

MILP for a variant of Pickup & Delivery Problem for both passengers and goods transportation*

Alexis Godart¹, Hervé Manier¹, Christelle Bloch² and Marie-Ange Manier¹

Abstract—Pickup and delivery problems have been declined in many variants. In these optimization problems, a fleet of vehicles must satisfy customers’ transport demand under specific constraints. This paper studies the feasibility of using a linear model including constraints that are not only related to the goods transport, but also combined with passengers transport in a more flexible way. We explore many characteristics simultaneously like transfers capabilities (with and without storage), time windows on sites, on transfer points and on vehicle depots, heterogeneous capacities and multiple visits. We use exact methods on some instances we created, using commercial solver IBM CPLEXTM. Results obtained show potential savings on two instances but also highlight the complexity and limits in terms of scalability.

I. INTRODUCTION

Over last decades, transport demand has been quickly growing around the world. These growths are especially located in cities that keep spreading. They have many consequences, already well-known, such as increased costs and environmental impacts. In most countries, stakeholders and governments have worked together to avoid such negative externalities without preventing the economical growth and social improvements. A fundamental factor is the optimization of transport and its infrastructure in urban areas [10]. Among the existing socio-economic studies dealing with urban transport, those conducted by Dacko [6] and Spikermann [16] highlight the expected benefits for all stakeholders in changing the urban mobility model. This would be made by promoting intermodal travel behavior, and require transport providers working together to offer multimodal mobility services. This collaboration implies pooling of transport infrastructures and resources. The proposed approach considers both *passengers mobility* and *goods mobility* into a single urban logistics system. The next section expose the related state of the art in operational research.

II. LITERATURE REVIEW

An efficient logistics system within urban environment implies a reasonable quality of service into an environment with low available space, shared infrastructures and a limited budget, keeping in mind that urban congestion can also be a significant source of delay, noise pollution and high-level

gas emissions. There have been lots of studies and research in recent years dedicated to the optimization of logistics problem, sometimes considering several characteristics independently. Problems in which a set of vehicles is used to fulfill transportation requests are called Vehicle Routing Problem (VRP), as introduced by Dantzig [7] in 1959. Pickup and Delivery Problems (PDP), also known as the Vehicle Routing Problem with Pickup and Delivery (VRPPD) have been surveyed and discussed by Savelsbergh [15] and more recently by Berbeglia in both static [1] and dynamic [2] context, within a general framework. In PDP, objects or people are moved between origins and destinations. These moves are made thanks to a fleet of homogeneous or heterogeneous vehicles. According to Cordeau [4] Dial-A-Ride Problems (DARP) can be considered as Pickup and Delivery Problems (PDP) in which origins and destinations for each demand are known beforehand. Additionally, DARP are problems that better fit for people transportation because of narrow time windows and objectives aiming better quality of services rather than cost reduction. The transport request is expressed with *one-to-one* relation where each demand has a single origin point and a single destination point.

In the following subsections we review some papers dealing with some variants of PDP including (but not limited to) time windows and transfers with characteristics such as heterogeneous fleet of vehicles, multiple depots or also multiple echelons. Our state of the art focuses on both static and dynamic problems with deterministic environment, integrating interesting features for the urban context.

Pickup and Delivery Problems with Time Tindows (PDPTW) are specific PDP problems with temporal constraints over sites, vehicles, demands and/or interactions between those. [9] present in 1991 a general framework and they exactly solve the problem using a column generation. [14] introduce a new formulation of general PDPTW. They tackle it with branch-and-cut algorithms and adding valid inequalities. They solve instances with 96 requests, 194 nodes and 8 vehicles to the optimality. One variant studied by [12] deals with coordination of heavy and light resources to save time and reduce global cost. This problem was solved with a two-staged heuristics. Initial solutions are first created. They are improved using ejection chains, and then an assignment stage with light resources is used until an optimal assignment. Multi-objective PDPTW has also been treated [11]. These authors solve a problem with several vehicles and three objectives (total travel cost, makespan, number of vehicles used) using an approximate ϵ -constraint solution to create the efficient Pareto fronts.

*Mobilitytech project funded by the region of Franche-Comté (CPER 2015-2020) and labeled by Pôle Véhicule du Futur

¹Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS, Rue Thierry-Mieg (UTBM), 90010 Belfort Cedex, France author@utbm.fr

²Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS, 1 Cours Leprince-Ringuet, 25200 Montbéliard, France author@univ-fcomte.fr

Nowadays PDP and DARP most often involve several vehicles. Benefits of transfer operations are intuitively guessed over cost reduction and distance savings. PDP with transfers (PDPT) was studied by [5] who propose general formulation for the static variant. An extra binary variable z keeps track of the exact position of requests. A branch-and-cut method using Benders decomposition is proposed. [3] formulate a dynamic PDPT involving additional demands arriving at arbitrary times. Given solutions use a shortest path algorithm on a dynamic graph, on instances with 200 to 2000 requests.

Regarding transfer operations arising in DARP (DARPT), [13] present a general formulation and an adaptive large neighborhood search method. The results prove savings due to transfers while minimizing the total distance traveled. [8] solve DARP with dynamic transfer points using insertion techniques and constraints propagation.

