

Pickup & Delivery Problem with Multiple Time Windows

Preben Bucher-Johannessen

February 27, 2019

Contents

1	Introduction	1
2	Problem Formulation	2
3	Final Remarks	7

1 Introduction

The automobile industry is one of the most significant industries in Europe today which is facing high cost-reduction pressure from fierce international competition and difficult conditions.[1] The supply chain process in automobile industry requires a lot of flexibility in their planning and modelling of transportation which often takes the form of some sort of vehicle transportation problem. In planning the most efficient way of organizing your transportation network the classical VRPTW problem is likely the first problem that comes to mind.

[18] The Vehicle Routing Problem with Time Window Constraints (VRPTW) is defined as the problem of minimizing costs when a fleet of homogeneous vehicles has to distribute goods from a depot to a set of customers satisfying time windows and capacity constraints [6]. Another classical problem is the PDPTW, Pickup and Delivery Problem with Time Windows, with multiple vehicles is another problem that has been widely researched and solved for example in [17].

[6] Presents a VRPTW Problem as an extension of the Capacitated Vehicle routing problem, with hard and soft timewindows, where soft can be broken against a cost and hard time-windows cannot. They present a multi-commodity network flow formulation of the problem and use approximation methods to derive upper bounds and use Lagrangean relaxation and column generation to derive lower bounds. To produce integer solutions they use cutting and branching strategies and finally they also present special cases and extensions. [20] formalized an algorithm for general PDP [16] insertion based heuristic [15] An algorithm for the travelling salesman problem with pickup and delivery of customers [9] pickup and delivery problem with time windows [10] speed optimization and reduction of fuel emissions [3] dynamic and static pickup and delivery problems survey and classification scheme [2] static [21] Inventory management and inbound logistics optimization [19] General pickup and delivery problem. [8] Properties and solution framework

[4] present a ship scheduling PDP problem of bulk cargoes with multiple time windows. They also handle specific shipping industry challenges regarding ship idle time, transport risks due to weather and unpredictable service time at ports. They use a set partitioning approach to solve the problem and found that you could increase the robustness of the schedules on account of costs. [5] Ship routing and scheduling [13] Consider a class of cargo ship routing and scheduling problems from tramp and industrial shipping industry. They provide solutions to a wide range of benchmark instances both optimal for smaller instances, using a commercial mixed-integer programming solver. And they present adaptive large neighbourhood search heuristics to larger instances. They also provide the benchmark instances and an instance generator. [14] A iterative two-phase hybrid metaheuristic for multi-product short sea inventory routing problem [12] variable neighbourhood search for vehicle routing problem with multiple time windows [7] Time constrained routing and scheduling

In this paper we will propose a mathematical formulation to a multivehicle PDPTW problem, with certain adjustments to satisfy the needs of an automobile manufacturing company. To these type of companies the inbound production side of the supply chain is more in focus and require adjustments on the classical PDPTW. In these type of industries companies often facing challenges related to the production side of the problem, which is often referred to as looking at the problem from an inbound perspective. Inbound oriented perspective means that you have less focus on the the supplier side of the of the production, and on the means of transportation as long as what is being transported arrives on time for production. This changes our problem away from the classical problems mentioned above. Instead of having a customer oriented view, we must instead shift our view to focus on getting a set of orders from a set of suppliers to a set of factories where the orders are needed for production.

Being more inbound oriented also leads to having more focus on the problems that sometimes occur within the factory and the delivery part of the problem. Some manufacturers might have a lot of traffic on their factories, and want to limit that, saying that a vehicle can only visit a certain amount of docks for each visit. This might lead to more ineffective solutions and more use of vehicles in some cases however, for the inbound oriented producer it is a necessary limitation. To our knowledge dock constraints has not been researched before in PDPTW problems.

Another result from the inbound thinker is that instead of using your own fleet of vehicles, you hire a logistics carrier to do your transports. The carriers might have different cost structures and payment methods which you have no influence over. Leading you to need a model that takes that into account. We focus here on dealing with a carrier that offers a varying cost per kilometer. Meaning that the cost parameter of the standard PDP will change depending on how far a vehicle is travelling. Using a carrier also changes the problem to that you don't care so much where the vehicle is travelling from to its first pickup. The vehicle rather starts at its first pickup point because the cost of the car manufacturer starts only from the time of the first pickup. Before pickup and after delivery the carrier has the costs. To our knowledge very little research has been made into using a logistics carrier as vehicle fleet.

