

A column generation-based algorithm for solving combined inventory and routing problems

Algoritmo basado en generación de columnas para resolver problemas combinados de ruteo e inventarios

Carlos Franco-Franco¹ Juan Carlos Figueroa-García²

Recibido 8 de enero de 2014, aceptado 29 de octubre de 2015

Received: January 8, 2014 Accepted: October 29, 2015

ABSTRACT

This paper presents a column generation algorithm for solving combined vehicle and inventory problems. This problem is based on the idea of coordinating customer inventory levels through a minimum routing cost. This is a combinatory decision problem since vehicle routing and inventory problems, are combined.

Using the column generation method, we can iteratively generate interesting routes to the system, based on their dual costs, this is routes that will improve the quality of the objective function because its reduced costs are negatives. The initial mixed integer problem has to be relaxed for getting its reduced costs. The sub problem is defined as the shortest path problem that returns a set of desirable routes. Finally, when the set of desirable routes is obtained, the mixed integer model should select a set of routes that fulfill both minimum shipping costs and the constraints of the system.

Keywords: Column generation, pricing, routing problem, inventory, shortest path problem, pulse algorithm.

RESUMEN

Este trabajo presenta un algoritmo basado en generación de columnas para la solución de problemas combinados de ruteo e inventarios. Este problema trata de coordinar los niveles de inventario del cliente mediante envíos a costo mínimo. Siendo este un problema de decisión combinatorio, ya que combina problemas de ruteo de vehículos (VRP) y problemas de inventario.

Utilizando el método de generación de columnas se pueden generar iterativamente rutas interesantes al sistema basadas en los costos duales, esto es rutas que mejoren la calidad de la función objetivo al presentar costos reducidos negativos. Para esto el problema entero mixto original se relaja para obtener los costos reducidos y se establece un subproblema encargado de generar las rutas. El subproblema se modela como un problema de ruta más corta. Finalmente cuando se tiene un conjunto de rutas atractivas para el modelo, el problema entero mixto es el encargado de seleccionar aquellas rutas que minimicen costos y satisfagan las restricciones establecidas.

Palabras clave: Generación de columnas, costeo ruteo de vehículos, inventarios, ruta más corta, algoritmo pulse.

¹ Universidad Distrital Francisco José de Caldas. Cra 7ª # 40-53, Piso 5°. Bogotá, Colombia. E-mail: cafra87@gmail.com

² Facultad de Ingeniería. Universidad Distrital Francisco José de Caldas. Cra 7ª # 40-53, Piso 5°. Bogotá, Colombia. E-mail: jcfigueroag@udistrital.edu.co

INTRODUCTION AND MOTIVATION

Nowadays many companies has to deal with the problem of coordinate inventory policies and transportation management. Due to this problem a variety of strategies are used for minimizing the logistics costs. One of these strategies is the Vendor Managed Inventory System (VMI) where the suppliers decide the quantity that will be sent to the customers, the period of time that will be made the delivery and how will be distributed the product. When these decisions are made by retailers the logistics cost are reduced and the customer can reduce the ordering cost.

The VMI can be model as an Inventory Routing Problem (IRP), which is an attractive problem due to its complexity and for the applicability that it has. In fact, this problem is NP-Hard because it is an extension of the vehicle routing problem (VRP) and involves inventory decisions.

The Inventory Routing Problem (IRP) was introduced by Federguen & Zipkin [1]. This problem can be seen as an extension of the classical Vehicle Routing Problem (VRP). The problem consists in designing a set of vehicle routes of minimal cost, starting and returning to the depot, while satisfying capacity constraints and customer requirements. Both types of decisions concerning inventory and routing are taken simultaneously.

The problem shifts responsibilities for customers and suppliers (Savelsbergh & Song [2]). Customers transfer the inventory control to the supplier. The suppliers monitor the inventory level and decide when to deliver, how much to deliver and how to deliver. This change has advantages like less resources required for inventory management for the customers, and the freedom for the supplier to decide the routes, when, how and how much to deliver. It is common to use a so-called order up to level policy or maximum level policy in the inventory routing problem.