III. MATHEMATICAL FORMULATION

A. Characteristics and assumptions

A fleet of vehicles V with heterogeneous capacity Q_v is considered. Vehicles are either dedicated to the mobility of passengers or to the mobility of goods. Vehicles have heterogeneous speed v_v equal to the mean velocity while visiting nodes through edges, and an associated depot w_v , from which each vehicle begins and ends its trip. Each vehicle can only do one trip per period.

The graph $G = (N, E)$ contains 3 sets of nodes, organized as follows. The set W of vehicle depots dedicated to host empty vehicles. The set Ω of Origin/Destination sites expresses origins and destinations for transportation requests. Each site can be origin and destination as it can be involved in more than one request. Neither vehicle depots nor sites have storage capacity. The set T of transfers points with a storage capacity sc_t . This capacity is only considered for vehicles transporting goods. It can not be expressed for passengers. Indeed, in urban context, infrastructures are assumed to be large enough for welcoming as many passengers as needed at any time, on any node. Storage capacity can be fixed to zero, to imply no storage at all. Then the transfer point requires both vehicles to meet during a same time window. This induces some precedence and synchronization constraints, which are discussed later in this paper. Two sets T^+ and T^- are created, resulting from the duplication of set T , representing respectively the incoming flow capture and outgoing flow capture at any transfer point $t \in T$.

Any node i has an opening time window, expressed by an earliest date e_i and a latest date l_i . Any loading/unloading operation must be realized within this time window $[e_i; l_i]$. The graph G is assumed to be complete, oriented and asymmetric. The set of edges E connects nodes in $N = W \cup \Omega \cup T \cup T^+ \cup T^-$. A subset $N' = N \setminus W$ contains all nodes except vehicle depots. A cost d_{ij} is assigned to each edge. It corresponds to the shortest path between an origin node $i \in N$ and a destination node $j \in N$ such that $i \neq j$.

Finally, transport demands are stored in a set of known transportation requests R . For each request $r \in R$ we know

the origin node S_r^+ , where request r will be picked up, and the destination node S_r^- where it will be dropped-off. A set B_r lists vehicles that can handle r . The quantity q_r is the number of entities moving from the origin to the destination. Service times for pickup δ_r^+ and for drop-off δ_r^- are taken into account respectively for loading operations and unloading operations. r can be handled only within a time window $[p_r^+; p_r^-]$ where p_r^+ is the earliest date for the pickup at S_r^+ , and p_r^- is the latest date for the drop-off at S_r^- . A vehicle can visit several times a same node within its trip.

The maximum number of visits θ_i in an Origin/Destination site $i \in \Omega$ is equal to the number of occurrences where node i appears as origin S_r^+ plus the number of occurrences where node i appears as destination S_r^- for all $r \in R$.

The maximum number of visits in a transfer site $t \in T$ by all vehicles $v \in V$ is θ_t .

B. Decision variables

$$x_{ijab}^v = \begin{cases} \mathbf{1} & \text{if vehicle } v \in V \text{ is moving directly from} \\ & i \in N \text{ on the visit number } a \in \{1, \dots, \theta_i\} \text{ to} \\ & j \in N \text{ on the visit number } b \in \{1, \dots, \theta_j\}, \\ \mathbf{0} & \text{else.} \end{cases}$$

$$A_{ijab}^v : \begin{cases} \text{Arrival date of vehicle } v \in V \text{ at node } j \in N \text{ on} \\ \text{visit number } b \in \{1, \dots, \theta_j\} \text{ when coming from} \\ \text{node } i \in N \text{ on visit number } a \in \{1, \dots, \theta_i\}. \end{cases}$$

$$D_{ijab}^v : \begin{cases} \text{Departure date of vehicle } v \in V \text{ at node } i \in N \text{ on} \\ \text{visit number } a \in \{1, \dots, \theta_i\} \text{ when going to node} \\ j \in N \text{ on visit number } b \in \{1, \dots, \theta_j\}. \end{cases}$$

$$z_{ijab}^{rv} = \begin{cases} \mathbf{1} & \text{if vehicle } v \in V \text{ is handling request } r \in R \\ & \text{from } i \in N \text{ on the visit number } a \in \{1, \dots, \theta_i\} \\ & \text{to } j \in N \text{ on the visit number } b \in \{1, \dots, \theta_j\}, \\ \mathbf{0} & \text{else.} \end{cases}$$

$$\tau_{ta}^{vv'} = \begin{cases} \mathbf{1} & \text{if request } r \in R \text{ is transferred with storage} \\ & \text{operation from vehicle } v \in V \text{ to vehicle} \\ & v' \in V \mid v' \neq v \text{ while } v' \text{ is leaving transfer} \\ & \text{node } t \in T \text{ on visit } a \in \{1, \dots, \theta_t\}, \\ \mathbf{0} & \text{else.} \end{cases}$$

Decision variables A_{ijab}^v and D_{ijab}^v are both defined in \mathbb{R}^+ . Binary variable z_{ijab}^{rv} allows us to track requests independently of the transport modes being used. Requests are tracked on edges. We denote M a big number, used to linearize some constraints.

C. Transfer operations

Any request can be transferred from one vehicle to another if there is at least one transfer point available. These transfer operations can take place under two major conditions. First, the transfer point must be available so vehicles can operate on it within its opening time window. Second, operations must take place respecting all time constraints. Hence we define two kinds of transfer operations we consider along this paper

: transfer **with storage** and transfer **without storage**. Storage isn't considered in the context of passengers transportation.