In car manufacturing business you might have production periods that go over several days with strict opening times on delivery. This leads to the problem of multiple time windows. The manufacturer might only want the parts delivered in the morning on one of three days but have a break in the middle of each time window (lunch). There might also be advantageous to move the delivery of one order to the next day to bundle orders together and we want our model to be able to handle such problems.

In [11] they propose a time window model to handle multiple time windows and multiple visits. We are using a modified version of this model to handle the multiple time windows in this paper to handle the time window constraints. However as they focus on satisfying a set of customers instead of production companies our model differs greatly when it comes to the rest of the constraints.

To make a model that is as realistic as possible it is important to integrate each of the above aspects and take them all into account. One aspect of the model could greatly influence the decision of another so it is important to build a model that satisfies all the aspects above. Research often focuses on solving single issues, like the multiple time windows, to handle this specific problem for a certain industry. However handling all of the above mentioned aspects have to our knowledge never been researched before and makes it therefore an important problem to solve.

The goal of this paper is to present a realistic mathematical model that can be further developed into a realistic model solving pickup and delivery problems on a daily/weekly basis for manufacturing companies. The goal is also to have a mathematical basis to be used in a solver such as AMPL to solve a small instance, and finally to make a heuristic model that can solve the problem efficiently and be used by companies in similar sort of industries.

2 Problem Formulation

Like mentioned above to take the perspective of a vehicle manufacturing company, we see that we have a certain set of factories where the products are produced, each with a set of demanded parts/orders for production. The parts, or orders (typically named transport orders but we refer to orders here), can be delivered from a set of suppliers. At a given point in time (could be delivery for one day or a week) we consider a planning problem with a demand for orders that should be satisfied with the given suppliers using vehicles provided by different logistic carriers.

	Sets
N	- nodes $\{1, \dots, 2n\}$ where n is amount of orders
V	- vehicles
E	- edges
E_v	- edges visitable by vehicle v
N_v	- nodes visitable by vehicle v
N^P	- pickup nodes
N^D	- delivery Nodes
F	- factories
N_f	- delivery nodes for factory f
A_v	- set of pairs (α, β) , repr. index of elements in the cost matrix of vehicle v
P_i	- set of time windows at node i , $\{1, \dots, \pi_i\}$
T_i	- set of time parameters $[T_{ip}, \overline{T_{ip}}]$ at node i where $p \in P_i$
S	- set of stops indicating a pickup/delivery location
L_s	- Sets of nodes sharing a stop location $s \in S$

The vehicles have different capacities, cost structures, incompatibilities and start at the first pickup location at the first pickup time (ie. costs to get from start to first pickup are not relevant to the manufacturer). The cost structure depend on the total distance a vehicle is driving and is calculated per km.

When a delivery is assigned to a vehicle, the vehicle must load the delivery from the supplier, and deliver the delivery at the factory dock. The docks (each represented by a different node) in each factory can differ according to which order is being delivered and there is a limit to how many docks each vehicle can unload at per visit to the factory.

Each delivery/pickup can have several time windows, lapsing sometimes over several days. If a car arrives before a time window it has to wait.

The mathematical formulation of the problem will now follow. The problem can be viewed as a graph $G(E, N)$ where $N = \{0, \dots, 2n\}$ are the vertices and n is the number of orders in the problem, and $E = \{(i, j) : i, j \in N, i \neq j\}$ represent the edges in the graph.

Parameters		
n	-	amount of orders
K_v	-	weight limit of vehicle $v \in V$
$o(v)$	-	starting node of vehicle v
$d(v)$	-	ending node of vehicle v
Q_i	-	weight of order at node $i \in N$
H_f	-	docking limit at factory $f \in F$
T_{ijv}	-	travel time for vehicle $v \in V$ over edge $(i, j) \in E_v$
π_i	-	amount of time windows at node $i \in N$
\overline{T}_{ip}	-	upper bound time of time window $p \in P_i$ at node $i \in N$
T_{ip}	-	lower bound time of time window $p \in P_i$ at node $i \in N$
$\overline{C}_{v\alpha\beta}^{km}$	-	cost per distance unit (km) in cost matrix element $(\alpha, \beta) \in A$ for vehicle v
$\overline{C}_{v\alpha\beta}^{kg}$	-	cost per weight unit (kg) in cost matrix element $(\alpha, \beta) \in A$ for vehicle v
$\overline{C}_{v\alpha\beta}^{fix}$	-	fixed cost in index $(\alpha, \beta) \in A$ for vehicle v
\overline{C}_i^{stop}	-	costs of making a stop at node i
\overline{C}_i	-	cost of not transporting order at node $i \in N^P$
D_{ij}	-	distance between node $i \in N$ and $j \in N$
B_α	-	distance for interval α in cost matrix A column index
Z_β	-	weight for interval β in cost matrix A row index