According to order up to level policy introduce by Bertazzi, Paletta and Speranza [17], every time a customer is visited, the quantity delivered by the supplier is such that the maximum level of the inventory is reached at the customer. In the maximum level policy the quantity delivered to a customer is

the minimum between the maximum quantity that can be delivered without exceeding the maximum capacity of the customer, the residual capacity of the vehicle and the quantity available at the supplier.

The difference between these two policies are that in the order up to level policy when a customer is visited the inventory level reaches exactly its maximum level, meanwhile in the maximum level policy when a customer is visited the quantity delivered cannot exceed the maximum inventory level. In other words the maximum level policy is more flexible than order up to level policy. In case these policies are not used, it means that the quantity shipped to the customers is not restricted by any constraint. In other words the total amount of quantity is sent to the customers can take any value that do not exceed the capacity of the vehicle and do not exceed the maximum level of inventory of the customer.

This problem is not only attractive for its computational complexity but also for its industrial applicability. This model can be used in wholesales, where they have to decide the distribution of products to its stores. Another application is the distribution of drugs between centers of attention, we can also see this problem in distribution of blood between hospitals (Hemmelmayr, Doerner & Hartl [3]) or animals between farms (Aghezzaf, Raa & Landeghem [4]) or applications in the maritime transportation (Henrik [5]) or perishable products [13] or bulk gas distribution [14]. A huge number of distribution problems can be modeled as an Inventory Routing Problem.

There are several algorithms proposed to solve the problem like Savelsbergh & Song [2] who proposed an integer programming based optimization algorithm with local search procedures in order to improve solutions. Zhao, Wang & Lai [6] develop a tabu search in order to find the optimal regional partition wich indicates the customer that must be visited. A hybrid genetic algorithm for a multi product IRP was developed by Moin, Salhi & Aziz [7], Aghezzaf, Raa & Landeghem [4] proposed a nonlinear mixed integer formulation and solve the problem using a column generation method where the subproblems are solved using saving based approximation method. Coelho, Cordeau & Laporte [8] proposed a matheuristic with and

adaptative large neighborhood search for solving the problem. Solyali and Sural [15] developed a branch and cut algorithm for solving the problem using a tour based heuristic and Coelho, Cordeau and Laporte [16] developed and cutting plane algorithm for obtain the exact solutions of several instances. Also real applications were developed by Oppen, Lokketangen & Desrosiers [9] who model the problem of livestock collection as an IRP, where for solving the problem a column generation approach was proposed, also Hemmelmayr, Doerner & Hartl [3] model the problem of blood distribution as an IRP with uncertainty of the demands.

THE INVENTORY-ROUTING PROBLEM

Mathematical formulation

The Inventory-Routing problem can be defined as a mixed integer decision problem in the sense that the supplier needs to define inventory levels alongside with routing decisions.

The problem can be represented by a graph consisting of a set of nodes $N = \{0, \dots, n\}$, where the node 0 is the depot, and the subset $N' = N \setminus \{0\}$ represents the customers. At each discrete time over a finite horizon $t \in T = \{1, \dots, \tau\}$ a quantity r_t will become available at the depot and a quantity d_{it} is consumed by customer $i \in N'$ at each time $t \in T = \{1, \dots, \tau\}$. The inventory holding costs are h_i units for each node $i \in N$ per time period. The inventory level at node $i \in N$ at the beginning of time $t \in T' = T \cup \{T + 1\}$ is represented by $I_{(i,t)}$. There is an initial inventory level at the depot of B and the customers $i \in N'$ have also an initial inventory defined by S_i . For each customer $i \in N'$ there is a maximum and minimum level of inventory denoted by U_i and L_i respectively. The set $K = \{1, \dots, k\}$ defines available vehicles at the depot, each one of them has a capacity Q . The mathematical formulation is based on column generation where the set R contains all the possible routes, and each route $r \in R = \{1, \dots, \pi\}$, is associated to a cost C_r . There is a binary parameter a_{ir} that is 1 if and only if a customer $i \in N'$ is on the route $r \in R$, and 0 otherwise. The binary variable x_{rkt} is equal to 1 if the route $r \in R$ is used by vehicle $k \in K$ at time $r \in T$, and 0 otherwise. Besides we introduce a variable that models the quantity of product sent to customer $i \in N'$ using the vehicle $k \in K$ at time $t \in T$, as q_{ikt} .

