A Hybrid Genetic and Simulation Annealing Approach for a Multi-period Bid Generation Problem in Carrier Collaboration

Elham Jelodari Mamaghani, Haoxun Chen and Christian Prins
Industrial Systems Optimization Laboratory, Charles Delaunay Institute and UMR CNRS 6281, University of Technology of Troyes, Troyes 10004, France

Keywords: Carrier Collaboration, Bid Generation, Periodic Vehicle Routing Problem, Pickup and Delivery, Profit.

Abstract: In this article, a new vehicle routing problem appeared in carrier collaboration via a combinatorial auction (CA) is studied. A carrier with reserved requests wants to determine within a time horizon of multi periods (days) which requests to serve among a set of selective requests open for bid of the auction to maximize its profit. In each period, the carrier has a set of reserved requests that must be served by the carrier itself. Each request is specified by a pair of pickup and delivery locations, a quantity, and two time windows for pickup and delivery respectively. The objective of the carrier is to determine which selective requests may be served in each period in addition of its reserved requests and determine optimal routes to serve the reserved and selective requests to maximize its total profit. For this NP-hard problem, a mixed-integer linear programming model is formulated and a genetic algorithm combined with simulated annealing is proposed. The algorithm is evaluated on instances with 6 to 100 requests. The computational results show this algorithm significantly outperform CPLEX solver, not only in computation time but also in solution quality.

1 INTRODUCTION

In collaborative logistics, carriers may exchange some of their transportation demands in order to improve their profitability (Hernández et al., 2011). In this article, we consider collaboration among multiple carriers through exchanging some of their requests. The goal of this collaboration is to maximize the total profit of all carriers and generate more profit for each carrier. The carrier collaboration is usually realized in two steps. The first step is the re-assignment of a part of requests called selective requests among carriers and the second step is the sharing of the profit among carriers (Dai et al., 2015).

Combinatorial Auction (CA) is an approach for request re-assignment among carriers. In a multi-round CA, in each round (iteration), the service price for each selective request is updated by an auctioneer (Dai et al., 2014). Each carrier determines which selective requests to serve in addition to its reserved requests to maximize its own profit by solving a bid generation problem. In real world applications, carriers usually plan their pickup and delivery operations and use of vehicle resources in advance (several days ago) and in a rolling horizon way (Wang et al., 2014), (Wang et al., 2015). This requires that each carrier considers multiple periods (days) when it determines which transportation requests to bid and serve in each period (day). Moreover, requests open for bid (requests to be exchanged among carriers) may span across multiple periods (days). That is, instead of fixing a day for serving each of the requests, each request is allowed to be served within a service day window consisting of multiple consecutive days. An important application of multi-period BGP is in e-commerce. For example, goods ordered on-line by a customer on Monday is asked to deliver to the home of the customer within three days from Tuesday to Thursday. This gives rise to a multi-period combinatorial auction (CA) problem. In this article, a multi-period Bid Generation Problem (BGP) for a carrier is considered. In the problem, there are two different types of requests, reserved requests of the carrier and selective requests. The carrier is committed by contracts with its shippers to serve all reserved requests by itself. The selective requests are offered by other carriers and are opened for bid by the carrier. Each request is specified by a pair of pickup and delivery locations, a pickup/delivery quantity, and two time windows for pickup and

Mamaghani, E., Chen, H. and Prins, C.
A Hybrid Genetic and Simulation Annealing Approach for a Multi-period Bid Generation Problem in Carrier Collaboration.
DOI: 10.5220/0007369203070314
In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems (ICORES 2019)*, pages 307-314
ISBN: 978-989-758-352-0

