International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:12, No:3, 2018

# A Hybrid Algorithm for Collaborative Transportation Planning among Carriers

Elham Jelodari Mamaghani, Christian Prins, Haoxun Chen

***Abstract*—**In this paper, there is concentration on collaborative transportation planning (CTP) among multiple carriers with pickup and delivery requests and time windows. This problem is a vehicle routing problem with constraints from standard vehicle routing problems and new constraints from a real-world application. In the problem, each carrier has a finite number of vehicles, and each request is a pickup and delivery request with time window. Moreover, each carrier has reserved requests, which must be served by itself, whereas its exchangeable requests can be outsourced to and served by other carriers. This collaboration among carriers can help them to reduce total transportation costs. A mixed integer programming model is proposed to the problem. To solve the model, a hybrid algorithm that combines Genetic Algorithm and Simulated Annealing (GASA) is proposed. This algorithm takes advantages of GASA at the same time. After tuning the parameters of the algorithm with the Taguchi method, the experiments are conducted and experimental results are provided for the hybrid algorithm. The results are compared with those obtained by a commercial solver. The comparison indicates that the GASA significantly outperforms the commercial solver.

***Keywords*—**Centralized collaborative transportation, collaborative transportation with pickup and delivery, collaborative transportation with time windows, hybrid algorithm of GA and SA.

## I. Introduction

THE number of freight vehicles moving within a city is growing and expected to continue to grow at a steady rate particularly due to the current distribution practices with timely deliveries and the explosive growth of business-to-customer electronic commerce that generates significant volumes of personal deliveries. Carrier collaboration, which can reduce the number of freight vehicles moving in a city, improves the efficiency of freight movements and reduces the empty vehicle-km, is emerging as a key strategy for realizing efficient urban distribution. By exchanging transportation requests among carriers in a transportation network, carriers can reduce their transportation costs, improve their profitability, and capture more business opportunities. In collaborative transportation, multiple companies (carriers) construct an alliance to improve their transportation operations by sharing vehicle capacities and transportation tasks. Such collaboration is to reduce transportation costs by eliminating empty backhauls and increasing vehicle utilization rates so as to increase the profits of all partners.

In carrier collaboration, multiple carriers collaborate with each other to maximize their total profit or to minimize their total transportation cost by optimally serving their requests. In practice, less than truckload (LTL) transportation services are often afforded to customers. For LTL transportation, whereas more studies adopt a decentralized planning approach for maximizing the individual profit of each partner, in this article, a centralized planning approach to the multi carrier collaborative problem is presented. Such a centralized planning approach has also been used in collaborative transportation service trading in B2B e-commerce logistics [1].

The objective of this study is to develop a mathematical model and a solution method for carrier collaboration in urban distribution. There is a focus on the horizontal collaboration among carriers via order sharing. Through order sharing, carriers can improve their efficiency and profitability because of an increase in vehicle capacity utilization, a reduction in empty vehicle repositions, and a reduction in total transportation costs due to an improved transportation planning. This order sharing or request exchange among carriers can be realized by using a centralized approach based on a global mathematical programming model or by using a decentralized approach such as combinatorial auctions. Centralized planning for order sharing means that customer orders from all participating carriers are combined and collected in a central pool, and efficient routes are set up for all requests. In this article, carriers in LTL transportation are considered where each request is a pick and delivery request with a time window for performing each pickup or delivery operation. Each carrier has a limited number of vehicles initially located at its own vehicle depot. These depots may be in different locations. The authors assumed that each carrier has both reserved requests, which must be served by itself because of its commitments to its customers (shippers), and shareable (exchangeable) requests that can be outsourced to other carriers. The collaborative transportation-planning problem with pickup and delivery requests and time windows is NP-hard, so metaheuristic algorithms are necessary to solve large instances of the problem. The algorithm proposed in this paper is based on GASA. On the one hand, this algorithm takes advantages of a population base algorithm with exploring multiple solutions at the same time, and on the other hand, by applying simulated annealing in each iteration of the genetic algorithm, there is a chance to choose a new solution

Elham Jelodari Mamaghani is with LOSI, ICD (UMR CNRS 6281), Université de Technologie de Troyes, Troyes Cedex 10004, France (corresponding author, e-mail: Elham.jelodari_mamaghani@ utt.fr).

Christian Prins is with LOSI, ICD (UMR CNRS 6281), Université de Technologie de Troyes, Troyes Cedex 10004, France (e-mail: christian.prins@utt.fr).

Haoxun Chen is with LOSI, ICD (UMR CNRS 6281), Université de Technologie de Troyes, Troyes Cedex 10004, France (e-mail: haoxun.chen@utt.fr).

worse than the current one for diversification purpose. Furthermore, this algorithm can be viewed as a simulated annealing algorithm, in which the mutation and crossover operators are used to generate neighborhoods, and in each iteration, the temperature decreases. Comparison of the centralized approach with a decentralized approach proposed in [2] is implemented and shows that the suggested GASA algorithm is effective in reaching optimal or near-optimal solutions with very small gaps.

## II. Literature Review

One of the important roles of collaborative transportation is increasing operational efficiency via sharing the requests among carriers. This collaboration not only has an effect on the system efficiency increasing but also has a vital role in the environment protection by decreasing the usage of the vehicles fuel [3]. Reference [4] modelled a carrier collaboration problem as a network flow problem by integrating the transportation networks and demands of all participating carriers to create a large network and a large pseudo-carrier, where the network of each carrier is determined according to the available capacity of cargo transportation in each flight leg operated by that carrier. The demand of each carrier is a set of loads that the carrier accepts for delivery. Reference [5] studied a problem of centralized collaborative transportation. In the transportation network of the problem, each node has a maximum capacity on its passing flow. Therefore, the objective of their problem is to find the maximum flow in the logistic network.