The mathematical model can be described as follows:

$$\min \sum_{i \in N} \sum_{t \in T'} h_i I_{it} + \sum_{r \in R} \sum_{k \in K} \sum_{t \in T} C_r x_{rkt} \quad (1)$$

$$\sum_{r \in R} x_{rkt} \leq 1 \quad \forall k \in K, t \in T \quad (2)$$

$$\sum_{r \in R} \sum_{k \in K} a_{ir} x_{rkt} \leq 1 \quad \forall i \in N', t \in T \quad (3)$$

$$q_{ikt} \leq Q \sum_{r \in R} a_{ir} x_{rkt} \quad \forall k \in K, i \in N', t \in T \quad (4)$$

$$\sum_{i \in N'} q_{ikt} \leq Q \quad \forall k \in K, t \in T \quad (5)$$

$$I_{0t} = I_{0,t-1} + r_{t-1} - \sum_{i \in N} \sum_{k \in K} q_{ik,t-1} \quad \forall t \in T' \setminus \{1\} \quad (6)$$

$$I_{01} = B \quad (7)$$

$$I_{it} = I_{i,t-1} + \sum_{k \in K} q_{ik,t-1} - d_{i,t-1} \quad \forall i \in N', t \in T' \setminus \{1\} \quad (8)$$

$$I_{i1} = S_i \quad \forall i \in N' \quad (9)$$

$$I_{it} \geq L_i \quad \forall i \in N', t \in T' \quad (10)$$

$$I_{it} \leq U_i \quad \forall i \in N', t \in T' \quad (11)$$

$$I_{it} \geq 0 \quad \forall i \in N', t \in T' \quad (12)$$

$$q_{ikt} \geq 0 \quad \forall i \in N, k \in K, t \in T \quad (13)$$

$$x_{rkt} \in \{0, 1\} \quad \forall k \in K, r \in R, t \in T \quad (14)$$

The objective function (1) comprises the inventory and transportation cost, the objective is to minimize the total cost. Constraints (2) guarantee that each vehicle k can be used at most once for each time t . Constraint (3) shows that every customer can be visited at most once for each period. Constraints (4) represents that the maximum quantity shipped to each customer could be de capacity of each vehicle if the customer is on the route otherwise the quantity is zero and constraints (5) guarantee that the quantity shipped for each route cannot exceed the capacity of the vehicle. Constraint (6) ensures the inventory balance equation at the depot while constraint (7) assures the initial inventory level; similarly, constraint (8) represents the inventory balance equation at each customer where stock outs are not allowed, and constraint (9) guarantees the initial inventory level

per customer. Constraints (10) and (11) enforce the inventory level to stay between lower bound (10) and upper bound (11). Finally, constraints (12), (13) and (14) define the type of variables.

THE PROPOSED APPROACH

The proposed approach for solving this kind of problems is based on a merge-and-conquer strategy, commonly used in ordering algorithms. In our case, we divide the problem into two steps:

1. A strategy for finding feasible routes.
2. An optimization routine for finding the optimal routes and the optimal deliveries quantities.

Therefore, the main idea is to find a subset of the possible routes using the dual information by finding an optimal solution of each subproblem (feasible route). The subproblem is then formulated as a shortest path problem (SPP) where each node represents a customer, and the depot is represented by the initial and the final node.

Finally, the solution is provided by the best routes of the subproblems after solving all of them as SPPs.

