve the new and increasing requirement of urban construction, additional mixing plants which should be located in a suitable location are needed to increase the supplement.

At present, the research on LCMP is less comparatively in academic realm. There are some similar researches: Authors in Ref. [1] proposed a reliable location routing problem (RLRP), in which sites are disrupted randomly. Then Ref. [1] built up a linear program to describe RLRP and proposed a metaheuristic algorithm for solving RLRP. Authors in Ref. [2] researched on the fuzzy-random-environment based location allocation problem, and proposed three different fuzzy random programming models. The hybrid intelligent algorithm based on genetic algorithm was used to solve the proposed models. Authors in Ref. [3] proposed another location allocation problem that based on continuous multi-facility (MFLP). And three general heuristics were proposed to solve MFLP. Authors in Refs. [4, 9] considered a new inventory location routing problem (ILRP), in which the sites and vehicles are capacitated. A hybrid metaheuristic has been designed to solve ILRP. The experimental results show the proposed algorithm is effective. Authors in Ref. [5] proposed a transportation location routing problem (TLRP). There are two objectives of TLRP. One is to reduce the total costs, and another is to balance the workload. A mathematical model based on priorities was designed to solve TLRP.

Traditional location problem belongs to one-to-many mode, with one central site serving many service centers and a service center building up connection with only one central site. One-to-many is the key mode of distribution location problems and the location of first-aid center problems. They are considered as the traditional location problems and belong to the integer programming problem. These draw several researchers, interest in modeling and developing an algorithm to solve $p$-median problem. However, LCMP is very different from traditional one, its many-to-many model. In LCMP problem, one central site can serve many service site; also at the same time, one service site can accept the service from many central sites.

$^*$Author to whom correspondence should be addressed.
The major task to solve LCMP problems is to determine the proportion from 0 to 1, which reflects the supplement from central site to service site. The solution to the traditional one is not suitable for LCMP, because it could not satisfy the requirement of LMS.

Another essential difficulty is the selection of the candidate mixing plants. Especially, as the size of the candidate mixing plants increases, it’s difficult to enumerate and analyze every plant. To solve this problem, many heuristic algorithms such as ant colony algorithm, particle swarm optimization, genetic algorithm and differential evolution can be used to overcome the difficulties.

In order to solve the LCMP, we proposed that a local search should be added into the traditional genetic algorithm, which is first provided by John Holland. The rest of this paper is organized as follow: the problem definitions and formulation will be presented in Section 2. Genetic algorithm with local search will be introduced in details in Section 3. The experimental analysis will be discussed in Section 4. Finally, the conclusion will be drawn in the last section.

2. PROBLEM DEFINITIONS AND FORMULATION

2.1. Problem Analysis

Different from the classic location problems, LCMP is a many-to-many problem with two objectives. Therefore, we take the location of concrete mixing plants in Nanning as the research object. To shorten the route of supplying transportation, several new mixing plants will be built in Nanning. However, as we discussed in the last section, LCMP is different with p-median distribution problems. The features of LCMP are listed as follow:

(1) Bi-objective Programming. There are two objectives of LCMP problem. One is to minimize the total weighed length of the concrete mixing plants, and the other is to minimize the number of new added plants. The first objective is similar to the traditional p-median problem. If this objective is achieved in LCMP problem, a good solution could shorten the route of transporting concrete, alleviate the traffic jam and reduce vehicle emissions. After realizing the latter goal, not only we preserve land resources, but also avoid the vicious competition.

(2) The limits on the transportation length. Given its chemical features, the concrete cannot be sent from afar.

(3) The limit on the supplying ability. In common p-median problem, there is no limit for supplying ability while the supplying capability of the concrete plant in LCMP is limited. Virtually, the actual supplement of each concrete mixing plant should not be too large, otherwise it would cause the diseconomy of scale. In addition, a large number of vehicles in and out of the city can aggravate the traffic jam. As a result, the supplying ability of each concrete mixing plant can not be expanded indefinitely.

(4) The uncertain number of adding plants. In LCMP, the number of adding plants is uncertain because the longest transporting length is limited by the number of new adding plants. Also, it is determined and limited by the gap between supply and demand and the distribution of the original sites, candidate plants, as well as the construction group. Therefore, we cannot decide the sites of plants only according to the gap between supply and demand.

(5) The relationship between groups and the concrete mixing plants is many-to-many mode. One group can receive service from many concrete mixing plants and each mixing plant can serve many groups. In this way, the efficiency of the mix station can be improved. However, it increases the complexity of the modeling process.

