Near Optimal Solution for Continuous Move Transportation with Time Windows and Dock Service Constraints

Fabián López-Pérez[1] and Carlos Cantú-Aguillén[2]

[1] Centro de Desarrollo Empresarial y Postgrado, CEDEEM-UANL, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, NL, México

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey, ITESM Campus Monterrey, Monterrey, NL, México

fabian.lopez@e-arca.com.mx, ccantu@itesm.mx

Abstract. We consider a pickup and delivery vehicle routing problem (PDP) commonly found in the real-world logistics operations. The problem includes a set of practical complications that have received little attention in the vehicle routing literature. There are multiple vehicle types available to cover a set of transportation orders with different pickup and delivery time windows. Transportation orders and vehicle types must satisfy a set of compatibility constraints. In addition, we include some dock service capacity constraints as required in real-world operations when there are a large number of vehicles to schedule. This problem requires to be attended on large scale instances: transportation orders ≥ 500, single-haul vehicles ≥ 100. Exact algorithms are not suitable for large scale instances. We propose a model to solve the problem in three stages. The first stage constructs initial solutions at the aggregated level relaxing time windows and dock service constraints of the original problem. The other two stages impose time windows and dock service constraints within a cut generation scheme. Our results are favorable in finding good quality solutions in relatively short computational time.

Keywords. Vehicle routing optimization, logistics and transportation planning, time windows, PDP-TWDS.

Solución casi óptima para transportación de movimiento continúo con restricciones de ventana de tiempo y de servicio de andenes

Resumen. Se considera un problema de vehículos dedicados a la carga y descarga de producto (PDP) el cual es comúnmente encontrado en las operaciones logísticas. El problema incluye un conjunto de complejidades prácticas encontradas en el mundo real y que han recibido relativamente poca atención en la literatura científica dedicada a los problemas de ruteo de vehículos. Existen múltiples tipos de vehículos disponibles para cubrir un conjunto de órdenes de transporte con diferentes ventanas de atención tanto en la carga como también en la descarga. Las órdenes de transporte así como los vehículos deben satisfacer ciertas restricciones de compatibilidad. Además, se incluyen algunas restricciones de capacidad de andenes de servicio en los nodos de carga y descarga. Este problema requiere ser resuelto para instancias de gran tamaño: ordenes de transporte ≥ 500, vehículos ≥ 100. Los algoritmos de solución exacta no son adecuados para este tipo de instancias. Por tanto se propone un modelo de tres etapas. La primera etapa construye las soluciones iniciales de manera agregada mediante la relajación temporal de las restricciones de ventanas de horario y andenes disponibles. Las otras dos etapas van añadiendo dichas restricciones al problema dentro de un esquema iterativo de generación de cortes. Los resultados obtenidos son favorables tanto lo que respecta a la calidad de las soluciones como en los tiempos computacionales requeridos.

Palabras clave. Optimización y ruteo de vehículos, planeación logística y transportación, ventanas de horario, PDP-TWDS.

1 Introduction

The Pickup and Delivery Problem with Time Windows and Dock Service Constraints (PDP-TWDS) is a variant of the well-known Vehicle Routing Problem with Time Windows (VRP-TW).

Vehicle routing plays a central role in logistics management. Different vehicle routing problems address different practical situations but focus on a common 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 the task to identify which transportation orders should be covered by each vehicle and at what time, in order to minimize the total transportation cost subject to a variety of constraints and complications. This paper illustrates the potential of the proposed approach in the context of a case study of a large soft drink company. *Embotelladoras ARCA* is the second largest bottler of *Coca-Cola* products in Latin America and the fifth in the world. The company faces a PDP-TWDS in transportation operations in the northern territory of Mexico. With each head haul move of a truck, the goods are transported from their origin to their destination so that revenue is generated. However, without the goods, the truck moves an empty haul, and in such cases only costs are incurred. Any attempt to enforce a transportation order from a given destination location back to the origin location results in unsuccessful practice. Pooling transportation orders among several dispatchers may avoid simple trips by replacing the empty return of a simple trip with a transportation order of another dispatcher. Thus, the collaboration of two or more dispatchers allows important cost savings for the company. The cost of an overall route is smaller when two trips are pooled together in comparison with making them independently. If trips $T_{i1,j1}$ and $T_{i2,j2}$ are combined then the following will be fulfilled:

- 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 destination *j2*.
- Return to the initial origin *i1*.

It is estimated that at least 38% of truck movements of the company are empty haul moves. This means millions of kilometers 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. Since our PDP-TWDS is NP-hard, combined with the fact that the real world PDPs are very big, there is no much hope for finding an optimal model that will work acceptably fast in practice. Thus, we propose a Hybrid Mixed Integer Programming (HMIP) approach to this problem. 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 introduce some notation and present our model approach structured in three stages. Section 5 contains a description of some empirical results we obtained in our implementation. Model contributions and applicability are explained in Section 6. We present some concluding remarks in Section 7.