We use the dual information to solve the subproblem by using a LP relaxation of (1) to (14). To do so, the PULSE algorithm (see next Section) allows us to solve the subproblem by generating more than one route per iteration. Once we complete a subset of feasible routes, the mixed integer problem is solved given an inventory policy (in this case, inventory levels).

Subproblem formulation

We will introduce the following notation obtained for the master problem; it is defined for a specific vehicle and day as follows:

- θ is the dual variable associated with the maximum uses of a vehicle, Constraints (2).
- π_i are the dual variables associated with the maximum visit of a customer, Constraints (3).
- φ_i are the dual variables associated with the quantity shipped, Constraints (4).

The goal of the subproblem is to find routes with negative reduce cost that can be added to the master problem. We model it as a network to solve it. It

can be defined by a graph consisting of a set of nodes $\mathcal{M} = \{0, \dots, N + 1\}$, where the node 0 and $N + 1$ correspond to the depot, and the subset $\mathcal{M}' = \mathcal{M} \setminus \{0, N + 1\}$ represents the customers. For every node $i \in \mathcal{M}$ the parameter b_i represents the transition demand, where $b_0 = 1$ is the demand of the depot, $b_i = 0$, $i \in \mathcal{M}'$ representing the customers demand, and $b_{N+1} = 1$ representing the depot. There is a cost C_{ij} associated with every arc $(i, j) \in \mathcal{M}$. The binary variable w_{ij} is equal to one if the arc $(i, j) \in \mathcal{M}$ is on the route and the binary variable y_i is equal to one if the customer $i \in \mathcal{M}'$ is visited. The variable u_i is used for subtour elimination and models the order in which a customer $i \in \mathcal{M}'$ is visited.

$$\min \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{M}} c_{ij} w_{ij} - \theta - \sum_{i \in \mathcal{M}'} \pi_i y_i + Q \sum_{i \in \mathcal{M}'} \varphi_i y_i \quad (15)$$

$$\sum_{j \in \mathcal{N}} w_{ij} - \sum_{j \in \mathcal{N}} w_{ji} = b_i \quad \forall i \in \mathcal{M} \quad (16)$$

$$\sum_{j \in \mathcal{N}} w_{ji} = y_i \quad \forall i \in \mathcal{M}' \quad (17)$$

$$u_i \geq 2 \quad \forall i \in \mathcal{M} \quad (18)$$

$$u_i \leq |\mathcal{M}| \quad \forall i \in \mathcal{M} \quad (19)$$

$$u_i - u_j + 1 \leq (|\mathcal{M}|)(1 - w_{ij}) \quad \forall i, j \in \mathcal{M} \quad (20)$$

$$w_{ij}, y_i \in \{0, 1\} \quad \forall i, j \in \mathcal{M} \quad (21)$$

$$u_i \geq 0 \quad \forall i \in \mathcal{M} \quad (22)$$

The objective function (15) comprises the routing cost and the dual information, the objective is to minimize the reduced costs. Constraints (16) represent the flow conservation constraints. Constraints (17) guarantee that if a customer i is visited then the variable y_i takes the value of one, and zero if the customer is not visited. Constraints (18), (19) and (20) are the Subtour elimination constraints. Finally Constraints (21) and (22) restrict the type of variables.

The dual information is obtained from solving the relaxed version of the master problem. With this information we solve the subproblem. The

subproblem then obtains routes that are added to the master problem.

The PULSE algorithm

The PULSE algorithm (Lozano & Medaglia, [10]), is an exact method for solving the constrained shortest path problem. The algorithm consists in send pulses through the network and use pruning strategies to avoid that the pulses continues propagating through the network. If the pulse reached the end of the network it contains all the information for a feasible path.

The pulse algorithm uses 4 types of pruning strategies: by cycles, by dominance, by infeasibility and by boundaries.

In the pruning by *cycles*, the algorithm uses a binary vector that keep the nodes already visited in the partial path, with this vector the pulses cannot go back to a node previously visited.

