International Journal of Operations Research

International Journal of Operations Research Vol. 8, No. 1, 36−56 (2011)

Hybrid MIP method for a Pickup and Delivery Problem with Time Windows and Dock Service Constraints

Fabian Lopez-Pérez *

Universidad Autónoma de Nuevo León, CEDEEM, Ave Universidad 66451, Monterrey, México

Received July 2010; Revised December 2010, January 2011; Accepted January 2011

Abstract — We consider a pickup and delivery vehicle routing problem (PDP) commonly find in real-world logistics operations. The problem includes a set of practical complications that have received little attention in the vehicle routing literature. In this problem, there are multiple vehicle types available to cover a set of transportation orders, each of which has pickup time windows and delivery time windows. Transportation orders and vehicle types must satisfy a set of compatibility constraints that specify which orders can or cannot be covered by which vehicle types. In addition we include some dock service capacity constraints as is required on common real world operations when there is a large quantity of vehicles to schedule. This problem requires to be attended on large scale instances (transportation orders ≥ 500), (single-haul vehicles ≥ 100). As a generalization of the traveling salesman problem, clearly this problem is NP-hard. The exact algorithms are too slow for large scale instances. The PDP-TWDS is both a packing problem (assign order to vehicles), and a routing problem (find the best route for each vehicle) with several side constraints. We propose a model to solve the problem in three stages. The first stage constructs initial solutions at aggregated level relaxing time windows and dock service constraints on the original problem. The other two stages imposes time windows and dock service constraints within a cut generation scheme. Our results are favorable in finding good quality solutions in relatively short computational times.

Keywords —Optimization, vehicle routing, logistics & distribution planning, scheduling, time windows.

1. INTRODUCTION

Multiple Vehicle Pickup and Delivery Problem with Time Windows and Dock Service Constraints (PDP-TWDS) is an important problem in logistics and transportation management. The PDP-TWDS is a variant of the well-known Vehicle Routing Problem with Time Windows (VRP-TW). Particularly, our real-world application deals with the schedule of a transportation operation on a network with several plants and distribution centers. Vehicle routing plays a central role in logistics management. A wide variety of vehicle routing problems have been studied in the literature. Different vehicle routing problems address different practical situations but focus on a common and a simple problem, the efficient use of a fleet of vehicles that must pick up and/or deliver a set of transportation orders within a time window framework. This implies to identify which transportation orders should be covered by each vehicle and at what times so as to minimize the total transportation cost subject to a variety of constraints and complications.

The model we propose on this work is integrated in an interactive and user-friendly Geographic Information System (GIS) application, named MAPINFO. This paper illustrates the potential of the proposed approach as an ease of use decision tool in the context of a study case developed on a large soft drinks company that operates in the city of Monterrey, México. Embotelladoras ARCA (www.e-arca.com.mx) is a company dedicated to the production, distribution and sale of soft drinks brands owned by The Coca-Cola Company, some own-labels and third parties. ARCA was formed in 2001 by integrating three of the oldest bottlers in Mexico and become the second largest bottler of Coca-Cola products in Latin America and the fifth in the world. The company distributes its products in the north region of the Mexican Republic and since 2008, in the northeastern region of Argentina and in the entire republic of Ecuador. ARCA also produces and distributes branded salty snacks Bokados. Thus, the company has an enormous market that makes us think that it could be better achieved by taking into account an operation research model. The

* Corresponding author's email: fabian.lopez@e-arca.com.mx

company faces a Pickup and Delivery Problem with Time Windows and Dock Service Constraints (PDP-TWDS). On this work we are going to focus our application in the transportation operation on the north territory of México.

Our business application considers that the available vehicle fleet is represented on a node basis. In other words, at the beginning of the planning stage, each plant or distribution center provides the expected number of available trucks per type and at a specific starting time hour. This information defines the consolidated transportation capacity. Because of carrier requirements contract, we start and finish a route at the origin depot. Indeed, contract payment used in practice by the industry fix the transportation price on the basis that a route starts and finishes at the first pickup site. In our PDP context, each transportation order has a single time window. This is an earliest pickup time at origin and a latest delivery time at destination. From a practical standpoint for our business application, we have a set of transportation orders with an origin and a destination (O-D). Usually transport planners at the company attempt to determine first the route for each O-D pair, and later assign trucks to these predetermined routes. The problem of determining the best assignment of trucks to O-D routes is typically referred as an assignment problem where trucks are assigned to routes or transportation lanes such that all transportation orders are covered and transportation costs are minimized.

It is easy to verify that with each head haul move of the truck, goods are transported from its origin to its destination and revenue is generated. However, without goods, the truck moves an empty haul, in which only costs are incurred and no revenue is generated. Attempt to enforce a transportation order from a destination location back to its origin location results on an unsuccessful practice. This is because the truck will run an empty haul. These empty hauls represent a serious problem for transportation operations, as well as the country's economic system. This is clearly true because an empty haul does not generate any economic value. Thus, we can verify that the least efficient route that can be planned by a dispatcher is the one of simple trips where the vehicle travels loaded from the origin to the delivery site and then returns empty. On this case, half of the hauling distance is traveled empty. Moreover, we have another serious company issue. If a dispatcher tries to avoid simple trips, the actual structure of transportation flows that he is responsible for in the company, does not always permit it. It is clear in this situation, that pooling these transportation orders with those of another dispatcher may avoid simple trips by replacing the empty return of a simple trip with a transportation request of another dispatcher. Thus, the new structure of transportation flows generated by the collaboration of two or more dispatchers will allow important transportation cost-savings for the company. The empty part of the overall route is smaller when two trips are pooled together compared when making them independently.

It is estimated that at least 36% of truck movements in the company are empty haul moves. This means millions of kilometers of empty haul moves and also millions of liters of fuel lost per year. This is a major economic loss for the company, especially in the current situation where fuel prices have skyrocketed. On the country context, the Department of Land Transportation in México note that over 160,000 tons of pollution is released to the environment directly as a result of empty haul moves. Thus, empty hauls are a serious problem which needs immediate attention. Due to our PDP-TWDS is NP-Hard, combined with the fact that real world PDP's are very large, having hundreds of transportation requests to serve, there is no much hope for finding an optimal model that will work acceptably fast in practice. We propose a Hybrid Mixed Integer Programming (HMIP) approach to this complex problem which is focused on finding good solutions in reasonably short computational times. The paper is organized as follows. In Section 2 we introduce the problem definition and its associated complications. In Section 3 we briefly sketch some related problems and previous research work. In Section 4 we proceed to introduce some notation and present our model approach structured on three stages. Section 5 contains a description of some empirical results we found on our implementation. On section 6 we present some concluding remarks.

2. PROBLEM DEFINITION

As is defined, in a general PDP problem a set of routes must be generated in order to satisfy a set of transportation requests at a total minimum cost (or a similar objective function) and subject to a set of constraints. Each transportation request (i.e. a transportation order) specifies a volume of product, a site of origin and a destination site. Each request must be transported by only one vehicle. However we consider that some trans-shipments can occur across a route sequence from one node to the next. For all this operation, a previous defined fleet of vehicles is available. These vehicles are spread throughout a set of specific depot sites. This fleet of vehicles may consist of different vehicle-types, each with a unique set of transportation relevant characteristics. Indeed, in a PDP-TW problem, time windows constraints are usually added to the transportation requests. This is specifying a time interval for pickup and/or delivery operation at the origin or destination site.

The PDP is a generalization of the VRP, which is a generalization of the TSP, the well-known hard combinatorial optimization problem. Considering also that the problem in practice is, usually, of a large-scale, it is obvious why the problem is a challenge. The general pickup and delivery problem (GPDP) is a problem of finding a set of optimal routes, for a fleet of vehicles, in order to serve a set of transportation requests. Each vehicle from the fleet of vehicles has a given capacity, a start location, and an end location. Each transportation request is specified by a load to be transported, an origin, and a destination location. In other words, the pickup and delivery problem deals with the construction of optimal routes in order to visit all pickup and delivery locations and satisfy precedence and pairing constraints. From here we can move on to include some others considerations. That is, the problem deals with a number of transportation orders that are to be served by a fleet of vehicles while a number of constraints must be observed. Each vehicle has a limited capacity (the capacity constraint). Each vehicle starts and ends at a specified depot. A request must be picked up from a pickup location to be delivered to a corresponding delivery location. In addition, every request must be served within a predetermined time window (TW) interval (the time window constraint). A vehicle may serve multiple transportation orders as long as time windows and other capacity constraints are satisfied. A solution to the problem should assign requests to vehicles and find a route for each vehicle, such that the total service cost is minimized and all problem constraints (precedence, capacity, time windows and dock service) are adhered with. The total volume of product to deliver on some nodes may exceed the capacity of all types of truck. Thus a site within the same route could be visited more than once. In addition, in the classical PDP, when a delivery has been made, no pickup is allowed until the truck is empty. However in our problem's case, when a delivery has been made, we allow pickup even if the truck is not completely empty. This makes routing much more complex than classical PDP. The problem can be outlined in: (1) objective function and (2) operation constraints.

2.1 Objective Function

The goal of our model is to determine the optimum route for a multiple vehicles dedicated for a given physical distribution operation. A route is defined as the arrival sequence of a vehicle (i.e. single or double trailer) which has to attend to a set of nodes or warehouses waiting for service. This service can be defined as a delivery or pickup of any kind of item (i.e. product). In a typical operation we arrive to a node, make a delivery for product A and then afterwards pickup for product B that is required on another point that is ahead on the route sequence. On any case, the vehicle departs from an origin node (i.e. a distribution center) and then returns to the same node at the end of the route. An optimal route is obtained when we achieve the minimal cost (or distance or time) in order to attend all the customer nodes waiting for service.