Reference [6] considered time-dependent centralized collaborative transportation. Their model has multi carriers and its objective function is to minimize the total cost of all carriers. Their chosen solution approach is a branch and cut algorithm. Reference [7] studied a sub-problem (bid generation problem) for carrier collaboration with two different types of requests: reserved requests and shared requests, where each request is a pickup and delivery request with time windows. After presenting a mathematical model for the single carrier selective vehicle routing problem with profits, they solved the model with Adaptive Large Neighborhood Search (ALNS).

## III. Problem Description and Mathematical Model

The multi-carriers collaborative transportation-planning problem (Multi-carrier CTP) with pickup and delivery requests, hard time windows, and reserved requests can be defined on a directed graph $G = (V, E)$ where $V$ is the set of all nodes and $E$ is the set of edges, where $V$ contains the vehicle depots of all carriers. In the problem, there is a set $M$ of carriers, each carrier $m \in M$ has a set of reserved requests, $R_{rm}$, that must be served by itself and a set of exchangeable (shareable) requests, $R_{sm}$, that can be served by any carrier. The set of exchangeable requests of all carriers is represented by $R_s$ with $R_s = \bigcup_{m \in M} R_{sm}$, and the set of all requests of all carriers is denoted by $R$ with $R = (\bigcup_{m \in M} R_{rm}) \cup R_s$. Each request has a pickup node and a delivery node with time windows for performing pickup and delivery operations. This problem extends the multi-depot pickup and delivery problem with time windows by introducing an additional constraint that each carrier must serve its reserved requests by itself. In the problem, each carrier $m \in M$ has a finite set of homogenous vehicles denoted by $K_m = \{1,2, \dots, k_m\}$. The capacity of each vehicle is denoted by $Q$. The carriers form an alliance to collaboratively plan their transportation operations. This collaboration is realized by exchanging requests among the carriers. Each carrier has a vehicle depot. Let $DC$ denote the set of depots of all carriers, $W = V \backslash DC$ denote the set of all nodes excluding the depot nodes of all carriers. $P = \{1,2, \dots, n\}$ is the set of pickup nodes of all requests, and $D = \{n+1, \dots, 2n\}$ is the set of all delivery nodes. For convenience of presentation, the requests of all carriers are indexed by 1, 2, …,$n$ and the pick node of request $i$, $i$=1,.., $n$, is denoted by pickup node $i$, and node $n+i$ represents the delivery node of request $i$. The demand of the pickup node of request $i$ is denoted by $d_i$ , whereas the demand of the delivery node of the same request is denoted by $d_{i+n}$, with $d_{i+n} = -d_i$. Each pickup and delivery node $i$ is associated with a time window $[e_i, l_i]$. The traveling time and cost from node $i$ to node $j$ are represented by $t_{ij}$ and $c_{ij}$, respectively. The maximum duration of each route is denoted by $T$. The objective of the problem is to make optimal decisions about the allocation of all exchangeable requests among the carriers under the constraint that each reserved request must be served by its own carrier and other constraints of the standard vehicle routing problem to minimize the total transportation cost of all carriers. Among the other constraints, the time window associated with each pickup/delivery node must be respected, the load of each vehicle cannot exceed its capacity, and the delivery node of each request must be visited after its pickup node on the same route.

The multi-carrier CTP problem can be formulated as a mixed-integer linear programing model. In the model, parameters $BM_{ij} = l_j - e_i$ and $CV_i = Q + d_i$ are introduced to formulate linearly the time window constraints and the vehicle capacity constraints. The decision variables of the model include binary variables, $x_{ijkm}$ and $y_{ikm}$and integer variables $U_{ikm}$and $L_{ikm}$ are defined as follows.

$$x_{ijkm} = \begin{cases} 1 & \text{if and only if vehcile k of carrier m visits directly node j after node i} \\ 0 & \text{else} \end{cases}$$

$$y_{ikm} = \begin{cases} 1 & \text{if and only if request i is served by vehicle k of carrier m} \\ 0 & \text{else} \end{cases}$$

$U_{ikm}$ = arriving time of vehicle k of carrier m at node i

$CV_{ikm}$ = Load of vehicle k of carrier m when it leaves node i

The problem can be formulated as the following mixed integer-programming model:

$$\min \sum_{m\in M}\sum_{k\in k_m}\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ijkm}$$

*Subject to:*

$$\sum_{k\in K}\sum_{j\in V,\, j\neq i}\sum_{m\in M} x_{ijkm} = 1 \quad \forall i \in W \tag{1}$$

$$\sum_{j\in V,\, j\neq i} x_{jikm} - \sum_{j\in V,\, j\neq i} x_{ijkm} = 0 \quad \forall i \in W, \forall k \in K_m, \forall m \in M \tag{2}$$

$$\sum_{j\in p,\, j\neq i} x_{ijkm} \leq 1 \quad \forall k \in K_m, \forall i = m \in M \tag{3}$$

$$\sum_{j\in P,\, j\neq i} x_{ijkm} = 0 \quad \forall k \in K,\ \forall i \in M \setminus \{m\} \tag{4}$$

$$\sum_{j\in D,\, j\neq i} x_{jikm} \leq 1 \quad \forall k \in K_m, \forall i = m \in M \tag{5}$$

$$\sum_{j\in D,\, j\neq i} x_{jikm} = 0 \quad \forall k \in K_m, \forall i \in M \setminus \{m\} \tag{6}$$