The second strategy used is pruning by *dominance*. When a pulse is propagating through the network it stores all the information about the resource consumption and the accumulative costs. When a new pulse is propagating it is compared with another path that has already propagated. If the new pulse is dominated by another path it is pruned and it cannot continue propagating.

A third method to prune is by *infeasibility*. When a pulse is propagating it is calculated the minimum resource consumption from the actual node to the final node, in order to calculate the minimum resource to reach the end of the network. If the sum of the partial consumption and the minimum resource consumption to the final node exceed the resource availability, then the pulse is pruned.

Finally, pruning by *boundaries* uses two strategies, pruning by primal boundary and pruning by primal and dual boundary. In the first case when a pulse reaches the final node of the network, the best total cost is updated. When a new pulse is exploring the network if the cost of the partial path exceed the best known solution it is pruned. In the primal and dual bound pruning strategy, the minimum cost to the final node is calculated and is pruned the pulse if the partial cost and the minimum cost to the final node is greater than the best known solution.

EXPERIMENTS AND RESULTS

The proposed algorithm is tested using the instances proposed by Archetti, Bertazzi, Laporte & Speranza [11], where we only consider one vehicle available at the depot. These instances are also used by Coelho, Cordeau, & Laporte [8] and Archetti, Bertazzi, Hertz & Speranza [12] who used metaheuristics for solving the problem, whose optimal solutions were computed in order to compare their results. Instances with a time horizon of 3 are denoted by $H3$ and instances with time horizon of 6 are referred to as $H6$. The amount of customers varies between 5 and 50, represented by $N = 5k$ where k varies between 1 and 10 if the time horizon is 3, and k varies between 1 and 6 if the time horizon is 6. The demand d_{it} for each customer $i \in N'$ at each period $t \in T$ is generated randomly, but it is assumed to be constant over the whole time horizon. Hence we will also refer to the customers demand per time period as d_i . There is a quantity r_t becoming available at the depot at each time $t \in T$, represented by $\sum_{i \in N} d_{it}$. There is a maximum inventory level for each customer $i \in N'$ represented by U_i , it is given by $d_i g_i$ where g_i is selected randomly from $\{2, 3\}$. The initial inventory level at the depot is $\sum_{i \in N} U_i$ and the initial inventory level for each customer $i \in N'$ is set as $U_i - d_i$. The instances are divided into two classes depending of their inventory holding cost: high cost and low cost. The inventory holding cost at the depot is $h_0 = 0.03$ when the inventory holding costs at the customers are low and $h_0 = 0.3$ when the inventory holding costs at the customers are high. The capacity for vehicles is given by $\frac{3}{2} \sum_{i \in N} d_i$. Finally the transportation cost $c_{ij} \forall (i, j) \in N$ is an Euclidean distance.

The methodology of the study

The study is performed through some experiments (different instances of well-known case studies), using the following key steps:

1. Vary the number of routes generated per iteration between [2-12]
 - a. Run column generation
 - b. Run PULSE algorithm
2. Set an appropriate number of routes provided in Step 1.
- c. Run SPP

3. Aggregate by number of vehicles
 - d. Obtain results for High and Low holding costs (for PULSE and SPP)
4. Compare PULSE against SPP

Firstly, the column generation algorithm is run alongside with the pulse algorithm, varying the number of routes generated per iteration. We have varied the amount of routes considered (between 2 and 12) per instance. After solving these problems, the average results for different classes of instances and results are aggregated for all instances with the same amount of vehicles. Once an appropriate number of routes to be generated per iteration are selected, two comparisons are performed: the first one compares the pulse algorithm to the approach that solves the subproblem as a shortest path problem exactly which generates only one route. The second comparison consists in contrasting our algorithm to the results published in the literature that consider a similar problem that uses an order up to level policy.

Once our algorithm is run, the results are summarized separately for instances of high and low inventory holding cost.