2.2 Operation Constraints

1. We have a network with pickup and delivery locations. Let’s define N as the set of nodes i on the network (i.e. plants, distribution centers or customers) where $\forall i \in N$.
2. We have a set M of different vehicles, *where* $m \in M$, that are considered as the available fleet in order to perform the transportation process. Several origin nodes are defined on the network where vehicles start from. For each vehicle m one only origin node is defined. Let’s define $P(i)$ as the subset of vehicles located at node i where $P(i) \subseteq M$ and $\forall i \in N$.
3. At the start of the day, each vehicle leaves from the origin node. Then each vehicle attends to a set of geographically scattered nodes i (i.e. customers). At the end of the route, each vehicle returns to its origin point.
4. We have a set of transportation orders R to be executed from origin nodes (i.e. plants) to destination nodes (i.e. warehouses). Each order $r \in R$ consists of a pickup at some location i and a delivery at some other location j in the underlying transportation network. Precedence constraints must be considered which imply that a vehicle m should visit the pickup location i before the delivery location j of each transportation order r.
5. In addition, we have several types of products that are required to transport. Let’s define K as the set of different SKUS k including regular and returnable products, where $k \in K$. Thus, we define parameter D_{ijk} as the total planned demand to transport from node i to node j for each SKU k, where $(i,j,k) \in R$, $\forall i,j \in N$. Furthermore, we can aggregate the total demand to transport from node i to node j. Let’s define V as the subset of transportation lanes where some volume has to be delivered or picked up where $(i,j) \in V$, $\forall i,j \in N$.
6. Each vehicle has a finite load capacity. Vehicle Capacity is modeled as the quantity of boxes, pallets or weight that the vehicle can load taking in mind the space constraints as well. Vehicle capacity is defined at a SKU level in such a way we can cubic a capacity requirement to transport any given mixture load. A mixture load is any set of different volumes per SKU k to complete a full transportation order. Let’s define parameter H_k as the quantity of cases of SKU k that can be loaded per cubic meter $\forall k \in K$.
7. Each trailer has a loading & unloading access by the sides. Thus, this design is not affected by the nested precedence constraints we find on the general freight PDP in which loading and unloading access is restricted by the truck trailer rear door.
8. Each order $r \in R$ is a specific mix of products (i.e. different SKU’s) which has a weight and space requirement. Capacity constraints guarantee that any mixture load of items on a vehicle m should be less than the vehicle capacity. Let’s define parameter Q_m as the quantity of cubic meters on the vehicle m, where $m \in M$. According to the sequence of the route, all the time we must observe the load capacity of the vehicle m.

9. Certain compatibility constraints must be satisfied in real-world distribution operations because of physical and legal restrictions. For each vehicle *m* we have some nodes where the vehicle can operate for pickup or delivery operations. Thus, the use of a vehicle can be constrained at the transportation lane level. Let's define $A(i,j)$ as the subset of compatible vehicles *m* that can be used for transportation lane (i,j) where $A(i,j) \subseteq M$, $m \in M$, $(i,j) \in V$ and $i,j \in N$. In other words, a vehicle cannot arrive to nor departs from any node not included on that defined sub set.
10. The quantity of time (i.e. hours) required to accomplish the delivery and the pickup service in each node *i* depends mostly on the vehicle capacity. This consideration is true because vehicle capacity is close related to the volume of product that is delivered or pickup at any given node. Thus, we define the parameter *TS* as the service time for any single trailer configuration. By the other hand, we have parameter *TF* as the service time for any double trailer.
11. Each node has a particular time window for service. Due to any location (e.g., plant, warehouse, retail store, etc) has a specific working period, the pickup or delivery of a transportation order *r* at a location *i* can only take place during its working period. A time window is defined by an open & close time that should be considered for make a deliver or pickup on the node. Time windows constraints make sure that a service has to be given between the earliest arrival time and the latest arrival time. Let's define parameters $IN(i)$ and $CN(i)$ as the opening time and closing time at node *i* respectively where $i \in N$.
12. The same constraint about time windows applies at a vehicle level. This means that any given vehicle *m* cannot operate before its open window neither after its close window. Thus, a transportation order *r* is associated with a specific time interval within pickup or deliver operation must be done. The wide of the time window at each node *i* or vehicle *m* is equal to the difference between the closing and opening time for service. Each time window has different wide depending on the characteristics of the location (e.g., plant, warehouse, retail store, etc) or the vehicle as is corresponds. Let's define parameters IVm and CVm as the opening time and closing time for the vehicle *m* respectively where $m \in M$.
13. According to the sequence of the route, we will obtain arrivals and departures times for each vehicle across the locations on the network. However, we define for each location a specific quantity of docks available for service. Indeed, this capacity service at each location is not constant because is constrained depending on the hour of the day. Our approach to deal with this dock service capacity is to constraint the quantity of vehicles that can arrive at each node and at each hour of the day. As we can verify here, dock service capacity imposes new time windows constraints which emerge according the traffic of vehicles waiting for service at any location and at any hour. Let's define the parameter S_{ih} as the quantity of docks available for service at node *i* at working hour *h* where $i \in N$, $h \in \{1, ..., 24\}$.
14. Finally, we have a cost matrix and a time matrix that defines the cost and time required to go from each node to all others on a distribution network. Moreover, transportation cost and time for each transportation lane (i, j) depends on the type of vehicle. Let's define the following network parameters:
    ST_{ij} = transportation time on arc (i, j) for a single trailer $\forall (i, j) \in V$
    FT_{ij} = transportation time on arc (i, j) for a double trailer $\forall (i, j) \in V$
    SC_{ij} = transportation cost on arc (i, j) for a single trailer $\forall (i, j) \in V$
    FC_{ij} = transportation cost on arc (i, j) for a double trailer $\forall (i, j) \in V$

3. PREVIOUS RELATED WORK

Time constrained sequencing and routing problems arise in many practical applications. Typically, computational difficult for those type of problems has been measured in terms of its size. However the difficult for PDP-TWDS depends strongly on the structure of the time windows that are defined around the nodes and vehicles as well. Indeed, multiple vehicles environment generates some dock service capacity constraints. Both the PDP and PDP-TW are generalizations of the classical Vehicle Routing Problem (VRP) and are thus NP-hard. As a result, the development of solution methods for these problems has focused on heuristics J.-F. Cordeau, G. Laporte, and M.W.P. Savelsbergh (2006).

There are well known and extensively studied routing problems which are special cases of the General-PDP. The Dial a Ride Problem (DARP) is a routing problem in which the loads to be transported represent people. Therefore, we usually speak of clients or customers instead of transportation requests and all load sizes are equal to one. The Vehicle Routing Problem (VRP) is a routing problem in which either all the origins or all the destinations are located at the same depot. The research of time constrained pickup and delivery problems emerged in the last 15-20 years. Researchers have developed a variety of heuristics and optimization methods. The development of optimization methods started in the early 1980s and lasted almost a decade. Heuristics for solving real-life pickup and delivery problems began to appear in the literature in the 1970s. The majority of published work on General-PDP is on dial-a-ride problems (DARP). In contrast to this, very little work has been done on pickup and delivery of packages and goods with time windows constraints (PDP-TW).

In regard to routing applications, we found that the variant with less research work corresponds to physical product distribution (Mitrovic 1998). We have the basic model named Traveling Salesman Problem with Time Windows constraints (TSP-TW). Christofides et al. (1976) describe a branch-and-bound algorithm in which the lower bound computation is performed via a state space relaxation in a dynamic programming scheme. Problem instances were solved up to 50 nodes with "moderately tight" time windows. Dumas et al. (1995) present a dynamic programming algorithm for the TSP-TW. They were able to solve problems of up to 200 nodes with "fairly wide" time windows. We refer now about the work presented by Ascheuer et al. (2001) for the TSP-TW. They tested instances up to 233 nodes. For an instance of 69 nodes was required 5.95 minutes of computational time. In general, all larger instances required more than 5 hours of computational time to converge in a feasible solution. The experimental results with the TSP-TW made by Ascheuer et al. proved that this problem is particularly difficult to resolve for instances with more than 50% of active nodes with time window constraints.

We move our research now from the typical TSP-TW to a more sophisticated problem named as Vehicle Routing Problem (VRP). The most widely studied extensions of the VRP are the capacitated vehicle routing problem (C-VRP) and the vehicle routing problem with time windows (VRP-TW). The basic model C-VRP assumes that all the vehicles are homogeneous with the same capacity and located initially at the same node (i.e. depot) and customers have no specific service time windows (i.e. can be covered at any time). A more complex model is the VRP-TW. On VRP-TW customers have time windows within which they must be covered. Solomon (1984) developed 87 test instances for the VRP-TW. Indeed, the largest instance he solved was about 100 nodes. Until year 1999 there were 17 instances that still remained without being solved. In that year in Rice University, were solved 10 of these instances (Cook & Rich 1999). VRP with multiple pickup and delivery locations have been studied by Savelsbergh (1998) and Hasle (2003).

The most general model is the Pickup and Delivery problem with Time Windows Constraints (PDP-TW). PDP-TW is more difficult to solve than VRP–TW. This is true because, the first problem is a generalization of the second (Palmgren 2001). According with Savelsbergh (1995), we have a variant for one alone vehicle (SPDP-TW) and one another for multiple vehicles (MPDP-TW). The first case is considered a restrictive TSP-TW while the second variant is considered a restrictive VRP-TW. The PDP-TW is NP-hard since the VRP and PDP is NP-hard (Desrosiers, Dumas, Solomon, & Soumis, 1995). Indeed, it is strongly NP-complete to find a feasible solution for the PDP. Furthermore, Tsitsiklis (1992) showed that even the basic TSP-TW is strongly NP-complete. Our PDP-TWDS is less studied than the classical vehicle routing problems. Indeed, this problem is a generalization of the vehicle routing problem (VRP) and the pickup and delivery problem (PDP). The problem involves a set of practical features that are commonly seen in practice but have received little attention in the vehicle routing literature. Some complex features involved in the PDP-TWDS such as dock service capacity and compatibility constraints, have not been addressed in the vehicle routing literature. For PDP-TWDS extension we just add some constraints on dock capacity service at each node and at each hour of the day. Therefore, the PDP-TWDS is more general and more complex to solve than any existing VRP-TW or a single PDP model. Furthermore, no existing model has incorporated dock service capacity constraints explicitly.

