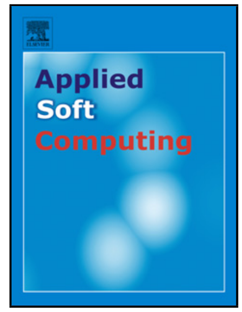

Title: Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm

Authors: Qiang Zhang, Shengwu Xiong

PII: S1568-4946(18)30438-1
DOI: https://doi.org/10.1016/j.asoc.2018.07.050
Reference: ASOC 5018

To appear in: *Applied Soft Computing*

Received date: 13-9-2017
Revised date: 24-7-2018
Accepted date: 25-7-2018

Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm

Qiang Zhang[1, 2] Shengwu Xiong[1]

[1] School of Computer Science and Technology, Wuhan University of Technology, Hubei Wuhan, 430070, China
[2] College of Information Science and Engineering, Henan University of Technology, Henan Zhengzhou 450001, China

Highlights

- The grain emergency vehicle scheduling model was established and solved by using the improved and optimized IACO algorithm.
- A hybrid algorithm was proposed to address the shortcomings of standard metaheuristic algorithms.
- Several common optimization algorithms were employed to assess IACO performance.
- This study demonstrated the effectiveness of IACO for emergency grain distribution optimization problem.

Abstract: The routing optimization problem of grain emergency vehicle scheduling with three objectives is studied in this paper. The objectives are: maximizing satisfaction of the needs at the emergency grain demand points, minimizing total cost of grain distribution and minimizing the distribution time. A hybrid algorithm is present to solve the proposed problem based on combining artificial immune and ant colony optimization (ACO) algorithms. This hybrid algorithm calculates the degree of crowding and conducts non-dominated sorting of the population in the ant colony optimization algorithm by applying a Pareto optimization model. A better solution set is quickly generated by making use of the fast global convergence and randomness of the improved immune algorithm together with the distributed search ability and positive feedback of the ACO algorithm. A better solution set obtained as the initial pheromone distribution, is solved further by using ACO until the approximate optimal solution set is obtained. A comparison of the proposed algorithm with several common optimization algorithms on the Solomon benchmark dataset demonstrates that this method obtains better performance in shorter time, and is an efficient way to solve the vehicle routing problem in emergency grain distribution scenarios.

Keywords: grain emergency logistics; immune ant colony optimization; routing optimization

1 Introduction

After centers for grain distribution in a disaster area has been established, vehicles are distributed from the distribution centers to the emergency grain demand points in the surrounding area, which is also known as "the last kilometer" problem [1]. Compared with traditional vehicle routing problem (VRP) problems [2,3], emergency grain vehicle distribution should meet the timeliness and dynamic changes in demand for grain at the demand points. In addition, the environment of the distribution process affects the routing of the distribution vehicles. Therefore, scheduling and distribution should be completed within the least amount of time when planning distribution routes according to the practical demand for food at different times and locations. It is necessary to find the optimal distribution route with the development of optimization techniques according to the needs of the demand points, delivery cost, and delivery time form a multi-objective optimization problem [4].

At present, due to its importance and complexity, routing optimization problem of vehicle scheduling has been studied extensively and many researches have been devoted to solving this problem by using multi-objective optimization techniques over the last decades. In terms of multi-objective approaches, metaheuristics approaches such as genetic algorithms (GA), ACO and IA have been widely developed to obtain approximately solutions for multi-objective problems.

Stodola et al. [5] present the capacitated multi-depot vehicle routing problem, heuristic and metaheuristic algorithms were studied, and an original solution of the MDVRP problem by using ACO algorithm was obtained. Ben et al. [6] present multi-criteria optimization problem, in which three objectives including minimizing total travel distance, total tardiness time and the total number of vehicles, are considered. Huang et al. [7] proposed several novel hybrid ACO algorithms to resolve multi-objective job-shop scheduling problem, PSO algorithm was used for adaptive tuning of parameters, the real-world data was employed to analyze the performance of the developed algorithms. Khanra et al. [8] proposed a hybrid heuristic algorithm combining ACO and GA for single and multi-objective imprecise traveling salesman problems. The performance of the algorithm is tested and analyzed. Kumar et al. [9] developed MMPPRP-TW with two objectives. This study studied those objectives (minimization of the total operational cost and the total emissions) and solved the model by hybrid SLPSO algorithm.

On the other hand, because of unconventional emergencies occurred frequently, which caused tremendous economic damages and attracted widespread concerns in the international community. Toregas et al. [10] present linear programming method to solve the locating problem of emergency facilities. Camacho-Vallejo et al. [11] presented a non-linear mathematical model for humanitarian logistics, which optimizes the decisions related to post-disaster international aids after a catastrophic disaster in a mixed integer linear programming format. Afshar et al. [12] proposed a comprehensive model to control the flow of relief goods. Tofighi et al. [13] brought up a two stage stochastic fuzzy model for emergency resource distribution in response to natural disasters, they considered the warehouse locations and distribution policy. Yi and Kumar [14] developed an ant colony optimization method to solve the logistics distribution planning. Therefore, it is significant that delivering relief materials should be distributed to the affected area using optimization models and appropriate algorithms.

In spite of the majority of the researches in the field of response to the emergency distribution problem, few studies have considered optimization algorithm for the grain emergency distribution problem. In this study, we combine the immune with ACO algorithm to propose a new method, the immune ACO (IACO) algorithm. Results on the Solomon benchmark dataset show that it outperforms existing optimization methods.

2 Mathematical model for the optimization of emergency grain distribution routing

The routing of emergency grain distribution vehicles should be reasonably planned under all the limits of the environment to meet the needs of the emergency grain demand points. The routing model for emergency grain distribution vehicles is designed using the following three objective functions: 1) satisfaction of the needs at the emergency grain demand points, 2) the total cost of grain distribution, and 3) control of the distribution time. The total cost includes transportation cost, penalty cost (the difference between the distribution time and demand time), and the cost of the vehicles themselves (additional vehicles increase the vehicle cost). The demand points have a higher degree of satisfaction if all vehicles that participate in the emergency grain distribution fulfill their task in the shortest time possible. The lowest distribution time is the longest time that the vehicles spend travelling from the food distribution center to the emergency grain demand points after the grain has been distributed to all of the demand points.

2.1 Assumptions and constraints of the model

The following assumptions are used in this study.

(1) The information about the locations and distance of the emergency grain distribution points and demand points is known.

(2) Grain demand is determined according to the population at the demand point.