delivery, respectively. The pickup/delivery time window of a request specifies the earliest and the latest time at which the pickup/delivery operation of the request must be performed in each period. In addition, each selective request has a period window which specifies the earliest period and the latest period between which the request must be served. Moreover, each selective request is associated with a profit that is the price for serving the request provided by a shipper. By considering multiple periods in CA, the carrier can plan its transportation operations in advance and in a rolling-horizon way. A carrier must make two important decisions in its BGP: Which requests are chosen to bid and serve within their service period windows and how the routes are constructed to maximize its total profit. This leads to a new periodic pickup and delivery problem with time windows, profits and reserved requests. According to Wang and Kopfer (2014), the presented problem is NP-hard and it is impossible to get an optimal solution for large instances by using a commercial solver like CPLEX. Hence, a hybrid approach combined genetic algorithm and simulated annealing (GASA) is proposed to solve the problem. The numerical results demonstrate the proposed algorithm can find a good feasible solution in a reasonable computation time for large instances.

The rest of the paper is organized as follows. Section 2 is devoted to literature review. A detailed description of a mathematical model is given in Section 3. In section 4, the GASA algorithm is described. In section 5, detailed numerical results of solving the model by GASA and CPLEX solver on instances is presented and compared. The final section concludes this paper with some remarks for future research.

2 LITERATURE REVIEW

Collaborative Transportation Management (CTM) is achieved through the horizontal collaboration between multiple shippers or carriers by either sharing transport capacities or transportation orders. With the collaboration, all actors involved can improve their profitability by eliminating empty backhauls and raising vehicle utilization rates (Dai and Chen, 2011). (D'Amours and Rönnqvist, 2010) present a survey of previous contributions in the field of collaborative logistics. Indeed, efficient utilization of vehicle capacity and reducing the number of vehicles through carrier collaboration is noticeable in Less than Truck Load (LTL) transportation. With this type of collaboration, operation efficiency will increase (Hernández et al., 2011). The considered problem in the current paper is a bid generation problem with multi periods in collaborative transportation. The bid generation problem (BGP) which is considered from the perspective of each carrier is the request selection problem and a key decision problem for auction-based decentralized planning approaches in CTP. (Lee et al., 2007) study the carrier's optimal BGP in combinatorial auctions for transportation procurement in TL (truckload) transportation. Carriers employ vehicle routing models to identify sets of lanes to bid for based on the actual routes. (Buer, 2014) proposes an exact strategy and two heuristic strategies for bidding on subsets of requests. The model proposed in this paper is a multi-period extension of the model proposed in (Li et al., 2016). Both of them assume the BGP of a carrier, but the BGP considered in this paper involves multi periods. There are two interesting studies in multiple periods BGP: (Wang et al., 2014), (Lau et al., 2007). In these papers, each carrier considers multiple periods (days) when it determines which transportation requests to bid and serve in each period (day). Moreover, requests open for bid may span across multiple periods (days). Other works related to ours include studies on the Team Orienting Problem (TOP). Multiple vehicle routing problem with profits is called Team Orienting Problem (TOP) (Chao et al., 1996) focus on the TOP by considering multiple tour maximum collection problem and multiple tour VRP with profits. (Yu et al., 2010) utilize a simulated annealing algorithm to solve a capacitated location routing problem.