The first optimization algorithm for the PDP-TW was a branch-and-price algorithm presented by Dumas, Desrosiers, & Soumis (1991). A column generation approach was proposed. Indeed, a set partitioning formulation is solved by a branch-and-price method in which columns of negative reduced cost are generated by a dynamic programming algorithm. The method has been successful in solving instances with tight capacity constraints and a small number of requests per route. They show that this approach is capable of solving some instances with up to 22 vehicles and 190 requests. Savelsbergh & Sol (1995) presented an integer programming formulation of the general pickup and delivery problem (GPDP) which considered several pickup and delivery locations of a transportation orders. Savelsbergh and Sol (1998) proposed a branch-and-price algorithm for the PDP-TW using both a heuristic algorithm and a dynamic programming algorithm for the column generation problem. They applied a new branching scheme based on assignment rather than routing decisions. In the past two decades, a tremendous amount of research results on these models have been published. Recent books and survey papers include, among others, Laporte (1992), Desrosiers et al. (1995), Fisher (1995), Savelsbergh and Sol (1995), Powell et al. (1995), Bramel and Simchi-Levi (1997), and Crainic and Laporte (1998).

Cordeau et al. (2003), developed a branch-and-cut algorithm for the DARP, based on a three-index formulation with a polynomial number of constraints. It uses several families of valid inequalities that are either adaptations of existing inequalities for the TSP and the VRP. However, direct implementation of methods for solving DARP is not a solution for GPDP. The GPDP is mostly capacited and the time windows are wider. These differences seem to imply that the set of feasible solutions is larger in GPDP than in the problems where people are transported. More recently, a branch-and-cut algorithm for the capacitated multiple-vehicle PDP and PDP-TW was later described by Lu and Dessouky (2006). Their formulation contains a polynomial number of constraints and uses two-index flow variables, but relies on extra variables to impose pairing and precedence constraints. Instances with up to 5 vehicles and 25 requests were solved optimally with this approach. By using appropriate inequalities, Ropke et al. (2006) introduced a new formulation for the PDP-TW which do not require the use of a vehicle index to impose pairing and precedence constraints. They report computational experiments on several sets of test instances and show that this approach is capable of solving some instances with up to 8 vehicles and 96 requests. In general, the best results found on literature are obtained by column generation methods. Instances of up to 880 requests and 53 vehicles can be solved with this method.

Many solution methods have appeared for vehicle routing problems. In general, heuristics can solve problems with larger scales in less computation times than optimization methods. For example, the recent progress in meta-heuristics such as Tabu Search, simulated annealing, and genetic algorithms (Gendreau et al., 1997; Golden et al., 1998) can solve vehicle routing problems with wide time windows with nearly 500 transportation requests. However, as pointed out by Fisher (1995), heuristics usually lack robustness and their performance is very much problem dependent. Fisher states that "It's not uncommon that a heuristic developed for a particular geographic region of a company's operation will perform poorly in another region served by the same company."

It is not easy to compare different approaches to the PDP-TW. Moreover, in most of the cases authors only use randomly generated data. It is not clear what their findings mean for "real-world instances" which is actually our case. The existing vehicle routing models are useful for various practical applications. However, many important practical issues have not been addressed in these models, as pointed out by Fisher (1995), "Real vehicle routing problems usually include complications beyond the basic model....". Given the enormous complexity of the PDP problems, it is not realistic to apply pure optimization methods. Instead, we focus on a strategy that can not only be as robust as optimization methods but also are capable of finding good solutions within acceptable computation time. Thus, we develop hybrid approach to integrate fast heuristics into an optimization framework of a cut generation method (e.g., Barnhart et al., 1998; Wolsey, 1998).

4. PROPOSED MODEL

A very important characteristic of routing problems is the way in which transportation requests become available. In a static situation all requests are known at the time the routes have to be constructed. In a dynamic situation some of the requests are known at the time the routes have to be constructed and the other requests become available in real time during execution of the routes. We focus on the static stage. An optimization method may benefit from the presence of time constraints since the solution space may be much smaller. To prevent transportation requests from being served long before (or after) their desired delivery (or pickup) time, we can either construct closed time windows or take an objective function that penalizes deviations from the desired service time. In general, we can figure out two kind of objective functions related to multiple vehicle pickup and delivery problems:

- Minimize the total time, distance or cost that all vehicles need to execute all the set of transportation requests.
- Minimize the number of vehicles. This function is almost always used. Because drivers and vehicles are the most expensive parts in a system, minimizing the number of vehicles to serve all requests is usually the main objective.

Our PDP model is focused on a continuous move strategy implementation. On this strategy attempts are made to match multiple truckload pickups and deliveries to one truck in sequential order such that the prior delivery is made before the next pickup in the sequence. The benefit of continuous moves derives from the overall reduction in empty haul distances. Careful planning can ensure that the relocation of a truck from the prior delivery location to the next pickup location will minimize the overall empty haul distances for the entire network. So, we focus our attention on finding optimal routes for the continuous move problem, using a large-scale mathematical model. A continuous move (i.e. c-move) trip occurs when two or more truckload trips are sequentially combined. That is, if trips $T_{i1,j1}$ and $T_{i2,j2}$ are combined, then a c-move trip will require as follows:

- Deliver goods from origin *i1* to destination *j1*.
- Make an empty haul move to a new origin *i2*.
- Pick up goods from origin *i2* and deliver them to a final destination *j2* and
- Return to the initial origin *i1*.

For each trip, we compute its total cost, which includes the summation of all costs including those associated with the empty hauls. Some assumptions are considered in our case. We consider only a daily operation. All trips are planned for one day of operation in order to enforce and simplify truck location requirements. In other words, all trucks starts the day at an origin *i* and then return to the same origin at the end of the day. Another assumption excludes stochastic and dynamic considerations. This is justifiable as the model that we propose is meant as a planning tool, not as an operational tool. We have chosen a hybrid MIP approach to solve our problem. The PDP-TWDS is formulated as a mixed integer linear program. We propose to solve the problem in three stages. The first stage constructs initials solutions at aggregate level relaxing some constraints on the original problem. The other two stages imposes time windows and dock service constraints respectively.

4.1 Relaxed Capacitated Vehicle Routing Problem (C-VRP) Model

Here we assume different vehicles capacities that are initially located at different nodes (i.e. depots). At this first stage our model constructs initials solutions at aggregate level. Particularly, we relax time windows and dock service constraints. This means that transportation orders have no specific service time windows constraints to satisfy. The objective is to find an optimal cost solution that completes all the transportation workload orders at aggregate level taking in mind vehicle cubic capacity constraints, vehicle compatibility constraints and 24-hours of operation per vehicle per day constraints. The main output of this relaxed C-VRP model is to identify an optimal assignment of the vehicles to cover all the transportation orders. In transportation operation, the regular case is when we operate a single trailer with just one haul. However, our first C-VRP model considers the case when we decide to operate a route with a vehicle *m1* grouped with another vehicle *m2*. As a result we obtain one new vehicle with a summed capacity. This is a double trailer case, in other words, a vehicle operating with two hauls. Thus, our C-VRP model includes identifying if one vehicle *m1* should be grouped (hooked) to operate a route with another vehicle *m2*. We present our first stage C-VRP model as follows:

<u>Sets and parameters:</u>

$A(i,j)$ = subset of compatible vehicles m that can be used on transportation lane (i,j) $\forall$ $(i,j) \in V$, where $A(i,j) \subseteq M$, $m \in M$
H_k = quantity of cases of SKU k per cubic meter, $\forall$ $k \in K$
K = set of different SKUS k including regular and returnable products, where $k \in K$
M = set of vehicles (trailers), where $m \in M$
N = set of nodes on the network (i.e. plants, distribution centers or customers), where $i \in N$
$P(i)$ = subset of vehicles located at node i, where $P(i) \subseteq M$ and $i \in N$
Q_m = quantity of cubic meters on vehicle m, where $m \in M$
R = set of transportation orders to satisfy of regular or returnable products from node i to node j, where $r \in$ R
TS = service time for single trailer configuration
TF = service time for double trailer configuration
ST_{ij} = transportation time on arc (i, j) on single trailer, $\forall$ $(i,j) \in V$ and $i, j \in N$
FT_{ij} = transportation time on arc (i, j) on double trailer, $\forall$ $(i,j) \in V$ and $i, j \in N$
SC_{ij} = transportation cost on arc (i, j) on single trailer, $\forall$ $(i,j) \in V$ and $i, j \in N$
FC_{ij} = transportation cost on arc (i, j) on double trailer, $\forall$ $(i,j) \in V$ and $i, j \in N$
D_{ijk} = planned demand to transport from node i to node j for SKU k, $\forall$ $(i,j) \in V$ and $(i, j, k) \in$ R
$IN(i)$ = opening time at node i, $\forall$ $i \in N$
$CN(i)$ = closing time at node i, $\forall$ $i \in N$
IV_m = opening time of vehicle m, $\forall$ $m \in M$
CV_m = closing time of vehicle m, $\forall$ $m \in M$
UB = number of times for demand covering (upper bound). Volume of product covered in advance or excess.

