

Two-Echelon Multi-Trip Multi-Traffic Pickup and Delivery Problem with Time Windows and Synchronization

Quang Ngoc Nguyen*, Nghia Nguyen Duc, Phuong Khanh Nguyen

Hanoi University of Science and Technology, Hanoi, Viet Nam

*Email: quangnn@etc.vn

Abstract

In City Logistics context, multi-echelon distribution system normally supports multi-traffic: inbound, outbound, and intra-city traffic. Tackling multi-echelon model with multi-traffic simultaneously will take benefit from saving time and money. Therefore, we study a new problem class, the Two-Echelon Multi-Trip Multi-Traffic Pickup and Delivery Problem with Time Windows and Synchronization. We consider the synchronization process of transshipment at satellites between vehicles with the temporary storage mechanism. As the proposed problem consists of two echelons with two separate fleets of vehicles, we thus propose the bottom-up approach to solve the problem. The experiment is to study the proposed model and analyze the satellite synchronization.

Keywords: Two-Echelon, MTT-PDTWS, ALNS

1. Introduction and Literature Review

City Logistics (CL) literature and projects often address inbound movements only, reflecting the dominant position of the traffic proceeding from the exterior of the city towards its center. Yet, the volumes of freight produced within the city and shipped to locations outside it may be significant. Tackling multi-echelon model with multi-traffic simultaneously will therefore take benefit on saving time and money. Consequently, addressing the needs of these all different types of transportation demands in a multi-echelon freight distribution system may greatly contribute to achieving the CL mobility, environmental, and quality-of-life objectives.

To our best knowledge, Crainic et al. was the first to introduce the Multi-Trip Multi-Traffic Pickup and Delivery Problem with Time Windows and Synchronization (MTT-PDTWS) addressing all three types of traffic: the inbound movement (external-to-customer demands, e2c), outbound movement (customer-to-external demands, c2e), and internal movement within the city (customer-to-customer, c2c) [1]. In this paper, we aim to extend the MTT-PDTWS by introducing a new problem, the Two-Echelon Multi-Trip Multi-Traffic Pickup and Delivery Problem with Time Windows and Synchronization (2E-MTT-PDTWS). The extensions are three folds. First, the distribution system is two-echelon, thus requiring the need of using two different sizes of vehicles: the bigger size, fleet of urban-trucks transporting freight on the first echelon outside the city; and the smaller size, fleet of city-

freighter operating on the second echelon within the city. Second, the e2c and c2e demands are synchronized at satellites by not only exchanging them directly from the vehicles as in the MTT-PDTWS but also using temporary storage to store them for later transshipment in case the vehicles arrive late at satellites. It thus helps to reduce empty vehicles on the road but makes the problem harder to tackle. Third, we include the capacity of each satellite as a new constraint for the problem.

The decomposition approach is proposed to handle the problem. Experimental results show the solution quality and examine the satellite synchronization.

The remainder is organized as follows. Section 2 contains the problem description. The proposed methodology is described in Section 3. Computation results are then reported and analyzed in Section 4, while conclusions are considered in Section 5.

2. Problem Setting

In the 2E-MTT-PDTWS setting, two-echelon system consists of two homogeneous fleets of vehicles operating on each echelon to transport freight (1) between the central distribution center (CDC) located on the outskirts of the city and a set of customers C within the city through a set of satellite facilities S , and (2) between customers of set C . No direct shipping between the CDC and customers is allowed. Each satellite $s \in S$ has a no-wait, hard opening time window $[t(s) - \eta, t(s)]$ specifying the earliest and latest times a vehicle may arrive at s , respectively. Each customer may request different services at different periods of time: 1) receive e2c demands from different satellites, possibly within different time windows; 2) ask for c2e demands to be

picked up and transported to one of a given subset of satellites; 3) specific pairs of customers may ship c2c demands between them. We model the time dependency by identifying each above particular demand as customer demand and thus define:

- The set of delivery-customer demands, $d \in C^D$, each being characterized by the wish-list of satellites $S_d \in S$ where customer wants to receive, and the time window when the delivery must be performed, the choice of a particular satellite being part of the decisions characterizing the problem;
- The set of pickup-customer demands, $p \in C^P$, each characterized by the customer shipping it, the time window within which the pickup must be performed, as well as by the set of admissible satellites $S_p \in S$ to which the demand may be delivered, the choice of a particular one also being part of the problem decisions;
- The set $(\bar{p}, \bar{d}) \in R$ of c2c-customer-demands, each request requiring a demand to be transported from a c2c-pickup-customer-demand $\bar{p} \in C_{c2c}^P$ to a c2c-delivery-customer demand $\bar{d} \in C_{c2c}^D$.

Let $(i, q_i, \delta(i), [e_i, l_i])$ stand for the quantity $q_i > 0 (q_{\bar{p}} = -q_{\bar{d}} \text{ for } (\bar{p}, \bar{d}) \in R$ need to be delivered or picked up at the customer demand $i \in \{C^P \cup C^D \cup C_{c2c}^P \cup C_{c2c}^D\}$ within its hard time window $[e_i, l_i]$, with a service time $\delta(i)$.