3 PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

In this problem, we consider a carrier who wants to determine which requests to bid (select) among all requests open for bid (offered by all carriers) in a combinatorial auction to maximize its own profit by solving a bid generation problem. Since the carrier plans its transportation operations in advance and in a rolling horizon way as mentioned in the introduction, this bid generation problem involves multiple periods. We consider the problem in the less-than-truck load transportation, where each transportation request is a pickup and delivery request with time windows, two types of requests-reserved requests and selective requests are involved, and each request is associated with a profit which is the revenue provide by a shipper to serve the request. Formally, the multi-period bid generation problem can be defined on a directed graph $G = (N, E)$ where N is the set of all nodes comprising all pickup, delivery nodes and the depot node of the carrier and E is the set of edges. The node set is defined as $N = \{0, \ldots, 2n+1\}$, where n represents the number of requests, 0 and $2n+1$ both denote the depot of the carrier, i and $n+i$ represent the pickup point and the delivery point of request $i = 1, 2, \ldots, n$. Let W denote the set of nodes excluding the depot node. The set of periods denoted by H. In the problem, the carrier has a finite fleet of homogenous vehicles whose index set is given by $K = \{1, 2, \ldots, VK\}$ where VK is the maximum number of vehicles. The capacity of each vehicle is denoted by Q and the load of each vehicle cannot exceed its capacity. t_{ij} and c_{ij} are the travelling time and the transportation cost from node i to node j, respectively. We assume $t_{ij} = c_{ij}$. The set of pickup and delivery nodes of all requests are denoted by $P = \{1, 2, \ldots, n\}$ and $D = \{n+1, \ldots, 2n\}$, respectively. Each request i has its pickup node i and its delivery node $n+i$. The demand of the pickup node of request i is denoted by d_i, while the demand of the delivery node of the same request is denoted by d_{i+n}, $d_{i+n} = -d_i$. The delivery node of each request must be visited after its pickup node on the same route. The set of all requests is denoted by R, where $R = (\bigcup_{l \in H} R_{rl}) \cup R_s$. R_{rl} is the set of reserved requests that must be served in period l, R_s is the set of selective requests and H is the set of periods. Each selective request has a service period window and two time windows. The service period window determines which periods the selective request can be served, and the two time windows determine at which times in each period the pickup node and the delivery node of the request can be visited by a vehicle that serves the request. Both selective and reserved requests are associated with two time windows, whereas only selective requests are associated with a service period window (the period in which each reserved request must be served is pre-specified). The time window of pickup node i and delivery node $i+n$ of request i are denoted by $[e_i, l_i]$ and $[e_{i+n}, l_{i+n}]$, respectively. The service period window for each selective request i is represented by $[E_i, L_i]$. Each reserved request $i \in R_{rl}$ must be served in its pre-specified period l, $l \in H$. The maximum duration of each route is limited by T. The multi-period bid generation problem can be formulated as a mixed-integer linear programing model. In the model, parameters $BM_{ij} = l_j - e_i$ is used to formulate linearly the time window constraints. The decision variables of the model include binary variables, x_{ijkh} and y_{ikh} and real variables U_{ikh} and CV_{ikh} are defined as follows.

$$x_{ijkh} = \begin{cases} 1 & \text{if and only if vehicle k visits directly node j after node i in period h} \\ 0 & \text{else} \end{cases}$$

$$y_{ikh} = \begin{cases} 1 & \text{if and only if request i is served by vehicle k in period h} \\ 0 & \text{else} \end{cases}$$

$U_{ikh} =$ arrival time of vehicle k at node i in period h

$CV_{ikh} =$ Load of vehicle k when it leaves node i in period h

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

$$\max \sum_{i \in R} \sum_{k \in K} \sum_{h \in H} p_i y_{ikh} - \sum_{k \in N} \sum_{i \in N} \sum_{j \in N} \sum_{h \in H} c_{ij} x_{ijkh}$$

Subject to:

$$\sum_{j \in N, j \neq i} x_{jikh} - \sum_{j \in N, j \neq i} x_{ijkh} = 0 \quad \forall i \in (P \cup D), \forall k \in K, \forall h \in H \tag{1}$$

$$\sum_{k \in K} \sum_{h \in H} \sum_{j \in P, j \neq 0} x_{0jkh} \geq 1 \tag{2}$$

$$\sum_{\forall k \in K} \sum_{h \in H} \sum_{i \in D, i \neq 2n+1} x_{i,2n+1,k,h} \geq 1 \tag{3}$$

$$\sum_{k \in K} y_{ikl} = 1 \quad \forall i \in R_{rl}, \forall l \in H \tag{4}$$

$$\sum_{h \in [E_i, L_i]} \sum_{k \in K} y_{ikh} \leq 1 \quad \forall i \in R_s \tag{5}$$

$$\sum_{j \in N, j \neq i, 0} x_{j,n+i,h} = y_{ikh} \quad \forall i \in P, \forall k \in K, \forall h \in H \tag{6}$$

$$\sum_{j \in N, j \neq i, 2n+1} x_{ijkh} = y_{ikh} \quad \forall i \in P, \forall k \in K, \forall h \in H \tag{7}$$