$$\sum_{k\in k_m} y_{ikm} = 1 \quad \forall i \in R_{rm}, \forall m \in M \tag{7}$$

$$\sum_{k\in k_m} y_{ikm} = 0 \quad \forall i \in R \setminus R_{rm},\ \forall m \in M \tag{8}$$

$$\sum_{m\in M}\sum_{k\in k_m} y_{ikm} = 1 \quad \forall i \in R_s \tag{9}$$

$$\sum_{j\in V,\, j\neq i, m'} x_{ijkm} = y_{ikm} \quad \forall i \in P, \forall k \in K_m, \forall m \in M \tag{10}$$

$$\sum_{j\in V,\, j\neq i, m'} x_{j(i+n)km} = y_{ikm} \quad \forall i \in P, \forall k \in K_m, \forall m \in M \tag{11}$$

$$U_{ikm} + t_{i(n+i)} + s_i \leq U_{(n+i)km} \quad \forall i \in P, \forall k \in K_m, \forall m \in M \tag{12}$$

$$U_{jkm} \geq U_{ikm} + t_{ij} + s_i - BM_{ij}(1 - x_{ijkm}) \quad \forall i \in V, \forall j \in W, \forall k \in K_m, \forall m \in M \tag{13}$$

$$e_i \leq U_{ikm} \leq l_i \quad \forall i \in W, \forall k \in K_m, \forall m \in M \tag{14}$$

$$U_{ikm} + t_{ij} - BM_{ij}\left(1 - x_{ijkm}\right) \leq T \quad \forall i \in W, \forall k \in K_m, \forall j, m \in M \tag{15}$$

$$CV_{jkm} \geq CV_{ikm} + d_i - Q\left(1 - x_{ijkm}\right) \quad \forall i, j \in V, \forall k \in K_m, m \in M \tag{16}$$

$$\max\{0, d_i\} \leq Q_{ikm} \leq \min\{Q, Q + d_i\} \quad \forall i \in V, \forall k \in K_m, m \in M \tag{17}$$

$$\sum_{j\in W}\sum_{k=1} x_{ijkm} \leq K_m \quad \forall i = m \in M \tag{18}$$

$$x_{ijkm} \in \{0,1\} \quad \forall i, j \in V, \forall k \in K_m, \forall m \in M \tag{19}$$

$$y_{ikm} \in \{0,1\} \quad \forall i \in R, \forall k \in K_m, \forall m \in M \tag{20}$$

$$U_{ikm} \geq 0 \quad \forall i \in V, \forall k \in K_m, \forall m \in M \tag{21}$$

$$CV_{im} \geq 0 \quad \forall i \in V, \forall m \in M \tag{22}$$

The objective function minimizes the total transportation cost of the routes of all carriers. Constraint (1) describes that each customer node (pickup node or delivery node) is visited exactly once by a vehicle. Equation (2) is the flow preservation equation describing when a vehicle arriving at a customer node must leave it. Equations (3)-(6) indicate that each vehicle of any carrier leaves the carrier's depot and returns to the same depot. Constraints (7)-(9) are about the reserved and shared requests. Equations (7)-(8) ensure that each reserved request must be served by its own carrier, whereas (9) implies that each exchangeable request can be served by any carrier. Constraints (10) and (11) ensure that if a request is served, there must be a vehicle leaving its pickup node and arriving at its paired delivery node. Constraint (12) is time constraint between pickup and delivery nodes. Constraint (13) ensures time feasibility, i.e. vehicle $k$ cannot start to serve the customer of node $j$ before completing service at the previous customer of node $i$. The time window constraint is formulated by (14). The maximum route duration constraint is given by (15). Equations (16)-(18) ensure that the capacity of each vehicle and the maximum number of vehicles of each carrier will not be exceeded. The constraints from (19) to (22) describe the domain of each variable.

## IV. GASA Approach to the Multi-Carrier CTP

In this section, a metaheuristic approach is proposed to solve the multi-carrier collaborative transportation-planning problem. NP-hard multi-depot vehicle routing problem is a special case of the multi-carrier CTP, so the latter problem is also NP-had. Therefore, solving a large instance of the problem in an acceptable time by a commercial solver like Cplex is impossible. Hence, it is required to develop a metaheuristic algorithm to solve the problem.

### *A. Genetic Algorithm*

GA is one of the popular metaheuristic algorithms firstly proposed in [8]. It is one of the most efficient methods to solve various vehicle routing problems. For this reason, GA is chosen as a solution method for CTP. GA is based on natural selection, where each solution of a problem is represented by a chromosome, and the population of GA at each iteration consists of a set of chromosomes (solutions). The fitness of each chromosome (solution) is evaluated by its objective function value. Similar to the real world, descendants in the population of the next generation are generated by a reproduction process unifying the genes of two parents through crossover operation, and altering the genes of a chromosome in the process is similar to the mutation in the real world. The choice of parents and children to be transferred to the next generation is made according to their fitness, i.e. the objective function value of each individual (solution). In this process, the individuals with lower fitness

will be removed from the population. By mimicking the natural selection process, a well-designed GA ensures that a near-optimal solution of a problem can be reached after a limited number of iterations. In the rest of this section, the components of our genetic algorithm for the multi-carrier CTP will be presented and explained. The general structure of the presented method is given by Algorithm 1 in Fig. 1. The first loop of the procedure starting with line 2 produces the initial population of GASA by using a heuristic approach. In the approach, firstly, all reserved requests of each carrier are assigned to the carrier itself, and then, each exchangeable request is allocated to its most efficient carrier, i.e. the carrier that can serve it at the minimal additional cost. The details of the construction of an initial solution are given in subsection B of this section. The step of line 5 initializes the temperature of simulation annealing (SA). The second loop starting with line 7 is the main loop of GASA. In each iteration of GASA, the minimum cost of the problem firstly is updated according to the achieved costs of the latest generation. A probability is then calculated (defined) for the solution choice. In the sub-loop starting with line 10, crossover and mutation operations are applied to the solutions of the current generation. After the operations of crossover and mutation, all the offspring generated is merged with the current population to create a new population. In the next step, the solutions in the new population are sorted according to their costs. The last sub-loop staring with line 15 is to choose a solution by applying a SA rule. The SA rule is applied to each solution in the current population. The step on line 31 reduces the temperature of SA. The details of using SA to choose a solution in the hybrid algorithm are given in Section V.