Freights in the 2E-MTT-PDTWS is thus transferred as follows:

- A fleet of urban-truck K_1 with capacity Q_1 and a fixed cost F_1 is operated within the first echelon. Each urban-truck might do either one or both following activities: (1) leave the CDC with e2c demands, it then delivers freight to a subset of satellites. Once empty, the vehicle may either return to the CDC to complete its task or traveling to satellites to load c2e demands and then bringing it back to the CDC; Thus, the urban-truck performs only one 1st-level delivery leg in the former case (see Fig.1a), while a 1st-level delivery leg and then a 1st-level pickup leg in the latter (see Fig.1b) (The dashed lines stand for an empty move); (2) leave the CDC empty, it then travels to satellites to collect c2e demands and then returns to the CDC; In this case, the urban-truck operates only one 1st-level pickup leg (see Fig.1c). Each such sequence is called a 1st-level work assignment.
- A fleet of city-freighters K_2 with capacity Q_2 and a fixed cost F_2 is operated within the second echelon. Each city-freighter performs following activities: (1)

loads e2c demands from a satellite to deliver them to customers inside the city, generating a route called a 2nd-level delivery leg; (2) picks up c2e demands from the customers to bring them to a satellite, creating a respective a 2nd-level pickup leg; and (3) transfers c2c demands between customers within the city, making a 2nd-level c2c leg. Each c2c leg must follow the last-in-first-out (LIFO) policy to ensure that no handling is required while unloading freight from the vehicle. Thus, each city-freighter departs at the garage g to perform one or several 2nd-level delivery and/or pickup and/or c2c legs, and returns g to finish its activity. Such sequence performed by each city-freighter is called a 2nd-level work assignment.

Vehicles operate according to the Pseudo-Backhaul strategy in which any leg must be completed before another one may start, i.e., vehicles need to be empty when finishing a leg [2].

In general, the activity of a vehicle at a satellite is “unload only”, “load only” or “unload and load”. Fig.2 and Fig.3 represent these activities for the urban-truck and the city-freighter respectively. Striped and empty disks stand for pickup and delivery customer demands, respectively, dashed lines indicate empty moves.

Fig.2a and Fig.2b depict instances of the “load only” operation in which, after arriving at satellite s , the urban-truck loads c2e demands, then it leaves s for its next satellite to continue loading additional c2e demands or to go to the CDC to end its activity. The two instances differ in the status of urban-truck arrival at the satellite only. The urban-truck arrives empty at s in Fig.2a, while with c2e demands in Fig.2b. Fig.2c and Fig.2d represent instances of the “unload only” activity when the urban-truck arrives at s with e2c demands. At satellite s , if it unloads all freights, Fig.2c, it then leaves s empty. Otherwise, it unloads part of its e2c demands, Fig.2d, then leaves s for its next satellite s' to continue unloading its e2c demands. Fig.2e depicts the “unload and load” case in which, after unloading all e2c demands, the urban-truck starts to load c2e demands at the same satellite s , and leaves s to either bring the freight back to the CDC or to go to the next satellite for a “load only” activity.

Fig. 3a depicts the “unload only” case of a city-freighter in which, after arriving at the satellite s with the collected c2e demands, the city-freighter unloads all freight, then it leaves s empty. Fig.3b represents the “load only” case when the city-freighter arrives empty at s and loads e2c demands, it then delivers it to designated delivery customer demands. Fig.3c depicts instance of “unload and load” operation in which, after unloading all c2e demands collected from pickup-customer demands, the vehicle loads e2c demands and leaves to travel to delivery-customer demands to deliver it.