2 Problem Definition

PDP-TW solutions are case-specific, since each one of them has its own constraints and objectives. Due to this fact, it is virtually impossible to create an algorithm that can be applied to all situations. Rather, we present the building blocks of broad applicability.

2.1 Objective Function

The goal of our model is to determine an optimum route for multiple vehicles dedicated to physical distribution operations. A route is defined as the arrival sequence of a vehicle (i.e., a single or double trailer) which has to attend a set of nodes or warehouses waiting for service. A service can be defined as delivery or pickup of any type of items (in our case, products). 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. Let us define N as a set of nodes for pickup and delivery operations, and $i \in N$. Set M is defined for different vehicles, and $m \in M$.
2. We define $P(i)$ as a subset of vehicles located at node i, where $P(i) \subseteq M$ and $\forall i \in N$. At the start of the day, each vehicle departs 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.

3. We have a set of transportation orders R. Each order $r \in R$ consists of a pickup at a location i and a delivery at a location j. Precedence constraints imply that a vehicle m should visit i before j for each transportation order r.
4. Let us define K as a set of different products (stock keeping units, SKUs) to be transported, where $k \in K$. The parameter Dijk is the total demand to transport from node i to node j for SKU k, where $(i,j,k) \in R$, $\forall$ $i,j \in N$. We define V as a subset of transportation lanes along which some volume has to be delivered or picked up, where $(i,j) \in V$, $\forall$ $i,j \in N$.
5. Since trailers have a loading and unloading access by the sides, such design is not affected by nested precedence constraints.
6. We define the parameter Hk as the quantity of cases of SKU k that can be loaded per cubic meter, $\forall$ $k \in K$.
7. Each order $r \in R$ has a specific mix of SKUs. The capacity constraints guarantee that any mixture load of items on a vehicle m should be less than the vehicle capacity. We define the parameter Qm as the cubic meter capacity for vehicle m, where $m \in M$.
8. The use of a vehicle is constrained at the transportation lane level. Let us define $A(i,j)$ as a 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 $i,j \in N$.
9. We define TS as the service time for a single trailer configuration and TF for a double trailer.
10. Each node has a particular time window for service. Let us define parameters INi and CNi as the opening time and the closing time, respectively, at a node i, where $i \in N$.
11. A vehicle m cannot operate either before its window opens or after its window closes. Let us define parameters IVm and CVm as the opening time and closing time for a vehicle m, where $m \in M$.
12. The dock service capacity is constrained as the quantity of vehicles that can be attended at each node and at each hour of the day. Let us define the parameter S_{ih} as the quantity of docks available for service at a node *i* at a working hour h, where *i* □ *N, h* □ *{1, ..., 24}.*
13. Let us define the time and cost required to go from each node to all the others on the distribution network as follows:
    ST_{ij} = transportation time for single trailer □ *(i,j)*□ *V*
    FT_{ij} = transp. time for double trailer □ *(i,j)*□ *V*
    SC_{ij} = transp. cost for single trailer $\forall$ (i,j) $\in$ V
    FC_{ij} = transp. cost for double trailer $\forall$ (i,j) $\in$ V

3 Related Research

PDP-TW is more difficult to solve than VRP-TW (Vehicle Routing Problem with Time Windows) and TSP-TW (Traveling Salesman Problem with Time Windows). J.N. Tsitsiklis [13] showed that the basic TSP-TW is strongly NP-complete. M.M. Solomon [12] developed 87 test instances for the VRP-TW. The biggest instance he solved included about 100 nodes. Until the year 1999, there had been 17 instances that still remained unsolved Ascheuer *et al.* [1, 2] tested TSP-TW instances containing up to 233 nodes. For an instance of 69 nodes, 5.95 minutes of solution time was required. All bigger instances required more than 5 hours of solution time to converge to a feasible solution. Dumas *et al.* [6] presented a dynamic programming algorithm for the TSP-TW. These authors were able to solve problems of up to 200 nodes.

PDP-TW is a generalization of VRP-TW M. Palmgren [8] and is NP-hard Desrosiers *et al.* [6]. The first optimization algorithm for the PDP-TW was a branch-and-price algorithm presented by Dumas *et al.* [5]. A set partitioning formulation is solved by a branch-and-price method in which columns of the negative reduced cost are generated by dynamic programming. This approach is capable of solving some instances with up to 22 vehicles and 190 requests. Savelsbergh and Sol [11] 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.

More recently, a branch-and-cut algorithm for the PDP-TW was described by Lu and Dessouky [7]. 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. Ropke *et al.* [10] introduced a new formulation for the PDP-TW which did not require the use of a vehicle index to impose pairing and precedence constraints. They showed that this approach was capable of solving some instances with up to 8 vehicles and 96 requests.