RESULTS

Using the pulse algorithm, we obtain better results in average when generating seven routes per iteration. The comparison between the pulse algorithm and the classic shortest path problem are provide in Table 1. The computational results are presented in Table 2 and 3, where only one vehicle is available at the depot. Each table provides the average results per class of instances, organized as follows: the first column presents the time periods and the type of cost (where H3 is 3 time periods an H6 is 6 time periods). Second column contains the amount of customers, and the results of the objective function obtained by the proposed algorithm are presented in the third column where $z(P7)$ indicates the objective function using the pulse algorithm with seven routes. Finally, the Gap (as percentage) and the Computational Time (in seconds) are presented in columns four and five respectively, the gap is obtained by comparing the best result report in the literature vs the solution obtained by our algorithm.

Remark 1. Table 3 compares the results of our proposal to the optimum obtained by Archetti,

Bertazzi, Laporte & Speranza [11], Coelho, Cordeau, & Laporte [8] and Archetti, Bertazzi, Hertz & Speranza [12] but relaxing the order up to level policy. This allows us to obtain a negative gap in most of experiments which means an improvement of the solutions. Also this kind of problems without this policy were not published so far, so our results are the first reported experiments in those conditions.

Note that the relaxation of the order up to policy allows us to find better results, so in practice, the use of the order up policy leads to higher transportation costs.

Table 1 shows that in overall instances the pulse algorithm is able to obtain better solutions that using a classical shortest path problem. Table 2 shows that our proposal is able to obtain the same solutions as reported in the literature in most of the instances. The obtained average gap is 0.04% on H3 low cost instances, 0.31% on H6 low cost instances, 0.39% on H3 high cost instances and 0.24% on H6 high cost instances. Our model is able to obtain high quality solutions on instances available in the literature. Our approach is effective and stable.

Table 3 shows the average results of the algorithm without order up policy, which obtains better solutions in most of the classes of instances than others reported in the literature. If the order up to level policy is relaxed, we can find better solutions using the PULSE algorithm.

Our proposal obtains better solutions in all the instances for H3 and H6. The average computational time of the algorithm is 376.11 seconds for H3 low cost instances and 1045.75 seconds for H3 high cost instances. For H6 low instances, our algorithm spent 800.72 seconds on average and 1249.98 seconds for the H6 high cost instances.

We improve solutions on average 3.35% on H3 low cost instances and 3.22% on H3 high cost instances. On H6 low cost instances we improve solutions on average 12.23% while on H6 high cost instances we obtain 25.40% on average. These improvements can be obtained because we relaxed the order up to level policy.

Table 1. Average results comparing a pure shortest path problem and the pulse algorithm.

	n	z (sp)	z (p)	Gap (%)
H3 Low cost	5	1529,24	1416,50	-7,37%
	10	2420,81	2028,77	-16,19%
	15	3016,98	2514,69	-16,65%
	20	3758,64	2870,76	-23,62%
	25	4500,76	3225,37	-28,34%
	30	5530,13	3546,02	-35,88%
	35	7026,36	3720,99	-47,04%
	40	8214,37	4001,93	-51,28%
	45	8711,60	4109,88	-52,82%
	50	11544,84	4808,19	-58,35%
Average		5625,37	3224,31	-33,76%
H6 Low cost	5	3124,41	2572,30	-17,67%
	10	4649,36	4269,81	-8,16%
	15	5564,97	4865,86	-12,56%
	20	6878,03	6139,21	-10,74%
	25	8295,41	6957,99	-16,12%
	30	9071,88	7041,86	-22,38%
Average		6264,01	5307,84	-14,61%
	n	z (sp)	z (p)	Gap (%)
H3 High cost	5	2320,57	2273,70	-2,02%
	10	4540,87	4252,73	-6,35%
	15	5699,73	5698,46	-0,02%
	20	7710,88	7493,00	-2,83%
	25	10647,60	9043,66	-15,06%
	30	13749,35	10878,49	-20,88%
	35	15304,90	11765,54	-23,13%
	40	18051,35	12850,79	-28,81%
	45	18092,30	13934,34	-22,98%
	50	19502,60	15256,75	-21,77%
Average		11562,02	9344,75	-14,38%
H6 High cost	5	5137,61	4790,30	-6,76%
	10	8405,39	6672,31	-20,62%
	15	11154,18	8412,01	-24,58%
	20	14301,74	10242,78	-28,38%
	25	17894,00	12532,19	-29,96%
	30	21784,22	14918,38	-31,52%
Average		13112,86	9594,66	-23,64%