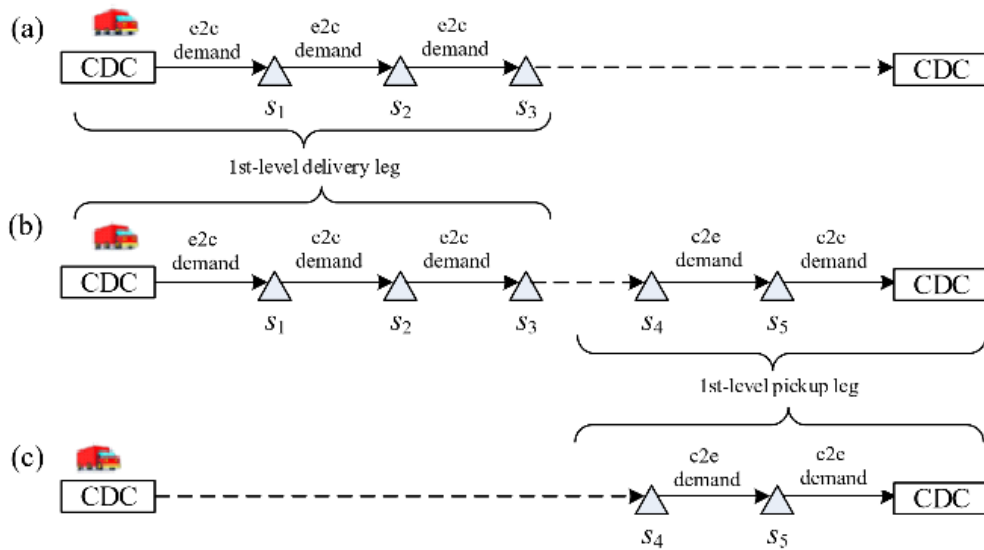


Fig. 1. The 1st-level work assignment illustration

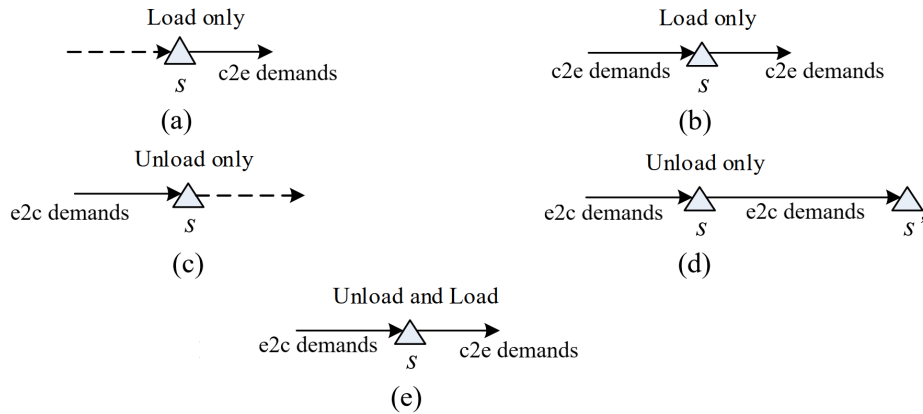


Fig. 2. Activities of an urban-truck at the satellite

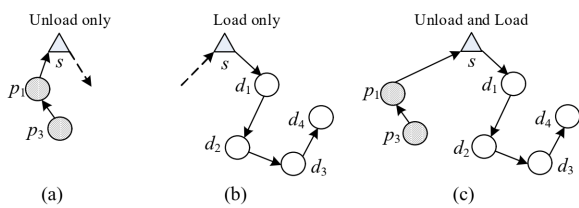


Fig. 3. Activities of a city-freighter at the satellite

Let T_U, T_L be the times required, respectively to unload and load an urban-truck at a satellite. Similar times, T'_U, T'_L , are assumed for a city-freighter. For simplicity, we assume that these time durations are equal at all satellites and independent on the transferred quantity.

2.1. The Temporary Storage and Synchronization at Satellites

At satellites, the e2c demand is transferred from urban-trucks to appropriate city-freighters, while the transshipment of c2e demand from city-freighters into

urban-trucks is performed. The temporary storage at the satellite is thus taken into account in case vehicles could not arrive concurrently for transshipment. Fig.4 gives an illustration of this necessary. Lines stand for the freight flow between vehicles.

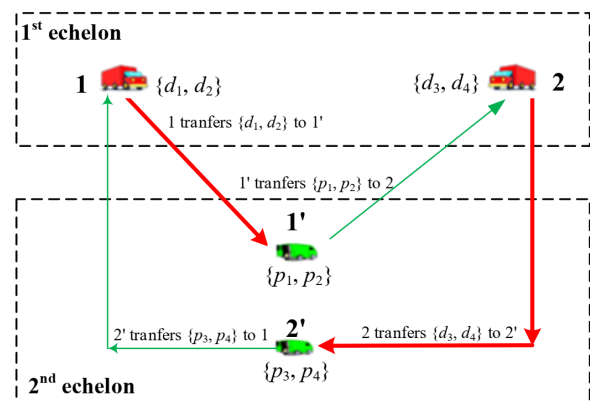


Fig. 4. An example of waiting cycle caused by the absence of the temporary storage

Two urban-trucks $\{1, 2\}$ and two city-freighters $\{1', 2'\}$ all perform the “unload and load” activity at the same satellite. At the first echelon, urban-truck 1 unloads its e2c demands $\{d_1, d_2\}$ and then loads c2e demands $\{p_3, p_4\}$ of city-freighter 2', while urban-truck 2 unloads e2c demands $\{d_3, d_4\}$ and loads c2e demands $\{p_1, p_2\}$ from city-freighter 1'. At the second echelon, city-freighter 1' unloads its c2e demands $\{p_1, p_2\}$ and loads e2c demands $\{d_1, d_2\}$ of urban-truck 1, while city-freighter 2' unloads $\{p_3, p_4\}$ and loads $\{d_3, d_4\}$ of urban-truck 2. In this case, the absence of the temporary storage makes these four vehicles trapped into a waiting cycle, *i.e.*, each unloading vehicle must wait for the appropriate loading vehicle to be empty in order to transfer its freight (urban-truck 1 waits for city-freighter 1', city-freighter 1' waits for urban-truck 2, urban-truck 2 waits for city-freighter 2', city-freighter 2' waits for urban-truck 1) while none of these vehicles could be emptied without using the temporary storage. One could make a vehicle empty by placing its freight in the temporary storage, the waiting cycle is therefore broken.