<u>Decision variables:</u>

$W_{m1,m2} \Rightarrow$ = 1 if vehicle $m1$ is linked to vehicle $m2$; =0 otherwise, $\forall$ $(m1, m2) \in P(i)$
$X_{ij}^{m1,m2} \geq 0$, integer $\Rightarrow$ number of trips from node i to j using vehicle $(m1, m2)$, $\forall$ $(i,j) \in V$, $(m1, m2) \in A(i,j) \subset P(i)$
$F_{ijk} \geq 0 \Rightarrow$ quantity of cases to transport from node i to j of SKU k, $\forall$ $(i, j, k) \in$ R

The C-VRP can be formulated as the following mixed integer model:

$$(CVRP\ Relaxed)\ \ Minimize \sum_{i \in N} \sum_{j \in N} \left[\sum_{(m1=m2) \in M} X_{ij}^{m1,m2} \cdot SC_{ij} + \sum_{(m1 \neq m2) \in M} X_{ij}^{m1,m2} \cdot FC_{ij} \right] \tag{1.1}$$

$$(Alternatively)\ \ Minimize \sum_{m1 \in M} \sum_{m2 \in M, m1 \leq m2} W_{m1,m2} \tag{1.2}$$

Subject to:

$$\sum_{m2 \in M} W_{m1,m2} \leq 1, \quad \forall\, m1 \in M \tag{1.3}$$

$$\sum_{w\in M,\ w\neq m} (W_{w,m} + W_{m,w}) \le 1, \quad \forall\, m \in M \tag{1.4}$$

$$\sum_{(i,j)\in V} (\mathrm{TS} + ST_{ij}) \cdot X_{ij}^{m1,m2} \le 24 \cdot W_{m1,m2}, \quad \forall\, (m1 = m2) \in M, \qquad where\ (m1,m2) \in A(i,j) \subset P(i) \tag{1.5}$$

$$\sum_{(i,j)\in V} (\mathrm{TF} + FT_{ij}) \cdot X_{ij}^{m1,m2} \le 24 \cdot W_{m1,m2}, \quad \forall\, (m1 \neq m2) \in M, \qquad where\ (m1,m2) \in A(i,j) \subset P(i) \tag{1.6}$$

$$\sum_{k\in K} \frac{F_{ijk}}{H_k} = \sum_{\forall\,(m1=m2)\in M} X_{ij}^{m1,m2} \cdot Q_{m1} + \sum_{\forall\,(m1\neq m2)\in M} X_{ij}^{m1,m2} \cdot (Q_{m1} + Q_{m2}), \quad \forall\, (i,j) \in V \tag{1.7}$$

$$\sum_{i\in N} F_{ijk} - \sum_{h\in N} F_{jhk} \ge \sum_{i\in N} D_{ijk}, \quad \forall\, \mathrm{j} \in \mathrm{N}, \qquad \mathrm{k} \in \mathrm{K}, \qquad \text{where } (i,j,k) \in R \tag{1.8}$$

$$\sum_{i\in N} F_{ijk} - \sum_{h\in N} F_{jhk} \le \mathrm{UB} \sum_{i\in N} D_{ijk}, \quad \forall\, \mathrm{j} \in \mathrm{N}, \qquad \mathrm{k} \in \mathrm{K}, \qquad \text{where } (i,j,k) \in R \tag{1.9}$$

$$\sum_{i\in N} X_{ij}^{m1,m2} = \sum_{i\in N} X_{ji}^{m1,m2}, \quad \forall\, j \in N, \qquad \forall\, m1, m2 \in M, \qquad where\ (m1,m2) \in A(i,j) \subset P(i), \quad (i,j) \in V \tag{1.10}$$

Objective function (1.1) is formulated to minimize the variable cost (i.e. distance) of vehicles that are needed to execute the set of transportation requests. This is taking in consideration transportation cost on single and double trailer operation. Alternatively we have another objective function (1.2) which is formulated to minimize the total number of vehicles required to execute the set of transportation orders. Constraints (1.3 - 1.4) assure that each vehicle can be assigned exclusively to a single or a double trailer operation only. Constraints (1.5) restrict the maximum quantity of trips that a single trailer can perform on a 24 hours time horizon. Something similar applies on constraints (1.6) for a double trailer operation. Constraints (1.7) assure that the quantity of cubic meters used to transport SKU products from node *i* to node *j* is equal to the total cubic meters of available capacity considering single and double trailer operation. Constraints (1.8) correspond to balance flow constraints that assures that total transportation volume from node *i* to node *j* is sufficient to cover the total demand at each SKU level as is required. Constraint (1.9) is similar to (1.8) but this is used to restrict the maximum volume of product to transport from node *i* to node *j* as an upper bound. Finally, (1.10) corresponds to the balance flow constraints imposed at vehicle level.

4.2 Pickup and Delivery Problem with Time Window Constraints (PDP-TW) Model

As a result from the previous model we obtain the optimal assignment of the vehicles. That is, binary variable $W_{m1,m2}$ identify which vehicles is going to operate a single trailer (i.e. with just one haul) and which others will operate on double trailer (i.e. a vehicle operating with two hauls). From here to the end, all double trailers will be modeled as one only vehicle with a summed capacity. Indeed, we can verify on the previous model that integer variable $X_{ij}^{m1,m2}$ calculates the optimal quantity of trips required on each final vehicle and on each arc between origin nodes and destination nodes. Our next PDP-TW model is implemented in order to take advantage from the previous information. Thus, on this model we add time windows constraints. We model as follows:

<u>Added sets and parameters:</u>
L = set of stops on a given route
X_{ij}^{m} = number of trips from node *i* to *j* using vehicle *m*, $\forall\ (i,j) \in V$, $m \in A(i,j)$
IN(i) = opening time at node *i*, $\forall\ i \in N$
CN(i) = closing time at node *i*, $\forall\ i \in N$
IV_m = opening time of vehicle *m*, $\forall\ m \in M$
CV_m = closing time of vehicle *m*, $\forall\ m \in M$
TC_{ij}^{m} = transportation cost for transportation lane (*i*,*j*) on vehicle *m*, $\forall$ (i,j) $\in$ V, $m \in A(i,j)$
Z_{ij}^{m} = total transportation and service time for transportation lane (*i*,*j*) on vehicle *m*, $\forall\ (i,j) \in V$, $m \in A(i,j)$

<u>Decision variables:</u>

Y_{ij}^{ml} ⇒ = 1 if vehicle *m* is routed from node *i* to *j* on sequence *l*; = 0 otherwise. ∀ $(i,j) \in V$, $m \in A(i,j)$, $l \in L$
$T_{ij}^{ml} \geq 0$ ⇒ arrival time at node *j* from node *i* on vehicle *m* at sequence *l*, ∀ $(i,j) \in V$, $m \in A(i,j)$, $l \in L$

The PDP-TW can be formulated as the following mixed integer model:

$$(PDP.TW)\ \ Minimize \sum_{i\in N}\sum_{j\in N}\sum_{m\in M}\sum_{l\in L}\left[Y_{ij}^{ml}\cdot TC_{ij}^{m}+T_{ij}^{ml}\right] \tag{2.1}$$

Subject to:

$$\textstyle\sum_{l\in L} Y_{ij}^{ml} = X_{ij}^{m}, \quad \forall\,(i,j)\in V, m\in M\in A(i,j) \tag{2.2}$$

$$\sum_{i\in N}\sum_{l\in L} Y_{ij}^{ml} = \sum_{i\in N}\sum_{l\in L} Y_{ji}^{ml}\,, \quad \forall\, j\in N, m\in M\in A(i,j), \qquad where\ \ (i,j)\in V \tag{2.3}$$

$$T_{ij}^{ml} \geq IN_i\cdot Y_{ij}^{ml}, \qquad \forall\, i\in N, (i,j)\in V, m\in M\in A(i,j), l\in L \tag{2.4}$$

$$T_{ij}^{ml} \leq CN_i\cdot Y_{ij}^{ml}, \qquad \forall\ i\in N, (i,j)\in V, m\in M\in A(i,j), l\in L \tag{2.5}$$

$$T_{ij}^{ml} \geq IV_m\cdot Y_{ij}^{ml}, \qquad \forall\, i\in N, (i,j)\in V, m\in M\in A(i,j), l\in L \tag{2.6}$$

$$T_{ij}^{ml} \leq CV_m\cdot Y_{ij}^{ml}, \qquad \forall\ i\in N, (i,j)\in V, m\in M\in A(i,j), l\in L \tag{2.7}$$

$$\sum_{i\in N} Y_{ij}^{ml} \leq 1, \qquad \forall\ j\in N, m\in M\in A(i,j), l\in L, \qquad where\ (i,j)\in V \tag{2.8}$$

$$\sum_{i\neq j\in N, m\notin P(i)} T_{ij}^{ml} + \sum_{\substack{i\neq j\in N\\ \notin P(j)}} Z_{ij}^{m}\cdot Y_{ij}^{ml} \leq \sum_{h\neq j\in N, m\notin P(h)} T_{jh}^{ml} + \sum_{h\neq j\in N, m\in P(h)} T_{jh}^{m,l+1} \quad \forall\, j\in N, m\in M, l=1, m \tag{2.9}$$

$$\sum_{i\neq j\in N} T_{ij}^{ml} + \sum_{\substack{i\neq j\in N\\ \in P(j)}} Z_{ij}^{m}\cdot Y_{ij}^{ml} \leq \sum_{h\neq j\in N, m\notin P(h)} T_{jh}^{ml} + \sum_{h\neq j\in N, m\in P(h)} T_{jh}^{m,\{l+1,1\}\in L} \quad \forall\, j\in N, l\neq 1\in L, m \tag{2.10}$$

Expression (2.1) is formulated as a multi-term objective function. The first part of the function is used to minimize the total transportation variable cost of the vehicles required to execute all the set of transportation orders. The second part minimizes all the set of arrival times that corresponds for each individual trip *(i,j,m)*. With this formulation we calculate the earliest arrival time for each trip. Constraints (2.2) assure that all the set of trips obtained on model 1 are fully covered on model 2. These constraints are imposed for each vehicle and for each pair of origin and destination nodes. Constraints (2.3) correspond to the balance flow constraints imposed at vehicle level. Constraints (2.4 - 2.5) are formulated for time windows constraints required on each node. Constraints (2.6 - 2.7) correspond to time windows formulation for each vehicle. Constraints (2.8) assures that each vehicle must depart from just one only origin node at each trip. Constraints (2.9 - 2.10) are formulated in order to calculate the arrival times for the entire set of trips considering all the nodes and all the vehicles. In other words, these constraints correspond to the time windows precedence for each trip and for each vehicle.