**Algorithm 1**. Pseudo-code of the proposed GASA algorithm

```
1:  f(S°) ← ∞
2:  for np = 1 to npop do
3:      call initial soltion procedure
4:  Sort population and store best solution as so
5:  T ← T0
6:  end for
7:  for it = 1 to nit do
8:      Update minimum cost
9:      Calculate selection probabilities
10:     for subit = 1 to MaxSubit do
11:         Perform crossover
12:         Perform Mutation
13:         merge offsprings population and create newpop
14:         sort newpop
            (compare using SA rule according to the following procedure)
15:         for i = 1 to npop do
16:             if newpop(i).cost ≥ pop(i).cost then
17:                 pop(i) = newpop(i)
18:             else
19:                 DELTA = (pop(i).Cost − newpop(i).Cost)/pop(i).Cost
20:                 P = exp(−DELTA/T)
21:                 if rand <= P
22:                     pop(i) = newpop(i)
23:                 end
24:             end
25:             if pop(i).Cost >= BestSolution.Cost
26:                 Update Best Solution Ever Found
27:             end
28:         end for
29:     end for
30:     store best cost
31:     T = T * co  (temp reduction)
32: end for
```

Fig. 1 The pseudo code of the presented GASA algorithm

### B. The Construction of Initial Solutions

We construct initial solutions for GASA by using a sequential insertion heuristic. In Genetic Algorithm hybrid algorithms, the number of initial solution is equal to the number of population.

In each population, the cost of initial solution must be preserved. After execution of the initial solution in the last population, all of the achieved costs are sorted and the minimum cost is selected as a best achieving solution.

For the presented problem, the reserved requests of each carrier must be served by the carrier itself, whereas exchangeable requests can be served by any carrier. In the proposed heuristic, firstly, each reserved request is assigned to its own carrier. After that, each exchangeable request will be inserted into an existing route of a carrier. Before assigning all requests to their own carriers or other carriers according to the type of each request, all reserved requests and all exchangeable requests are sorted respectively in two lists along with their profits. Both reserved requests and exchangeable requests are assigned one by one in the order of their list. After assigning all reserved requests to their own carriers, exchangeable requests are assigned to carriers according to the following two policies:

*Policy one:* Each exchangeable request is assigned to a route of its closest carrier. That is, each exchangeable request is assigned to the carrier with the minimum total distance of pickup and delivery nodes of the exchangeable request to the depot of the carrier.

*Policy two:* Each exchangeable request is assigned to a route of its best carrier, i.e. the carrier that can serve this request at the minimal additional cost (the lowest insertion cost).

In the heuristic, the two sorted lists, list of reserved requests and list of exchangeable requests are explored successively. If it is not feasible to insert both of the pickup and of delivery nodes of a request into the current route, a new route is created to insert the request. This process will continue until all requests are served and a feasible solution is reached.

### C. Solution Representation

To achieve good results of the proposed hybrid algorithm, the structure (coding) of each solution (chromosome) has an important role. We code a solution of the multi-carrier CTP by three vectors X, K, and Y.

Vector X specifies an order of pickup nodes and delivery nodes of all requests. The dimension of the vector is $|P \cup D|$. Each component of vector K corresponds to a request, which indicates the index of the vehicle that serves the request. Each component of vector Y also corresponds to a request, which indicates the carrier to which the request is assigned, i.e. the carrier that serves the request. The dimension of both K and Y is the same as the number of all requests. Note that the index of each request is the same as its pickup node index. To construct a solution from the three vectors, we take pickup and delivery nodes one by one from X, from its first component to its last component. For each pickup node taken, its assignment to each carrier and a route of the carrier respectively is determined according to vector Y and vector K. If a pickup node is assigned to a route of a carrier, its paired delivery node

must also be assigned to the same route while satisfying the precedence relation between the two nodes. If the insertion of a request into a route leads to an infeasible solution that violates either the vehicle capacity constraint or time window constraint, the request will be served by anew route if such new route can be created.

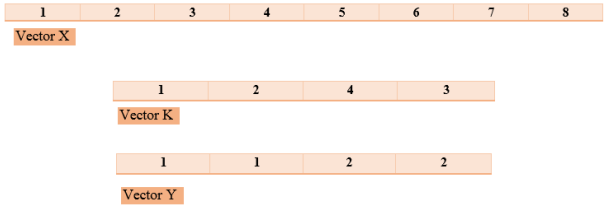


Fig. 2 Representation of Chromosome's structure

The coding of a solution by the three vectors is illustrated in Fig. 2 for a small example of multi-carrier CTP. In Fig. 2, X, K, and Y are vectors of dimension 8, 4, and 4, respectively. Vector Y indicates that request 1 and request 2 are assigned to carrier 1, and request 3 and request 4 are assigned to carrier 2. Vector K indicates that request 1 is served by vehicle (route) 1, request 2 is served by vehicle 2, request 3 is served by vehicle 4, and request 4 is served by vehicle 3. Actually there are totally 4 vehicles. Vector X indicates a sequence of pickup nodes and delivery nodes of all requests. The total number of pickup and delivery nodes of the example is eight. Therefore, the dimension of the vector X is eight. Note that for each request *i*, its pickup node and delivery node are node *i* and node *i*+4, respectively.