Let Q_s be the maximum temporary storage capacity of the satellite $s \in S$. The unloading (resp. loading) time at the temporary storage is assumed to be zero. Freights transfer between vehicles through satellites is assumed to follow the operational characteristics given as below:

- No pre-emption: The unloading or loading of a vehicle cannot be interrupted;
- As temporary storage is allowed, freight can be unloaded from a vehicle before the appropriate receiving vehicles are available;
- The unloading operation of a vehicle must be initiated as soon as it arrives at the satellite;
- The arrival time of a vehicle must be in the hard opening time window of the satellite;
- The loading operation of a vehicle can start only after its unloading operation is completed (if any) and all freight to be loaded on is ready. Otherwise, the freight that arrived previously is placed in temporary storage.

We below only provide the synchronization process of transshipment in the direction from city-freighters to urban-trucks. The reverse direction would be similar. The synchronization is thus detailed based on the category of transshipment at a satellite s as follows:

- (1-1) case: The transshipment is performed from an urban-truck k to a city-freighter k' : We assume

the vehicle k arrives at s at time t . The vehicle k starts unloading from time t , thus finishes unloading at the time $t + T_U$. If the city-freighter k' is ready to load at the time $t + T_U$, freight is transferred directly to this vehicle, the city-freighter k' then leaves s at the time $t + T_U + T_L'$ once it finishes loading. Otherwise, freight is placed in the temporary storage;

- (1- m) case: A single urban-truck k transfers freight to m city-freighters k'_1, \dots, k'_m : We assume the arrival time of the urban-truck k is t . The urban-truck k thus performs unloading at time t for a duration T_U . The appropriate freight transferring between each pair of vehicles (k, k'_i) where $i = 1, \dots, m$ is considered separately as a representation of the previous case (1-1);

- (n -1) case: n urban-trucks k_1, \dots, k_n transfer freight to a single city-freighter k' : We assume urban-trucks k_1, \dots, k_n arrive at s at times urban-trucks t_1, \dots, t_n , respectively. Urban-trucks k_1, \dots, k_n start unloading respectively from times t_1, \dots, t_n , for a duration T_U . If all urban-trucks arrive at s simultaneously, then freight is either transferred directly to city-freighter k' if city-freighter k' is ready to load at time $t_1 + T_U$, or placed in the temporary storage if city-freighter k' is not ready. Otherwise, in the case these urban-trucks do not arrive simultaneously, the freight that arrived previously is placed in the temporary storage;

- (n - m) case: n urban-trucks k_1, \dots, k_n transfer freight to m city-freighters k'_1, \dots, k'_m : Depending on the transshipment performed, one could consider this case as a combination of any two or all three previous cases. For example, the transshipment in which urban-trucks k_1, \dots, k_{n-1} transfer freight to a single city-freighter k'_1 while a single urban-truck k_n transfers freight to city-freighters k'_2, \dots, k'_m could be considered as a combination of (n -1) and (1- m) cases.

Given an illustrative example for the case (n -1) in Fig.5, we consider a satellite s , two urban-trucks $\{1, 2\}$ and one city-freighter $\{1'\}$. Urban-truck 1 arrives at s at time t_1 to unload e2c demands $\{d_1, d_2, \dots, d_5\}$, while the arrival time of urban-truck 2 is t_2 and it unloads e2c demands $\{d_6, d_7, \dots, d_9\}$, the city-freighter 1' is empty at time t' and it loads $\{d_1, d_2\}$ of urban-truck 1 and $\{d_6, d_7\}$ of urban-truck 2. Since the arrival times t_1 and t_2 play a symmetric role in this example, four possibilities of the consolidation process are following:

- $t_1 = t_2$ and $t_1 + T_U \geq t'$: Urban-truck 1 and urban-truck 2 start to unload their demands at time t_1 . After unloading for a time T_U , they leave the satellite. At this moment (i.e., $t_1 + T_U$), as city-freighter 1' is already empty, demands $\{d_1, d_2, d_6, d_7\}$ are transferred directly to city-freighter 1' without using temporary storage, the remaining demands $\{d_3, d_4, d_5\}$ of urban-truck 1 and demands $\{d_8, d_9\}$ of urban-truck 2 are placed in the temporary storage. city-freighter 1' leaves the satellite at time $t_1 + T_U + T'_L$ once it finishes loading;