The set of vehicles used is denoted by V and capacity of each vehicle $v \in V$ is denoted by K_v . The set of Edges that each vehicle can traverse is represented by E_v . Since n is the number of orders in the problem, then if i is a specific pickup-node then $i + n$ corresponds to the delivery node for the same order. The set of pickup Nodes (supplier docks) we denote using N^P and each delivery node (Factory dock) is denoted by N^D . All nodes are therefore equivalent to $N = N^P \cup N^D$. Each Factory, $f \in F$, also has a set of Nodes belonging to the same factory which we denote N_f . Since all factories are delivery nodes these sets only include delivery nodes. Each vehicle has a set of Nodes it can travel to represented by N_v . This set also includes an origin node, $o(v)$ and a destination node $d(v)$ corresponding to the first pickup and the last delivery of vehicle v . The factory docking limit is denoted by H_f . Each vehicle has a given amount of docks per factory it wants to deliver to represented by h_f .

Each node has a set T_i of time windows represented by $[T_{ip}, \overline{T}_{ip}] \in [0, T]$ where $p \in P_i = \{0, 1, \dots, \pi_i\}$ and all nodes should be picked up and delivered after time 0 and before time T . Each node has a current time based on when its being served, denoted by t_i and where $i \in N$. The distance from node i to node j is denoted by D_{ij} . The cost per distance unit of vehicle v for a cost structure interval s is represented by \overline{C}_{vs} and the corresponding time between the nodes T_{ijv} . The total distance travelled by vehicle v will be denoted by the variables d_{vs} for each $s \in S$, where only one variable will have the value equal to the total distance of vehicle. Which d_{vs} has a value will be determined by the binary b_{vs} and the distance interval parameter B_s , where B_s indicated which distance interval we are in which again results in a different \overline{C}_{vs} .

Variables

x_{ijv}	-	binary indicating travel from node $i \in N$ to $j \in N$ of vehicle $v \in V$
y_i	-	binary indicating no pickup of order at node $i \in N^P$
l_{iv}	-	load of vehicle v at node i
h_i	-	docking times in factory at node $i \in N_f$
t_i	-	time of visit at node $i \in N$
u_{ip}	-	binary indicating usage of time window $p \in P_i$ at node i
$d_{v\alpha\beta}$	-	total distance travelled of vehicle $v \in V$ if it fits in interval $(\alpha, \beta) \in A_v$
$b_{v\alpha\beta}$	-	binary indicating interval $(\alpha, \beta) \in A_v$ for vehicle $v \in V$
$l_{v\alpha\beta}$	-	maximum weight carried by vehicle $v \in V$ for interval $(\alpha, \beta) \in A$

l_{iv} is the load of vehicle v leaving node i and x_{ijv} is a binary variable indicating if vehicle v is travelling between i and j node. The cost of not transporting an order will be represented by C_i for each node i , with a corresponding binary variable y_i indicating if an order is not picked up.

$$\min \sum_{v \in V} \sum_{(\alpha, \beta) \in A} (C_{v\alpha\beta}^{km} d_{v\alpha\beta} + C_{v\alpha\beta}^{kg} l_{v\alpha\beta} + C_{v\alpha\beta}^{fix} b_{v\alpha\beta}) + \sum_{v \in V} \sum_{s \in S} \sum_{\substack{i \in L_s \\ j \in N_v \notin L_s}} C_i^{stop} x_{ijv} + \sum_{i \in N^P} C_i y_i \quad (1)$$

subject to:

$$\sum_{v \in V} \sum_{j \in N_v} x_{ijv} + y_i = 1, \quad i \in N^P \quad (2)$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{jiv} = 0, \quad v \in V, i \in N_v \notin \{o(v), d(v)\} \quad (3)$$