(6) The already existing plants must take part in the supplying service. To meet the requirement of exploiting the supplying ability as much as possible, the address of new adding concrete mixing plants should be carefully considered with the original plants.

2.2. Symbol Description

In this subsection, we proposed the definition and description of the sets, parameters, and deciding variables.

(i) Set. Let \( K = \{1, 2, \ldots, k\} \) be the set of construction group; \( I = \{1, 2, \ldots, i\} \) be the set of existing plant; \( J = \{i+1, i+2, \ldots, J\} \) be the set of candidate plants.

(ii) Parameters. \( d_{ik} \) denotes the required supplying ability for group \( k \); \( d_{jk} \) denotes the distance from plant \( i \) to the construction group \( k \); \( t_d \) denotes the upper bounds of the limited transporting distance (in this paper, \( t_d = 25 \) km); \( news_j \) represents the supplying capability of candidate plant \( j \), and the supplying ability of all candidate plants is 500000 m³/year; \( olds_j \) represents the existing plant \( j \)'s concrete supplying capability; \( c_i \) denotes the supply of plant \( i \) to group \( j \).

(iii) Decision variables. In this model, a variable \( sel_j \) is designed to mark the selection of each candidate plant. If the \( j \) plant is selected, the value of \( sel_j \) is 1. Otherwise, the value of it is 0.

2.3. Modeling

According to the definition of symbols and variables above, the objectives of this model are as follows:

\[
\begin{align*}
\min(Z_1) &= \sum_{i,k} \left( \sum_{j,l} c_{ik} \cdot d_{ij} + \sum_{j,l} c_{jk} \cdot d_{jk} \cdot sel_j \right) \\
\min(Z_2) &= \sum_{j,l} sel_j
\end{align*}
\] (1)

constraints are as follow:

\[
\begin{align*}
\sum_{i,l} c_{ik} + \sum_{j,l} c_{jk} \cdot sel_j & \geq dem_k \quad \forall k \in K \\
\sum_{i,k} c_{ik} \cdot sel_j & \leq news_j \quad \forall j \in J \\
\sum_{i,k} c_{ik} & \leq olds_i \quad \forall i \in I
\end{align*}
\] (3) (4) (5)

\[J.\text{Comput.\ Theor.\ Nanosci.}\ 13, 1–7, 2016\]
problems because of its NP-hard feature. At the same time, makes LCMP more difficult than the existing plants. Therefore \( p \)-median belongs to \( n \)-median problem.

Although the hybrid integer programming model above belongs to \( p \)-median problem, the number of newly added plants is uncertain. Further, the fact that every construction group could accept service from several mixing plants and every mixing plant could serve several groups at the same time, makes LCMP more difficult than the \( p \)-median problems because of its NP-hard feature.

Genetic algorithm,\(^{13}\) a widely adopted method for solving real-world engineering problems, has an adaptable fast convergence speed, but the solution accuracy is not good enough to solve LCMP problem. To address LCMP and to improve the accuracy of solution, we add a local search to genetic algorithm to improve the quality of solution and to make good use of genetic algorithm feature.

Equations (1) and (2) are the target evaluation functions which require that the solution should find a shortest transportation distance, and at the same time require as less new plants as possible. Here Eqs. (2) to (7) are the constraint conditions. The requirements which are represented by the equations are as follow: Eq. (3) shows that every group should have enough supplying ability; Eqs. (4) and (5) are the limits on each plants’ supplying ability. Moreover, Eq. (6) guarantees that all existing plants have been selected. Finally, Eqs. (7) and (8) represent that the transportation distance, and at the same time require as less new plants as possible.

The process of genetic algorithm with local search (GALS) is presented in Algorithm 1.

\[ 
\begin{align*}
\sum_{k \in K} c_{ik} & \geq 0 \quad \forall i \in I \\
d_{ik} & \leq td \quad \forall i \in I, k \in K \\
d_{jk} & \leq td \quad \forall j \in J, k \in K 
\end{align*}
\]

\[ 
\begin{align*}
\sum_{k \in K} c_{ik} & > 0 \quad \forall i \in I \\
d_{ik} & \leq td \quad \forall i \in I, k \in K \\
d_{jk} & \leq td \quad \forall j \in J, k \in K 
\end{align*}
\]

3. THE IMPROVED GENETIC ALGORITHM

Although the hybrid integer programming model above belongs to \( p \)-median problem, the number of newly added plants is uncertain. Further, the fact that every construction group could accept service from several mixing plants and every mixing plant could serve several groups at the same time, makes LCMP more difficult than the \( p \)-median problems because of its NP-hard feature.