On one hand, transfers can be accomplished with storage, and time constraint applied is only a precedence constraint between two vehicles and their drop-off and pickup operations. On the other hand, transferring without storage implies a synchronization constraint. Both vehicles must be at the same transfer node within a common time window, and this time window must be at least wide enough to realize their respective operations.

In both cases (for passengers and for goods), requests are not necessarily transferred, and therefore the unload time is only the sum of service times of requests being transferred.

Transfer points are duplicated twice (fig. 1 & 2), so we can separate incoming flow from outgoing flow. Since decision variable z tracks requests **on edges**, pickup and drop-off operations are executed by switching the binary value z between two edges.

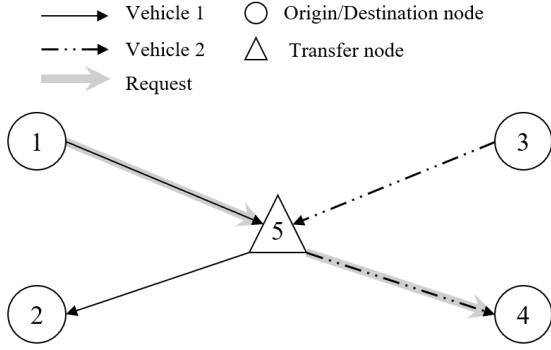


Fig. 1. Abstracted physical flow of a request being transferred from vehicle 1 to vehicle 2

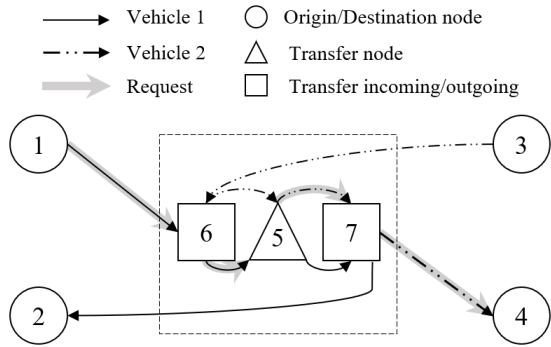


Fig. 2. Same operation with our modeling. Transfer point is duplicated twice.

D. Objective function

$$\text{minimize } \sum_{i \in N} \sum_{j \in N | i \neq j} \sum_{a=1}^{\theta_i} \sum_{b=1}^{\theta_j} \sum_{v \in V} (x_{ijab}^v * d_{ij}) \quad (1)$$

The problem stated is solved with mono-objective function. The criteria is to minimize the overall distance traveled by vehicles (1).

E. Constraints

$$\sum_{v \in B_r} \sum_{j \in N'} \sum_{a=1}^{\theta_i} \sum_{b=1}^{\theta_j} z_{ijab}^{rv} = 1 \quad \forall r \in R; i = S_r^+; j \neq i \quad (2)$$

$$\sum_{v \in B_r} \sum_{i \in N'} \sum_{a=1}^{\theta_i} \sum_{b=1}^{\theta_j} z_{ijab}^{rv} = 1 \quad \forall r \in R; j = S_r^-; i \neq j \quad (3)$$

$$x_{ijab}^v * M \geq \sum_{r \in R} z_{ijab}^{rv} \quad \forall i, j \in N; \forall v \in V \quad (4)$$

$$\forall a \in \{1, \dots, \theta_i\}; \quad \forall b \in \{1, \dots, \theta_j\}$$

We choose here to deal with the non-selective variant of PDPTW, meaning that all requests must be honored. The variable z is used to track requests during transportation, and constraints (2) and (3) force the request r to be picked up from the origin node S_r^+ and to be dropped-off at the destination node S_r^- , regardless of the vehicle $v \in B_r$ being used.

As mentioned in 3.2, we use decision variable x to track the route described by each vehicle. Constraint (4) ensures that if the request r is handled by vehicle v when going from node i on visit number a to node j on visit number b (i.e. when $z_{ijab}^{rv} = 1$) then this vehicle had to move on that edge the exact same way.

$$\sum_{j \in N'} \sum_{b=1}^{\theta_j} x_{hj1b}^v \leq 1 \quad \forall v \in V; h = w_v \quad (5)$$

$$\sum_{j \in N'} \sum_{c=1}^{\theta_j} x_{hj1b}^v * M \geq \sum_{i \in N'} \sum_{j \in N' | j \neq i} \sum_{a=1}^{\theta_i} \sum_{b=1}^{\theta_j} x_{ijab}^v \quad (6)$$

$$\forall v \in V; \quad h = w_v$$

$$\sum_{j \in N'} \sum_{b=1}^{\theta_j} x_{hj1b}^v = \sum_{i \in N'} \sum_{a=1}^{\theta_i} x_{iha2}^v \quad \forall v \in V; h = w_v \quad (7)$$

Constraint (5) allows vehicles to be used or not. We ensure with constraint (6) that each vehicle has to leave its associated vehicle depot if and only if this vehicle is being used anywhere in the graph, while constraint (7) forces this same vehicle to finish his route at this same vehicle depot.