(3) The demand point has the last arrival time constraint in the process of emergency grain distribution, and reverse distribution is not allowed.

(4) The number of emergency grain distribution vehicles are sufficient and the vehicle attributes (fuel consumption, capacity, and speed) are uniform.

(5) Emergency grain distribution vehicles set off from the supply center to the demand points, and finally return to the emergency grain distribution point, but each demand point can receive grain from only one vehicle.

(6) The demand points and routes should be composed into a transport map and the emergency road conditions are estimated. If road conditions are poor, the speed is

slow and the travel time is long.

2.2 Penalty function

In emergency grain distribution scheduling, the time constraint has always been a focus for researchers in China and internationally. In this article, a penalty function is used to describe the timeliness of the emergency grain distribution scheduling. The optimal distribution time window of each grain demand point is set according to the different levels of disaster in the disaster-affected areas and different levels of damage to infrastructure and demand for food. If grain distribution vehicles finish the distribution within the set time window, they will not be penalized. If distribution vehicles finish the distribution before the set time window or later than the set time window, distribution cannot be completed or emergency scheduling results are affected, and they must pay a penalty. The penalty cost function is designed as follows:

$$P_i(s_i) = p\max(e_i - s_i, 0) + q\max(s_i - l_i, 0) \tag{1}$$

This equation can be rewritten to more intuitively reflect the meaning of the penalty cost as follows:

$$P_i(s_i) = \begin{cases} p(e_i - s_i) & if \;\; s_i < e_i \\ 0 & if \;\; e_i \le s_i \le l_i \\ q(s_i - l_i) & if \;\; s_i > l_i \end{cases} \tag{2}$$

Here, e_i is the earliest time at which emergency grain is distributed to the demand points, l_i is the latest time at which the emergency grain is distributed to the demand points, s_i is the travel time from the emergency grain distribution center to the demand point, p is the waiting cost incurred if the vehicles arrive ahead of schedule, and q is the penalty cost incurred when the vehicles arrive later than the set time window.

2.3 Modeling

In this section, the multi-objective optimization approach for emergency grain distribution logistics [1] [15] is presented.

2.3.1 Parameters and variables

The node set is denoted as $V = \{0, 1, 2, \ldots, n\}$, where 0 is emergency grain distribution center, which is the starting point and ending point of a grain distribution route, and the demand nodes are numbered as $\{1, 2, \ldots, n\}$. Moreover,

$M = \{1, 2, \ldots, m\}$ is the set of emergency grain distribution vehicles. Other parameters are defined as follows:

· q_i is the demand of demand point *i*.

· Q_k is the load capacity of emergency distribution vehicle *k*, which should meet $Q_k > \max\{q_i, i \in V\}$.

· d_{ij} is the distance between demand points *i* and *j*.

· α_{ij} is the road condition coefficient between nodes *i* and *j*, and it takes a value in the range [0, 1]. Worse road conditions are indicated by smaller values.

· v is the speed of an emergency grain distribution vehicle.

· t_{ij} is the time that the emergency grain distribution vehicle takes to travel from demand point *i* to demand point *j*.

· *c* is the cost an emergency vehicle incurs when moving a unit distance on good roads.

· f_{ij} is the cost from demand point *i* to *j*.

· f_r is the fixed cost of the emergency delivery vehicle.

· $[0, T_i]$ refers to the distribution time window of emergency point *i*.

· t_i is the time at which the emergency distribution vehicle arrives at the demand point.

· ST_k refers to the total delivery time of vehicle *k*.

Finally, $\mu_i(t_i)$ refers to the degree of satisfaction of demand point *i* in the emergency grain distribution process. This function is expressed as:

$$\mu_i(t_i) = \begin{cases} \dfrac{(T_i - t_i)}{T_i}, 0 \le t_i \le T_i \\ 0, t_i \ge T_i \end{cases} \tag{3}$$

If the delivery vehicles fulfill the emergency distribution tasks on time, the degree of satisfaction of the demand points increases. In contrast, as the delivery time is increasingly delayed, the degree of satisfaction gradually reduces.

$$x_{ij}^k = \begin{cases} 1, \textit{distribution vehicle k travels from i to j.} \\ 0, \textit{otherwise} \end{cases}$$

$$y_i^k = \begin{cases} 1, \textit{grain emergency point is distributed by vehicle } k \\ 0, \textit{otherwise} \end{cases}$$

2.3.2 Objective function

Considering these assumptions, three objectives are defined for the mathematical model including total costs of grain distribution, total distribution time and satisfaction of the needs at the emergency grain demand points. Since the objective isn't one summation, the minimization-maximization format is used in the objective function. Using on the above parameters and variables, the multi-objective function of the emergency grain distribution vehicle routing model is as follows:

$$\max f_1 = \frac{1}{n}\sum_{i=1}^{n} u_i \tag{4}$$

$$\min f_2 = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m} f_{ij} x_{ij}{}^{k} + \sum_{j=1}^{n}\sum_{k=1}^{m} f_r x_{0j}{}^{k} + \sum_{j=1}^{n}\sum_{k=1}^{m} P_i(s_i) \tag{5}$$

$$\min f_3 = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m} t_{ij} x_{ij}{}^{k} \tag{6}$$

The first objective function of the model, which is shown in equation (4), maximizes the satisfaction degree of an emergency grain distribution scheduling demand point. The second objective function, which is shown in (5), consists of three parts. The first part is related to the cost from demand point, transportation costs from demand point to distribution centers, the second part is related to the invariability of the total cost, and the last part is related to the penalty costs of the model. The third objective function, which is shown in equation (6), minimizes the delivery process time through the minimization of the sum of the transportation time among the demand points.