- $t_1 = t_2$ and $t_1 + T_U < t'$: As the previous case, urban-truck 1 and urban-truck 2 both finish unloading at time $t_1 + T_U$. At this moment, city-freighter 1' has not arrived yet, all demands $\{d_1, d_2, \dots, d_9\}$ are therefore placed in the temporary storage. Once arrive, city-freighter 1' performs loading from t' for a duration T'_L , i.e., demands $\{d_1, d_2, d_6, d_7\}$ are transferred from the temporary storage to this vehicle. Thus, from time t' , there are only $\{d_3, d_4, d_5, d_8, d_9\}$ in the temporary storage;

- $t_1 < t_2$ and $t_2 + T_U \geq t'$: Urban-truck 1 and urban-truck 2 perform unloading respectively at times t_1 and t_2 , for a duration T_U . In fact, city-freighter 1' is ready to load at time $t_2 + T_U$ (i.e., when urban-truck 1 and urban-truck 2 both finish unloading). However, demands $\{d_1, d_2\}$ of urban-truck 1 and $\{d_6, d_7\}$ of urban-truck 2 do not arrive at the satellite simultaneously. Therefore, the demand of urban-truck 1 that arrived previously is placed in temporary storage. At time $t_2 + T_U$, city-freighter 1' performs loading, i.e., demands $\{d_6, d_7\}$ of urban-truck 2 are transferred directly from urban-truck 2 to it, while $\{d_1, d_2\}$ of urban-truck 1 are moved from the temporary storage and loaded onto it. At the same time, the remaining demands of urban-truck 2 (i.e., $\{d_8, d_9\}$) is placed in the temporary storage. City-freighter 1' leaves the satellite at time $t_2 + T_U + T'_L$ once all freight is loaded, and there are only $\{d_3, d_4, d_5, d_8, d_9\}$ in the temporary storage from this time;

- $t_1 < t_2$ and $t_2 + T_U < t'$: As in the previous case, Urban-truck 1 and urban-truck 2 finish unloading respectively at times $t_1 + T_U$ and $t_2 + T_U$. As city-freighter 1' is not available at satellite when they both finish unloading, all demands $\{d_1, d_2, \dots, d_5\}$ of urban-truck 1 are placed in the temporary storage

from time $t_1 + T_U$, while $\{d_6, d_7, d_8, d_9\}$ of urban-truck 2 are also placed from time $t_2 + T_U$. Once arrive at time t' , city-freighter 1' starts loading demands $\{d_1, d_2, d_6, d_7\}$ from the storage. Then, city-freighter 1' leaves the satellite at time $t' + T'_L$, and there are only $\{d_3, d_4, d_5, d_8, d_9\}$ in the storage from this time.

The capacity of each satellite $s \in S$ is measured in the maximum number of urban-trucks (π_s) and city-freighters (π'_s) that it can accommodate simultaneously. Let W be the set of waiting stations. The vehicle cannot go directly to a satellite $s \in S$, but has to wait at a waiting station $w \in W$ instead, if either its arrival time at s is before the opening time $[t(s) - \eta, t(s)]$, or there is no capacity available at s .

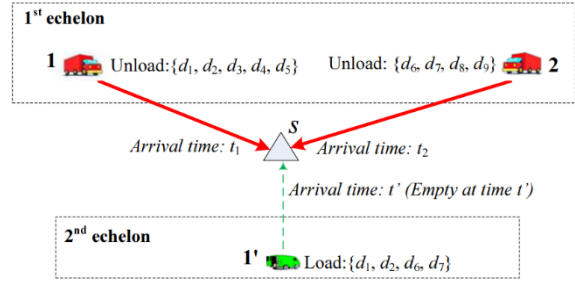


Fig. 5. An example of activities of vehicles from both echelons at the satellite s

2.2. Problem Definition

An example of the solution to the problem is represented in Fig.6 which models a partial solution with four work assignments performed by two urban-trucks $\{1, 2\}$ on the first echelon and two city-freighters $\{1', 2'\}$ on the second echelon. Dashed lines stand for the empty travel. The urban-truck number 1 performs a 1st-level work assignment that consists of a sequence of two legs $\{r_1, r_2\}$ where $r_1 = \{CDC, s_1, s_2\}$ is a 1st-level delivery leg and $r_2 = \{s_2, s_3, CDC\}$ is a 1st-level pickup leg. In details, it first leaves the CDC to deliver e2c demands to s_1, s_2 . At the satellite s_2 , after finishing unloading to complete leg r_1 , the urban-truck number 1 becomes empty. It then starts to perform leg r_2 by loading e2c demands at s_2 , then s_3 , and finally brings them back to the CDC. Similarly, the urban-truck number 2 performs a 1st-level delivery leg $r_3 = \{CDC, s_4\}$ and then a 1st-level pickup leg $r_4 = \{s_4, CDC\}$. On the second echelon, the city-freighter 1' performs a sequence of four legs $\{r'_1, r'_2, r'_3, r'_4\}$ where $r'_1 = \{s_1, d_5, d_6\}$ and $r'_3 = \{s_4, d_7, d_8\}$ are 2nd-level