In general, heuristics can solve larger scale problems in less solution time than exact methods. Recently, some progress has been achieved in meta-heuristics by developing tabu search and genetic algorithms. An extensive survey on methods to solve the PDP-TW can be found in Parragh *et al*. [9]. In this work, the best results are obtained by column generation methods. Instances of up to 500 requests and 53 vehicles can be solved with this method. Given the enormous complexity of PDP problems, it is not realistic to apply pure optimization methods. Thus, we develop a hybrid approach to integrate some heuristics into an optimization method based on a cut generation strategy. In the next section, we present our model. Briefly speaking, there are two main differences between our approach and the previous methods: (1) dock capacity constraints handling and (2) iterative cut generation strategy.

4 Proposed Model

We can figure out two objective functions: (1) minimize the total time, distance, or cost of vehicles needed to execute all the set of transportation orders or (2) minimize the number of vehicles. Our model is based on a continuous move strategy. Here, attempts are made to match multiple truckload pickups and deliveries to one truck. The benefit of continuous moves is derived from the overall reduction of empty haul distances. For each trip, we compute its total cost, including trips associated with empty hauls. All trips are planned for one day of operation in order to enforce and simplify truck location requirements. We propose to solve the problem in three stages. The first stage constructs initial solutions at the aggregated level relaxing some constraints of the original problem. The other two stages impose time windows and dock service constraints, respectively.

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

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

Sets and parameters:

N = set of nodes (plants and distribution centers), $i \in N$
M = set of vehicles (trailers), where $m \in M$
$P(i)$ = subset of vehicles at node i, $P(i) \subseteq M$ and $i \in N$
K = set of different SKUs k, where $k \in K$
H_k = # of cases of SKU k per cubic meter, $\forall k \in K$
R = set of transportation orders, where $r \in R$
Q_m = # of cubic meters on vehicle m, where $m \in M$
$A(i,j)$ = subset of vehicles that can be used on transp. lane (i,j), $\forall (i,j) \in V$, where $A(i,j) \subseteq M$, $m \in M$
TS = service time for single trailer configuration
TF = service time for double trailer configuration
ST_{ij} = transp. time for single trailer $\forall (i,j) \in V$, $i,j \in N$
FT_{ij} = transp. time for double trailer $\forall (i,j) \in V$, $i,j \in N$
SC_{ij} = transp. cost for single trailer $\forall (i,j) \in V$, $i,j \in N$
FC_{ij} = transp. cost for double trailer $\forall (i,j) \in V$, $i,j \in N$

D_{ijk} = demand from i to j for SKU k, $\forall (i,j) \in V, (i,j,k) \in R$
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 = demand covering factor (upper bound).

Decision variables:

$W_{m1,m2}$ binary ⇒ (1) if vehicle m1 is grouped with vehicle m2, (0) otherwise, $\forall (m1,m2) \in P(i)$.
$X_{ij}^{m1,m2} \geq 0$, integer ⇒ # of trips from node i to node j using vehicle (m1,m2), $\forall (i,j) \in V$, $(m1,m2) \in A(i,j) \subset P(i)$
$F_{ijk} \geq 0$, ⇒ quantity of cases to transport from node i to node j of SKU k, $\forall (i,j,k) \in R$

$$\text{Obj. Fun}_{\min} \sum_{i \in N} \sum_{j \in N} \left[\sum_{(m1=m2) \in M} X_{ij}^{m1,m2} \cdot SC_{ij} + \sum_{(m1 \neq m2) \in M} X_{ij}^{m1,m2} \cdot FC_{ij} \right] \quad (1.1)$$

$$\text{Obj. Fun}_{\min} \sum_{m1 \in M} \sum_{m2 \in M, m1 \leq m2} W_{m1,m2} \quad (1.2)$$

subject to:

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

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

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

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

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

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

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

$$\sum_{i \in N} X_{ij}^{m1,m2} = \sum_{i \in N} X_{ji}^{m1,m2}, \quad \forall\, j \in N, \forall\, m1, m2 \in M, \quad (1.10)$$

$$where\ (m1,m2) \in A(i,j) \subset P(i), (i,j) \in V$$

Objective Function 1.1 is formulated to minimize the variable cost (i.e., distance) needed to execute the set of transportation orders. 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 is assigned exclusively to a single or double trailer operation only. Constraints 1.5-1.6 restrict the maximum quantity of trips so that a single and double trailer can perform on a 24-hour time horizon. Constraint 1.7 assures that the quantity of cubic meters used to transport products from node *i* to node *j* is equal to the total cubic meters of available capacity considering single and double trailer operation. Constraint 1.8 corresponds to the balance flow constraint which assures that the total transportation volume from node *i* to node *j* is sufficient to cover the total demand at each SKU level. Constraint 1.9 restricts the maximum volume of the product to be transported from node *i* to node *j* (i.e., the upper bound). Finally, constraint 1.10 corresponds to the balance flow constraints imposed at the vehicle level.