The constraint conditions are expressed by the following equations.

$$\sum_{i} q_i y_i{}^{k} \le Q_k, \forall k \tag{7}$$

$$t_{ij} = d_{ij} / \alpha_{ij} v \tag{8}$$

$$f_{ij} = d_{ij} c / \alpha_{ij} \tag{9}$$

$$\sum_{i} y_i{}^{k} = 1, i \in v \tag{10}$$

$$\sum_{k=1}^{m} x_{0j}{}^{k} \le m, j = 1, 2, \ldots, n \tag{11}$$

$$\sum_{i} x_{ij}{}^{k} = y_{j}{}^{k}, j \in v, \forall k \tag{12}$$

$$\sum_{j} x_{ij}{}^{k} = y_{j}{}^{k}, i \in v, \forall k \tag{13}$$

$$\sum_{i,j \in S\text{xs}} \sum x_{ij}{}^{k} \leq |s| - 1, S \subseteq v, 2 \leq |s| \leq n-1 \quad x_{ij}{}^{k} = (0,1), y_{ij}{}^{k} = (0,1) \tag{14}$$

Constraint condition (7) defines the capacity limits of the emergency distribution vehicles. Constraint condition (8) defines the time that a distribution vehicle needs to travel from demand point *i* to demand point *j* considering the traffic conditions. Constraint condition (9) defines the cost that a distribution vehicle incurs traveling from demand point *i* to demand point considering the traffic conditions. Constraint condition (10) specifies that the distribution task of each demand point is fulfilled by only one vehicle. Constraint condition (11) limits the number of emergency grain distribution vehicles. Constraint conditions (12–14) specify that after an emergency distribution vehicle starts from the distribution center and fulfills the distribution tasks at the emergency demand points, it returns to the distribution center, forming a loop.

3 IACO algorithm for emergency grain distribution routing

This section presents the IACO algorithm for the vehicle scheduling model, which maximizes demand point satisfaction and minimizes cost and delivery time. The hybrid algorithm uses an IA [16－18] to distribute pheromones and uses ACO [19,20] to determine the optimal solution. First, non-dominated sorting and the degree of crowding are calculated for the population. The initial solution is determined using an immune mechanism to extract a vaccine. The algorithm accelerates its convergence speed through the immunity calculations and avoids premature convergence. ACO is then used to obtain the Pareto solution.

3.1 Immune antibody design for ACO

3.1.1 Non-dominated sorting

In order to ensure the diversity and convergence of a Pareto optimization model [21], this study uses the rapid non-dominated sorting algorithm to design the immune antibodies of the ant colony. Non-dominated sorting genetic algorithm II is introduced by Deb et al [22], is one of the most widely used algorithms for multi-objective problem, which has the two following specific features: a non-dominated sorting method and crowding degree estimation. The former feature that ranks the population members in non-dominated front classes, the members are not dominated by the other population member, while the first non-dominated front class with rank 1. The next front class with rank 2, which is dominated only by the members in the first class. The third front class with rank 3 and the members are dominated only the first and the second class members, and so on. Another characteristic is the crowding degree estimation that determines whether a member is located in a crowded area of known Pareto solutions or not. The search toward a uniformly distributed Pareto front is determined by crowding degree. The steps of NSGA II are as follows:

(1) Random population P_0, then sorting P_0 based on the Pareto frontier, the rank of each member of the population is calculated. The crossover, mutation operator and tournament selection mechanism are used to handle the initial population, and a next population Q_0 can be obtained

(2) Generating the next population of $R_t = P_t \cup Q_t$, then sorting P_0 based on the Pareto optimization and the non-dominated frontiers F_1, F_2, ….are obtained.

(3) According to the crowding degree, sorting the entire members in F_i, and selecting the best *NA* members to generate population P_{t+1}.

(4) Performing the crossover and mutation operator, generating the next population Q_{t+1}.

(5) Satisfying the termination conditions and the algorithm will be ended. Otherwise, $t = t + 1$, go to step (2).

The main pseudo codes of this algorithm are listed in Algorithm 1.

Algorithm 1. Rapid non-dominated sorting

Inputs: p: an antibody, which is a feasible solution, in population *P; N*: the number of antibodies in population *P*; S_P: the antibody set dominated by antibody *p* in population *P*; np: the number of antibodies that dominate antibody *p*.

for each $p \in P$

$S_p = 0$, $n_p = 0$ //initialize S_p and n_p

for each $q \in P$

if *p* dominates *q* then

add *q* to $S_p, S_p = S_p \cup \{q\}$

else if *q* dominates p then

$n_p = n_p + 1$

If $n_p = 0$ //there is no antibody dominating *p*

then

Mark rank $p_{rank} = 1, NDset_1 = NDset_1 \cup \{p\}$

End

Initialize counter $i = 1$

While $NDset_i \neq 0$ // $NDset_i$ is the first layer non-dominated antibody set; when it is circulated i times, $NDset_i \neq 0$

Initialize $Q = 0$

for each $p \in NDset_i$

for each $q \in S_p$ //S_p is the antibody set dominated by each p in the present $NDset_i$

$n_q = n_q - 1$

If $n_q = 0$ then

mark rank $q_{rank} = i + 1, updating\, Q = Q \cup \{q\}$

$i = i + 1,$ $NDset_i$ =Q

3.1.2 Computation of the crowding degree

The crowding degree refers to the surrounding antibody density for a given antibody in the population; The crowding degree of members is the distance difference of the two objectives between adjacent two solutions in the same rank. The crowding degree was presented to preserve the diversity of population. That is, the average length of the biggest rectangle in the area of the antibody itself. This calculation is performed mainly to ensure the diversity of population in the algorithm. After non-dominated sorting has been performed, the crowding of each antibody in each non-dominated set $NDset_i$ needs to be computed and pseudo-code of the crowding degree method in a non-dominated set are shown in Algorithm 2.

Algorithm 2. Crowding degree computation

Inputs: $I(i)_{distance}$: the crowding degree of population antibody *i*, *n*: the number of antibodies, *m*: the number of objective functions.

for each *i*

Initialize $I(i)_{distance} = 0$

for each *m*

I = sort [*I*, *m*] //ranking each objective's antibodies *i*

$I(1)_{distance} = I(n)_{distance} = \infty$

for *i*= *2* to *(n-1)*

$$I(i)_{distance} = I(i)_{distance} + \frac{I(i+1).m - I(i-1).m}{f_m^{max} - f_m^{min}}$$

// *I(i).m* stands for the number *m* objective function values of antibody *i*.

After Algorithms 1 and 2 have been computed, each antibody population p has two properties: p_{rank} and p_d. Using these two properties, the dominance relation between any two antibodies (i.e., antibodies *i* and *j*) can be distinguished. Only when condition $i_{rank} < j_{rank}$ or $i_{rank} = j_{rank}$ and $i_d > j_d$ is met can antibody *i* be considered to be better than antibody *j*.

3.2 Algorithm flow

Combining the characteristics of the emergency grain distribution vehicle routing optimization problems and the optimized mathematical model, the main flow of the algorithm is as follows.