$$\sum_{j \in N} \sum_{b=1}^{\theta_j} x_{ijab}^v \leq 1 \quad \forall i \in N; \forall v \in V; \forall a \in \{1, \dots, \theta_i\} \quad (8)$$

$$\sum_{j \in N} \sum_{b=1}^{\theta_j} z_{ijab}^{rv} \leq 1 \quad \forall r \in R; \quad \forall i \in N' \quad (9)$$

$$\forall v \in B_r; \quad \forall a \in \{1, \dots, \theta_i\}$$

$$\sum_{v \in B_r} \sum_{a=1}^{\theta_t} z_{t'taa}^{rv} \leq 1 \quad \forall r \in R; \forall t \in T; t' = t + |T| \quad (10)$$

A vehicle v or a request r on node i must only pass through one edge $[i, j]$ for each visit number a to ensure one unique flow for vehicles (8) and for requests (9) departing from any node. Any given request can only enter once at most in each transfer point (10).

$$\sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} x_{ija'b}^v \leq \sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} x_{ijab}^v \quad \forall v \in V \quad (11)$$

$$\forall i \in \Omega \cup T^+ \cup T^-; \forall a \in \{1, \dots, \theta_i - 1\}; a' = a + 1$$

$$\sum_{v \in V} \sum_{j \in N} \sum_{b=1}^{\theta_j} x_{ija'b}^v \leq \sum_{v \in V} \sum_{j \in N} \sum_{b=1}^{\theta_j} x_{jt'jab}^v \quad \forall j \in W \cup \Omega \quad (12)$$

$$\forall t \in T; \forall a \in \{1, \dots, \theta_t - 1\}; t'' = t + 2 \cdot |T|; a' = a + 1$$

Constraints (11) and (12) ensure that any solution must use lowest visit index order while visiting an origin/destination site or a transfer point.

$$\sum_{h \in N' | h \neq i} \sum_{a=1}^{\theta_h} x_{hiab}^v = \sum_{j \in N | j \neq i} \sum_{c=1}^{\theta_j} x_{ijbc}^v \quad \forall v \in V \quad (13)$$

$$\forall b \in \{1, \dots, \theta_i\}; \forall i \in N$$

$$\sum_{h \in \Omega \cup T^- | h \neq i} \sum_{a=1}^{\theta_h} z_{hiab}^{rv} = \sum_{j \in \Omega \cup T^+ \cup T^- | j \neq i} \sum_{c=1}^{\theta_j} z_{ijbc}^{rv} \quad (14)$$

$$\forall r \in R; v \in V; b \in \{1, \dots, \theta_i\}; \forall i \in N'$$

$$\sum_{v \in B_r} z_{t'taa}^{rv} = \sum_{j \in N'} \sum_{b=1}^{\theta_j} \sum_{v \in B_r} z_{t'tjab}^{rv} \quad \forall r \in R; \forall t \in T \quad (15)$$

$$\forall a \in \{1, \dots, \theta_t\}; t' = t + |T|; t'' = t + 2 \cdot |T|$$

$$\sum_{a=1}^{\theta_t} z_{t'tab}^{rv} - \sum_{c=1}^{\theta_t} z_{t't'bc}^{rv} \geq 0 \quad \forall r \in R; \forall t \in T \quad (16)$$

$$\forall b \in \{1, \dots, \theta_t\}; \forall v \in B_r; t' = t + |T|; t'' = t + 2 \cdot |T|$$

$$\sum_{a=1}^{\theta_t} z_{t't'ab}^{rv} - \sum_{j \in N} \sum_{c=1}^{\theta_j} z_{t't'jbc}^{rv} \geq 0 \quad \forall r \in R; \forall t \in T \quad (17)$$

$$\forall b \in \{1, \dots, \theta_t\}; \forall v \in B_r; t' = t + |T|; t'' = t + 2 \cdot |T|$$

Route continuity for vehicles is defined by constraint (13) which ensures that any vehicle arriving on any site (except vehicle depots) will leave for the next destination. Requests $r \in R$ do need the same flow conservation (14)

but only on Ω sites. However note that origin node S_r^+ and destination node S_r^- are excluded since physical flow must be interrupted to make pickup and drop-off operations happen. Any request $r \in R$ entering a transfer point must leave it, no matter which vehicle is transporting it (15). Thereby, transfer operations can be achieved for one or more requests when vehicles are visiting a transfer point. Equations (16) and (17) allow respectively pickup and drop-off operations only on the incoming transfer node and the outgoing transfer node (see fig. 2).

$$\sum_{v \in V} \sum_{b=1}^{\theta_t} x_{t'tab}^v \leq 1 \quad \forall t \in T; \forall a \in \{1, \dots, \theta_t\}; t' = t + |T| \quad (18)$$

$$\sum_{v \in V} \sum_{b=1}^{\theta_t} x_{t't''ab}^v \leq 1 \quad \forall t \in T; \forall a \in \{1, \dots, \theta_t\}; t'' = t + 2 \cdot |T| \quad (19)$$

Constraints (18) and (19) associate exactly one vehicle at maximum to one visit index. Thus for any transfer point $t \in T$, the maximum number of visits θ_t concern all vehicles, and not per vehicle. These two constraints are used to ensure that temporal constraints on transfer points are valid (eg. precedence, operations order between two vehicles during a transfer...).