4.3 Pickup and Delivery Problem with TW and Dock Service Constraints (PDP-TWDS) Model

As a result from the previous PDP-TW model we obtain the optimal assignment of the vehicles considering vehicles capacity and time windows constraints as well. That is, binary variable Y_{ij}^{ml} identify if a vehicle m is routed from node i to j on sequence l. This is the route sequence for each vehicle. At the same time, positive variable T_{ij}^{ml}, calculates the arrivals time at each node for all the vehicles. With this in mind, we can proceed now to apply dock service capacity constraints on our final model. Our previous model works as the master model. Then, the logic we apply here is to iteratively generate cuts in a Brach & Cut scheme. For that purpose we identify in the incumbent solution, at each arrival node and at each working hour, the subset of vehicles that are violating the dock service constraint. For that purpose we compare the quantity of vehicles that are being dispatched simultaneously at a given node and at a given hour versus the docks quantity that the node is capable to attend at a given hour. Then we add these cuts to the master model. The generated cuts are kept in a pool of constraints that are managed separately of the rest of the cuts generated automatically by the B&C scheme. The procedure continues until is found the first optimal solution for the problem that does not violate the dock service capacity on all nodes and at each 24-hour planning day. We model as follows:

Added sets and parameters:
S_{jh} = number of docks available for service at node *j* at working hour *h*, where $j \in N$, $h \in \{1, ..., 24\}$
E = set of cases where vehicle *a* is violating the dock service constraint at node *j* at hour *h*
OT = minimal offset time between arrivals of vehicles *a* and *β* at node *j*

$$e(j\alpha, j\beta, h) \in E \;\rightarrow\; \textit{if and only if} \begin{cases} \left| T_{ij}^{\alpha,l} - T_{ij}^{\beta,l} \right| < OT \quad and \\ \#\ of\ vehicles\ arriving\ at\ node\ j\ at\ hour\ h\ >\ S_{jh} \\ where\ \{\alpha \dots \beta\} \in M \end{cases}$$

Decision variables for dock service constraint at node *j* at hour *h* ($e \in E$):
$B^+_e \geq 0 \Rightarrow$ Case e → time difference between arrival of vehicle *a* and arrival of vehicle *β* to node *j* at hour *h*: $e(j\alpha, j\beta, h) \in E$
$B^-_e \geq 0 \Rightarrow$ Case e → time difference between arrival of vehicle *β* and arrival of vehicle *a* to node *j* at hour *h*: $e(j\alpha, j\beta, h) \in E$
$U_e \Rightarrow$ Case e → = 1, if vehicle *a* arrives before vehicle *β* to node *j* at hour *h*; = 0 otherwise, where $e \in E$

Subject to:

$$T_{ij}^{\alpha,l} - T_{ij}^{\beta,l} = B_e^{\pm B_e^-}, \quad \forall\ e(j\alpha, j\beta, h) \in E, \quad where\ i,j \in N, (i,j) \in V, \quad \alpha, \beta \in M \in A(i,j), l \in L \quad (3.1)$$

$$B_e^+ + B_e^- \geq OT, \quad \forall\ e \in E \quad (3.2)$$

$$B_e^+ \leq 24 \cdot U_e, \quad \forall\ e \in E \quad (3.3)$$

$$B_e^- \leq 24 \cdot (1 - U_e), \quad \forall\ e \in E \quad (3.4)$$

Constraints (3.1) deal with a set of deviational variables to calculate the offset time on the arrival times to node *j* for each combination of vehicles α and β. We impose these constraints for any combination of vehicles α and β that may be arriving at a given node and at the same time. As a consequence it might be exceeding dock service capacity at the location and thus some constraints would be required. Constraints (3.2) assure that the offset time for any given pair of vehicles α and β arriving at node *j* looking for dock service capacity must be at least of size OT (e.g. one hour). Constraints (3.3 – 3.4) correspond to upper bounds imposed for deviational variables. In our case we define 24 hours as the time frame horizon. As can be verified on the previous model, these constraints grow exponentially as the number of nodes and vehicles are large. Thus on the last model we add these constraints on an iterative scheme only as is required. We model a linear relaxation of the PDP-TW problem resulting in a master problem solved very efficiently by a MIP solver. On this stage we fully apply the time windows constraints but we relax the dock service capacity constraints. Thus, at each iteration an integer feasible solution is obtained for time windows constraints on all nodes and all vehicles. An iteration procedure is performed within the MIP solver framework to add dock capacity constraints as necessary. We found that our approach is capable of obtaining near-optimal solutions in acceptable computational times for real business instances around 160 vehicles and 500 transportation orders.

5. COMPUTATIONAL RESULTS

We present some results indicating the efficiency of our method for solving large scale instances. CPU configuration used in our implementation is Win X32, 2 Intel Cores at 1.4GHz. We implement our model on X-PRESS MIP Solver from FICOTM (i.e. Fair Isaac, formerly Dash Optimization). The first stage of our solution method corresponds to a Relaxed Capacitated Vehicle Routing Problem (C-VRP) model. On this stage we relax all the time window and dock service constraints. Instead, some side constraints at aggregate level are included in order to assure feasibility on the original problem. This heuristic stage works at aggregate level to create a network simplification for the original PDP-TWDS in order to reduce the search space. The basic idea of our heuristic is to identify a sub set of incumbent decision variables in such a way that transportation orders and vehicles are required to be compatible. However, the idea to include only a small percentage of the possible links as decision variables in the first stage MIP model is based not only on a geographic distance criterion but also on other compatibility issues. That is, we point out about compatibility between transportation orders and vehicles types, compatibility between transportation lanes and vehicles types and compatibility between transportation nodes and vehicles types. Thus, this heuristic is very important to estimate the number of links (i.e. decision variables) necessary for each transportation lane and for each vehicle in order to assure feasibility for the original disaggregated problem. Thus, one of the main contributions of our work is to develop a model that is stable on input data in order to find a way to dismiss enough links to make the solution of the first aggregated MIP model very efficient. With this in mind, the first stage model can be solved more efficiently. This strategy reduces the search space because it decreases the number of possible solutions for transportation orders assignment to vehicles. This trade-off on optimality is going to be detailed on the next section.

On the following, table 1 shows the optimal solutions that we find for our first stage model using different combinations on input values for parameters: (*F1*) quantity of vehicles to be considered for each transportation lane and (*F2*) quantity of transportation lanes to be considered for each vehicle. Basically these two heuristic parameters *F1* and *F2* affect the matrix size of the decision variables and the complexity to be considered by our model at the first stage when feasible solutions are obtained at aggregate level. Particularly, table 1 shows the results obtained with an objective function implemented with equation (1.1). This objective function minimizes just the total transportation cost which is in fact a variable cost. As can be verified on the last column, the number of tractors or single hauls is not actually minimized when compared with table 2.

From table 1 we can verify that with an appropriate setting on parameters *F1* and *F2*, we can obtain good quality solutions in short computational times. However, the trade off we have to pay with this strategy is that we may have an over constrained solution space. By the other hand, when we set *F1* = 40 and *F2* = 40, our problem size is larger (see No. of binary variables). Thus, we can obtain better solutions but more time is required to solve the problem. Column "% of Gap to optimality" corresponds to the gap optimality expressed as a percentage when we compare the best MIP solution found on column # 6 versus the best bound obtained at any given computational time by our solver. Now on table 2, we present the results obtained with an objective function implemented with equation (1.2). This objective function minimizes the total number of tractors and single hauls (i.e. vehicles) that are required to cover the entire set of transportation orders.

As we can verify on both tables 1 and 2, as more time is available we can improve our solutions. However as is expected, on table 2 we obtain better solutions on the number of vehicles than we have on table 1. Our empirical results show that this heuristic has no impact on the optimal solution for the original problem we will find afterwards on the next stage. From our business perspective, we believe that it makes sense to use the objective function (1.2) reported on table 2 instead of (1.1) because it reflects more closely the transportation cost. That is, the fixed operative cost is very depending on the number of trailers and trucks that are required to attend the set of transportation orders. Accordingly with the contract, the bottler company has to pay to the third party provider for the rental of each truck running at operation independently of the number of trips that each truck performs during the workday. Taking in mind this consideration, the best solution we find on table 1 and 2 for different values on parameters *F1* and *F2* corresponds to the solution with 39 tractors only (71 single-hauls). Thus, with this solution obtained so far at aggregate level, we have an optimization around 34% when compared with the actual number of single-hauls in use and 27% when compared with the actual number of trucks on rental. Our challenge is to assure this optimization on the next model stage when time windows and dock capacity constraints are included.

From this aggregated and relaxed solution we move to process the model for the next two stages. Indeed, it is important to consider that these two stages are actually implemented in just one single model. That is, the 2nd model is the master model and the 3rd model runs iteratively adding the cuts to consider dock service constraints only as necessary. Thus the 3rd model iteratively runs until we find a solution that fully satisfies all the dock service constraints. All the computational experiments we perform from here are done taking in consideration a value of 1% for our solver MIP optimality tolerance. This optimality tolerance is set in the solver engine in order to identify a true near-optimal solution for each instance tested. This optimality tolerance operates at each iteration within the cut generation strategy on the 3rd model.