3.2.1 Generating the initial antibody population

All the population antibodies are encoded using natural numbers, where 0 is the distribution center and $1,2,\cdots,n$ are the demand points that need distribution. Supposing that there are *m* distribution vehicles in the distribution center, we can obtain at most *m* distribution paths. First, an initial antibody is randomly generated. If the randomly generated initial antibody has an illegal solution, repair strategies can be used to compute a legal solution or a penalty is incurred and the solution is deleted. The penalty strategy is used in this article, that is, illegal solutions are simply deleted.

3.2.2 Antibody affinity assessment

Antibody affinity is evaluated to determine the standards for the selection of antibodies, which is mainly affected by the objective function and constraint conditions [23]. Non-dominated sorting and crowding degree are used to determine affinity in the proposed method. Hence, to assess affinity, considering the objective function of the constraint conditions is sufficient to calculate the target value of each antibody.

3.2.3 Memory cell generation

The IA enables the best antibodies to be stored in the memory. After each iteration, the antibodies saved in the memory are replaced by the better antibodies generated in this iteration. This memory functionality effectively avoids the possibility that the best feasible solution is lost during evolution. After non-dominated sorting and crowding degree calculation, the proposed method treats the antibodies for which P_{rank} is small and crowding degree is high as optimal.

3.2.4 Antibody Metabolism

In this section, the crossover and mutation functions [24] in the proposed method are described.

For crossover, the antibodies in memory are chosen randomly from the global solution set based on the probability P_m. These antibodies are exchanged with other antibodies to generate new antibodies. The aim is to improve the convergence speed of the algorithm and obtain the optimal solution as soon as possible.

For example, suppose there are two antibodies H1 and H2, and two numbers (3 and 6) are randomly selected as the insertion point. Then, H1：3 8|4 2 5 6 |1 9 7 H2：2 5|8 7 1 6|3 4 9

Exchanging antibody H1 with antibody H2, we obtain:

A1：3 8|8 7 1 6|1 9 7 A2：2 5|4 2 5 6|3 4 9

Then, repeated elements in A1 and A2 are deleted and the elements that do not appear in the antibody are added. Finally, two legal solutions are obtained:

A1：3 8|4 7 1 6|5 9 2 A2：7 1|4 2 5 6|3 8 9

For mutation, suppose the mutation probability of a certain point in an antibody is P_m. When mutation occurs, one or more points are randomly chosen and randomly inserted into other positions. For example: suppose there is an antibody H and variation occurs at points (4, 7). If H is 3 6 5 4 7 8 9 1 2, extracting the two underlined numbers obtains A, which is 3 6 5 7 8 1 2. Next, two numbers (3 and 6) are randomly generated. If the two numbers are inserted simultaneously, A becomes 3 6 4 5 7 9 8 1 2 1.

3.2.5 Vaccine operation

It is difficult to find a vaccine that is suitable for solving the entire food emergency vehicle scheduling problem described in this paper. Only some parts can be adjusted, that is, each target value of the antibodies is improved as much as possible. Specifically, the distance between each demand point is calculated first. At each iteration, for the demand points with low satisfaction, the vehicles that are the closest to those demand points are scheduled to distribute grain to them in order to save transportation time and cost. This is the so-called vaccine operation in the IA.

For example: suppose satisfaction at demand point S_i is low ($\mu < 0.6$), the antibody of demand point S_i is encoded as $S = \{S_0, S_1, \cdots, S_i \cdots\}$. If the same vehicle should be used to transport grain to demand point S_0 and demand point S_i, the positions of S_0 and S_i are switched and $S = \{S_i, S_1, \cdots, S_0 \cdots\}$ is obtained. If satisfaction is higher at demand point S_i ($\mu > 0.6$) and the distance between demand points S_j and S_i is the smallest, the randomly generated antibody is encoded as $S = \{S_0, S_1, \cdots, S_i, S_{i+1}, \cdots S_j, S_{j+1} \cdots\}$. To save delivery time and cost, the same distribution vehicle is used and only the positions of S_{i+1} and S_j are switched.

3.2.6 Immune selection

The non-dominated sorting and crowding degree is calculated again for the antibodies. The results are used to update the memory and the ACO algorithm is further used to solve the non-dominated solutions in memory to distribute pheromone.

3.2.7 Parameter initialization for pheromone distribution

In the ACO algorithm, the following parameters need to be initialized: τ_c is the pheromone constant; τ_G denotes the pheromone value, which was changed by the IA, N is the number of the ants, α indicates for the importance of edge (i, j), β is the importance of visibility η_{ij} and pheromone concentration τ_{ij}, ρ is the evaporation coefficient of a pheromone track, and Q is the pheromone strength. The initial value of the pheromone is set using the following equation:

$$\tau_{ij}(0) = \begin{cases} \tau_c + \tau_G,\ \textit{If sides (i, j) are on the better} \\ \textit{feasible route obtained by immune selection} \\ \tau_c, \textit{otherwise} \end{cases} \tag{15}$$

3.2.8 Calculating of the vehicle route for each ant

All ants are placed at the distribution center to select the demand points to which they will travel. When an ant is at demand point *i*, the next demand point *j* needs to be selected. First, the selection probability of a demand point that has not been scheduled is calculated, and then the roulette selection algorithm is used to select the next point that needs to be scheduled. The probability that a demand point can be selected is influenced by two factors: i) the intimacy between demand points *i* and *j*, namely the visibility $\eta_{ij} = 1/d_{ij}$, where d_{ij} stands for the length of edge (*i*, *j*) and ii) the feasibility from demand point *i* to demand point *j*, namely, pheromone concentration τ_{ij}. The probability that ant *k* travels from demand point *i* to demand point *j* at time *t* is represented by

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}(t)^\alpha . \eta_{ij}(t)^\beta}{\sum_{K \in A_k} \tau_{ik}(t)^\alpha . \eta_{ik}(t)^\beta}, & j \in A_k; \\ 0 & , j \notin A_k; \end{cases} \tag{16}$$

Here, A_k is a demand point that has not been visited by ant *k* in the ACO solution.

In the process of dispatching grain the demand points using ants to represent vehicles, when the remaining vehicles cannot meet the demand of the next distribution point, it returns to the distribution center for more grain and continues to dispatch grain to demand points that still need grain. When the ants have traversed to all demand points, that is, one cycle has ended, a distribution is finished. Moreover, every time an ant fulfills a distribution, all pheromones in the corresponding paths are added and updated until all distributions have been completed.