$$\sum_{b=1}^{\theta_j} D_{ij1b}^v \geq e_i * \sum_{b=1}^{\theta_j} x_{ij1b}^v \quad \forall v \in V; i = w_v; \forall j \in N \setminus \{i\} \quad (20)$$

$$\sum_{a=1}^{\theta_h} A_{hia2}^v \leq l_i * \sum_{a=1}^{\theta_h} x_{hia2}^v \quad \forall v \in V; i = w_v; \forall h \in N \setminus \{i\} \quad (21)$$

$$A_{ijab}^v \geq D_{ijab}^v + x_{ijab}^v * d_{ij} / v_v \quad \forall i \in N; \forall j \in N | i \neq j \quad (22)$$

$$\forall a \in \{1, \dots, \theta_i\}; \forall b \in \{1, \dots, \theta_j\}; \forall v \in V$$

Vehicles $v \in V$ are allowed to leave their associated depot w_v only after the earliest date e_{w_v} (20) and must return in it before the latest date l_{w_v} (21). Temporal continuity between 2 nodes is ensured with constraint (22) considering average speed v_v and distance between i and j called d_{ij} .

$$\sum_{j \in N \setminus T^- | j \neq i} \sum_{c=1}^{\theta_j} D_{ijbc}^v \geq \sum_{h \in N \setminus \{T \cup T^+\} | h \neq i} \sum_{a=1}^{\theta_h} A_{hiab}^v \quad (23)$$

$$+ \sum_{r \in R | S_r^+ = i} (\delta_r^+ * \sum_{j \in N \setminus T^- | j \neq i} \sum_{c=1}^{\theta_j} z_{ijbc}^{rv})$$

$$+ \sum_{r' \in R | S_{r'}^- = i} (\delta_{r'}^- * \sum_{h \in N \setminus \{T \cup T^+\} | h \neq i} \sum_{a=1}^{\theta_h} z_{hiab}^{r'v})$$

$$\forall i \in \Omega \cup T^+; b \in \{1, \dots, \theta_i\}; \forall v \in V$$

$$\sum_{c=1}^{\theta_t} D_{tt''bc}^v \geq \sum_{a=1}^{\theta_t} A_{t'tab}^v + \sum_{r \in R} (\sigma_r^- (\sum_{a=1}^{\theta_t} z_{t'tab}^{rv} - \sum_{c=1}^{\theta_t} z_{tt''bc}^{rv}))$$

$$\forall t \in T; \forall b \in \{1, \dots, \theta_t\}; v \in V; t' = t + |T|; t'' = t + 2 \cdot |T| \quad (24)$$

$$\sum_{j \in N | j \neq t''} \sum_{c=1}^{\theta_j} D_{t''jbc}^v \geq \sum_{a=1}^{\theta_t} A_{tt''ab}^v$$

$$+ \sum_{r \in R} (\sigma_r^+ (\sum_{j \in N | j \neq t''} \sum_{c=1}^{\theta_j} z_{t''jbc}^{rv} - \sum_{a=1}^{\theta_t} z_{tt''ab}^{rv})) \quad \forall v \in V$$

$$\forall t \in T; \forall b \in \{1, \dots, \theta_t\}; t' = t + |T|; t'' = t + 2 \cdot |T| \quad (25)$$

Temporal continuity on nodes is conditioned by taking into account pickup and drop-off operations on origin/destination nodes in Ω (23) but also on incoming transfer points in T^+ (24) and outgoing transfer points in T^- (25). If multiple loading/unloading operations begin at the same time we consider the sum of all service times, and vehicles are free to move once all operations performed.

$$A_{ijab}^v \leq x_{ijab}^v * M \quad \forall i \in N; \forall j \in N | j \neq i$$

$$\forall a \in \{1, \dots, \theta_i\}; \quad \forall b \in \{1, \dots, \theta_j\}; \quad \forall v \in V \quad (26)$$

$$D_{ijab}^v \leq x_{ijab}^v * M \quad \forall i \in N; \forall j \in N | j \neq i$$

$$\forall a \in \{1, \dots, \theta_i\}; \quad \forall b \in \{1, \dots, \theta_j\}; \quad \forall v \in V \quad (27)$$

$$\sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} D_{ijab}^v \leq \sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} D_{ija'b}^v$$

$$+ M(1 - \sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} x_{ija'b}^v)$$

$$\forall i \in N \setminus T; \forall a \in \{1, \dots, \theta_i - 1\}; a' = a + 1; \forall v \in V \quad (28)$$

$$\sum_{v \in V} \sum_{b=1}^{\theta_j} D_{t'tab}^v + M(1 - \sum_{v \in V} \sum_{b=1}^{\theta_j} x_{t'tab}^v)$$

$$\leq \sum_{v \in V} \sum_{b=1}^{\theta_j} D_{t'ta'b}^v + M(1 - \sum_{v \in V} \sum_{b=1}^{\theta_j} x_{t'ta'b}^v) \quad (29)$$

$$\forall t \in T; \quad \forall a \in \{1, \dots, \theta_t - 1\}; \quad a' = a + 1$$

Constraints (26-27) enforce null values for arrival date A and departure date D when vehicle isn't passing on that edge. For all origin/destination sites in Ω and any vehicle $v \in V$, visit order is sorted chronologically thanks to equation (28). Same constraint is applied on transfer points in T but **regardless of vehicles** (one visit corresponds to one vehicle at most)

$$\sum_{r \in R} \sum_{j \in N | j \neq i} \sum_{c=1}^{\theta_j} (z_{ijbc}^{rv} * q_r) \leq Q_v \quad \forall i \in N \quad (30)$$

$$\forall b \in \{1, \dots, \theta_i\}; \quad \forall v \in V$$

Capacity constraint (30) allows vehicles to welcome a limited amount of entities on board at any time, corresponding to the defined capacity Q_v .