In order to effectively stress our model, we generate several instances using different values for dock service capacity that is available on each transportation node. Thus, as we have an instance with less available docks for service we generate a more difficult problem to solve and a larger computational time is expected to find a solution. We have 34 physical different transportation nodes in total in our problem. Between each pair of transportation nodes (i.e. origin and destination) we may have several transportation orders to be attended with different time windows requirements. Thus, the number of transportation nodes that are considered at the model level is much larger. By the other hand, the number of docks required for service at each node depends mainly on the volume of transportation orders to attend inbound or outbound operations at each node. In our instances, the number of docks available for service at each node ranges from one only up to eight. On table 3 as follows we present the instances and some results we obtain. For description purposes we detail each different instance indicating the number of docks available for service on each node. That is, on the first 10 columns of the table we have the number of docks on each node. For example label "5..6" indicates the same given capacity on both nodes. "Total Docks" is the total number on docks on the entire instance considering all the transportation nodes. The next columns are "# ITERs" and "# CUTs". The first is the number of iterations and the second the number of constraints that are required on each instance by the MIP solver to converge on a feasible near-optimal solution. Next we have the solution time on seconds that is required for each instance. Column "Max.Docks" refers to the maximum number of transportation orders on any given iteration where dock service capacity is violated. Indeed, this maximum number is mostly reached on the first iteration(s). Column "Avg.Docks" is the average number of transportation orders where dock capacity is violated. Furthermore, this average is weighted by the length of computational time that the solver spends on each iteration to solve the incumbent dock constraints. The last column is a speed measure that indicates the quantity of dock constraints that are solved by unit of time (i.e. by second).

We can verify on table 3 that in general, as we have a larger quantity of docks available for service, shorter is the number of iterations and constraints that are required to add on the cut generation stage. As a consequence of less iterations and constraints to add on the master problem, this means less solution time to solve the problem. Furthermore, a larger number on the transportation orders where dock service is violated (i.e. 15^{th} column) means a larger quantity of cuts required to solve the problem. Something different happens with the "Avg.Docks" indicator. This measure does not depend much on the number of available docks nor is correlated to the other indicators. In fact, this measure ranges from 2.4 up to 10.4 and 5.1 in average. For example 2.4 means that, during the computational time when the progress take place, the most of the time the solver was trying to solve 2.4 constraints in average where the dock capacity is violated. In other words, the most part of the time required to converge on a fully feasible solution, we have a cuasi-feasible solution with only 2.4 transportation orders that does not have any dock capacity available for service. This is just like the solver's average backlog. Finally, for the efficiency indicator on the last column, as we have a more constrained instance (i.e. with less available docks), the quantity of dock constraints solved by second is reduced. On the following we present table 4 showing the same set of instances as on table 3 accordingly with the total docks available for service. On this table we focus on present some activity measures for the vehicles operation.

The third column on table 4 corresponds to the second part of the objective function presented on equation (2.1). It is the total sum of the arrival times of all the vehicles used to attend the entire set of transportation orders. This indicator is very useful in order to estimate how much efficiency and time delays we have on the vehicles. As we have a more constrained instance (i.e. with less available docks), we have larger waiting times on the vehicles. The waiting time can occur on the origin or on the destination node. Either way, this delay on the vehicle has a negative impact on its efficiency and also on the finish time when each vehicle completes its route at the end of the working day. The 4^{th} column on table 4 corresponds to the sum of all the finish times of the vehicles on each instance. Thus, if we divide the sum of all the vehicles finish times between the total of vehicles we obtain in column 5 the average finish time of the entire fleet. In according with the last idea, we have on the next column the number of vehicles that are running on or after the 22^{nd} hour. From the bottler's operation perspective, there are several reasons they would like to avoid these times on the vehicles. It is preferable that all the waiting times of a vehicle take place at the end of the working day. Indeed, this strategy would allow to the planning people to have a more clear status of the vehicles location for the next operation cycle. Thus, the last column of the table is in accordance with the idea of measure the length of time (i.e. a percentage) that the vehicle is waiting during its route and just before the last stop. We can verify on table 4 that as we have a more constrained instance (i.e. with less available docks), we have larger values for all the previous mentioned indicators. Indeed, we obtain a large negative correlation coefficient about 89% between the number of docks and the total sum of the vehicles arrival times. Similar correlation coefficients are obtained for the sum of the vehicles finish times and for the percentage of waiting time on the vehicles.

On table 5 we present statistical distribution providing some evidence about how constrained is the dock capacity for each transportation node. As we have more cuts added on a transportation node, we have a clear indicator about how many vehicles asking for service are violating the dock service capacity (a bottleneck). This information is useful for the business. Top management can be advised to make some changes on infrastructure (e.g. open more docks) in order to assure transportation service. The last row of the table 5 corresponds to a calculated average for each indicator. Thus, for our instances we have solution times that range from 9 up to 378 seconds and 91 seconds in average. Furthermore, our cut generation strategy adds 108 constraints in average for each instance. Finally, we can verify on the next columns the average number of constraints that are added to the master problem for each transportation node. It would seem that the node # 1 is the one that has the larger volume of transportation orders and vehicles asking for service (24.2 cuts added in average). In a far distance second place we have the nodes # 4 and 14. However this assumption is not necessarily true, because this indicator is correlated with the actual number of docks available for service at each node. What we can conclude from this average measure is that the actual dock capacity at transportation nodes # 1, 4 and 14 is not well balanced according with the volume of transportation orders that are asking for inbound or outbound service. Thus, this is a clear and useful advice for the business management.

6. MODEL APPROACH CONTRIBUTION AND APPLICABILITY

As can been seen in section 3, there are a lot of applications and approaches, each being slightly different from the other, requiring a different model. The solutions are case-specific, since each one of them has its own constraints and objectives, making it virtually impossible to create an algorithm that can be applied to all PDP applications. Rather, we present building blocks of a broad applicability. We can point out here that one of the features that the end-users ask for the model understanding results is the developing of a simple map where we can outline the entire transportation operation. That is, a so-called "table map" where we can verify the volume of transportation orders that require to be serviced at each node and at each hour of the day. For simplification, on table 6 we show a 24 hour timetabling just for some of the transportation nodes of our problem. Each value on the table means the number of outbound transportation services at each node and at each working hour. This "table-map" solution corresponds to the more constrained instance (i.e. with less available dock capacity) and operating with 39 tractors only (71 single-hauls) defined on the relaxed C-VRP model. This instance is good to show how the vehicles operation is stressed when dock capacity is not available or is scare. The last row of the table is the total number of outbound services for each node. This is a true indicator about the volume of operation at each node, for example see nodes # 1, 2, 4 and 14. By the other hand, the last column is the total number of outbound services that occur at each given hour of the day. It is easy to understand why the largest number of outbound services takes place at the very first hour of the day. Also, we can verify that on the last 3 hours of the working day we just have 13 services only on the entire operation. Thus, the main part of the vehicles finishes the operation before at an earlier time. From this very constrained instance we obtain a statistical distribution for the vehicle effective utilization. The minimum utilization is 72.9% and 92.8% in average for the entire fleet. We have 28 vehicles with at least 90% of utilization and 16 vehicles with at least 95%.

In general, for PDP applications is very important to identify which criteria should be viewed as a (hard) constraints and which should be optimized. However, the model we present on this paper is flexible enough to cope with different combinations of objectives and constraints that are very common to find on typical PDP problems. The novelty of our model approach presented in this paper is the combination of three basic stages that interact in order to solve effectively the PDP-TWDS. Our three models are presented on section 4 and we can outline as follows:

- A Relaxed Capacitated Vehicle Routing Problem (C-VRP) model. This heuristic stage works at aggregate level to create a network simplification for the original PDP-TWDS in order to reduce the search space. On this stage we relax all the time window and dock service constraints. Instead, some side constraints at aggregate level are included in order to assure feasibility on the original problem. Our empirical results show that this heuristic has no impact on the optimal solution we find afterwards on the disaggregated stage for the original PDP-TWDS when time windows and dock capacity constraints are fully included.
- A Pickup and Delivery Problem with Time Window Constraints (PDP-TW) model. Typical vehicle capacity, precedence and time windows constraints are fully considered on this stage. A feasible near-optimal solution is obtained at each iteration assuming infinite dock service capacity. For objective function (2.1) there is no need to make a tradeoff between cost and time because the cost obtained before on (1.1) is fully satisfied on (2.1). Instead we focus on optimize the second part of the objective function (2.1) which accounts for the total sum of the arrival times of all the vehicles used to attend the entire set of transportation orders (see table 4).
- A Pickup and Delivery Problem with TW and Dock Service Constraints (PDP-TWDS) model. Dock service constraints are included within an iterative cut strategy scheme. This heuristic is used on this stage to add constraints only as is required at each iteration. As a result we speed up the MIP search for a near-optimal solution.

A main contribution of our work is the implementation of dock service constraints. Particularly our implementation is based on a cut generation strategy. Empirical results show the efficiency of these valid inequalities to constraint connected routes considering dock service constraints on each node. To the best of our knowledge, this is the first time that these valid constraints are implemented. Our implementation indicates that the considered model provides with an appropriate trade-off for the solution quality and computational time. The proposed model not only address the difficulties embedded in the common PDP applications but also some practical concerns about pre-defined and/or forbidden route assignments at the node and vehicle level. Pre-assigned or forbidden requirements arise from business issues like routing realignment. From the practical standpoint, the issue of routing realignment is as how the model could efficiently accommodate for changes on transportation orders additions or dropouts trying not to disrupt the previous design considerably. All these features are very important if we consider how easy this model could be extended to other cases.

It is important to point out that our methodology presents a HMIP model that ensures time windows feasible solutions at each iteration. Thus, it is interesting to verify how rapidly our implementation can converge on cuasi-feasible solutions for dock service constraints (see table 3). However, a future research opportunity exists in order to prove the viability of this paradigm when gap optimality is required to confirm. Our computational results are only to give some evidence to our arguments. They are not intended to be an in–depth comparison of available methods for PDP. Finally, it is true the convenience for integrate our OR model into a GIS environment application in order to complete a transportation planning framework. Clearly transportation routing design process cannot be completely automated, but GIS is an appropriate tool to help in this process. However, a pure GIS does not offer much support for the design and optimization of the vehicle routes. Therefore a hybrid combination of heuristics and MIP exact models for optimization are required. We believe that this kind of optimization featured applications will be the future trend of the GIS transportation industry.