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

As a result from the previous model, we obtain an optimal assignment of the vehicles. Binary variable $W_{m1,m2}$ identifies which vehicles are going to operate on a single trailer basis and which ones will operate on a double trailer basis. From here to the end, all double trailers will be modeled as a single vehicle with the combined capacity. Integer variable $X_{ij}^{m1,m2}$ calculates an optimal number of trips for each vehicle between the origin and destination nodes. The next PDP-TW model will take advantage from the previous information. Thus, for this model we add time window constraints. We model as follows:

Sets and parameters:

L = set of stops on a given route (1,…, 9)
X_{ij}^{m} *= trips i-j with vehicle m,* $\forall (i,j) \in V$*,* $m \in M \in A(i,j)$
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} *= transp. cost (i,j) of vehicle m,* $\forall (i,j) \in V$*,* $m \in M \in A(i,j)$
Z_{ij}^{m} *= transp. and service time for lane (i,j) on vehicle m*

Decision variables:

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

$$\text{Obj. Fun}_{\min} \sum_{i\in N}\sum_{j\in N}\sum_{m\in M}\sum_{l\in L}\left[Y_{ij}^{ml}\cdot TC_{ij}^{m} + T_{ij}^{ml}\right] \quad (2.1)$$

Subject to:

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

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

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

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

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

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

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

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

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

Expression 2.1 is formulated as a multi-term objective function. The first part is used to minimize the transportation cost of the vehicles. The second part minimizes the entire set of arrival times that corresponds to each individual trip *(i,j,m)*. Constraint 2.2 assures that the whole set of trips obtained by Model 1 are fully covered by Model 2. Equation 2.3 shows the balance flow constraints imposed at the vehicle level. Constraints 2.4-2.7 are formulated for time window constraints for each node and vehicle. Constraint 2.8 assures that each vehicle departs from only one origin node at each trip. Constraints 2.9-2.10 calculate the arrival times for the entire set of trips.

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 an optimal assignment of the vehicles considering the vehicle capacity and time window constraints as well. Binary variable Y_{ij}^{ml} identifies if vehicle *m* is routed from node *i* to node *j* on sequence *l*. Positive variable T_{ij}^{ml} calculates the arrival time at each node for all the vehicles. Our previous PDP-TW model works as the master model. Then, the logic we apply here is to iteratively generate cuts in a branch and cut scheme. 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. We compare the quantity of vehicles that are being dispatched simultaneously at a given node and at a given hour versus the quantity of docks that the node is capable to attend at a given hour. Then, we add these cuts to the master model accordingly. This procedure continues until we find the first feasible-optimal solution for the problem that does not violate the dock service capacity. We model as follows:

Sets and parameters:

Sjh = docks available at node j at hour h, j∈N, h∈{1..24}

E = set of cases where vehicle α is violating the dock service constraint at node j at hour h

OT=min. offset time between arrivals of vehicles α and β

$e(j\alpha, j\beta, h) \in E$ → *if and only if*

$$\begin{cases} \left| T_{ij}^{\alpha,l} - T_{ij}^{\beta,l} \right| < OT \quad and \\ \#of\ vehicles\ arriving\ at\ node\ j\ at\ hour\ h\ > Sjh \\ where\ \{\alpha \ldots \beta\} \in M \end{cases}$$

Decision variables:

dock constraints for node *j* at hour *h*

Case e: where e(jα,jβ,h)∈E

$B^{+}_{e} \geq 0$ *time gap between arrivals of vehicles α and β to node j at hour h,*

$B^{-}_{e} \geq 0$ *time gap between arrivals of vehicles β and α*

U_e *binary ⇒ (1) if vehicle α arrives before vehicle β to node j at hour h, (0) otherwise, where e ∈E*

subject to:

$$T_{ij}^{\alpha,l} - T_{ij}^{\beta,l} = B_e^+ - B_e^-, \quad \forall\ e(j\alpha, j\beta, h) \in E,$$

where

$$(i,j) \in V, \alpha, \beta \in M \in A(i,j), l \in L \tag{3.1}$$

$$B_e^+ + B_e^- \geq OT, \quad \forall\ e \in E \tag{3.2}$$

$$B_e^+ \leq 24 \cdot U_e, \quad \forall\ e \in E \tag{3.3}$$

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

Constraint 3.1 deals with a set of deviational variables to calculate the gap on the arrival times to node *j* for each pair of vehicles α and β. Constraint 3.2 assures that the gap time for any given pair of vehicles α and β arriving at node *j* asking 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 on deviational variables. We define 24 hours as the time frame horizon.