Genetic algorithm,\(^{13}\) a widely adopted method for solving real-world engineering problems, has an adaptable fast convergence speed, but the solution accuracy is not good enough to solve LCMP problem. To address LCMP and to improve the accuracy of solution, we add a local search to genetic algorithm to improve the quality of solution and to make good use of genetic algorithm feature.

\[ 
\begin{bmatrix}
6 & 6 & 4 & 0 & 0 \\
0 & 0 & 2 & 4 & 6 \\
2 & 3 & 4 & 6 & 5 \\
0 & 0 & 0 & 2 & 0
\end{bmatrix}
\]

3.1. Encoding

In this paper a matrix with \( k \) rows and \( m \) columns is adopted for each individual, where \( m \) is the input number of plants and \( k \) represents the group number. According to Eq. (6), each individual must contain all of existing plants. Therefore \( m > |I| \) where \( |I| \) is the number of existing plants. The encoding of individual \( x_i \) is as follow:

\[ 
x_i = 
\begin{bmatrix}
  c_{r_1, 1} & c_{r_2, 1} & \cdots & c_{r_m, 1} \\
  c_{r_1, 2} & c_{r_2, 2} & \cdots & c_{r_m, 2} \\
  \vdots & \vdots & \ddots & \vdots \\
  c_{r_1, k_1} & c_{r_2, k_1} & \cdots & c_{r_m, k_1}
\end{bmatrix}
\]

where \( r_j \) is the index of the plant, and \( r_1, r_2, \ldots, r_m \) is the non-repeat random integer sequence from 1 to \( i + j \). \( c_{i, k} \) denotes the amount of supply from plant \( r_m \) to group \( k \). Although LCMP is a two-objective problem, we noticed that the number of candidate plants are limited (in the simulation experiments \(|J| = 20\)). Therefore we ran our proposed algorithm with all possible values of \( m \) as the input to find the non-dominated solutions.

3.2. Initialization for the Population

Each individual in the population is a solution to the LCMP problem. Every individual containing the plants should thus satisfy all the constraining conditions. In addition, to maintain the diversity of solutions, a random
selection is adopted to select the plant. The details are presented as follow:

**Step 1:** Pick up \( m \) plants randomly for each individual \( x_i \), including all existing plants and several candidate plants (the number of candidate plants depends on \( m \)). If the total supplying amount of \( x_i \) is no less than the sum of all groups’ requirement, then Step 2. Otherwise, should be return to Step 1, until all individual are served.

**Step 2:** Check out all distribution amounts for the groups. If there is any groups whose requirement could not be satisfied, then randomly select one of those groups without satisfaction of requirement and allocate one site with all of its unallocated supplement to that group, until requirement of all groups have been satisfied. Otherwise, allocation ends.

The process of initialization for population is presented in Algorithm 2.

**Algorithm 2:** Initialization for population

Input: population size \( n \)

Output: initial population \( X_0 \)

population size counter \( s = 1 \)

while \( s < n \) do

  randomly select \( m \) mixing plants (include all existing plants) for individual \( x_i \)

  if \( \text{Supplying amount of } m + k \text{ mixing plants} > \text{requirement} \) then

    while not every group has been satisfied do

      Randomly allocate one plant with all its superfluous Supplying to the unsatisfied group

      end

    end

  end

  \( s = s + 1 \)

end

3.3. Selection

Roulette wheel selection is adopted as the selection operation which reflects the idea of the GALS that the higher fitness the individual holds, the larger probability for existence it has. The target function in this paper is to count the cost of transportation for concrete. The lower cost solution can be considered as a better solution for the LMS. Equation (8) is the fitness function to evaluate the quality of a solution.

\[
fit(i) = \left( \frac{g_{\text{max}} - \beta (g_i - g_{\text{max}})}{g_{\text{max}}} \right)^2
\]

where \( g_i \) is the fitness value of \( i \)-th individual while \( g_{\text{max}} \) is the minimum fitness value among population. Here, \( \alpha \) and \( \beta \) are designed to control the convergence speed.