$$U_{ikh} + t_{i,n+i} \leq U_{i+n,k,h} \quad \forall i \in P, \forall k \in K, \forall h \in H \tag{8}$$

$$U_{jkh} \geq U_{ikh} + t_{ij} x_{ijkh} - BM_{ij}(1 - x_{ijkh}) \quad \forall i, j \in N, \forall k \in K, \forall h \in H \tag{9}$$

$$e_i y_{ikh} \leq U_{ikh} \leq l_i + (1 - y_{ikh}) \sum_{j \in N} BM_{ij} \quad \forall i \in N, \forall k \in K, \forall h \in H \tag{10}$$

$$U_{ikh} + t_{ij} - BM_{ij}(1 - x_{ijkh}) \leq T \quad \forall i \in W, \forall k \in K, \forall h \in H \tag{11}$$

$$CV_{jkh} \geq CV_{ikh} + d_j - CV_j(1 - x_{ijkh}) \quad \forall i, j \in N, \forall k \in K, \forall h \in H \tag{12}$$

$$\max\{0, d_i\} \leq CV_{ikh} \leq \min\{Q, Q + d_i\} \quad \forall i \in N, \forall k \in K, \forall h \in H \tag{13}$$

$$x_{ijkh} \in \{0,1\} \quad \forall i, j \in N, \forall k \in K, \forall h \in H \tag{14}$$

$$y_{ikh} = 0 \text{ for any } i \in R_s, h \notin [E_i, L_i] \text{ and } y_{ikh} = 0 \text{ for any } i \in R_rl, h \neq l \tag{15}$$

$$U_{ikh} \geq 0 \quad \forall i \in N, \forall k \in K, \forall h \in H \tag{16}$$

$$CV_{ikh} \geq 0 \quad \forall i \in N, \forall k \in K, \forall h \in H \tag{17}$$

The objective function represents the total profit of the carrier, which is equal to the difference between the total payments of serving requests in all periods and the total transportation cost. Constraint (1) ensures that when a vehicle arrives at a node in a period, it must leave from the node in the same period. Constraints (2) and (3) signify that each vehicle leaves its depot in a period must return to the depot in the same period. Equation (4) implies that each reserved request must be served in its pre-specified period. Equation (5) indicates that each selective request can be served in a period within its service period window or not served. Constraints (6) and (7) guarantee if a request is served in a period, its delivery node must be visited after its pickup node with the same vehicle in the same period. Equations (8)-(11) specify time window constraints on the pickup and delivery nodes of each request, and the constraint on the maximum duration of each route. Constraints (12)-(13) ensure vehicle capacity constraints. Equations (14)-(17) describe the variables.

4 METAHEURISTIC APPROACH TO SOLVE MULTI-PERIOD BID GENERATION PROBLEM

The Multi-Period Bid Generation Problem based on Pickup and Delivery with Time Windows, reserved requests and profits is NP-hard (Wang and Kopfer, 2014a) and special case of vehicle routing problem. Consequently, it is required to enforce metaheuristic algorithms to solve the problem. GASA is a metaheuristic algorithm used to solve the problem.

4.1 Initial Solution Construction Procedure

In the proposed hybrid genetic algorithm with simulated annealing (GASA), the size of population determines the number of initial solutions to construct. The initial solutions are constructed in the following three ways:

1. Only the reserved requests are served. All reserved requests are sorted in the decreasing order of their profits and the reserved request with the highest profit is firstly served. All reserved requests must be served in this case.
2. All reserved requests are served firstly and then selective requests are served. Both types of requests are sorted in the decreasing order of their profits and the reserved request with the highest profit is served at first. After serving all reserved requests, selective requests are served in the decreasing order of their profits within their service period windows. In this case, all selective requests must be served.
3. All reserved requests must be served first and selective requests are served only if its assigned period is not zero. That is, if the period assigned to the request is zero, it will be not served.