$$\text{Carrier1:}\begin{cases}\text{the index of the reserved request: } 1\\ \text{the pickup node index of reserved request: } 1\\ \text{the delivery node index of reserved request: } 5\\ \text{the index of the exchangeable request: } 2\\ \text{the pickup node index of exchangeable request: } 2\\ \text{the delivery node index of exchangeable request: } 6\end{cases}$$

$$\text{Carrier2:}\begin{cases}\text{the index of the reserved request: } 3\\ \text{the pickup node index of reserved request: } 3\\ \text{the delivery node index of reserved request: } 7\\ \text{the index of the exchangeable request: request: } 4\\ \text{the pickup node index of exchangeable request: } 4\\ \text{the delivery node index of exchangeable request: } 8\end{cases}$$

### *D.Genetic Operators*

Considering the structure of each solution of the proposed model, two crossover operators and two mutation operators are proposed.

1. Crossover Operator of Vector X

Since the vector X has a permutation structure, a single point crossover can be defined without requiring any extra operation to make its generated offspring's chromosomes feasible. Actually, it is necessary to use such single point crossover operator to avoid any gene repetition. The crossover point is chosen randomly in the vector. To generate the first offspring, all of the genes of a chromosome before the crossover point are transferred sequentially and compose the first part of the offspring's chromosome. In order to compose the rest genes of the offspring's chromosome, at first, the genes of the second parent are compared with the first part of the offspring. All of the repetitive genes of the second parent in the offspring will be ignored, and non-repetitive genes of the second parent complete the rest genes of the first offspring's chromosome. To generate the second offspring, the above-mentioned approach of gene selection is done in the opposite direction. Actually, in the second offspring, the parent selected to produce the first part of the offspring before the crossover point is the second parent. Fig. 3 shows the crossover of two X vectors and the generation of their two off-springs after the application of the crossover operator. It is necessary to mention that, in Fig. 3, the two arrows indicate the crossover point. According to the single point crossover specification over permutation chromosomes, in the first offspring, after transferring the genes of the first parent before the crossover point, the rest genes of the offspring after the crossover point will be chosen from the genes of the second parent by comparing with the genes of the offspring and removing the repetitive genes. In Fig. 3, the genes of the offspring after the crossover point are the last three genes.

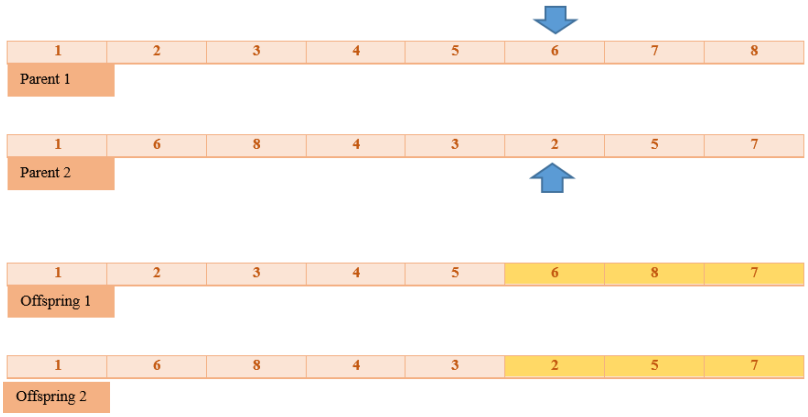


Fig. 3 The single point crossover over two X vectors

2. Crossover Operator of Vector Y and K

A crossover operator suitable for K and Y is uniform crossover. For the uniform crossover, after producing a mask with the same size of the parent vector and with zero and one genes, each offspring's gene is selected according to the value of the mask vector and the same indexed gene of parent. Fig. 4 illustrates the crossover over two K vectors. The crossover over two Y vectors is the same as the crossover over two K vectors. To generate the first offspring, if a gene of the mask vector is zero, the gene of the first offspring is chosen from the second parent and if the gene of the mask vector is one, the gene of the first offspring is chosen from the first parent. To create the second offspring, the same operation is performed but in the opposite direction. It means that, to produce the second offspring, when the gene of the mask vector is one, the gene of the second offspring is chosen from the second parent with the same index of the chosen gene and when the gene of the mask vector is zero, the gene selected to create the second offspring's gene is the first parent's gene.

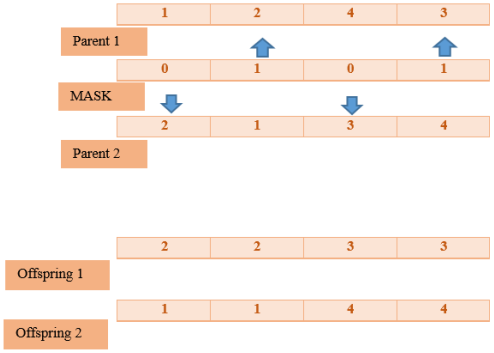


Fig. 4 The crossover over two K vectors

3. Mutation over Vector X

Because vector X has a permutation structure, a mutation operation with the following two steps can be applied to generate a diversified feasible solution.

1) Choose two elements of the vector X randomly
2) Choose randomly one of the insertion, swap and reversion operations and perform the operation on the selected elements.