$$\sum_{h \in N | h \neq i} \sum_{a=1}^{\theta_h} A_{hiab}^v \geq e_i * \sum_{h \in N | h \neq i} \sum_{a=1}^{\theta_h} x_{hiab}^v \quad \forall i \in \Omega$$

$$v \in V; \forall b \in \{1, \dots, \theta_i\} \quad (31)$$

$$\sum_{j \in N | j \neq i} \sum_{c=1}^{\theta_j} D_{ijbc}^v \leq l_i \quad \forall i \in \Omega \quad (32)$$

$$\forall v \in V; \forall b \in \{1, \dots, \theta_i\}$$

$$\sum_{h \in N | h \neq t'} \sum_{a=1}^{\theta_h} A_{ht'ab}^v \geq e_t * \sum_{h \in N | h \neq t'} x_{ht'ab}^v \quad (33)$$

$$\forall t \in T; t' = t + |T|; \forall v \in V; \forall b \in \{1, \dots, \theta_t\}$$

$$\sum_{j \in N | j \neq t''} \sum_{c=1}^{\theta_j} D_{t''jbc}^v \leq l_t \quad (34)$$

$$\forall t \in T; t'' = t + 2 \cdot |T|; \forall v \in V; \quad \forall b \in \{1, \dots, \theta_t\}$$

$$\sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} D_{ijab}^v - \delta_r^+ + M(1 - \sum_{j \in N | j \neq i} \sum_{b=1}^{\theta_j} z_{ijab}^{rv}) \geq p_r^+$$

$$\forall r \in R; \quad \forall v \in V; \quad \forall a \in \{1, \dots, \theta_i\}; \quad i = S_r^+ \quad (35)$$

$$\sum_{j \in N | j \neq i} \sum_{c=1}^{\theta_j} D_{ijbc}^v - \delta_r^- - M(1 - \sum_{h \in N | h \neq i} \sum_{a=1}^{\theta_h} z_{hiab}^{rv}) \leq p_r^-$$

$$\forall r \in R; \quad \forall v \in V; \quad \forall b \in \{1, \dots, \theta_i\}; \quad i = S_r^- \quad (36)$$

Constraints (31-32) apply on all vehicles so any visit matches time windows $[e_i; l_i]$ on any origin/destination site $i \in \Omega$. Constraints (33-34) are also time window constraints arising for transfer points $t \in T$, such all vehicles can only visit them during time window $[e_t; l_t]$. Finally, requests $r \in R$ can only be handled within a specific time window $[p_r^+; p_r^-]$ on origin node S_r^+ and destination node S_r^- (35-36).

$$\begin{aligned}
& \sum_{j \in N | j \neq t''} \sum_{b'=1}^{\theta_j} D_{t''j a' b'}^{v'} + M(1 - \sum_{j \in N | j \neq t''} \sum_{b'=1}^{\theta_j} z_{t''j a' b'}^{r v'}) \\
& \geq \sum_{b=1}^{\theta_t} D_{t t'' a b}^v + \delta_r^+ - M(1 - \sum_{b=1}^{\theta_t} z_{t t'' a b}^{r v}) \\
& \forall v \in V; \quad \forall v' \in V; \quad \forall r \in R; \quad \forall t \in T; \quad t' = t + |T| \\
& t'' = t + 2 \cdot |T|; \quad \forall a \in \{1, \dots, \theta_t\}; \quad \forall a' \in \{1, \dots, \theta_t\}
\end{aligned} \tag{37}$$

$$\begin{aligned}
& \sum_{a'=1}^{\theta_t} A_{t t'' a' b'}^{v'} - M(1 - \sum_{j \in N | j \neq t''} \sum_{c=1}^{\theta_j} z_{t''j b' c}^{r v'}) - M * \tau_{t b'}^{r v v'} \\
& \leq \sum_{b=1}^{\theta_t} D_{t t'' a b}^v - \delta_r^- + M(1 - \sum_{b=1}^{\theta_t} z_{t t'' a b}^{r v}) - M * \tau_{t b'}^{r v v'} \\
& \forall r \in R; \quad \forall v \in B_r; \quad \forall v' \in B_r; \quad \forall t \in T; \quad t' = t + |T| \\
& t'' = t + 2 \cdot |T|; \quad \forall a \in \{1, \dots, \theta_t\}; \quad \forall b' \in \{1, \dots, \theta_t\}
\end{aligned} \tag{38}$$

$$\sum_{r \in R} \sum_{v \in B_r} \sum_{v' \in B_r} (\tau_{t a}^{r v v'} * q_r) \leq s c_t \quad \forall t \in T; \forall a \in \{1, \dots, \theta_t\} \tag{39}$$

Transfers operations have been discussed earlier in this paper, and two situations have been presented. In both cases, precedence constraint (37) must always be applied so vehicle $v \in V$ and $v' \in V$ can exchange one request $r \in R$, including service time δ_r^+ needed to exchange it. However if there is no storage, another temporal constraint has to be applied : synchronization constraint (38). Both vehicles are sharing a common time window so requests can be transhipped from on vehicle to another. In the other case where storage arises, constraint (39) ensures the respect of storage capacity at any visit (since one visit correspond to one vehicle at most, visiting the transfer point).