If the probability of choosing each of the three ways to construct an initial solution is denoted by α_1, α_2 and α_3, respectively, then $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

4.2 Hybrid Genetic Algorithm with Simulated Annealing (GASA)

One of the well-known metaheuristic algorithms inspired from the nature is Genetic Algorithm (GA) that is suggested by (Holland and H., 1992). Another well-known metaheuristic algorithm is Simulated Annealing (SA) that accepts a worse solution with a probability. Actually, diversification of search space is included in the algorithm by decreasing the probability of accepting worse solution. We apply hybrid algorithm of GA and SA that is called GASA to prevent GA from premature convergence. All components of GASA for multi-period CT with pickup and delivery and time window will be explained in the rest of this section. In each iteration of GASA, the profit is updated thanks to the obtained profit in the latest generation. SA is applied on each solution of the present population. Each solution is selected with a probability based on a simulated annealing procedure.

Simulated Annealing in GASA

Simulated Annealing gives a chance to worse solutions, which accepts a worse new solution with a probability. The new solution acceptance probability is given by $e^{-(f(s)-f(s'))/T}$ where $f(s')$ is the objective value of the new solution and $f(s)$ is the objective value of the current solution and $f(s') < f(s)$. The acceptance probability depends on both the profit decrease $f(s) - f(s')$ and the temperature parameter T which is decreased at each iteration. The temperature reduction is performed by multiplying T with a cooling factor $co \in (0,1)$. To attain a slow cooling, the cooling factor must be set close to one. At the beginning of the GASA algorithm, the temperature parameter T is set to T_0 and a solution with profit 30% lower than that of the initial solution is accepted with a given possibility pr. In GASA, to produce the next generation solutions, at the end of each GA iteration, the solutions generated by crossover and mutation are sorted and merged with the current population. GASA utilizes the SA rule to determine whether each solution in the sorted list becomes a solution (chromosome) in the next generation.

4.2.1 Solution Representation in GASA

The chromosome of multi-period BGP is defined by three vectors X, Y and Z. Vector X is included all pickup nodes and delivery nodes and its dimension is $|P \cup D|$ as the index of each request is the same as its pickup node index in three vectors. The size of vector Y is equal to the number of all requests, and each component of vector Y indicates the period assigned to a request. Since a selective request is not compulsory to be served, if it is not profitable, it will not be served. Vector Z is similar to vector Y and the dimension of Z is equal to the number of all requests. In vector Z, each component indicates the index of the vehicle serving a request. An extra period is introduced to indicate a selective request is not served. The extra period is referred to period 0.

4.2.2 Operators of GASA

The proposed chromosome has three vectors which two of them have same structure. Therefore, two different crossover and mutation operators are applied.

Crossover Operator of Vector X

A single point crossover is suitable to the solution structure with permutation specification. In the suggested chromosome to GASA, vector X has a permutation structure and single point crossover is used without need to use any extra operation to make its produced offspring's chromosomes feasible. In fact, by using single point crossover, it is prevented from creating repetition genes. To generate the first offspring after chosen the crossover point randomly in the vector X, all genes of first parent chromosome before the crossover point are sequentially transferred and creates the first part of the offspring's chromosome. To generate the genes after the crossover point, in the beginning, all genes of second parent are compared with the first part of first offspring chromosome. All non-repetitive genes of the second parent are transferred to construct the rest gens of first offspring and complete the chromosome of first offspring. This approach of gene selection in the opposite direction is done to produce second offspring.

Crossover operator of *vector Y and Z*

According to the structure of vector Z and Y, uniform crossover is suitable crossover operator. At first, a mask vector with the same size of parents is created with zero and one genes. The value of each offspring's gene is generated according to the value of mask vector and the same indexed gene of parent. The gene of first parent is transferred to the first offspring gene with the same index if the mask gene is one and the gene of second parent is transferred to first offspring if the mask gene is zero. To produce second offspring, the gene of second offspring has the same value of first parent when the mask vector gene with the same index is zero and the value of second offspring gene is chosen from second parent if the value of mask in the same index is one.