According to the Eq. (8), the smaller \( g_i \) is, the larger \( fit(i) \) will be, meaning that an individual with higher fitness value holds a higher probability for being selected. However, variable \( \beta \) is designed to control the distance between the \( g_i \) and \( g_{\text{max}} \) because of the uncertain range of \( g_i \). For example, while \( g_i - g_{\text{min}} = 1000 \), \( \beta \) can make the range of \( g_{\text{min}} - \beta (g_i - g_{\text{min}}) \) be \((-\infty, 1]\). With adjusting the value of \( \beta \), \( g_{\text{min}} - \beta (g_i - g_{\text{min}}) \) can reflect the information of distance on the basis of custom intention.

Further, to better control the convergence speed of the GALS, variable \( \alpha \) is also added into the fitness function. Obviously, a smaller value of \( \alpha \) reflecting the diversity of selected individual is better, which can be considered that individual with long distance has a high probability to be selected. Also the convergence speed of GALS can slow down with a small \( \alpha \) value. In contrast, a high quality solution will be selected when the value of \( \alpha \) is large enough. That means the probability that individual with long distance will be selected is low and GALS has a fast convergence speed.

3.4. Crossover

Two-point crossover is adopted in this paper. It will randomly generate two crossover points in two individuals and then exchange all elements between them. After crossover, original individual considered as the parent will produce two offspring. To make sure that the offspring provided by crossover is still a qualified solution to LCMP, an improvement strategy for crossover should be added. Its detail will be presented as follow:

**Step 1:** Randomly generate two crossover points, point A and point B, for executing the crossover process and then generate offspring.

**Step 2:** Check out whether there is any reduplicated sites and replace them.

**Step 3:** Check out offspring satisfy the requirement or not. If the supplying amount of the offspring is larger than requirement it needs, adjust the allocation solution so that every group could get enough supply for achieving their requirement. Otherwise, there will be no crossover.

The adjustment for the allocation solution details are as follow:

**Remark 1.** Iterate all the groups. If the supplying amount of anyone group is larger than requirement, then the site with longest path will be reallocated until the supplying amount equals to the requirement.

**Remark 2.** Iterate all the groups. If there is any group suffering from the problem that the supplying amount is lower than the requirement, a site with shortest path for picking up from the newly allocated sites should be allocated to this group. Repeat the process until its requirement is satisfied.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Fig. 3.** Solution adjustment.
Assume that the crossover point is \( a = 2, b = 4 \), one of the off-spring is shown in Figure 4.

After crossover, the site 3 is duplicated. Then randomly select the sites among crossover point A and point B and replace one of the duplicate the sites, site 3, with it. Here site 4, site 5 and site 5 are qualified to finish this task. And the Figure 6 shows the result of the adjustment offspring. In addition, GALS will not change the sites among A to B for maintaining the gene diversity.

If the offspring can not satisfy the requirement, according to step 3, GALS adjusts the gene. One of the samples in this situation is presented in Figure 6.

3.5. Mutation
There are two mutation strategies for GASL in consideration of the LCMP feature.

3.5.1. Mutation Strategy 1
Replace a randomly selected site of the solution by the site which is not included in the individual solution, as

<table>
<thead>
<tr>
<th>Plant no.</th>
<th>Supply amount (cubic metres)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47</td>
<td>(97.87, 429.44)</td>
</tr>
<tr>
<td>2</td>
<td>76</td>
<td>(91.62, 389.48)</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>(505.34, 590.61)</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>(566.05, 530.27)</td>
</tr>
<tr>
<td>5</td>
<td>47</td>
<td>(783.74, 546.83)</td>
</tr>
<tr>
<td>6</td>
<td>66</td>
<td>(789.66, 533.82)</td>
</tr>
<tr>
<td>7</td>
<td>83</td>
<td>(858.28, 553.93)</td>
</tr>
<tr>
<td>8</td>
<td>66</td>
<td>(1088.16, 304.18)</td>
</tr>
<tr>
<td>9</td>
<td>76</td>
<td>(1066.73, 279.27)</td>
</tr>
<tr>
<td>10</td>
<td>47</td>
<td>(1157.48, 321.62)</td>
</tr>
<tr>
<td>11</td>
<td>32</td>
<td>(1072.78, 161.9)</td>
</tr>
<tr>
<td>12</td>
<td>56</td>
<td>(1031.61, 170.37)</td>
</tr>
<tr>
<td>13</td>
<td>52</td>
<td>(723.09, 204.25)</td>
</tr>
<tr>
<td>14</td>
<td>51</td>
<td>(345.57, 245.39)</td>
</tr>
<tr>
<td>15</td>
<td>37</td>
<td>(456.89, 272.01)</td>
</tr>
<tr>
<td>16</td>
<td>56</td>
<td>(488.35, 232.08)</td>
</tr>
<tr>
<td>17</td>
<td>173</td>
<td>(495.56, 204.49)</td>
</tr>
<tr>
<td>18</td>
<td>40</td>
<td>(478.27, 113.01)</td>
</tr>
<tr>
<td>19</td>
<td>47</td>
<td>(474.28, 103.88)</td>
</tr>
<tr>
<td>20</td>
<td>37</td>
<td>(503.93, 629.1)</td>
</tr>
<tr>
<td>21</td>
<td>50</td>
<td>(477.96, 95.54)</td>
</tr>
<tr>
<td>22</td>
<td>50</td>
<td>(615.49, 20.25)</td>
</tr>
<tr>
<td>23</td>
<td>50</td>
<td>(623.53, 12.22)</td>
</tr>
</tbody>
</table>