4. Mutation over Vectors K and Y

The mutation over vector *K* is realized in three steps. In the first step, the number of elements in *K* to be mutated, denoted by *l*, is first determined randomly. This number is obtained in the following way: firstly, an integer number *h* is randomly generated between 1 and dim[*K*], where dim[*K*] is the number of elements of vector *K*. This number is multiplied by a mutation rate *r*, leading to *hr*. The number *l* is then obtained by rounding *hr* to the least integer number larger than or equal to it. In the second step, *l* elements are randomly chosen from *K*. In the third step, for each element of *K* chosen, an integer number is randomly generated between 1 and *k*, and the element in *K* is changed to this number, where *k* is the total number of vehicles. The mutation over vector *Y* is similar to that of vector *K* except that in the third step, for each element of *Y* chosen, an integer number is randomly generated between 1 and *M*, and the element in *Y* is changed to this number, where *M* is the number of carriers. In Fig. 5, the mutation over vector K is illustrated. The number of elements of K to be mutated is one, and the second gene with green color is chosen to be mutated. After the mutation, an offspring chromosome is transformed from the parent chromosome with new values in some genes.

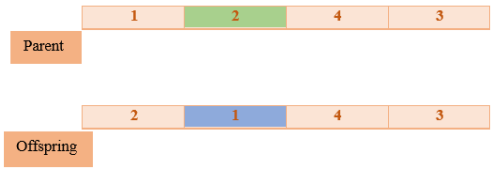


Fig. 5 The mutation operator over vector K

## V. Simulated Annealing in GASA

In comparison with most heuristics that only accept an improved solution, SA accepts a worse solution as well. If $f(so')$ is an objective function value in the new population and $f(so)$ is objective function value in the previous population, the acceptance probability of the new solution is $e^{-(f(o)-f(so'))/T}$ when $f(so') < f(so)$ for a maximization problem. The probability for profit decreasing depends on both the profit disparity $f(so') - f(so)$ and parameter $T$ called temperature. At the beginning of simulated annealing, $T$ is set to $T_{in}$. $T_{in}$ is computed so that a solution with profit 30% lower than that of the initial solution is accepted with a given possibility $pr$.The temperature $T$ of SA is decreased after each iteration. This reduction is done by multiplying $T$ and cooling factor $co \in (0,1)$. To achieve a slow cooling, the cooling factor must be close to one. Generally, at the end of each iteration of GA, after sorting the solution generated by crossover and mutation operations, we merge them with the current population in order to generate the next generation. In GASA, the transformation of offspring into a solution (chromosome) in the next generation is done by applying the SA rule. This means that, after merging the solutions obtained by crossover and the solutions obtained by mutation and sorting them, the solutions are compared with those in the current population by applying the SA rule. In this way, each solution has an opportunity to be selected. At the end of each iteration of the genetic algorithm, the temperature of SA is decreased like whatever is done in any simulated annealing algorithm.

## VI. Computational Experiments

To assess the performance of the GASA algorithm, the instances of [2] are utilized. All requests including reserved requests and exchangeable requests are the same as [2]. The GASA algorithm is executed on 50 instances with different specifications. The specifications of instances like the number of requests and the number of vehicles are different. The GASA is compared with Cplex 12.6 MILP solver and the

results of [2]. For large and medium size instances as the authors expected, it is not possible to get the solution with Cplex even by putting 5-hour time limitation, whereas the GASA algorithm could achieve the solution in a reasonable time. At last, the gap between the presented algorithm, GASA, the results of Cplex, and the results of [2] confirm the efficiency of the algorithm.

### *A. Calibration of Parameters*

The solution of GASA and its execution time depend on the parameters quantity. Taguchi tunes some critical parameters that have been illustrated in Table II. Genichi Taguchi, Japanese electrical engineer, has developed a method to optimize the process of producing the products. His proposed approach is a robust method to produce the products in a low cost. Moreover, the approach minimizes the deviation in the quality attributes of the products. The aim of the Taguchi Method (TM) is optimizing the effects of the factors on the mean value. The factors are controllable factors and have an effect on the robustness of the objective function as well. In fact, their effectiveness on the objective function value is consideration of the noise factors according to use statistical design of experiments (DOE). To decrease the variation of the objective function value, firstly, the levels of the control factors are determined. After the determination of the control factor levels, mean value have to be adjusted to the target size. There are two different experimental designs to validate the experimental procedure of TM. Inner and outer arrays are the name of the experimental designs. In the inner array, all of the control factors are placed, whereas all noise factors are considered in the outer array. Each factor level combination of control factors from the inner array is tested against various combinations of the noise factors in the outer array, which can be referred to as "quasi-repetition". In actual fact, by using the TM, combination of control factors in each level is examined. The important point to consideration is considering the combination of control factors under the influence of noise factors. There is necessary to mention that the control factors deviation is combined with the noise factors thanks to repeated measures. In other words, the related uncertainty is called an inner noise factor and the aim of the designing the system is being robust against these noise factors. By increasing the number of factors, the number of experiments are increased and a large number of experiments is necessary and the designing will be complex. To solve this problem, orthogonal array is applied in TM to decrease the number of experiments while entire parameter space is studied. Taguchi proposed to use loss function to measure the deviation of the performance of the attributes from the desired target value. The loss function value is converted to signal-to-noise (S/N) ratio. To analyze the S/N ratio, three types of the performance features are used. smaller-the-better, larger-the-better, and nominal-the-best are the three category [9]-[11]. In this paper, authors utilize the third category; it means that smaller is better. If $n$ denotes the orthogonal arrays and $Y$ indicates response value, $S/N$ ratio is formulated in Eq. (23) and S/N ratios are given in the decibel (db) scale. In the formulation, y is the response variable and $n$ is the number of experiments.

$$\frac{S}{N} = -10 \times \log\left(\frac{S(Y^2)}{n}\right) \tag{23}$$

Table I shows the domain of each candidate parameter in different levels to tuning by Taguchi approach. The results of Taguchi tuning approach of the selected parameters are depicted in Table II. Fig. 6 shows signal to noise figures of Taguchi approach in different parameters and different levels in Minitab. In signal to noise figures according to the feature of signal to noise, the maximum value in each parameter is acceptable.