As it can be verified, these constraints grow exponentially because the number of nodes and vehicles is big. Thus, in our third model we add these constraints on an iterative scheme only when required. In summary, we model a linear relaxation of the PDP-TW problem that results in a very efficient solution of the master problem by the MIP solver. At this stage, we fully apply the time window constraints but relax the dock service capacity constraints. Thus at each iteration, a feasible solution is obtained for the time window constraints at all nodes and for all vehicles. An iteration procedure is performed within the MIP solver framework to add the dock capacity constraints only when necessary. We have found that our approach is capable of obtaining competitive solutions in acceptable computational time for real business instances of around 160 vehicles and 500 transportation orders. Figure 1 presents the tree stages and their relationships. We indicate where heuristics is applied as well.

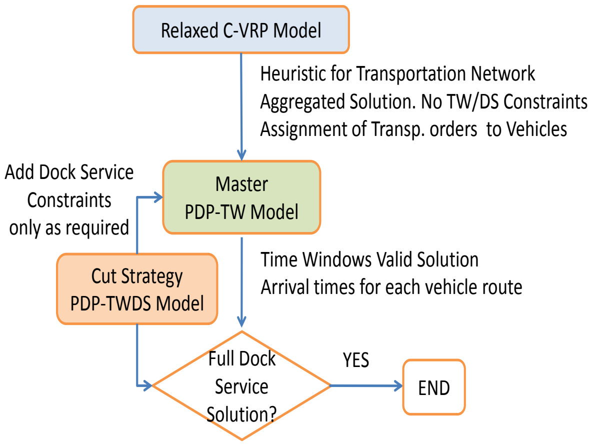

Fig. 1. Three-Stage Model for PDP-TWDS

5 Test Instances and Results

We implement our model for a real-world case. Although we test the model on different number of available vehicles, however, the number of transportation orders corresponds to a given real-world instance that the company faces regularly. Our experiments can be structured as follows:
(1) Aggregated solutions for the C-VRP model. Several instances are tested with different values of parameters that are used to setup the transportation network. Thus, the main feature that makes the difference for each instance corresponds to the number of vehicles that can be considered for each transportation lane and also the number of transportation lanes that can be considered for each vehicle.
(2) Detailed solutions for PDP-TWDS instances. We make tests with different values of the number of available docks for service. Here, we obtain some efficiency measures for vehicles operation as well. These different instances are justified as we can anticipate some requirements to the management of the firm in terms of transportation infrastructure (i.e., docks available for service at each node).

In the first model, we relax time windows and dock service constraints. Instead, some side constraints at the aggregated level are included in order to assure feasibility when solving the original problem. This heuristic stage works at the aggregated level to create a network simplification for the original PDP-TWDS in order to reduce the search space. The basic idea in our heuristic is to identify a subset of incumbent decision variables in such a way that transportation orders and vehicles are compatible. We point out the compatibility between transportation orders and vehicles types, the compatibility between transportation lanes and vehicles types, and the compatibility between transportation nodes and vehicles types. Thus, one of the main contributions of our work is the development of a model that is stable for input data in order to find a way to dismiss enough links to make the solution of the first aggregated MIP model very efficient. This trade-off on optimality will be detailed in the next section. We present some results showing that our method is efficient for solving large scale instances. The CPU configuration used in our implementation is Win X32, 2 Intel Cores 1.4GHz. We implement our model on X-PRESS MIP Solver from FICO™ (Fair Isaac, formerly Dash Optimization).

Table 1 shows optimal solutions found by our first stage model using different combinations of input parameters: *(F1)* the quantity of vehicles to be considered for each transportation lane and *(F2)* the quantity of transportation lanes to be considered for each vehicle. These two heuristic parameters *F1* and *F2* affect the matrix size of the decision variables and the complexity to be considered by our first stage model when feasible solutions are obtained at the aggregated level.

Particularly, Table 1 shows the results obtained for Objective Function 1.1. As it can be verified in the last column, the number of tractors or single hauls is not actually minimized when compared with Table 2. It is seen from Table 1 that with an appropriate setting of parameters *F1* and *F2*, we obtain good quality solutions in a short computational time. However, the trade-off we have to pay with this strategy is a possibility to have an over-constrained solution space. When we set *F1* = 40 and *F2* = 40, better solutions can be obtained but more time is required to solve the problem. The gap to optimality in Column #7 is expressed as a percentage comparing the best MIP solution found in Column #6 with the best bound obtained by the solver at a given iteration

Table 1. Aggregated-Level Solutions for the Relaxed CVRPmodel. Objective function minimizes the cost variable

(F1) # of Vehicles per Transp. Lane	***(F2) # of Transp. Lanes per Vehicle***	***# of Binary Variables***	***Linear Prog Solution***	***Comput. 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

In Table 2, we present the results obtained for Objective Function 1.2. This objective function minimizes the total number of tractors (i.e., vehicles) required to cover the entire set of transportation orders. As it is expected, Table 2 shows better solutions for the number of vehicles than Table 1. It makes sense to use Objective Function 1.2 reported in Table 2 instead of Objective Function 1.1 because the former reflects the transportation cost more precisely. Thus, the best solution we find from Tables 1-2 is about 39 tractors only (71 single hauls). At the aggregated level, we get an optimization around 34% when compared with the actual number of single hauls in use and 27% when compared with the actual number of rented trucks. Now the challenge is to assure this optimization by the next stage model when time windows and dock capacity constraints are included. From here, we set a value of 1% for our MIP solver optimality tolerance in order to identify a true near-optimal solution for each instance to test.