Table 2. Average results with order up to level policy.

	n	z (P7)	Gap (%)	Time (s)
H3 Low cost	5	1418,76	0,00%	2,43
	10	2228,67	0,00%	82,32
	15	2493,47	0,00%	97,00
	20	3053,02	0,00%	194,78
	25	3451,15	0,00%	238,57
	30	3643,22	0,00%	537,50
	35	3846,87	0,00%	541,21
	40	4125,70	0,00%	957,51
	45	4270,61	0,00%	1323,93
	50	4829,33	0,38%	1247,32
H6 Low cost	5	3299,98	0,00%	1,42
	10	4832,89	0,00%	24,24
	15	5566,39	0,00%	1428,48
	20	6833,29	0,00%	1905,56
	25	7592,33	1,85%	875,27
	30	7847,39	0,00%	2196,43
H3 High cost	5	2354,18	0,00%	0,00
	10	4690,46	0,00%	6,51
	15	5736,91	0,00%	43,55
	20	7619,91	0,00%	1116,66
	25	9460,74	0,00%	1566,77
	30	11320,63	0,00%	2053,46
	35	12101,32	2,30%	2057,90
	40	13011,46	0,00%	2316,68
	45	14317,82	0,00%	2702,39
	50	16209,32	1,63%	2592,90
H6 High cost	5	5538,02	0,00%	3,31
	10	8872,41	0,00%	772,22
	15	11721,83	0,00%	1193,38
	20	14863,85	0,00%	2541,00
	25	17170,81	0,00%	2941,25
	30	20949,32	1,41%	3210,28

CONCLUDING REMARKS

Our proposal combines column generation with the pulse algorithm, decomposing the mathematical problem into a master problem and a sub-problem.

Table 3. Average results without order up to level policy.

	n	z (P7)	Gap (%)	Time(s)
H3 Low cost	5	1416,50	-0,16%	1,20
	10	2028,77	-8,97%	2,04
	15	2514,69	0,85%	5,34
	20	2870,76	-5,97%	159,00
	25	3225,37	-6,54%	178,03
	30	3546,02	-2,67%	430,00
	35	3720,99	-3,27%	413,14
	40	4001,93	-3,00%	638,34
	45	4109,88	-3,76%	894,55
	50	4808,19	-0,06%	1039,43
H6 Low cost	5	2572,30	-22,05%	0,00
	10	4269,81	-11,65%	17,56
	15	4865,86	-12,59%	971,75
	20	6139,21	-10,16%	1270,37
	25	6957,99	-6,66%	729,39
	30	7041,86	-10,26%	1815,23
	n	z (P7)	Gap (%)	Time(s)
H3 High cost	5	2273,70	-3,42%	0,00
	10	4252,73	-9,33%	4,43
	15	5698,46	-0,67%	31,33
	20	7493,00	-1,67%	749,44
	25	9043,66	-4,41%	1243,47
	30	10878,49	-3,91%	1387,47
	35	11765,54	-0,53%	1559,01
	40	12850,79	-1,23%	1643,04
	45	13934,34	-2,68%	1889,78
	50	15256,75	-4,34%	1949,55
H6 High cost	5	4790,30	-13,50%	2,49
	10	6672,31	-24,80%	589,48
	15	8412,01	-28,24%	932,33
	20	10242,78	-31,09%	1789,44
	25	12532,19	-27,01%	1987,33
	30	14918,38	-27,78%	2198,82

The master problem selects routes, the quantity to be delivered to the customers and the resulting inventory levels. The sub-problem generates attractive routes to be included into the master problem.