Mutation over Vector X

To have a diversified feasible solution, a mutation operation with the following two steps is suitable to the permutation structure vector X.

1) Choose randomly two genes of vector X
2) Choose randomly one of the relocation, swap and reversion operations and execute the selected operation on the selected genes.
    - Swap Operator
      Select two components of vector X and swap their positions in the vector.
    - Reversion Operator
      Selects two components of vector X and reverse the order of the components between the selected components.
    - Relocation Operator
      Select two components in vector X and

relocates one of them to the front of another component.

Mutation over Vectors Y and Z

The mutation over vector Z is proceeded in three steps. At first, the number of mutated genes of vector Z is determined randomly and illustrated by l. The number of mutated genes are obtained in the following step: The primer step is generating randomly integer number h between 1 and dim $[Z]$, as dim $[Z]$ is the number of genes in vector Z where VK is the maximum number of vehicles. This number is multiplied by a mutation rate r, leading to hr. By rounding hr to the least integer number larger than or equal to hr, the number l is obtained. In the second step, from Z, l genes are randomly selected. Finally, for each selected gene of Z, an integer number is randomly generated between 1 and VK, and the gene in Z is changed to this value. The mutation over vector Z is the same as mutation operation over vector Z with a difference in third step. In vector Y, for each chosen gene, an integer number is randomly generated between 0 and T where T is the maximum periods and the selected gene in Y is changed to this value. After the mutation, an offspring chromosome is modified from the parent chromosome with new values in some elements.

5 EXPERIMENTAL RESULTS

To evaluate the performance of the proposed metaheuristic algorithm, we applied it to solve instances of taken from (Li et al., 2016) with the same reserved requests and the selective requests, and compared them with the MILP solver of CPLEX 12.6 in terms of profit and computation time. We consider there are 5 periods and each period has its own reserved requests according to the random function. Note that for the instances more than 20requests it was impossible to solve the MILP model optimally by CPLEX after 2 hours.

5.1 Parameter Setting

The values of some parameters of the algorithm are determined empirically and are given in Table 1. The values of other parameters are tuned by using the Taguchi method (Semioshkina and Voigt, 2006) are given in Table 2.

Table 1: Parameter values of GASA determined empirically.

Parameter	Description	Value
SubIt	Number of iteration of SA	110
μ_m	Mutation rate	0.26

Table 2: Parameter values of GASA determined by Taguchi.

Methodology	Parameter	Description	Value
GA	Npop	Size of population	150
	nIt	Number of iterations	800
	Pc	Crossover probability	0.6
	Pm	Mutation probability	0.05

5.2 Test Results

After the parameters calibration, we executed the GASA algorithm and CPLEX on all instances. For CPLEX, since the considered carrier collaboration problem is NP-hard (Wang and Kopfer, 2014b) it is very time consuming to solve optimally large size instances. For this reason, we set a maximum running time for CPLEX to solve large size instance. The time limitation is 2 hours. Our proposed algorithm is compared with CPLEX based on the criterions defined in Table 3, where Obj_{GASA} and is the profits of the studied problem obtained by the algorithm, respectively; UB_{MILP} and LB_{MILP} are the upper bound and the lower bound of the objective value of the problem obtained by CPLEX in a preset computation time.

Table 3: Criterions used for comparison of GASA and Cplex.

Criterion	Description
$\mathrm{Gap}_{\mathrm{MILP}}$	The relative gap between $\mathrm{UB}_{\mathrm{MILP}}$ and $\mathrm{LB}_{\mathrm{MILP}}$, defined as $\dfrac{\mathrm{UB}_{\mathrm{MILP}} - \mathrm{LB}_{\mathrm{MILP}}}{\mathrm{UB}_{\mathrm{MILP}}}$
$\mathrm{Gap}_{\mathrm{GASA}}$	The relative gap between $\mathrm{UB}_{\mathrm{MILP}}$ and $\mathrm{Obj}_{\mathrm{GASA}}$, defined as $\dfrac{UB_{\mathrm{MILP}} - \mathrm{Obj}_{\mathrm{GASA}}}{\mathrm{UB}_{\mathrm{MILP}}}$