In order to build our model successively, we generate several instances using different values for the dock service capacity available at each transportation node. Thus, since we have an instance with less available docks for service, we generate a more difficult problem to solve and a longer solution time is expected. There are 34 different transportation nodes in our problem. The number of docks available for service at each node ranges from one to eight. In Table 3, we present the instances and some obtained results. The first 10 columns of the table contain the number of available docks at each node. “Total Docks” is the total number of docks in the entire instance considering all transportation nodes. The next columns are “# ITERs” and “# CUTs”, the former is the number of iterations and the latter is the number of constraints required for each instance to converge to a feasible near-optimal solution.

The columns “Max. Docks” and “Avg. Docks” refer to the maximum and average number of transportation orders in which the dock capacity is violated. This average number of orders is weighted by the length of computational time that the solver spends to solve the incumbent dock

Table 2. Solutions for the Relaxed C-VRP model. Objective function minimizes # of vehicles

(F1) # of Vehicles per Transp. Lane	(F2) # of Transp. Lanes per Vehicle	Binary Variables	Computational 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

constraints. The last column is a speed measure that indicates the quantity of dock constraints solved per second. Since we have a larger quantity of docks available for service, Table 3 verifies that the number of iterations and constraints required to be added at the cut generation stage is less. Less iterations and constraints to be added to the master problem mean less solution time to solve to problem. Furthermore, a bigger number of the transportation orders where the dock service is violated (i.e., the 15th column) means a larger quantity of cuts required to solve the problem. The average number of orders is not correlated with the number of available docks. In fact, this measure ranges from 2.4 up to 10.4 and is 5.1 on average. For example, 2.4 means that, for the most part of the time required to converge to a fully feasible solution, we have a cuasi-feasible solution with only 2.4 orders which does not have dock capacity available for service. Finally, concerning the efficiency indicator in the last column, as we have a more constrained instance (i.e., with less available docks), the quantity of the dock constraints solved per second is reduced.

In Table 4, we present the same set of instances as in Table 3 according to the total docks available for service. In this table, we focus on some activity measures for vehicle operation. The third column of Table 4 corresponds to the second part of the objective function presented in Equation 2.1. This is the total sum of the arrival times of all vehicles used to attend the entire set of orders. This indicator is very useful in order to estimate how much efficiency and time delay we have for the vehicles. As this instance is more constrained (i.e., with less available docks), we have a larger waiting time for the vehicles. The waiting time can be observed at the origin or at the destination node. Either way, such delay of a 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 4th column of Table 4 corresponds to the sum of all finish times of the vehicles in each instance. Thus, if we divide the sum of all finish times of the vehicles by the total number of the vehicles, we obtain the average finish time of the entire fleet, indicated in the 5th column. In the next column, we give the number of vehicles that are running on or after the 22nd hour. From the bottler's operation perspective, it is preferable that all waiting times of a vehicle take place at the end of the working day. Indeed, this strategy would allow the planning managers to have a more clear status of vehicle locations for the next operation

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

cycle. The last column of the table includes a measure of the length of time (a percentage) that a vehicle spends in waiting during its route and just before the last stop. It can be observed in Table 4 that as we have a more constrained instance (i.e., with less available docks), the values for all previously mentioned indicators are higher. Indeed, we obtain a big negative correlation coefficient of about 89% between the number of docks and the total sum of the vehicle arrival times. Similar correlation coefficients are obtained for the sum of the vehicle finish times and for the % of the waiting time of the vehicles.

6 Model Contributions and Applicability

The novelty of our model presented in this paper is the combination of three basic stages that interact in order to solve the PDP-TWDS effectively. The main contribution of our work is the development of dock service constraints. Our implementation is based on a cut generation strategy. The empirical results show the efficiency of these valid inequalities to constraint connected routes considering dock service constraints at each node. To the best of our knowledge, our

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

Total Docks	*Solution Time in sec.*	*Total Sum for Vehicle Time*	*Vehicle Sum End Times*	*Vehicle 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%

work is the first to implement these valid constraints. Our implementation indicates that the considered model provides an appropriate trade-off for the solution quality and computational time. The proposed model not only addresses the difficulties embedded in 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 such business issues like routing realignment. From a practical standpoint, the issue of routing realignment is how the model can efficiently accommodate such changes as transportation order additions or dropouts trying not to disrupt the previous design considerably. All these features are very important if we consider how easily this model can be extended to other cases.