The update rules of the proposed ACO algorithm are as follows:

$$\tau_{ij}(t+1)=(1-\rho)\cdot\tau_{ij}+\Delta\tau_{ij} \tag{17}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{z_k}, If\ \ ant\ k\ passes\ sides\ (i,\ j)\ in\ this\ cycle\\ 0, otherwise\end{cases} \tag{18}$$

Here, $\Delta\tau_{ij}{}^{k}$ indicates that ant *k* ant assesses the pheromones and traverses edge (*i*, *j*) and Z_k is the path length of ant *k* in that cycle.

3.2.9 Population update

After the above steps are complete, a new antibody population is obtained. To ensure a wide global distribution of the Pareto model’s optimal solutions, non-dominated sorting and calculation of the crowding of the antibody population in memory need to be recalculated. Next, the memory needs to be updated again.

3.2.10 Algorithm termination

If the algorithm has reached the maximum number of iterations or the current antibody population contains the globally best individual, then the algorithm ends. Otherwise, the algorithm will return to step 4 and iterate again.

4 Experiments

4.1 Experimental data

To verify the effectiveness of the proposed IACO algorithm for solving the emergency grain distribution vehicle scheduling model, the data in the Solomon benchmark examples [25－27] were used for the numerical simulation with some slight modifications. The representative cases R101, C101, and RC101, which consist of 100 demand points, were selected for the experiment and 30 distribution points were randomly selected for the three cases. The demand of the emergency demand points, emergency vehicles, emergency distribution centers, the capacity of the distribution vehicles, and the emergency delivery times are consistent with the

Solomon benchmark examples. We set the fixed cost of distribution vehicle C_f = 20 RMB/vehicle, the average travelling cost of a vehicle to C = 6 RMB/km, and the average velocity of the vehicles to v = 40 km/h.

4.2 Results

First, the parameters of the algorithm need to be initialized. We set the size of the antibody population to N = 60, the maximum number of iterations to 220, and the number of memory cells in the antibody to 10, $P_m = 0.9$. Moreover, the initial value of the ant colony pheromone distribution was $\tau_c = 100$ and $\tau_G = 10$, and the other parameters were set as follows: $N_1 = 40$, $\alpha = 1$, $\beta = 4$, $\rho = 0.4$, and $Q = 100$. The steps of the algorithm flow described above were used to solve the model, and the final computational results of the IACO are shown in Table 1.

Table 1 Final computational results of the model

Examples	R101	C101	RC101
Pareto-optimal solution	9	8	8
Number of delivery vehicles	4	5	5
Maximum average degree of demand point satisfaction	0.95	0.81	0.79
Minimum average degree of demand point satisfaction	0.78	0.71	0.76
Shortest delivery time (min)	66.9	71.2	75.3
Longest delivery time (min)	80.8	86.2	90.3
Lowest cost (RMB)	176.3	159.3	154.7
Highest cost (RMB)	183.5	162.5	163.0

The characteristics of the time window and the geographical position of each demand point are different in each distribution process. Therefore, the average degree of satisfaction, number of vehicles, and delivery time and cost of the three examples are different. Figures 1–3 respectively show the scheduling scheme of three examples that were randomly chosen from the solution set of examples R101, C101, and RC101. The central point in the figure indicates the distribution center and all the other points indicate the demand points.

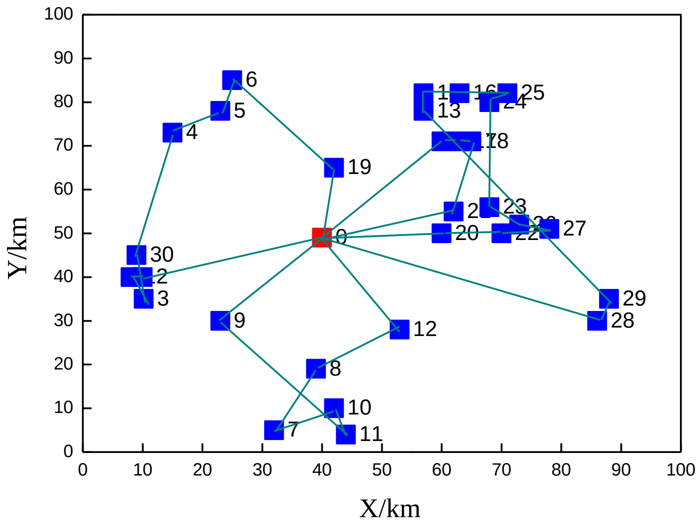

Figure 1 Scheduling route for example C101

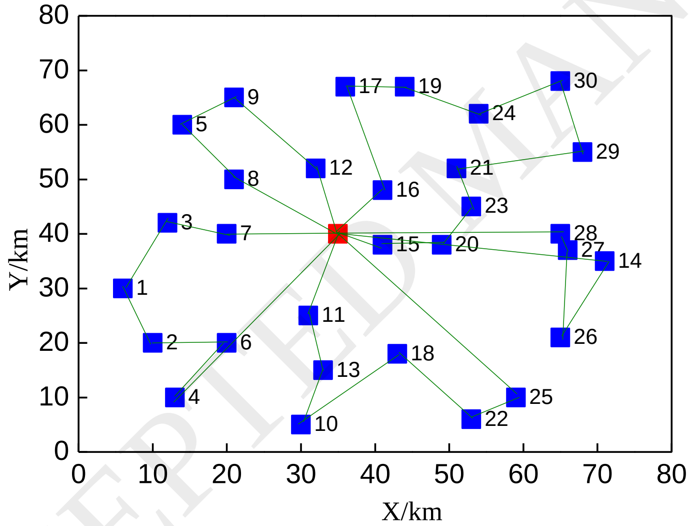

Figure 2 Scheduling route for example R101

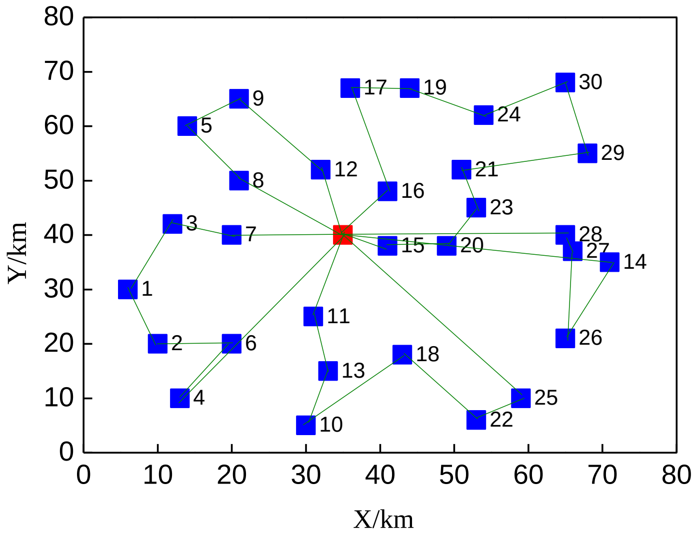

Figure 3 Scheduling route for example RC101

The optimal routes of the distribution schemes are shown in Table 2.