From practical business application standpoint, this operation research (OR) application was developed and implemented to optimize the transportation network between manufacturing plants and distribution centers. During the last years, the firm was interested in developing a better transportation & routing schedules. Indeed, this is the first OR application that has been implemented in the bottler company. It is important to point out that the overall results have been very positive. The firm's top management recognize that features included on the OR model implemented were truly outstanding. The project was a major undertaking, requiring a great deal of thought and effort. The first plans for transportation routes suggested by the optimization model were implemented eight months ago. Throughout the ramp-up and launch of the project, these plans for distribution operation were analyzed and the company found to be an extremely viable idea. Sometime after, during the course of the project, has resulted in a significant increase in productivity and direct savings to the firm. We can list some of the benefits that the company has achieved within this project:

- The firm identifies now a rational set of measures to target and balance on each truck resource. This results on an optimal fleet of trucks, drivers and quantity of warehouse workers for each plant and distribution center.
- An increase on effectiveness on the planning process required to set up an efficient transportation & route schedules. The typical fully-manual planning process time was reduced from 6 hours to less than 20 minutes using the new OR application. This permitted to the company to fine its truck capacity by season on a dynamic basis. As a result the company achieves an optimal capacity to attend the demand on each territory with an optimization of 18 trucks. This represents a 27% reduction from the original quantity of trucks working at the Mexico's operation.
- Streamline truck capacity to align it to a new transportation strategy. The added throughput allows the firm to defer investments on trucks and hauls that have been originally allocated. As a result of our continuous move model, the new routes are more efficient so the total travel time decreased, improving the productivity of the truck drivers. Our model achieves to optimize from actual 120 available hauls down to 71 only. Accordingly with this productivity, the management decided to rationalize the number of available hauls on the firm. The save on investments for hauls was about 15% of the current fleet.
- Identify & implement an optimal cost of service. This allowed the firm to set an optimal deliver frequency. This means less travel time between plants and depots and a 14% increase in volume delivered per route per day.

Besides all these business benefits, the new OR model will allow the company to speed up some others inventory optimization initiatives which are of special interest among Coca Cola bottlers around the world. The proposed model approach can extend the basic problem to address different specific business rules or additional planning criterion. Overall, we have provided a very valuable tool for a more efficient transportation planning according to the company business requirements. Nowadays, our model is being used by the firm to obtain a business solution with significant benefits.

7. CONCLUSIONS

This paper has addressed the PDP-TWDS as a critical component of the operational transportation process. Many logistics problems found in the manufacturing and service industry can be modeled as a PDP-TWDS application. Along with the increasing use of geographical information systems, companies seek to improve their transportation networks in order to tap the full potential of possible cost reduction. Transportation problems have been widely studied in the operations research literature. Over the last decades extensive research has been dedicated to modeling aspects as well as optimization methods in the field of vehicle routing. Still, there are areas and sub-problems, yet, to be researched. Several different objectives and constraints in the transportation design process (i.e. continuous move strategy) are identified and discussed. In this paper, we considered a particular PDP application that is frequently encountered in the real-world logistics operations. Our PDP-TWDS problem incorporated a diversity of practical complexities. Among those, we have a heterogeneous vehicle fleet with different travel times, travel costs and capacity, order/vehicle compatibility constraints, and different start and end locations for vehicles. Instead of assuming that each vehicle becomes available at a one only central depot, we modeled as each vehicle is given a start location where it becomes available at a specific time of the day. Particularly, on our PDP-TWDS extension we add some constraints for dock capacity service at each node and at each hour of the day.

PDP-TWDS is NP-hard since this is a generalization of the well-known PDP and VRP. Within OR various algorithmic approaches have been proposed, some based on integer linear programming, others on classical heuristics and, more recently, on some meta-heuristics. However, solving a real world PDP possesses a significant challenge for both researchers and practitioners. Real-world instances of this NP-hard combinatorial optimization problem are very large, so exact methods have failed even for relatively medium-size instances. MIP models when are used to solve instances as described, require a strong computational effort in time. Indeed, a pure MIP strategy usually compromises its practical implementation in business applications. With a real world application from the service industry, we present a rich featured PDP-TWDS model. We include some extensions that are very common to some of the problems encountered in the industry. Because of the characteristics of this PDP application, it is challenging to solve it within a reasonable computational time based upon the concrete business requirement. Furthermore, field people who are going to deploy the solution of our PDP application may have to pay more attention to the feasibility of the solution in practice than a pure optimal solution in terms of mathematics.

A particular emphasis is given to a business application case at Embotelladoras ARCA. One of the individual optimization problems arising here is the task to schedule the operation on a transportation network with several plants and distribution centers. In this case we aim to make an optimization over a full fleet of tractors vehicles. In particular, it is of interest to deal with large scale instances with a high presence of time windows constraints. Time windows constraints can be found on the nodes (i.e. plants, warehouses, etc) or on the vehicles as well. Thus, as can be verified, our PDP-TWDS considers the schedule of a large scale of vehicles simultaneously. As a result some real world difficulties arise for dock service capacity issues. In order to tackle these simultaneous and conflicting objectives, a hybrid MIP approach has been developed to accommodate to the particular business requirements. We present the components of the model and a step-by-step description of the solution procedure. We implement a three stage HMIP model. The last stage includes a cut generation strategy to add dock service capacity constraints on an iterative scheme only as is required. We believe that this is an important contribution of our work. Empirical results show the efficiency of these valid inequalities.

Computational results for a real-world instance around 100 single-haul vehicles and 500 transportation orders are reported, showing the suitable of our model to provide good quality solutions. Given the current state of the art for the solution of vehicle routing problems with time windows, it seems fair to say that these are large instances. A Relaxed Capacitated Vehicle Routing Problem (C-VRP) model is used to find a solution at aggregate level. On this stage we relax all the time window and dock service constraints. Instead, some side constraints at aggregate level are included in order to assure feasibility on the original problem. With the solution obtained at aggregate level we reduce the number of vehicles required and as a consequence the complexity of the original problem. At this aggregated level of results, we report an optimization of 34% when compared with the actual number of single-hauls in use and 27% when compared with the actual number of trucks on rental. The empirical results show that our simplification C-VRP model has no impact on the optimal solution we find for the original problem on the PDP-TWDS stage when time windows and dock capacity constraints are fully included. Thus, optimization and economic benefits for the company are assured. We report on this work good quality solutions (optimality tolerance $\leq$ 1%) in short computational times (total solution time $\leq$ 10 minutes). In general, the performance of a method is difficult to compare. Clearly, the diversity of theoretical and practical problems is immense. Consequently, there are not too many papers working on the same problem. Constraints can be different, objective functions can be different. Another possible way to compare a method is in checking the problem size that can solve and the amount of computer time and space it needs. It is clear that future research should be done in order to statistically test our method. This issue will be overcome of the subsequent paper. However the results obtained so far, indicate that our model is robust to solve this hard problem, reaching good solutions in short computational times.

We integrate our HMIP model into an advanced interactive tool based on a MAPINFO™ application. Thus, we achieve a practical functionality to the end-users. This GIS environment can be used in different contexts. At the operational level, it represents a valuable tool to quickly produce and deploy different solutions. At the tactical level it can be used to simulate alternative scenarios and evaluate the impact of changes in time windows on nodes and vehicles as well. It is important to point out the interest of the end users about how our model can easily take the already existing routes into account. Particularly, the model is ready prepared to consider any prescribed and forbidden vehicles routes. All these features can be extended for any case when some vehicle routing information is present at the beginning of the planning process. Thus, the company evaluates how our model efficiently accommodates for system changes like transportation orders additions or dropouts trying not to disrupt the previous routing design considerably.

Finally, with respect to the literature on routing and scheduling problems, it is interesting to observe that PDP have received far less attention than VRP applications. However, time constraints play an even more prominent role in PDP-TW. Furthermore, assigning transportation orders to vehicles in PDP-TW is much more difficult than in VRP-TW. In VRP-TW, all the origins of the transportation orders are located at the depot. Therefore, transportation orders with geographically close destinations are likely to be served by the same vehicle. In the PDP-TW, geographically close destinations may have origins that are geographically far apart and we cannot conclude that they are likely to be served by the same vehicle. The current situation in freight transportation reflects the need for improved efficiency, as the traffic volume increases much faster than the road network grows.

8. ACKNOWLEDGEMENT

The authors would like to thank the Managing Editor and the anonymous referees for their valuable comments and constructive suggestions.

REFERENCES

1. Ascheuer, N., Fischetti, M. and Grotschel, M. (2001), *Solving ATSP with time windows by branch-and-cut*, Springer-Verlag, Germany.
2. Cook, W. and Rich, J. (1999), *A parallel cutting-plane algorithm for the vehicle routing problem with time windows*, Computational and Applied Mathematics Rice University, Houston USA.
3. Dumas, Y., Desrosiers, J. and Soumis, F.(1991). The pickup and delivery problem with time windows. *European Journal of Operational Research,* 54: 7-22.
4. Dumas, Y., Desrosiers, J. and Solomon, M. (1995), An algorithm for the traveling salesman problem with time windows, *Operations Research,* 43(2): 23-25.
5. Desrosiers, J., Dumas, Y., Solomon, M. and Soumis, F.(1995). Time constrained routing and scheduling. Network Routing, *Handbooks in Operations Research and Management Science,* 8: 35-139. INFORMS: Elsevier Science.
6. Fisher, R. and Eijl Van, C. (1995), A polyhedral approach to the delivery man problem, *Technical Report 95-19*, Department of Mathematics and Computer Science, Eindhoven University of Technology, the Netherlands.
7. Palmgren, M. (2001), *A Column Generation Algorithm for the Log Truck Scheduling Problem*, Department of Science and Technology (ITN), Linköping University, Norrköping Sweden, 3.
8. Lu, Q. and Dessouky, M. (2006), A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research,* 172: 672-687.
9. Ropke, S., Cordeau, J. and Laporte, G. (2007), Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows. *Networks*, 49: 258-272.
10. Savelsbergh, M.W.P. and Sol, M. (1998). DRIVE: Dynamic routing of independent vehicles. *Operations Research,* 46: 474-490.
11. Solomon, M. (1984), On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints, *Report 83-05-03*, The Wharton School, USA.
12. Tsitsiklis, J. (1992), Special cases of traveling salesman and repairman problems with time windows, *Networks No. 22*, USA.