Figure 7 shown. To make sure that the mutated individual is still the qualified solution for the LCMP, a check should be executed of confirm that the requirement of the group is satisfied. Otherwise, no mutation will be executed.

3.5.2. Mutation Strategy 2
Randomly transform the path between group as Figure 8 shown.
Step 1: Randomly pick two path between site and group, \( l_{i, m} \) and \( l_{i, n} \), where \( i \) and \( f \) mean the site of mixing plant

\[
\begin{array}{|c|c|c|}
\hline
\text{Plant no.} & \text{Supply amount (cubic metres)} & \text{Location} \\
\hline
24 & 50 & (541.15, 25.95) \\
25 & 50 & (897.09, 51.13) \\
26 & 50 & (1135.02, 105.08) \\
27 & 50 & (280.14, 176.91) \\
28 & 50 & (921.64, 111.44) \\
29 & 50 & (1194.34, 401.34) \\
30 & 50 & (411.62, 627.07) \\
31 & 50 & (966.26, 606.32) \\
32 & 50 & (1196.33, 340.29) \\
33 & 50 & (606.32, 629.57) \\
34 & 50 & (710.24, 27.83) \\
35 & 50 & (1062.58, 63.84) \\
36 & 50 & (886.18, 519.49) \\
37 & 50 & (341.92, 111.18) \\
38 & 50 & (236.62, 429.49) \\
39 & 50 & (1174.19, 452.85) \\
40 & 50 & (1083.1, 276.17) \\
41 & 50 & (429.53, 23.57) \\
42 & 50 & (841.28, 16.27) \\
43 & 50 & (772.66, 20.65) \\
\hline
\end{array}
\]
### Table III. The concrete requirement and location of the construction groups.

<table>
<thead>
<tr>
<th>Group no.</th>
<th>Supply amount (cubic metres)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>112 (995.36, 215.66)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>700 (715.39, 129.56)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>168 (402.93, 219.25)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>98  (594.75, 341.93)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>140 (331.34, 447.38)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>140 (529.76, 521.88)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>462 (786.45, 357.31)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>140 (1038.13, 341.84)</td>
<td></td>
</tr>
</tbody>
</table>

### Table IV. Parameter setting.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Number of iteration</td>
<td>200</td>
</tr>
<tr>
<td>Number of local search</td>
<td>10</td>
</tr>
<tr>
<td>Threshold for local search</td>
<td>3.56e+05</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.02</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>2</td>
</tr>
<tr>
<td>$\beta$</td>
<td>3.5</td>
</tr>
<tr>
<td>Max number of sites</td>
<td>35</td>
</tr>
</tbody>
</table>

while $n$ and $m$ mean group. Assume that the distance of $l_{i,m}$ is shorter than $l_{i,n}$.

Step 2: Replace $l_{i,m}$ with $l_{i,n}$ and offer the amount of supply to them.

Step 3: Add a path $l_{i,n}$, adjust the supply amount of site $j$ to achievement the requirement of all groups.

### 3.6. Local Search

The local search strategy has the same component of mutation strategy 2. However, if the fitness value is less than one of original solutions after changing the path, the local search will be adopted. Hence, the local search is adopted conditionally.

### 4. EXPERIMENTAL ANALYSIS

#### 4.1. Experimental data

We adopt the GALS to solve the LCMP problem of Nanning city for meeting the demand of concrete in 2015. The experimental data contains 8 construction groups, 23 existing plants and 20 candidate plants. The supply capacity and the location of the existing plants are as shown in Table I; the supply capacity and the location of the candidate plants are as shown in Table II; The requirement and the location of the construction groups are as shown in Table III.