Table 2. Optimal route of the distribution scheme.

Dataset	**Route (Vehicle) No.**	**Optimal Route**
R101	1	GDC----DP19----DP6----DP5----DP4----DP30----DP3----DP1----DP2---- GDC
	2	GDC----DP9----DP11----DP10----DP7----DP8----DP12-----GDC
	3	GDC—DP26----DP29---DP13----DP14----DP16---DP25-----DP24---DP23---- DP26---DP27----DP22---DP20----GDC
	4	GDC----DP21----DP18----DP17----DP15-----GDC
C101	1	GDC----DP7----DP3----DP1---DP2----DP6----DP4----GDC
	2	GDC----DP11----DP13----DP10----DP18----DP22----DP25----GDC
	3	GDC----DP28----DP27----DP26----DP14----GDC
	4	GDC----DP16----DP17----DP19----DP24-----DP30----DP29----DP21----DP23----DP20----DP15----GDC
	5	GDC----DP8----DP5---DP9----DP12----GDC
RC101	1	GDC----DP3----DP2----DP1---DP4----GDC

	2	GDC----DP13----DP14----DP17----DP19----DP18----DP16-----GDC
	3	GDC----DP15----DP26---DP28----DP29----DP30---DP27----GDC
	4	GDC----DP21----DP23----DP24----DP25-----DP22----DP20----DP12----DP10----DP9----GDC
	5	GDC----DP11----DP8----DP7----DP6----DP5----GDC
GDC: Grain Distribution Center, DP: Demand Point		

4.3 Performance of the proposed algorithm

To evaluate the performance of the proposed method, an ACO, IA, and genetic algorithm (GA) [28,29] were also employed as reference methods. The same population size and iterations were used for each method in the experiment. Moreover, the parameters of the GA were as follows: the population size was 50, $P_c = 0.9, P_m = 0.1$, and the maximum number of iterations was 200. The above four algorithms were able to obtain Pareto solutions, which can be distributed evenly. The experiments used the modified version of R101 in the Solomon benchmarks problems. The above four algorithms found 5, 5, 4, and 9 Pareto solutions, respectively. Specifically, the procedure of forming the Pareto front for objectives 1 and 2 along with the Pareto front for the objectives 1 and 3 are shown in Figure 4. The results show that the performance of the proposed algorithm is much better than that of the other algorithms. Moreover, the IA, combined with ACO, is clearly an effective potential tool for the analysis and study of route planning for emergency grain distribution logistics.

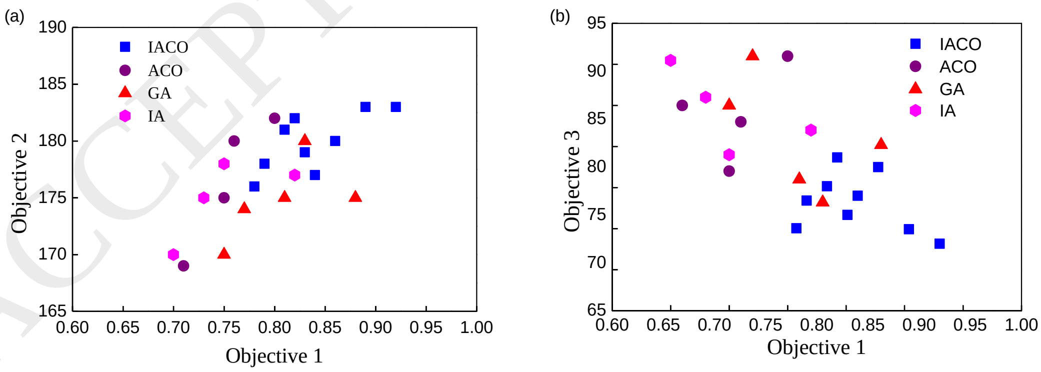

Figure 4 Comparisons of the Pareto fronts for objectives 1 and 2 (a) and objective 1 and 3 (b) obtained by different algorithms

In addition, to evaluate the proposed algorithm more deeply, the mean ideal distance (MID) [30], CPU running time, and convergence speed are presented for the IACO algorithm. MID, which indicates the closeness between the Pareto solution and the ideal point, calculates the average distance of the solutions in the Pareto fronts from an ideal point (minimum of each objective). It can be calculated by the formula:

$$MID = \sum_{i=1}^{n} \frac{c_i}{n} \quad (18)$$

Here, c_i is the distance of the Pareto solution from the ideal solution and n is the number of Pareto solutions. Better performance of the algorithm is indicated by lower MID values. Meanwhile, the CPU running time of the algorithm is an important criterion for assessing the performance of the implemented algorithms. Figure 5 shows the convergence of the experimental analysis for the ACO algorithm, IA, GA, and proposed algorithm. Each algorithm was calculated 10 times. It is clear that the proposed algorithm outperforms the others. The optimal value is obtained within 80 iterations; the proposed method has the advantages of high search efficiency and fast convergence performance.

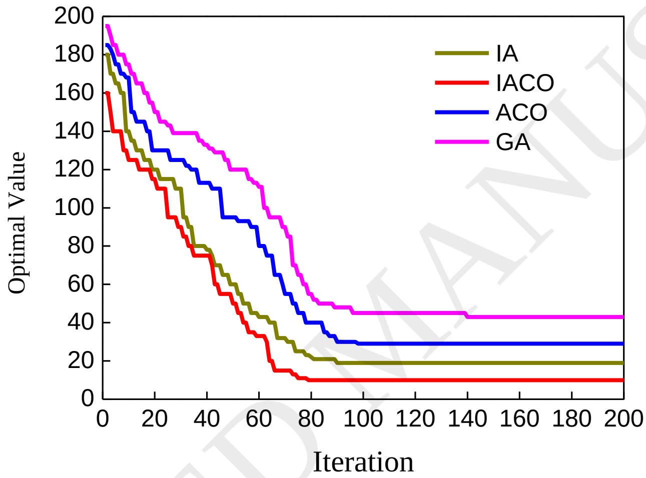

Figure 5 Convergence of the algorithms

The MID and CPU times were calculated for R101, and the results are shown in Figure 6.