TABLE I
PARAMETER VALUES OF GASA TO BE DETERMINED BY TAGUCHI

| Calibrated parameters of GASA | npop | $P_c$ | $P_m$ | nit |
|---|---|---|---|---|
| Level | 2 | 2 | 1 | 3 |
| Value | 150 | 0.6 | 0.05 | 300 |

TABLE II
DETERMINED LEVEL OF TUNING PARAMETERS OF GASA

| Parameter | Description | Value range | |
|---|---|---|---|
| *npop* | Number of population | *lower bound*(1): | 100 |
| | | *median* (2): | 150 |
| | | *upper bound*(3): | 200 |
| $P_c$ | Crossover probability | *lower bound*(1): | 0.4 |
| | | *median* (2): | 0.6 |
| | | *upper bound*(3): | 0.8 |
| $p_m$ | Mutation probability | *lower bound*(1): | 0.05 |
| | | *median* (2): | 0.175 |
| | | *upper bound*(3): | 0.3 |
| *nit* | Number of iteration | *lower bound*(1): | 100 |
| | | *median*(2): | 200 |
| | | *upper bound*(3): | 300 |

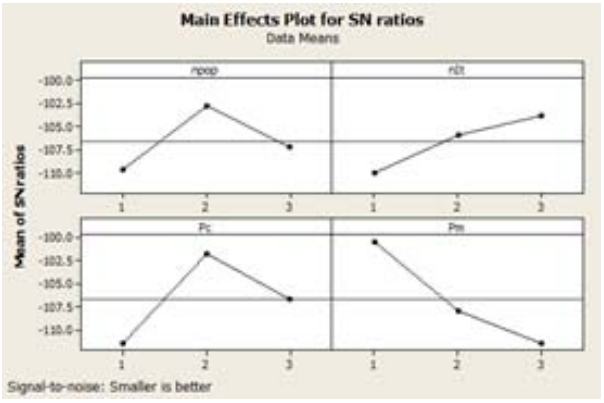


Fig. 6 The plot of S/N ratio at different levels of the selected parameters of GASA

### *B. Experimentation and Results*

After determining the value of parameters by calibration, all instances of [2] were executed with GASA algorithm and Cplex. Because carrier collaborative transportation is Np-hard, the time limitation of execution was set in Cpelx for large size and medium size instances to compare the solution of GASA

with Cplex results. Finally, the results of GASA and Cplex were compared according to their gap. Equations (24) and (25) are used to formulate the gap between GASA, Cplex, and the results of [2].

$$GAP_{GASA,Chen} = (\cos t_{GASA} - \cos t_{chen}) / \cos t_{chen} \quad (24)$$

$$GAP_{GASA,CPLEX} = (\cos t_{GASA} - \cos t_{cplex}) / \cos t_{cplex} \quad (25)$$

The results of different size instances are illustrated in the following tables. Tables III and IV indicate the comparison results of GASA with Cplex and [2] for small size instances. Table V includes the results of comparing the presented approach with Cplex and [2] for medium size instances. Tables VI and VII consists of the comparing results for large instances.

TABLE III
THE RESULTS OF SMALL SIZE INSTANCES (3×5×2)

| Instance number | $GAP_{GASA,Cplex}$(%) | $GAP_{GASA,Chen}$(%) | $CPU_{GASA}$(s) |
|---|---|---|---|
| **1** | 0 | 0 | 88.377 |
| **2** | 0 | 0 | 96.020 |
| **3** | 0 | 0 | 94.436 |
| **4** | 0 | 0 | 86.733 |
| **5** | 0 | 0 | 91.930 |
| **6** | 0 | 0 | 86.646 |
| **7** | 0 | 0 | 91.932 |
| **8** | 0 | 0 | 92.345 |
| **9** | 0 | 0 | 90.213 |
| **10** | 0 | 0 | 89.638 |

TABLE IV
THE RESULTS OF SMALL SIZE INSTANCES (3×8×2)

| Instance number | $GAP_{GASA,Cplex}$(%) | $GAP_{GASA,Chen}$(%) | $CPU_{GASA}$(s) |
|---|---|---|---|
| 1 | 0 | 0 | 263.644 |
| 2 | 0 | 0 | 252. 293 |
| 3 | 0 | 0 | 289.829 |
| 4 | 0 | 0 | 271.758 |
| 5 | - | 0 | 193.839 |
| 6 | - | 0.093 | 240.213 |
| 7 | - | 0 | 234.032 |
| 8 | - | 0 | 265.126 |
| 9 | - | 0.026 | 229.674 |
| 10 | - | 0 | 191.011 |

TABLE V
THE RESULTS OF SMALL SIZE INSTANCES (3×15×4)

| Instance number | $GAP_{GASA,Cplex}$(%) | $GAP_{GASA,Chen}$(%) | $CPU_{GASA}$(s) |
|---|---|---|---|
| 1 | | 0.071 | 691.537 |
| 2 | - | 0.086 | 594.417 |
| 3 | - | 0.122 | 638.526 |
| 4 | - | 0.033 | 631.245 |
| 5 | - | 0.018 | 654.513 |
| 6 | - | 0.177 | 597.204 |
| 7 | - | 0.088 | 576.208 |
| 8 | - | 0.094 | 619.129 |
| 9 | - | 0.042 | 647.118 |
| 10 | - | 0.053 | 612. 227 |