delivery legs, $r_2' = \{p_6, p_7, s_3\}$ is a 2nd-level pickup leg, and $r_4' = \{\bar{p}_5, \bar{d}_5, \bar{p}_6, \bar{d}_6\}$ is a 2nd-level c2c leg. More precisely, it starts empty from the garage g to perform delivery of leg r_1' : load e2c demands at s_1 and delivery them to delivery-customer demands d_5, d_6 . It is now empty, so it continues to do leg r_2' : go to pickup-customer demands p_6, p_7 to load c2e demands and bring them to satellite s_3 to unload there. After finishing leg r_3' , it services two requests $(\bar{p}_5, \bar{d}_5), (\bar{p}_6, \bar{d}_6)$ in the LIFO order $\bar{p}_5, \bar{d}_5, \bar{p}_6, \bar{d}_6$ before moving empty back to garage g . City-freighter 2' performs six legs sequentially $\{r_5', r_6', r_7', r_8', r_9', r_{10}'\}$, where $r_6' = \{p_1, p_2, s_2\}$ and $r_9' = \{p_3, p_4, p_5, s_4\}$ are 2nd-level pickup legs, $r_7' = \{s_2, d_1, d_2\}$ and $r_{10}' = \{s_4, d_3, d_4\}$ are 2nd-level delivery legs, $r_5' = \{\bar{p}_1, \bar{p}_2, \bar{d}_2, \bar{d}_1\}$ and $r_8' = \{\bar{p}_3, \bar{d}_3, \bar{p}_4, \bar{d}_4\}$ are 2nd-level c2c legs. One could see that once loading p_5 is finished, city-freighter 2' has to move to the waiting station w_1 and wait there in order to reach s_4 in time for synchronization.

The 2E-MTT-PDTWS is defined on a directed graph $G=(V, A)$, which represents the two-echelon system. The first echelon is defined by $G_1=(V_1, A_1)$ with $V_1 = \{CDC\} \cup S \cup W$ and $A_1 = \{(i, j) | i \in V_1 \setminus W, j \in V_1\} \cup \{(w, s) | w \in W, s \in S\}$. The second echelon is defined by $G_2=(V_2, A_2)$ with $V_2 = \{g\} \cup S \cup W \cup C^P \cup C^D \cup C_{c2c}^P \cup C_{c2c}^D$ and $A_2 = \{(g, i) | i \in S \cup C^P \cup C_{c2c}^P\} \cup \{(i, j) | i \in C^P, j \in C^P \cup S \cup W\} \cup \{(i, j) | i \in C^D, j \in C^P \cup C_{c2c}^P \cup C^D \cup S \cup W\} \cup \{(i, j) | i \in C_{c2c}^P, j \in C_{c2c}^P \cup C_{c2c}^D \cup S \cup W\} \cup \{(i, j) | i \in C_{c2c}^D, j \in C_{c2c}^P \cup C_{c2c}^D \cup C^P \cup S \cup W\} \cup \{(i, j) | i \in S, j \in C^P \cup C^D \cup C_{c2c}^P \cup W\} \cup \{(i, g) | i \in C^D \cup C_{c2c}^D \cup S\} \cup \{(w, s) | w \in W, s \in S\}$.

Each arc $(i, j) \in A_1 \cup A_2$ is associated with a travel cost c_{ij} and a travel time t_{ij} . Thus, the 2E-MTT-PDTWS can then be seen as the problem of (1) assigning pickup-customer and delivery-customer demands to satellites, and (2) finding efficient work assignments operated at each echelon. The objective is to minimize the total cost, which is comprised of the routing cost of operating the work assignments and the fixed cost of using the vehicles, while

capacities of temporary storage, satellites, vehicles, and time-related constraints are satisfied.

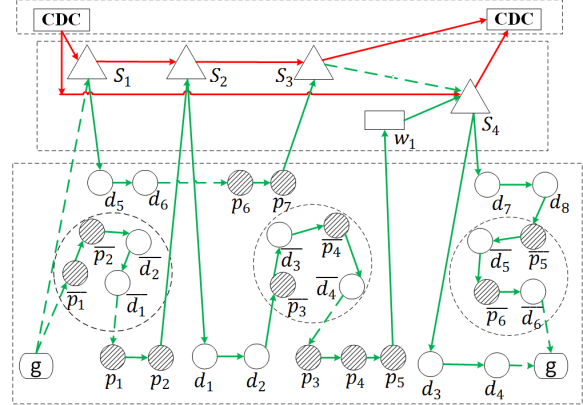


Fig. 6. An example of the solution

3. Solution Method

3.1. General Structure

For simplification, we only allow the search to explore feasible solutions. First, an initial feasible solution z is generated using a greedy method seeking to fully utilize vehicles and minimize the total cost. The solution z is then assigned as the current best solution z_{best} . As the 2E-MTT-PDTWS consists of two echelons with two separate fleets of vehicles, we divide the solution to the problem into two parts: the first part is the schedule of the urban-trucks on the first echelon, while the second part is the schedule of the city-freighters on the second echelon. We thus use the bottom-up approach to solve the problem. Consequently, there are two steps that need to be performed at each iteration of the algorithm. In the first step, to schedule the city-freighters we rebuild the second part of the current solution z by inheriting the adaptive large neighborhood search (ALNS) introduced in [3]. Given this schedule together with synchronization constraints at satellites, we then propose a heuristic algorithm, named QFE, to regenerate the first part of the current solution z in the second step. We then update the z_{best} if the solution z obtained after two steps is better than it. The overall number of iterations is IT_{max} . The structure of the proposed algorithm for the 2E-MTT-PDTWS is given in Algorithm 1.