Table 1 Aggregated-level solutions for the Relaxed C-VRP model. Objective function for minimize variable cost.

(F1) # of Vehicles per Transp. Lane	(F2) # of Transp. Lanes per Vehicle	# of Binary Variables	LP Solution	Comp. Minutes	Best MIP Solution	% of GAP to optimal.	# of Single Trailers	# of Double Trailers	#Tractors/ # Single hauls
20	20	8,751	172,569	2	211,876	14.10%	11	45	56 / 101
20	20			3	193,818	6.09%	9	41	50 / 91
20	20			5	192,238	5.32%	9	40	49 / 89
20	20			10	190,278	4.34%	9	39	48 / 87
30	30	15,915	168,632	2	191,382	6.82%	9	37	46 / 83
30	30			3	190,262	6.26%	9	37	46 / 83
30	30			5	189,076	5.67%	9	38	47 / 85
30	30			10	188,002	5.13%	9	38	47 / 85
40	40	21,534	166,062	2	NA	NA	NA	NA	NA
40	40			3	190,378	7.77%	10	43	53 / 96
40	40			5	187,860	6.53%	9	37	46 / 83
40	40			10	186,018	5.60%	9	38	47 / 85

Table 2 Aggregated-level solutions for the Relaxed C-VRP model. Objective function for minimize No. of Vehicles.

(F1) # of Vehicles per Transp. Lane	(F2) # of Transp. Lanes per Vehicle	Binary Variables	Comput. Minutes	% of GAP to optimal.	# of Single Trailers	# of Double Trailers	# Tractors / # Single hauls
20	20	8,754	3	21.89%	9	37	46 / 83
20	20		5	19.92%	9	36	45 / 81
20	20		10	17.86%	9	35	44 / 79
30	30	15,915	3	31.03%	11	40	51 / 91
30	30		5	20.55%	9	35	44 / 79
30	30		10	17.40%	8	34	42 / 76
40	40	21,534	3	25.99%	9	37	46 / 83
40	40		5	25.98%	9	37	46 / 83
40	40		10	17.01%	8	33	41 / 74
40	40		20	7.58%	7	32	39 / 71

Table 3 Instances for the complete PDP-TWDS model. Available docks for service and efficiency measures.

1	**2**	**3**	**4**	**5..6**	**14**	**15**	**16..27**	**28**	**29..34**	**Total Docks**	**# ITERs**	**# CUTs**	**Solution Time**	**Max. Docks**	**Avg. Docks**	**Docks /Secs.**
8	6	4	4	4	3	3	2	3	2	75	6	40	13	21	4.7	1.58
8	6	4	4	4	3	3	1	3	2	63	10	49	26	24	3.5	0.92
7	6	4	4	4	3	3	1	3	2	62	8	60	28	25	4.6	0.89
6	6	4	4	4	3	3	1	3	2	61	5	50	14	27	6.4	1.89
6	5	4	4	4	3	3	1	3	2	60	9	55	28	28	3.9	1.00
6	4	4	4	4	3	3	1	3	2	59	12	69	45	29	3.0	0.65
6	3	4	4	4	3	3	1	3	2	58	4	62	9	32	10.4	3.66
6	3	3	4	4	3	3	1	3	2	57	8	74	25	33	6.1	1.30
6	3	2	4	4	3	3	1	3	2	56	7	65	23	34	5.4	1.51
6	3	1	4	4	3	3	1	3	2	55	14	98	67	36	4.2	0.54
6	3	1	3	4	3	3	1	3	2	54	12	102	118	39	2.4	0.33
6	3	1	2	4	3	3	1	3	2	53	21	134	110	42	4.2	0.38
6	3	1	2	3	3	3	1	3	2	51	18	137	98	45	5.0	0.46
6	3	1	2	2	3	3	1	3	2	49	18	149	117	49	4.3	0.42
6	3	1	2	2	2	3	1	3	2	48	12	147	86	55	6.8	0.64
6	3	1	2	1	2	2	1	3	2	45	23	198	378	62	3.9	0.16
6	3	1	2	1	2	1	1	2	2	43	20	225	236	66	6.0	0.28
6	3	1	2	1	2	1	1	2	1	37	16	226	218	71	7.7	0.33

Fabian: *Hybrid MIP method for a Pickup and Delivery Problem with Time Windows and Dock Service Constraints*
IJOR Vol. 8, No. 1, 36–56 (2011)

Table 4 Instances for the complete PDP-TWDS model. Activity measures for vehicles operation.

Total Docks	Solution Time in secs.	Total Sum for Vehicles Time	Vehicles Sum End Times	Vehicles Average End Times	# Vehicles with End Time > 22	Vehicles % of Wait.
75	13	2,184.70	697.70	17.89	7	4.51%
63	26	2,136.90	688.83	17.66	6	3.41%
62	28	2,159.07	696.22	17.85	7	3.47%
61	14	2,213.75	702.68	18.02	11	4.32%
60	28	2,198.50	696.93	17.87	10	3.66%
59	45	2,168.37	694.53	17.81	7	4.06%
58	9	2,210.57	705.55	18.09	9	4.54%
57	25	2,228.33	710.00	18.21	9	4.32%
56	23	2,249.00	721.93	18.51	10	4.37%
55	67	2,300.70	728.50	18.68	13	5.33%
54	118	2,238.43	707.72	18.15	8	4.69%
53	110	2,258.85	728.78	18.69	11	5.08%
51	98	2,240.35	709.47	18.19	8	4.65%
49	117	2,311.65	727.12	18.64	12	5.77%
48	86	2,310.92	729.27	18.70	12	5.82%
45	378	2,385.28	748.45	19.19	12	6.65%
43	236	2,358.55	743.78	19.07	10	7.05%
37	218	2,416.90	754.80	19.35	14	7.26%

Fabian: *Hybrid MIP method for a Pickup and Delivery Problem with Time Windows and Dock Service Constraints*
IJOR Vol. 8, No. 1, 36−56 (2011)

Table 5 Instances for the complete PDP-TWDS model. Number of dock constraints added for each transportation node.

Total Docks	**Sol. Time**	**# of Cuts**	**1**	**2**	**3**	**4**	**5**	**6**	**14**	**15**	**16**	**19**	**20**	**27**	**28**	**30**	**31**	**32**	**34**
75	13	40	15	0	0	7	3	1	7	2	0	0	0	1	3	0	1	0	0
63	26	49	14	0	0	8	3	1	10	3	1	1	1	3	3	0	1	0	0
62	28	60	19	1	0	7	3	1	15	3	2	1	1	3	3	0	1	0	0
61	14	50	22	0	0	7	3	1	5	2	1	1	1	3	3	0	1	0	0
60	28	55	21	1	0	7	3	2	9	1	2	1	1	3	3	0	1	0	0
59	45	69	25	4	0	8	2	1	13	3	2	3	1	3	3	0	1	0	0
58	9	62	22	9	0	7	2	1	8	3	1	1	1	3	3	0	1	0	0
57	25	74	26	16	1	7	3	1	6	1	2	3	1	3	3	0	1	0	0
56	23	65	22	11	2	6	2	1	9	2	1	1	1	3	3	0	1	0	0
55	67	98	32	11	16	7	3	1	12	3	2	3	1	3	3	0	1	0	0
54	118	102	27	13	18	12	2	2	12	3	2	3	1	3	3	0	1	0	0
53	110	134	26	19	21	35	3	1	10	2	3	6	1	3	3	0	1	0	0
51	98	137	27	19	19	33	5	3	15	1	2	5	1	3	3	0	1	0	0
49	117	149	32	11	19	32	13	8	14	4	2	6	1	3	3	0	1	0	0
48	86	147	25	10	18	29	14	8	28	1	3	4	1	3	2	0	1	0	0
45	378	198	26	9	21	41	24	24	33	3	3	7	1	3	2	0	1	0	0
43	236	225	26	13	19	35	24	23	31	32	2	6	1	3	9	0	1	0	0
37	218	226	29	17	20	30	25	23	28	27	3	4	1	3	6	1	5	1	3
54.8	91.1	107.8	24.2	9.1	9.7	17.7	7.6	5.7	14.7	5.3	1.9	3.1	0.9	2.9	3.4	0.1	1.2	0.1	0.2

Table 6 Timetabling solution for the more constrained instance with the least available dock capacity

Node #	1	2	3	4	5	6	14	15	16	19	20	27	28	30	31	32	34	Total
0	6	3		2	1	1												13
1	4	1		1	1	1												8
2	2	2	1	1	1	1	1	1										10
3	2	2	1	1	1	1	1	1										10
4	3		1	2	1	1	1	1					1					11
5	2	1	1	1				1					1					7
6	2		1	2				1		1					1	1	1	10
7	1		1	1			2		1				2		1		1	10
8	2	2		2			2						2			1		11
9	2	1	1	2	1		1	1							1			10
10	3			1	1	1	1											7
11	1	3	1															5
12	3	2			1	1	2				1						1	11
13	1	2		2	1	1	1	1					1					10
14	3	2		1	1	1	1						1					10
15	3		1		1	1	2			1			1				1	11
16	2		1			1	1	1							1			7
17	2	1		1		1	1						1	1				8
18	2	1		1			1		1	1					1		1	9
19	1	1		1			2	1	1			1	1	1	1			11
20	1						2	1		1		1						6
21	4		1							1		1						7
22	3							1										4
23		1					1	1										3
24		1		1			2	1			1							6
Total	55	26	11	23	11	12	25	13	3	5	2	3	11	2	6	2	5	