$$\sum_{j \in N_v} x_{o(v)jv} = 1, \quad v \in V \quad (4)$$

$$\sum_{j \in N_v} x_{jd(v)v} = 1, \quad v \in V \quad (5)$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{(i+n)jv} = 0, \quad v \in V, i \in N_v^P \quad (6)$$

$$l_{iv} + Q_j - l_{jv} \leq K_v(1 - x_{ijv}), \quad v \in V, j \in N_v^P, (i, j) \in E_v \quad (7)$$

$$l_{iv} - Q_j - l_{(j+n)v} \leq K_v(1 - x_{i(j+n)v}), \quad v \in V, j \in N_v^P, (i, n+j) \in E_v \quad (8)$$

$$0 \leq l_{iv} \leq K_v, \quad v \in V, i \in N_v^P \quad (9)$$

$$h_i + 1 - h_j \leq (H_f + 1)(1 - x_{ijv}), \quad v \in V, f \in F, i \in N_f, j \in N_f, j \neq i \quad (10)$$

$$h_j \leq H_f \sum_{i \in N_v} (x_{ijv}), \quad v \in V, f \in F, j \in N_f, \quad (11)$$

$$h_j = \sum_{\substack{i \in N_v \\ i \notin N_f}} (x_{ijv}), \quad v \in V, j \in N_f \quad (12)$$

$$\sum_{p \in P_i} u_{ip} = 1, \quad i \in N \quad (13)$$

$$\sum_{p \in P_i} u_{ip} \underline{T}_{ip} \leq t_i, \quad i \in N \quad (14)$$

$$\sum_{p \in P_i} u_{ip} \overline{T}_{ip} \geq t_i, \quad i \in N \quad (15)$$

$$t_i + T_{ijv} - t_j \leq (\overline{T}_{\pi_i i} + T_{ijv})(1 - x_{ijv}), \quad v \in V, (i, j) \in E_v \quad (16)$$

$$t_i + T_{i(i+n)v} - t_{(i+n)} \leq 0, \quad v \in V, i \in N_v^P \quad (17)$$

$$\sum_{(\alpha, \beta) \in A} d_{v\alpha\beta} = \sum_{(i, j) \in E_v} x_{ijv} D_{ij}, \quad v \in V \quad (18)$$

$$\sum_{(\alpha, \beta) \in A} l_{v\alpha\beta} \geq l_{iv} \quad v \in V, i \in N_v \quad (19)$$

$$B_{(\alpha-1)} b_{v\alpha\beta} \leq d_{v\alpha\beta} \leq B_{\alpha} b_{v\alpha\beta}, \quad v \in V, (\alpha, \beta) \in A \quad (20)$$

$$Z_{(\beta-1)} b_{v\alpha\beta} \leq l_{v\alpha\beta} \leq Z_{\beta} b_{v\alpha\beta}, \quad v \in V, (\alpha, \beta) \in A \quad (21)$$

$$\sum_{(\alpha, \beta) \in A} b_{v\alpha\beta} \leq \sum_{j \in N_v} x_{o(v)jv}, \quad v \in V \quad (22)$$

$$h_i, t_i \geq 0, \quad i \in N \quad (23)$$

$$u_{ip} \in \{0, 1\}, \quad i \in N, p \in P_i \quad (24)$$

$$b_{v\alpha\beta} \in \{0, 1\}, \quad v \in V, (\alpha, \beta) \in A \quad (25)$$

$$d_{v\alpha\beta}, l_{v\alpha\beta} \geq 0 \quad v \in V, (\alpha, \beta) \in A \quad (26)$$

$$y_i \in \{0, 1\}, \quad i \in N^P \quad (27)$$

$$x_{ijv} \in \{0, 1\}, \quad v \in V, (i, j) \in E_v \quad (28)$$

The objective function (1) sums up to the cost of all vehicles given corresponding costs from their cost matrix. Costs could be variable per distance, weight, fixed and/or related to stops made. Loads not transported will be penalized with costs and the aim is to minimize the sum of all these costs. Constraint 2 secures that a load is either picked up or not, ie. either x_{ijv} has a value of 1 or y_i . 3 makes certain that if a load at a node j is travelled to it is also left. This is not the case for origin and final destination node. Then 4 makes sure that origin is only left once by each vehicle, and 5 ensures that destination is only arrived to once. Finally 6 is there to say if a load is picked up it is also delivered.