This figure shows that the performance of the proposed method is better than that of others with respect to MID and CPU time. The MID values of IA, ACO, and GA are higher than that of IACO; this is also true for the CPU running time. These comparisons show that IACO not only improves the quality of the solution, but also avoids stagnation and ensures the effectiveness and global property of the search. These results indicate that the IACO possess fast convergence speed compared to the standard ACO algorithm and IA. Therefore, the proposed hybrid algorithm is efficient and feasible for the emergency grain distribution vehicle routing problem.

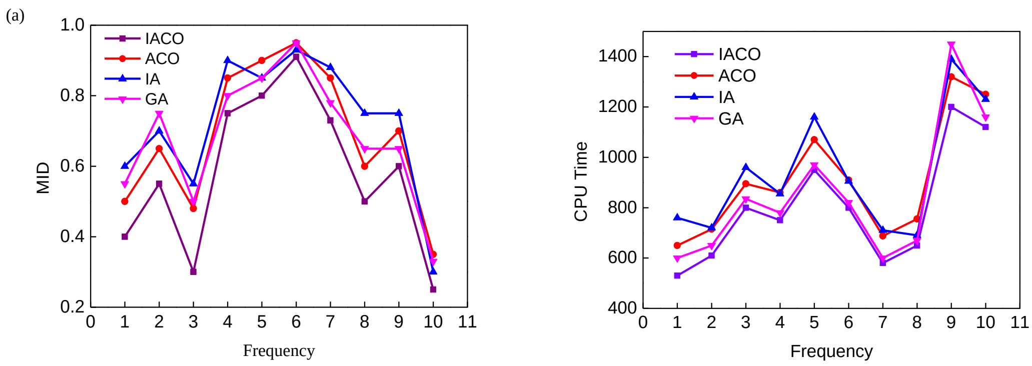

Figure 6 Comparison of the metaheuristic algorithms based on MID (a) and CPU time (b).

In order to assess the performance and make the experiment more comprehensive, we select the hypervolume (HV) to measure the characteristics of the algorithms; *HV* evaluates the convergence and the diversity of the obtained solutions. HV is calculated as follows:

$$HV = VOL(\bigcup_{x \in P}[f_1(x), z_1] \times \cdots f_m(x), z_m]) \tag{19}$$

Where $VOL(\bullet)$ indicates the Lebesgue measurement. and $z_1 \cdots z_m$ denotes reference points dominated by all Pareto-optimal solutions. The larger the HV value, the better the quality of obtained *P* for approximating the whole PF and better diversity.

Table 3 The comparison results of average HV for R101,C101 and RC101

Problems	AIACO	NSGAII	SPEA2
R101	0.264	0.257	0.224
C101	0.315	0.281	0.279
RC101	0.874	0.693	0.676

The hypervolume indicator was employed to compare the non-dominated solutions and the other two comparing algorithm (NSGAII and SPEA2).Table 3 shows the comparison results of average HV for R101,C101 and RC101 problems respectively. It can provide a qualitative measure not only of convergence but also of diversity. It can be seen that non-dominated solutions obtained by the proposed algorithm have larger hypervolume values on all the R101, C101 and RC101 problems.

According to the average hypervolume values, the proposed algorithm outperforms all the other on R101, C101 and RC101 problems. NSGA II and SPEA2obtains the relatively next-best performance in C101and RC101problems in terms of HV metric, while the two algorithms presented an almost unanimously values. The customers in R101, C101 and RC101 have small time windows and are randomly distributed; IACO has obvious advantages on these problems.

To better visualize the results and further investigate the performance on the IACO, the comparison results of HV with generations for the instance of RC101 for the five algorithms are present in Fig .7.

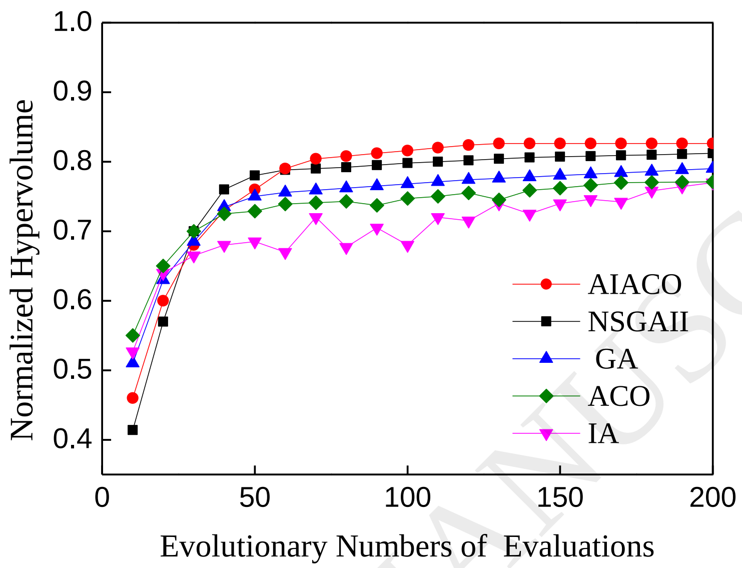

Figure 7 The comparison results of HV with generations during varied algorithms on RC101

As shown in Figure 7, HV indicates that the solutions are better spread and closer to the optimal set and the HV changes indicate that the performance of the algorithms differ during the evolutionary process. It is clear that AIACO always obtains the better HV value during the whole evaluations, which shows that the AIACO has a clear advantage and outperforms the others. IA and ACO have a better HV at first, but with the increasing evolutionary numbers, the performance is not very stable. While NSGA-II and GA produce the stable HV and obtain the similar results.

5 Conclusions

This article established a multi-objective mathematical model for emergency grain distribution route optimization. This problem is based on the actual problem of emergency food logistics optimization together with its specific requirements and characteristics. This model was solved by the improved and optimized IACO algorithm and experimental simulation data were used to obtain the optimal route. This approach nicely solves "the last kilometer" problem in emergency grain distribution logistics. A hybrid algorithm was proposed to address the shortcomings of standard metaheuristic algorithms. ACO, IA, and GA were employed to assess IACO performance, and the results show that the IACO algorithm performs better than the other algorithms in this paper. Therefore, this study demonstrated the effectiveness of IACO for multi-objective optimization problems, and it can be concluded that IA combined with ACO is a potential tool to obtain a vehicle routing solution for emergency grain distribution logistics.