IV. COMPUTATIONAL RESULTS

Problem formulation implies to create new instances as no similar ones have been found in literature to our knowledge. 4 initial fictive instances were created (F00-F03). Each instance is studied in the case with no transfer operation, in the case with transfer operation without storage and in the case with transfer and storage allowed. Finally these 3 new instances are studied in both mono-visit and multi-visit cases. Given results (fig. 3) are obtained with exact methods using IBM CplexTM solver on a Xeon E7-4850 @ 2.4 GHz with 16 cores and 32 threads.

As expected instances are solved widely faster considering mono-visit over multi-visit, and faster without transfer operations allowed. In mono-visit, saving for instance F01 obtained by allowing transfers without storage is 1.6% and raise to 3.7% with storage. Allowing several visits for each vehicle leads to 12.7% of saving which is significant for this size of instance. Saving for F02 is only 1.2% and is obtained by whether allowing transfer operations with

v: vehicles - w: depots - t: transfer - n: sites - r: requests	without transfer		transfer without storage		transfer with storage	
	CPU(s)	Obj.	CPU(s)	Obj.	CPU(s)	Obj.
F00: 2v 2w 1t	0.1	<i>Infeasible</i>	0.5	<i>Infeasible</i>	0.6	3698
4n 3r	0.2	<i>Infeasible</i>	1.7	3532	1.7	3532
F01: 2v 2w 1t	0.4	3939	1.1	3873	1.1	3792
6n 4r	0.5	3437	0.8	3437	0.8	3437
F02: 2v 2w 1t	3.2	3873	22.6	3873	27.4	3828
6n 6r	2632.9	3828	2640.0	3828	3118.0	3828
F03: 2v 2w 1t	0.7	5617	58.1	5617	54.6	5483
6n 4r	0.6	4380	5.7	4380	1.6	4380

Fig. 3. Computational results

storage or allowing vehicles to visit each site more than one time. Computational times are really high when solving multi-visit variant of this instance because of the amount of requests considered and so the potential number of visit increasing for each site. Instance F03 shows 2.3% saving when allowing transfer operations and storage in mono-visit, but saving is extended to 22% when allowing several visits.

Because of low scalability potential, exact methods can only deal with few amount of requests, vehicles and sites. Finally, let's illustrate results by considering one fictive instance F00 (fig. 4) composed of 2 vehicles, 2 depots, 4 sites and one transfer point. The 2 vehicles must satisfy 3 requests with heterogeneous quantities. Also each vehicle has its own time window ([0;60] for the first one since it's affected to vehicle depot number 0, and [40;100] for the second one associated to vehicle depot number 1). From this instance, 6

Vehicles V	1	2
Capacity (Q)	10	10
Speed (v)	50	50
Assoc. depot (w)	0	1

Depots W	0	1	Transfer p. T	6
Position x	200	1000	Position x	600
Position y	400	400	Position y	400
Opening date (e)	0	40	Opening date (e)	0
Closing date (l)	60	100	Closing date (l)	100

Sites Ω	2	3	4	5
Position x	400	800	400	800
Position y	200	200	600	600
Opening date (e)	0	0	0	0
Closing date (l)	100	100	100	100

Requests R	1	2	3
Origin (S ⁻)	2	4	4
Destination (S ⁻)	3	3	5
Quantity (q)	5	10	5
Pickup durat. (δ ⁻)	5	5	5
Deliv. durat. (δ ⁻)	5	5	5
Earliest pick. (p ⁻)	0	0	0
Latest deliv. (p ⁻)	100	100	100
Vehicles ass. (B)	{1,2}	{1,2}	{1,2}

Fig. 4. Fictive instance F00

different variants are generated, organized as follow. Transfer operations are forbidden (1), transfer operations are allowed only without storage (2) and finally allowing storage (3). One visit per vehicle is allowed for 1-3, i.e. in mono-visit, and we

duplicate these instances allowing multiple visits per vehicle (4 to 6), i.e. in multi-visit.

Numeric results obtained are shown in (fig. 3) and prove that this instance can be solved in mono-visit only if transfer operations with storage are allowed (solution in fig. 5).

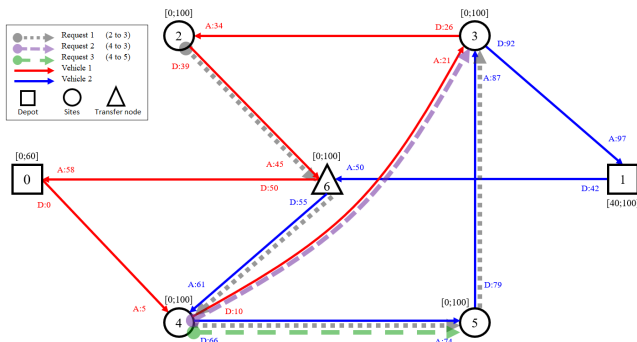


Fig. 5. F00 solved with mono-visit and transfer with storage allowed

However, better solution can be found if a vehicle can visit one site multiple times (multi-visit) and for this instance saving appeared to be almost 5% (solution in fig. 6). However, one can safely speculate about narrow relations between requests density and distribution among origin/destination sites, the number of vehicles being available (time window and load on board), and savings that can be made using multi-visit. Another speculation is about savings made by allowing transfer operations. In the case of 2 vehicles that are willing to proceed a transshipment, one first observation is towards their respective time windows. Indeed, if both vehicles are sharing a common time window (i.e. joint time windows) then transfer operation will more likely happen with synchronization and so without storage. One major interest in this particular case would be to allow storage because vehicle being transhipped could have a temporary lack of free space at the date considered. The other observation concerns the case where both vehicles aren't sharing a common time window (i.e. disjointed time windows). Vehicles will most likely transship request if there is a transfer point with storage capacity and of course free space in it, for the complete duration of storage.