It is important to point out that our methodology presents an HMIP model that ensures time window feasible solutions at each iteration. Thus, it is interesting to verify how rapidly our implementation can converge to cuasi-feasible solutions for dock service constraints (see Table 3). However, future research is necessary to prove viability of this paradigm when gap optimality is required to be confirmed. The obtained computational results serve only as evidence for our arguments. They are not intended to be used for in–depth comparison of available methods for PDP. From a practical business application standpoint, this operations research (OR) application is developed and implemented in order to optimize the transportation network between manufacturing plants and distribution centers. During the last years, the firm was interested in developing a better transportation and routing schedules. Indeed, this is the first OR application that has been implemented in a bottler company. It is important to point out that the overall results have been very positive. The first plans for transportation routes suggested by the optimization model were implemented eight months ago. Since then, a significant increase in productivity and direct savings to the firm has been achieved.

Some of the benefits the company obtained within this project: (1) the firm achieved an optimal truck capacity to attend the demand on each territory. This represents 27% of truck reduction. (2) An increase of effectiveness for the planning process of the transportation schedule. The fully manual planning process time was reduced from 6 hours to less than 20 minutes. (3) As a result of our continuous move model, the new routes are more efficient so the total travel time is decreased, thus improving the productivity of the truck drivers. The added throughput allows the firm to defer investments on trucks and hauls. Our model was able to optimize the number of hauls reducing the actual 120 available hauls to 71 only. In accordance with this productivity, the management decided to rationalize the number of available hauls in the firm. The save on investments for hauls was about 15% of the current fleet. Beside all these economic benefits, this new OR model allows the company to speed up others inventory optimization initiatives which are of special interest among *Coca Cola* bottlers. The proposed model approach can extend the basic problem to address different specific business rules or additional planning criteria. Nowadays, our model is being used by the firm to reach business solutions with significant benefits.

7 Conclusions

Many logistics problems found in the manufacturing and transportation industry can be modeled as a PDP-TWDS application. Along with the increasing fuel cost, companies seek to improve their transportation operations in order to tap the full potential of possible cost reduction. Transportation problems have been widely studied in the operations research literature. Still, there are some unstudied areas and sub-problems. Several different objectives and constraints in the transportation design process (i.e., continuous move strategy) are identified and discussed. In this paper, we consider a particular PDP-TWDS application that incorporates a diversity of practical complexities. Among those, we can mention a heterogeneous vehicle fleet with different travel times, travel costs and capacities, order/vehicle compatibility constraints, time window constraints, and different start and end locations for vehicles. Particularly, in 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 various OR algorithmic approaches that have been proposed, some are based on integer linear programming, others on classical heuristics, and more recent approaches are based on meta-heuristics. However, solving a real-world PDP poses 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. Furthermore, field people who are going to deploy the solution of our PDP application may have to pay more attention to feasibility of a solution in practice than to a pure optimal solution in terms of mathematics.

In this paper, a business application case at *Embotelladoras ARCA* is studied. With a real-world application from the service industry, we present a rich featured PDP-TWDS model. It is of interest to deal with large scale instances with a high presence of time window constraints. 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 meet 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 when required. We believe that this is an important contribution of our work. The obtained empirical results show the efficiency of these valid inequalities.

Computational results for a real-world instance of about 100 single-haul vehicles and 500 transportation orders are reported, showing that our model is suitable to provide good quality solutions. A Relaxed Capacitated Vehicle Routing Problem (C-VRP) model is used to find a solution at the aggregated level. At this stage, we relax time windows and dock service constraints. Instead, some side constraints at the aggregated level are included in order to assure feasibility for the original problem. With the solution obtained at the aggregated level, we reduce 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 rented trucks. The empirical results show that our simplification of the C-VRP model has no impact on the optimal solution found for the original problem at the PDP-TWDS stage when time windows and dock capacity constraints are fully included. Thus, optimization and economic benefits for the company are assured.

It is clear that optimal solutions for our tested instances are estimated using the gap to optimality information shown in Tables 1 and 2. This is a very pessimistic estimation because all these results correspond to the aggregated C-VRP model. Furthermore, when we apply the full PDP-TWDS model, the previous solutions are assured as seen in Table 3. In Table 2, a good quality solution reached (i.e., gap to optimality = 7.6%) in short computational time (total solution time ≤ 10 minutes) is presented. In general, it is difficult to compare the performance of methods. Obviously, the diversity of theoretical and practical problems is immense. Consequently, there are not too many papers devoted to the same problem. It is clear that future research should be done in order to test our method statistically. This issue will be covered in future work. However, the results obtained so far indicate that our model is robust to solve this hard problem, reaching good solutions in a short computational time.