Regarding weight, constraint 7 ensures that if a load at node j is picked up, the weight before plus the weight of the load at node j minus the weight leaving the node is equal to zero. If not the summation cannot exceed the capacity of the vehicle. Also 8 makes certain that at delivery of node j the weight before delivery minus the weight of the load at j minus the weight after,

is equal to zero, ie. the weight of the vehicle after is lighter than before by exactly the weight of the load at j . The final weight constraint 9 says that weight leaving node i always has to be between 0 and the capacity of the vehicle.

It follows from constrain 10 that within a certain factory, if you travel between two nodes, the amount of stops you have made should always be iterated by one. If you are not travelling between them, the summation cannot be bigger than the limitation of the factory-stops. The next constraint on factories 11, ensures that any node visited in a factory cannot exceed the docking limit. Then 12 makes sure that when a vehicle enters a factory from outside, the docking amount gets an initial value of 1.

In 13

-6 makes sure that all orders that are picked up are also delivered and that orders has to either be picked up or not. They also make sure that nodes that are visited are also left except for the starting and finishing nodes. 7-9 are weight capacity constraints. Constraint 10-12 are dock constraints making sure no vehicle visits more than the allowed amount of docks per visit. Then 13-16 are handling the multiple time windows to make sure an order is picked up within a given time window with regards to waiting time. Constraint 17-10 are deciding which interval the total distance is in which in return determines the calculation of the objective function.

3 Final Remarks

The model presented in this paper is ready for testing in a solver software such as AMPL. Making a small instance to test the model would then be the next step before adapting the model to an even more complex version of itself and developing a heuristic to solve the problem.

References

- [1] 4flow AG. Industries & references - automotive manufacturers, www.4flow.com. 2018.
- [2] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31, 2007.
- [3] G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15, 2010.
- [4] M. Christiansen and K. Fagerholt. Robust ship scheduling with multiple time windows. *Naval Research Logistics (NRL)*, 49(6):611–625, 2002.
- [5] M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation science*, 38(1):1–18, 2004.
- [6] J.-F. Cordeau and Q. Groupe d’études et de recherche en analyse des décisions (Montréal. *The VRP with time windows*. Montréal: Groupe d’études et de recherche en analyse des décisions, 2000.
- [7] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Time constrained routing and scheduling. *Handbooks in operations research and management science*, 8:35–139, 1995.
- [8] M. Dror, G. Laporte, and P. Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation science*, 23(3):166–176, 1989.

- [9] Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European journal of operational research*, 54(1):7–22, 1991.
- [10] K. Fagerholt, G. Laporte, and I. Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3):523–529, 2010.
- [11] D. Favaretto, E. Moretti, and P. Pellegrini. Ant colony system for a vrp with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, 10(2):263–284, 2007.
- [12] H. S. Ferreira, E. T. Bogue, T. F. Noronha, S. Belhaiza, and C. Prins. Variable neighborhood search for vehicle routing problem with multiple time windows. *Electronic Notes in Discrete Mathematics*, 66:207–214, 2018.
- [13] A. Hemmati, L. M. Hvattum, K. Fagerholt, and I. Norstad. Benchmark suite for industrial and tramp ship routing and scheduling problems. *INFOR: Information Systems and Operational Research*, 52(1):28–38, 2014.
- [14] A. Hemmati, L. M. Hvattum, M. Christiansen, and G. Laporte. An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem. *European Journal of Operational Research*, 252(3):775–788, 2016.
- [15] B. Kalantari, A. V. Hill, and S. R. Arora. An algorithm for the traveling salesman problem with pickup and delivery customers. *European Journal of Operational Research*, 22(3):377–386, 1985.
- [16] Q. Lu and M. M. Dessouky. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175(2):672–687, 2006.
- [17] W. P. Nanry and J. W. Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121, 2000.
- [18] S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(2):81–117, 2008.
- [19] M. W. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation science*, 29(1):17–29, 1995.
- [20] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.
- [21] H. Zhou. *Inventory Management and Inbound Logistics Optimization for a Food Processing Company*. PhD thesis, University of Cincinnati, 2013.