Algorithm 1 The proposed algorithm

- 1: Generate an initial feasible solution z ;
 - 2: $z_{best} \leftarrow z$;
 - 3: Initialize weights for destroy and repair neighborhoods (all equal to 1);
 - 4: **repeat**
 - 5: Probabilistically select a destroy and a repair based on their current weights;
 - 6: Apply the selected destroy and repair
-

```

neighborhood to the second part of  $z$  to obtain
the new solution  $z''$ ;
7:   if  $z''$  satisfies the acceptance criterion then
8:     Apply the heuristic algorithm QFE to
 $z''$  to obtain the new solution  $z'$ ;
9:      $z \leftarrow z'$ ;
10:    if  $z'$  is better than  $z_{best}$  then
11:       $z_{best} \leftarrow z'$ ;
12:    end if
13:  end if
14:  Adjust weights for all destroy and repair
neighborhoods;
15: until number of iterations  $IT_{max}$  has passed;
16: return  $z_{best}$ ;

```

3.2. Initialize Solution

An initial solution is created using the decomposition approach by scheduling the fleet of vehicles at each echelon. To schedule the fleet of city-freighters, we inherit the process of initiating the solution in [3]. This process consists of two steps. At the first step, each pickup-customer demand $p \in C^P$ are first assigned to a satellite s selected from S_p . The selection is aimed to not only reduce the required travel distance of vehicles but also balance the total delivery and pickup demand of each satellite, thus reduce the empty movements. A greedy algorithm is then applied on the second step to build each work assignment sequentially until all customer demands are serviced. At each iteration of this greedy algorithm, a suitable leg with the smallest average cost per unit demand will be concatenated to the current work assignment.

Once we get the schedule of city-freighters, we then use this information to schedule the fleet of urban-trucks by applying our proposed QFE heuristic algorithm.

3.3. The ALNS for the Second Echelon

In order to schedule vehicles of the second echelon, we inherit the proposed ALNS proposed: for destroy operators, we use random and worst cost destroy (customer level), worst utilization and time related destroy (leg level); for the repair operator, we use the random greedy [3]. Destroy and repair operators are selected probabilistically based on historical weights. To keep the diversification, the worse solutions are accepted with a probability given by the Boltzmann factor.

3.4. The QFE Heuristic Algorithm for the First Echelon

Once the ALNS is terminated, we obtain schedule of the city-freighters. As a result, for each e2c delivery- (c2e pickup-) customer demand $d \in C^D$

($p \in C^P$), we know the satellite and the time t^d (t^p) at which it needs to be loaded to a city freighter (urban-truck). Accordingly, we need to schedule urban-trucks to (1): bring each e2c delivery-customer demand $d \in C^D$ from the CDC to its appropriate satellite so that it could be loaded to the city freighters at time t^d , and (2): go to appropriate satellite to start loading each c2e pickup-customer demand $p \in C^P$ at the time t^p and bring them back to the CDC.

First, we sort all e2c delivery-customer demands $d \in C^D$ in ascending order of the time t^d and store them in the set $C_{unserved}^D$. Initially, these e2c demands in $C_{unserved}^D$ are not serviced by any urban-truck yet. We determine the satellite s_1 having the demand $d_1 \in C_{unserved}^D$ with the earliest t^{d_1} , then schedule an urban-truck k to do an itinerary from the CDC to s_1 to service (transfer) the demand d_1 directly. In addition, for each demand d' among e2c demands of satellite s_1 in $C_{unserved}^D$, we add d' to this itinerary if this vehicle could also service (transfer) d' either directly or through temporary storage without violation of capacity Q_1 . Once finish, we get all e2c demands such that the urban-truck k need to deliver to satellite s_1 . We then delete these demands from $C_{unserved}^D$ and find the next satellite s_2 having the demand $d_2 \in C_{unserved}^D$ with the earliest t^{d_2} so that urban-truck can go from s_1 to s_2 to service this demand directly without violation of capacity Q_1 . Repeat the above process we get a 1st-level delivery leg that consists a sequence of satellites that urban-truck k need to go through and quantity of e2c demands to deliver at each of these satellites. The process is terminated once there does not exist any satellite with unserved e2c demands for urban-truck k to go to without violation of time and capacity. As a result, we get a 1st-level delivery leg performed by the vehicle urban-truck k . Therefore, we then try to schedule this vehicle to perform a 1st-level pickup leg so that this vehicle will go back to the CDC with c2e demands rather than empty. The process to create 1st-level pickup leg for urban-truck k could also be done as what we have just done to create 1st-level delivery leg, but with c2e pickup-customer demands instead.