The computation results are given in Table 4 and Table 5. From Table 4, we can see, for small instances, CPLEX and GASA could find an optimal solution. However, for some 8 requests instances and CPLEX could not even find a feasible solution in a preset computation time, whereas GASA could find a feasible solution for all instances. So we compare the solutions obtained by GASA based on a relative gaps with the upper bound obtained by CPLEX, i.e., using the above-mentioned criteria. GASA can find an optimal solution for 6 requests instances, it can find a solution with the relative profit gap smaller than 0.34% for 8 request instances and 4.50% for 10 request instances. For instances with 20 requests, GASA can find a solution with the gap smaller than 6.98%. For instances with 30 requests, GASA can find a solution with the gap smaller than 10.31%. For instances with 40 and 50 requests, GASA can find a solution with the gap smaller than 11% and 13.2% sequentially. For the instances with 100 requests, GASA can find a solution with the gap smaller than 16.4%. From the results, we can see our proposed algorithms perform much better than CPLEX in terms of running time for medium and large instances.

Table 4: Computational results of GASA and CPLEX – part one.

Instance	LB_{MILP}	UB_{MILP}	$Profit_{GASA}$
6-3-3	256.5254	265.5254	265.5254
8-4-4	512.015	514.731	514.682
10-5-5a	909.810	984.660	974.614
10-5-5c	974.001	1028.626	1019.693
10-3-7d	1157.164	1257.153	1248.267
10-3-7f	-	1210.611	1183.689
20-10-10a	-	2619.785	2473.626
20-10-10c	-	3815.445	3651.975
20-5-15d	1832.804	2015.517	1991.431
20-5-15e	-	2568.903	2462.24
30-15-15c	-	7432.469	7003.483
30-10-20d	-	6629.284	6263.02
30-20-10g	-	11214.73	10452.39
40-20-20a	-	12065.1	11103.11
40-15-25d	-	12205.41	109481.51
40-25-15g	-	11115.574	99794.391
50-25-25a	-	23781.93	21105.171
50-20-30d	-	22083.155	19206.231
50-20-30f	-	20495.213	18004.147
100-50-50a	-	88406.782	75606.06
100-50-50b	-	90722.467	78110.724
100-25-75d	-	112103.589	93850.223
100-25-75f	-	90686.991	75815.239
100-75-25g	-	93663.471	78945.834
100-75-25h	-	82124.016	70237.949

Table 5: Computational results of GASA and CPLEX – part two.

Instance	Gap_{GASA}(%)	Gap_{MILP} (%)	CPU_{GASA}	CPU_{MILP}
6-3-3	0	0	85.117	67.649
8-4-4	0.009	0.333	100.794	93.365
10-5-5a	0.512	7.61	184.342	500
10-5-5c	0.869	5.311	173.110	500
10-3-7d	0.469	7.733	186.185	500
10-3-7f	0.729	2.003	177.961	500
20-10-10a	5.579	-	331.511	3600
20-10-10c	4.284	-	351.836	3600
20-5-15d	1.195	9.065	326.537	3600
20-5-15e	4.1520	-	322.419	3600
30-15-15c	5.771	-	547.992	3600
30-10-20d	5.524	-	5213.14	3600
30-20-10g	6.797	-	546.232	3600
40-20-20a	7.973	-	825.753	7200
40-15-25d	10.297	-	844.225	7200
40-25-15g	10.007	-	819.471	7200
50-25-25a	11.255	-	1719.124	7200
50-20-30d	13.027	-	1696.439	7200
50-20-30f	12.154	-	1690.491	7200
100-50-50a	14.479	-	2725.955	10800
100-50-50b	13.901	-	2728.801	10800
100-25-75f	16.398	-	2759.854	10800
100-75-25g	15.713	-	2916.492	10800
100-75-25h	14.473	-	2891.837	10800

6 CONCLUSIONS

In this article, a new vehicle routing problem appeared in collaborative logistics, the multi period pickup and delivery problem with time windows, reserved requests and selective, is considered. By solving this problem, a carrier determines which transportation requests to serve in a combinatorial auction. This problem has a new feature that each selective request has a service period window besides time windows to visit its pickup and delivery points. We have proposed a hybrid metaheuristic algorithm to solve the model. Numerical experiments on benchmark instances show that the algorithm can obtain optimal solutions for small instances and much better solutions for medium to large instances than CPLEX. CPLEX cannot find even a feasible solution for such instances in a preset computation time.