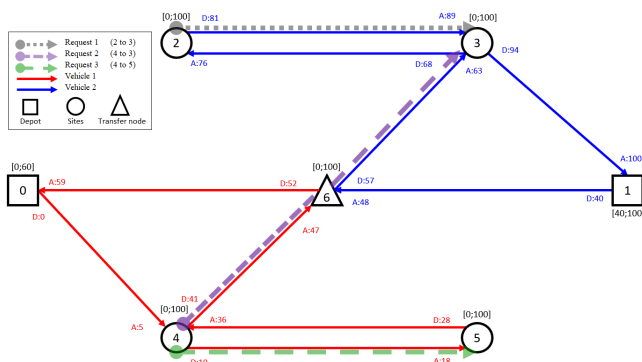


Fig. 6. F00 solved with multi-visit and transfer without storage (same solution obtained with storage allowed)

V. CONCLUSION

In this paper, a general formulation has been developed to answer new urban mobility challenges, considering both goods and passengers viewpoints. This variant of PDP with time windows includes transfer operations with and without storage, vehicles are allowed to visit one site several times, and it ensures that vehicles must visit sites and transfer points within defined time windows. Computational results show potential savings on some instances but also show complexity and limits in terms of scalability. We can observe that using transfer points does not always improve the objective function. Indeed, the location of these points can strongly impact the quality of the solutions. The next step in our work will be to develop rules that will determine the number of transfer points and their position. Solving large examples is very difficult with this model, but is still a starting point for defining metaheuristics to solve this PDP variant.

REFERENCES

- [1] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, Static Pickup and Delivery Problems: A Classification Scheme and Survey, 2007.
- [2] G. Berbeglia, J. F. Cordeau, G. Laporte, and H. Waisanen, Dynamic Pickup and Delivery Problem, *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 815, 2010.
- [3] P. Bours, D. Sacharidis, T. Dalamagas, T. Sellis, Dynamic pickup and delivery with transfers. *Lecture Notes in Computer Science*, Vol. 6849 LNCS, pp. 112129, Springer, Berlin, Heidelberg, 2011.
- [4] J.-F. Cordeau and G. Laporte, The dial-a-ride problem: models and algorithms, *Ann. Oper. Res.*, vol. 153, no. 1, pp. 2946, Jun. 2007.
- [5] C. E. Cortés, M. Matamala, C. Contardo, The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, vol. 200, 711724, Jan. 2009.
- [6] S. G. Dacko and C. Spalteholz, Upgrading the city: Enabling inter-modal travel behaviour, *Technol. Forecast. Soc. Change*, vol. 89, pp. 222 - 235, Nov. 2014.
- [7] G. B. Dantzig and J. H. Ramser, The Truck Dispatching Problem, *Manage. Sci.*, vol. 6, no. 1, pp. 8091, 1959.
- [8] S. Deleplanque, A. Quilliot, Dial-a-ride problem with time windows, transshipments, and dynamic transfer points. *IFAC Proceedings Volumes*, Vol. 46, IFAC, 2013.
- [9] Y. Dumas, J. Desrosiers, and F. Soumis, The pickup and delivery problem with time windows, *Eur. J. Oper. Res.*, vol. 54, no. 1, pp. 722, Sep. 1991.
- [10] I. Flores, I. Chatziioannou, E. Segura, S. Hernández, Urban transport infrastructure: A state of the art. *Proceedings of the European Modelling and Simulation Symposium*, pp. 8392, Athens, Sep. 2013.
- [11] L. Grandinetti, F. Guerriero, F. Pezzella, and O. Pisacane, The Multi-objective Multi-vehicle Pickup and Delivery Problem with Time Windows, *Procedia - Soc. Behav. Sci.*, vol. 111, pp. 203212, Feb. 2014.
- [12] C. K. Y. Lin, A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources, *Comput. Oper. Res.*, vol. 38, no. 11, pp. 15961609, Nov. 2011.
- [13] R. Masson, F. Lehuédé, O. Péton, The dial-a-ride problem with transfers. *Computers and Operations Research*, 41(1), 1223, 2014.
- [14] S. Ropke, J.-F. Cordeau, and G. Laporte, Models and a Branch-and-Cut Algorithm for Pickup and Delivery Problems with Time Windows, *Networks*, vol. 49, pp. 258272, 2007.
- [15] M. W. P. Savelsbergh and M. Sol, The general pickup and delivery problem, *Transp. Sci.*, vol. 29, no. 1, pp. 1729, 1995.
- [16] A. Spickermann, V. Grienz, and H. A. Von Der Gracht, Heading towards a multimodal city of the future? Multi-stakeholder scenarios for urban mobility, 2014.