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, assigning orders to vehicles in PDP-TW is much more difficult than in VRP-TW. In VRP-TW, all the origins of transportation orders are located at a depot. Therefore, transportation orders with geographically close destinations are likely to be served by the same vehicle. In 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 a need for improved efficiency, as the traffic volume increases much faster than the road network grows.

References

1. **Ascheuer, N., Fischetti, M., & Grotschel, M. (2001).** Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut. *Mathematical Programming, Series A,* 90(3), 475–506.
2. **Cook, W. & Rich, J.L. (1999).** *A parallel cutting-plane algorithm for the vehicle routing problem with time windows* (TR99-04). Houston, TX: Rice University, Computational and Applied Mathematics.
3. **Cordeau, J.F. (2006).** A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Operations Research*, 54(3), 573–586.
4. **Desrosiers, J., Dumas, Y., Solomon, M.M., & Sournis, F. (1995).** Time constrained routing and scheduling. *Handbooks in Operations Research and Management Science volume 8, Network Routing,* 35–139.
5. **Dumas, Y., Desrosiers, J., & Soumis, F. (1991).** The Pickup and delivery problem with time windows. *European Journal of Operational Research,* 54(1), 7–22.
6. **Dumas, Y., Desrosiers, J., & Solomon, M. (1995).** An optimal algorithm for the traveling salesman problem with Time Windows, *Operations Research,* 43(2), 367–371.
7. **Lu, Q. & Dessouky, M.M. (2006).** 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.
8. **Palmgren, M. (2001).** *Optimisation Methods for Log Truck Scheduling*, PhD Thesis, Linköpings Universitet, Linköping, Sweden.

9. **Parragh, S.N., Doerner, K.F., & Hartl, R.F. (2008).** A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations, *Journal für Betriebswirtschaft,* 58(2), 81–117.
10. **Ropke, S., Cordeau, J.F., & Laporte, G. (2006).** *Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows* Networks, 49(4), 258–272.
11. **Savelsbergh, M. & Sol, M. (1998).** DRIVE: Dynamic routing of independent vehicles. *Operations Research*, 46(4), 474-490.
12. **Solomon, M.M. (1984).** *On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints*, Boston, Massachusetts: College of Business Administration, Northeastern University.
13. **Tsitsiklis, J.N. (1992).** Special cases of traveling salesman and repairman problems with time windows, *NETWORKS,* 22(3), 263–282.

Fabián López-Pérez has been a Professor and Researcher at the Center of Business Development and Postgraduate Studies (CEDEEM) of Nuevo León Autonomous University (UANL) for 15 years. He is a member of the National System of Researchers (SNI), Level 1. In 2004, he was the first graduate from UANL within the doctoral program of Philosophy with specialization in Administration honored with "Summa Cum Laude". He has a M.Sc. degree in Administration and another M.Sc. degree in Industrial Engineering, both from the Monterrey Institute of Technology and Superior Studies (EGADE-ITESM). He also has a CPIM degree (Certified in Production and Inventory Management) from the American Production and Inventory Control Society (APICS). He has a B.Sc. degree in System Engineering, Control and Computation, Electronics and Communications from the Faculty of Mechanic and Electronic Engineering of UANL. He has more than 20 years of managerial experience in the areas of manufacturing, logistics, planning and systems in eight different industries. He is the author of various articles in indexed journals. His research interests include model development for business process optimization based on mixed integer programming and meta-heuristic search algorithms for combinatorial problems related to manufacturing, logistics planning, inventories, production and transportation programming, facility location, schedule assignment, territory districting, transportation and distribution vehicle routing.

Carlos Cantú-Aguillén is a Consultant for Innovation as well as a Coordinator of the academic program "The Sphere of Innovation" at the Monterrey Institute of Technology and Superior Studies (ITESM), Monterrey campus. He obtained his M.Sc. in Artificial Intelligence from ITESM in 1992, and his B.Sc. with honors in System Administration from the Nuevo León Autonomous University in 1988. He has developed more than 15 innovations for such companies as CYDSA, PEMEX, COPAMEX, among others. He was also an Assistant Coordinator of the academic program "The Sphere of Innovation" within which he has realized more than 60 projects for more than 35 organizations together with interdisciplinary high talented student teams of ITESM. His research interests include innovation, business, intelligent systems, artificial intelligence, analytic business intelligence, business intelligence, cellular automata, and the PVS NP-problem. He strongly believes that it is fundamental for organizations to create a healthy labor environment in order to realize high level innovation because only in this way the best ideas can appear and their authors can be recognized; it is also indispensable to apply the four principles of leadership defined by Jack Welch. He is recognized as a highly creative analytical expert in various areas. He is also interested in literature and writes about internal structures found in Scriptures dated more than three thousand years ago, the Book of Psalms.

Article received on 31/12/2010; accepted on 04/08/2011.