ACKNOWLEDGEMENT

This work is supported by the ANR (French National Research Agency) under the project

ANR-14-CE22-0017 entitled "Collaborative Transportation in Urban Distribution".

REFERENCES

Buer, T. (2014) 'An exact and two heuristic strategies for truthful bidding in combinatorial transport auctions'. Available at: http://arxiv.org/abs/1406.1928.

Chao, I.-M., Golden, B. L. and Wasil, E. A. (1996) 'The team orienteering problem', *European Journal of Operational Research*, 88(3), pp. 464–474.

D'Amours, S. and Rönnqvist, M. (2010) 'Issues in Collaborative Logistics', in. *Springer*, Berlin, Heidelberg, pp. 395–409.

Dai, B. *et al.* (2015) 'Proportional egalitarian core solution for profit allocation games with an application to collaborative transportation planning', *European J. Industrial Engineering*, 9(1), pp. 53–76.

Dai, B. and Chen, H. (2011) 'A multi-agent and auction-based framework and approach for carrier collaboration', *Logistics Research*. Springer Berlin Heidelberg, 3(2–3), pp. 101–120.

Dai, B., Chen, H. and Yang, G. (2014) 'Price-setting based combinatorial auction approach for carrier collaboration with pickup and delivery requests', *Operational Research*, 14(3), pp. 361–386.

Hernández, S., Peeta, S. and Kalafatas, G. (2011) 'A less-than-truckload carrier collaboration planning problem under dynamic capacities', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 47, Issue 6, pp.933-946

Holland, J. H. (John H. and H., J. (1992) Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press..

Lau, H. C. et al. (2007) 'Multi-Period Combinatorial Auction Mechanism for Distributed Resource Allocation and Scheduling', in 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07). IEEE, pp. 407–411.

Lee, C. G., Kwon, R. H. and Ma, Z. (2007) 'A carrier's optimal bid generation problem in combinatorial auctions for transportation procurement', Transportation Research Part E: Logistics and Transportation Review, 43(2), pp. 173–191.

Li, Y., Chen, H. and Prins, C. (2016) 'Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests', *European Journal of Operational Research*. Elsevier B.V., 252(1), pp. 27–38.

Mengzhu Shi et al. (2011) 'Hybrid Genetic Tabu Search Simulated Annealing Algorithm and its application in vehicle routing problem with time windows', in 2011 *2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC). IEEE*, pp. 4022–4025.

Semioshkina, N. and Voigt, G. (2006) 'An overview on Taguchi Method', J*ournal Of Radiation Research*, 47 Suppl A(2), pp. A95–A100.

Wang, C. et al. (2015) A Rolling Horizon Auction Mechanism and Virtual Pricing of Shipping Capacity for Urban Consolidation Centers Urban Consolidation Centers. Available at: http://ink.library.smu.edu.sg/sis_research).

Wang, C., Handoko, S. D. and Lau, H. C. (2014) 'An auction with rolling horizon for urban consolidation centre', in *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics. IEEE*, pp. 438–443.

Wang, X. and Kopfer, H. (2014a) 'Collaborative transportation planning of less-than-truckload freight: A route-based request exchange mechanism', *OR Spectrum*, 36(2), pp. 357–380.

Yu, V. F. et al. (2010) 'A simulated annealing heuristic for the capacitated location routing problem', Computers & Industrial Engineering. *Pergamon Press, Inc.,* 58(2), pp. 288–299.