TABLE VI
THE RESULTS OF SMALL SIZE INSTANCES (3×30×6)

| Instance number | $GAP_{GASA.Chen}$(%) | $CPU_{GASA}$(s) |
|---|---|---|
| 1 | 0.077 | 1346.392 |
| 2 | 0.108 | 1247.729 |
| 3 | 0.086 | 1308.046 |
| 4 | 0.150 | 1281.783 |
| 5 | 0.057 | 1319.637 |
| 6 | 0.066 | 1307.521 |
| 7 | 0.181 | 1384.930 |
| 8 | 0.179 | 1296.012 |

TABLE VII
THE RESULTS OF SMALL SIZE INSTANCES (3×50×8)

| Instance number | $GAP_{GASA.Chen}$(%) | $CPU_{GASA}$(s) |
|---|---|---|
| 1 | 0.152 | 8315.214 |
| 2 | 0.207 | 7987.192 |
| 3 | 0.032 | 8422.596 |
| 4 | 0.150 | 8721.154 |
| 5 | 0.027 | 8515.219 |
| 6 | 0.204 | 8852.186 |
| 7 | 0.118 | 8911.88 |
| 8 | 0.131 | 8687.173 |

According to the results and calculating gaps, the effectiveness of the proposed GASA algorithm is proven. In small size instances (3×5), the GASA could obtain the exact solution, and the gap between the algorithm and Cplex is zero. In other small size instances like (3×8), in spite of Cplex, GASA could reach the solution for all instances of (3×8). In rest of the instances, (3×15), (3×30), and (3×50), GASA could reach the solution, whereas Cplex, even by setting time limitations, could not. Furthermore, the gap between the proposed GASA algorithm and the results of [2] is not prominent in all type of instances. All these results describe the efficiency of the GASA in comparison with commercial solver and decentralized approach in a reasonable time.

## VII. CONCLUSION

In this paper, the authors focused on the multi-carriers collaboration transportation with pickup and delivery and hard time windows. In this article, a model was presented for centralized collaborative transportation according to the real-world constraints. In the presented model, all of the vehicles of each depot in all carriers are homogenous and the vehicles have limit capacity. The presented model was mixed integer linear programming mathematical model, and the object was minimization of costs. This problem is Np-hard because it is a special case of vehicle routing problem. Hence, it is not possible to reach the solution in large size instances by commercial solvers. A hybrid algorithm of GASA algorithm with heuristic approach to find initial solution solved the large sizes of the model. The Taguchi-tuning approach was applied to parameter calibration due to the important roles of parameters in heuristic and metaheuristic algorithm to reach the acceptable solution in the reasonable time. By comparing the results with the solution of Cplex and decentralized collaborative transportation of [2], the effectiveness of the

proposed GASA algorithm was confirmed. In all of the instances, Cplex cannot reach the solution, whereas the proposed GASA algorithm could achieve solution. Moreover, its effectiveness by comparing the results of [2] in less time is conspicuous. As a future work, it can be considered as periodic aspect of collaborative transportation that is much closer to the real world applications.

## REFERENCES

[1] M. Zhang. S. Pratap. G. Q. Huang. and Z. Zhao. "Optimal collaborative transportation service trading in B2B e-commerce logistics." *International Journal of Production Research.* vol. 55. pp. 5485–5501. 2017.

[2] H. Chen. "Combinatorial clock-proxy exchange for carrier collaboration in less than truck load transportation." *Transportation Research Part E*. vol. 91. pp. 152-172.

[3] S. D'Amours. and M. Rönnqvist. "Issues in collaborative logistics." *Energy. Natural Resources and Environmental Economics.* Part 3. Springer.

[4] L. Houghtalen. "Designing allocation mechanisms for carrier alliances." *PhD Thesis. Georgia Institute of Technology*. USA. 2007.

[5] E. Markakis. and A. Saberi. "On the core of the multi-commodity flow game ." EC 03: *Proceedings of the 4thACM Conference on Electronic Commerce*. *ACM Press*. New York. USA. PP. 93-97. 2003.

[6] S. Hernandez. and S. Peeta. "A centralized time dependent multiple less-than-truckload carrier collaboration problem." *Transportation Research Board*. 2007. Accepted. available at the website: http://amonline.trb.org/12juh5/12juh5/1

[7] Y. Li. H. Chen. and C. Prins. "Adaptive large neighborhood search for the pickup and delivery problem with time windows. profits. and reserved requests." *European Journal of Operational Research*. vol. 252. PP. 27–38. 2016.

[8] J. H. Holland. "Adaptation in natural and artificial systems:An introductory analysis with applications to biology." *control and artificial intelligence. Michigan: University of Michigan Press.* 1975.

[9] S. H. Karna. and R. Sahi. "An Overview on Taguchi Method." *International Journal of Engineering and Mathematical Sciences.* vol.1. pp.11-18. 2012.

[10] R. Rao. Sreenivas; R. S. Prakasham. K. Krishna Prasad. S. Rajesham. P. N. Sarma. and L. VenkateswarRao. "Xylitol production by Candida sp.: parameter optimization using Taguchi approach." *Process Biochemistry*. vol. 39. pp. 951-956. 2004.

[11] G. Taguchi. "introduction to Quality Engineering-Designing Quality into Products and Processes." *Asian Productivity Organization.* Tokyo. 1986.