Acknowledgements

This work was supported by the National "Twelfth Five-Year" Plan for Science & Technology Support (Grant No. 2013BAD17B04) and the National Key Research and Development Program of China (Grant No. 2017YFD0401004).

References

[1] D.Q. Jiang, Q.X. Zhu, Grain Emergency Vehicle Scheduling Problem with Time and Demand Uncertainty, Math Probl Eng, (2014).

[2] A. Moreno, D. Alem, D. Ferreira, Heuristic approaches for the multiperiod location-transportation problem with reuse of vehicles in emergency logistics, Comput Oper Res, 69 (2016) 79-96.

[3] I. Markov, S. Varone, M. Bierlaire, Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities, Transport Res B-Meth, 84 (2016) 256-273.

[4]J.F. Camacho-Vallejo, E. Gonzalez-Rodriguez, F.J. Almaguer, R.G. Gonzalez-Ramirez, A bi-level optimization model for aid distribution after the occurrence of a disaster, J Clean Prod, 105 (2015) 134-145.

[5] Stodola P, Mazal J. Applying the ant colony optimisation algorithm to the capacitated multi-depot vehicle routing problem. *Int J Bio-Inspir Com* 2016, **8**(4)**:** 228-233.

[6] Ben Alaia E, Dridi IH, Bouchriha H, Borne P. Genetic Algorithm for Multi-Criteria Optimization of Multi-Depots Pick-up and Delivery Problems with Time Windows and Multi-Vehicles. *Acta Polytech Hung* 2015, **12**(8)**:** 155-174.

[7] Huang RH, Yu TH. An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting. *Appl Soft Comput* 2017, **57:** 642-656.

[8] Khanra A, Maiti MK, Maiti M. A hybrid heuristic algorithm for single and multi-objective imprecise traveling salesman problems. *J Intell Fuzzy Syst* 2016, **30**(4)**:** 1987-2001.

[9] Kumar RS, Kondapaneni K, Dixit V, Goswami A, Thakur LS, Tiwari MK. Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Comput Ind Eng* 2016, **99:** 29-40.

[10] Toregas, C., Swain, R., ReVelle, C., Bergman, L., The location of emergency service facilities. Oper. Res. 1971, 19 (6), 1363-1373.

[11] Camacho-Vallejo, J.F., Gonzalez-Rodríguez, E., Almaguer, F.J., Gonzalez-Ramírez, R.G., A bi-level optimization model for aid distribution after the occurrence of a disaster. J. Clean. Prod. 2015, 105, 134-145

[12] Afshar, A., Haghani, A., Modeling integrated supply chain logistics in real-time large-scale disaster relief operations. Socio-Econ. Plan. Sci. 2012, 46 (4), 327-338.

[13] Tofighi, S., Torabi, S.A., Mansouri, S.A., 2016. Humanitarian logistics network design under mixed uncertainty. Eur. J. Oper. Res. 250 (1), 239-250

[14] Yi, W., Kumar, A.: Ant colony optimization for disaster relief operations. Transp. Res. Part E, 2007 43(6),660–672

[15] Qiang Zhang, Qiuwen Zhang, Optimization of grain logistics distribution center location based on genetic algorithm. Journal of Henan University of Techno logy (Natural Science Edition)Vo1.29.NO.5Oct.2008

[16] S. Ghandi, E. Masehian, A breakout local search (BLS) method for solving the assembly sequence planning problem, Eng Appl Artif Intel, 39 (2015) 245-266.

[17] V.N.S.A. Kumar, V. Kumar, M. Brady, J.A. Garza-Reyes, M. Simpson, Resolving forward-reverse logistics multi-period model using evolutionary algorithms, Int J Prod Econ, 183 (2017) 458-469.

[18] M. Zangari, A. Pozo, R. Santana, A. Mendiburu, A decomposition-based binary ACO algorithm for the multiobjective UBQP, Neurocomputing, 246 (2017) 58-68.

[19]S. Braaten, O. Gjonnes, L.M. Hvattum, G. Tirado, Heuristics for the robust vehicle routing problem with time windows, Expert Syst Appl, 77 (2017) 136-147.

[20] X.Y. Wang, T.M. Choi, H.K. Liu, X.H. Yue, Novel Ant Colony Optimization Methods for Simplifying Solution Construction in Vehicle Routing Problems, Ieee T Intell Transp, 17 (2016) 3132-3141.

[21] A.M. Mora, P. Garcia-Sanchez, J.J. Merelo, P.A. Castillo, Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal, Soft Comput, 17 (2013) 1175-1207.

[22] K. Dep, A. Pratap, S. Agawal, and T. Meyarivan.A fast and elitist multi-objective genetic algorithm: NSGAII IEEE Transactions on Evolutionary Computation, 6(2).2002 182-197.

[23] L.F. Hsu, C.C. Hsu, T.D. Lin, An Intelligent Artificial System : Artificial Immune based Hybrid Genetic Algorithm for the Vehicle Routing Problem, Appl Math Inform Sci, 8 (2014) 1191-1200.

[24] Q.Z. Lin, J.Y. Chen, Z.H. Zhan, W.N. Chen, C.A.C. Coello, Y.L. Yin, C.M. Lin, J. Zhang, A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems, Ieee T Evolut Comput, 20 (2016) 711-729.

[25] D.F. Zhang, S.F. Cai, F.R. Ye, Y.W. Si, T.T. Nguyen, A hybrid algorithm for a vehicle routing problem with realistic constraints, Inform Sciences, 394 (2017) 167-182.

[26] Solomon M .Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 35(1987)254–265

[27] R. Perez-Rodriguez, A. Hernandez-Aguirre, Simulation optimization for the

vehicle routing problem with time windows using a Bayesian network as a probability model, Int J Adv Manuf Tech, 85 (2016) 2505-2523.

[28] G.L. Storti, M. Paschero, A. Rizzi, F.M.F. Mascioli, Comparison between time-constrained and time-unconstrained optimization for power losses minimization in Smart Grids using genetic algorithms, Neurocomputing, 170 (2015) 353-367.

[29] K. Braekers, K. Ramaekers, I. Van Nieuwenhuyse, The vehicle routing problem: State of the art classification and review, Comput Ind Eng, 99 (2016) 300-313.

[30] H. Maghsoudlou, B. Afshar-Nadjafi, S.T.A. Niaki, A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem, Comput Chem Eng, 88 (2016) 157-169.