One could see that each 1st-level work assignment must has one of the three following forms: (1) a single 1st-level pickup leg; (2) a single 1st-level delivery leg; (3) a 1st-level delivery leg followed by a 1st-level pickup leg. Our QFE heuristic thus aims to create the latest form as many as possible to reduce the number of urban-trucks.

4. Computational Results

The objective of the numerical experimentation is to study the proposed model and analyze the satellite synchronization. The proposed algorithm is implemented in C++. Experiments are run on a 2.4 GHz Intel i7 processor with 16GB of RAM.

4.1. Instance Generation

We inherit instance generation proposed by Cranic et al. [1]. The CDC, supply points, waiting stations, and customers are uniformly distributed in a square condinating in the interval $[0,100]$. Each city-freighter has the fixed cost ($F_2 = 500$), capacity ($Q_2 = 100$), unloading time ($T'_U = 30$), and loading time ($T'_L = 30$). Each urban-truck then has $\{F_1, Q_1, T_U, T_L\} = \{2 * F_2, 2.5 * Q_2, 3 * T'_U, 3 * T'_L\}$, respectively. The number of satellites and waiting stations in each instance are all four. Each satellite has capacity ($\pi'_s = 5$ and $\pi_s = 10$) and maximum temporary storage capacity $Q_s = 1,000$. The instance set consists of three sets C_1, C_2, C_3 with five instances each. The number of customer demands for three sets are 594, 761, 1070, respectively. Each customer demand has service time $\delta(i) = 20$ and volume of demand q_i generated randomly in the range $[5, 25]$. The characteristics of these instances are summarized in the Table 1.

Table 1. Characteristics of the instances

Test set	$ S $	$ W $	$ C^D $	$ C^P $	R
C_1	4	4	400	44	150
C_2	4	4	400	171	190
C_3	4	4	400	400	270

4.2. Computational Results

Table 2 displays the average of the best cost and the average running time obtained over 10 runs for each IT_{max} within $\{100,000; 200,000; 500,000\}$. We have observed that executing the algorithm with 500,000 iterations yields an average improvement of the best solution of 9.9%, while requiring about 420% more time, when compared to the case of 200,000 iterations. It indicates that 200,000 iterations is the most appropriate value setting for IT_{max} .

We next examine the synchronization at satellites. On average, there is almost 33.15% of the cases, temporary storages at satellites need to be used as the freight could not be transferred directly between vehicles. It thus helps to reduce the number of vehicles running empty to satellites just for transshipment on time in the case temporary storages

are not allowed at satellites. For the remaining 66.85% of the cases without using temporary storages, at each satellite, every single urban-truck transfers freight directly to four city-freighters, while every single city-freighter does direct transshipment with three urban-trucks.

Table 2. Performance of the proposed algorithm

IT_{max}	Test set	Average best cost	Average running time (min.)
100,000	C_1	22,487.95	6.95
	C_2	24,433.64	9.26
	C_3	34,784.23	19.85
	Avg.	27,235.27	12.02
200,000	C_1	19,769.70	10.73
	C_2	23,501.81	11
	C_3	32,851.82	22.64
	Avg.	25,374.44	14.79
500,000	C_1	18,930.78	76.33
	C_2	21,266.02	65.23
	C_3	28,382.08	89.19
	Avg.	22,859.63	76.92

5. Conclusion

We addressed a new problem class that applied the two-echelon distribution system to the MTT-PDTWS problem described in [3]. We consider the synchronization process of transshipment at satellites between vehicles with the temporary storage mechanism. We proposed a decomposition method for the problem. At each step of the proposed algorithm, routing of the fleet of city-freighters is determined by the algorithm ALNS. Latter, the QFE heuristic algorithm is applied to obtain the schedule of urban-trucks. Experiments were run with the proposed instance datasets to study the problem and the satellite synchronization.

References

- [1]. T. Crainic, P. Nguyen, and M. Toulouse; Synchronized Multi-trip Multi-traffic Pickup & Delivery in City Logistics; Transportation Research Procedia tenth International Conference on City Logistics; Spain; 2015; 26-39. <https://doi.org/10.1016/j.trpro.2016.02.023>
- [2]. T. G. Crainic, F. Errico, W. Rei, and N. Ricciardi, Integrating c2e and c2c Traffic into City Logistics Planning, Procedia - Social and Behavioral Sciences, 39, 0, (2012) 47-60. <https://doi.org/10.1016/j.sbspro.2012.03.090>
- [3]. Nguyen, Quang Ngoc and Duc, Nghia Nguyen and Nguyen, Phuong Khanh; Proceedings of the Eighth International Symposium on Information and Communication Technology; SolCT 2017; Viet Nam; 2017; 439-446.