The existing plants’ supply amount cannot satisfy the requirement of all the construction groups. We have to add some candidate plants to meet the need of groups. We also need to minimize the number of adding plants and total distance between groups and plants, to reduce costs and improve quality of supply.

#### 4.2. Parameter Setting

The concrete requirement of Nanning city is adopted as the research object and the experiment is developed to simulate the programming of locating mixing plant to meet the concrete demand. For simulation, the parameter setting is introduced in Table I as above. Population size is set following the general GA algorithm setting. We limit the iteration on 200 and local search 10 in order to control the time consuming. The threshold for local search is set on value of 3.56e+05, which means the local search will be performed conditionally.

### Table V. Allocating amount of Figure 6.

<table>
<thead>
<tr>
<th>Group no.</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>Plant no. (supply amount)</td>
<td>8 (23)</td>
<td>12 (1)</td>
<td>2 (1)</td>
<td>4 (1)</td>
<td>1 (47)</td>
<td>3 (89)</td>
<td>3 (1)</td>
<td>8 (28)</td>
</tr>
<tr>
<td></td>
<td>9 (2)</td>
<td>13 (50)</td>
<td>14 (51)</td>
<td>17 (97)</td>
<td>2 (55)</td>
<td>4 (12)</td>
<td>4 (38)</td>
<td>9 (74)</td>
</tr>
<tr>
<td></td>
<td>10 (5)</td>
<td>16 (39)</td>
<td>15 (37)</td>
<td>30 (38)</td>
<td>20 (35)</td>
<td>5 (47)</td>
<td>10 (38)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 (32)</td>
<td>17 (63)</td>
<td>16 (16)</td>
<td>30 (38)</td>
<td>6 (66)</td>
<td>10 (38)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 (50)</td>
<td>18 (40)</td>
<td>17 (13)</td>
<td>37 (50)</td>
<td>8 (15)</td>
<td>10 (4)</td>
<td>12 (5)</td>
<td>13 (2)</td>
</tr>
<tr>
<td></td>
<td>19 (47)</td>
<td>21 (50)</td>
<td>22 (50)</td>
<td>23 (50)</td>
<td>24 (50)</td>
<td>25 (50)</td>
<td>28 (4)</td>
<td>35 (36)</td>
</tr>
<tr>
<td></td>
<td>34 (50)</td>
<td>42 (50)</td>
<td>43 (50)</td>
<td>30 (8)</td>
<td>33 (50)</td>
<td>36 (50)</td>
<td>31 (50)</td>
<td></td>
</tr>
</tbody>
</table>
In addition, GALS is implemented in Matlab 2014a. The experiment is conducted on operation system Windows 8.1 with 6.0 GB RAM and i5-2350 CPU.

### 4.3 Simulation Result
Tables II and III indicate that the gap between demand and supply of concrete is 580 cubic metres. Therefore $m$ must be no less than 35. We used every possible $m$ value ($m \in [35, 43]$) as the input to find the non-dominated solutions. Simulation result shows that when $m = 35$, the total distance traveled also reaches the minimum. Therefore, we set $m = 35$ and compared the performance of GALS with GA. Table V presents the best result and the average result of GALS which outperform the ones of original GA. Furthermore, Table VI introduces a solution detail presented in Figure 6 regarding allocating amount. The results of the solution presented in Figure 9 is solution of $3.46e+5$ m³/year.

The result is not invariant due to the random feature of crossover and mutation adopted by GA. However, according to the result, GALS can achieve average solution of $3.52e+05$ m³/year showing that the GALS has a greater stability.

From Table V, we can find out that the result of original genetic algorithm is higher than GALS. The result shows that with the help of specific encoding scheme, crossover, mutation strategy and local search, the GALS outperforms original GA.

### 5. CONCLUSION
We have adopted a new idea to study the LMS problem and provide an improved genetic algorithm with local search to solve the LMS problem in addition to finding out a good enough solution to meeting the requirement for Nanning city in 2015. Furthermore, the GALS overcomes the accuracy problem which the original algorithm suffers from.

For future work, we hope to control the convergence speed of GA algorithm so that the GALS can have a better convergence speed, so that it can find out a better solution and prevent itself from losing performance when premature convergence occurs.

**Acknowledgment:** This work is supported by National Natural Science Foundation of China (61170193, 61370102, 61370185), Guangdong Natural Science Foundation (2014A030306050, S2013010013432), the Fundamental Research Funds for the Central Universities, SCUT(2015PT022).

**References**


Received: 24 June 2015. Accepted: 30 September 2015.