We have obtained better solutions in most of the instances because we do not use order up to level policy while other algorithms use this policy. We also can obtain the optimal solution in most of the instances even considering the order up to level policy.

The mathematical model and the presented solution approach are generic. This means that other constraints could be added to the master problem, and the proposed algorithm is able to solve the problem and test its flexibility.

Finally, the strategy of divide and merge properly works for solving the problem (for both the relaxed and original problem), since the shortest path problem is efficient when solving well defined instances which are provided by the combination of the column generation and PULSE algorithms. This allows us to find better results than other compact algorithms.

REFERENCES

- [1] A. Federguen and P. Zipkin. "A combined vehicle routing and inventory allocation model". *Operations Research*. Vol. 32 N° 5, pp. 1019-1035. 1983.
- [2] M. Savelsbergh and J.-H. Song. "An optimization algorithm for the inventory routing problem with continuous moves". *Computers & Operations Research*. Vol. 35, pp. 2266-2282. 2008.
- [3] V.C. Hemmelmayr, K.F. Doerner and R.F. Hartl. "A variable neighborhood search heuristic for periodic routing problem". *European Journal of Operational Research*. Vol. 95 N° 1, pp. 91-802. 2009.
- [4] E.-H. Aghezzaf, B. Raa and H.V. Landeghem. "Modeling inventory routing problems in supply chains of high consumption products". *European Journal of Operational Research*. Vol. 169, pp. 1048-1063. 2006.
- [5] A. Henrik. "Industrial aspects and literature survey: Combined inventory management and routing". *Computers & Operations Research*. Vol. 37, pp. 1515-1536. 2010.
- [6] Q.-H. Zhao, S.-Y. Wang and K. Lai. "A partition approach to the inventory/routing problem". *European Journal of Operational Research*. Vol. 177, pp. 786-802. 2007.

- [7] N. Moin, S. Salhi and N. Aziz. "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem". *International Journal in Production Economics*. Vol. 133, pp. 334-343. 2011.
- [8] L.C. Coelho, J.-F. Cordeau and G. Laporte. "Consistency in multi-vehicle inventory-routing". *Transportation Research*. Vol. 24 (Part C), pp. 270-287. 2012.
- [9] J. Oppen, A. Lokketangen and J. Desrosiers. "Solving a rich vehicle routing and inventory problem using column generation". *Computers & Operations Research*. Vol. 37, pp. 1308-1317. 2010.
- [10] L. Lozano and A.L. Medaglia. "On an exact method for the constrained shortest path problem". *Computers & Operations Research*. Vol. 40, pp. 378-384. 2013.
- [11] C. Archetti, L. Bertazzi, G. Laporte and M.G. Speranza. "A branch-and-cut algorithm for a vendor-managed inventory-routing problem". *Transportation Sciences*. Vol. 41 N° 3, pp. 382-391. 2007.
- [12] C. Archetti, L. Bertazzi, A. Hertz and M.G. Speranza. "A hybrid heuristic for an inventory routing problem". *INFORMS Journal on Computing*. Vol. 24 N° 1, pp. 101-116. 2011.
- [13] S. Mehmet, M. Jacqueline, Bloemhof-Ruwaard, H. Rene and G.A.J. Jack Van der Vorst. "Modeling an Inventory Routing Problem for perishable products with environmental considerations and demand uncertainty". *International Journal of Production Economics*. Vol. 164, pp. 118-133. 2015.
- [14] D. Hugues, D. Pierre, N. Nicoleta and Y. Thomas. "A GRASP for real life inventory routing problem: application to bulk gas distribution". 9th International Conference on Modelling, Optimization & Simulation. Bordeaux, France. June, 2012.
- [15] O. Solyali and H. Sural. "A branch and cut algorithm using a strong formulation and a priori tour based heuristic for an inventory-routing problem". *Transportation Science*. Vol. 45, pp. 335-345. 2011.
- [16] L.C. Coelho, J.-F. Cordeau and G. Laporte. "The exact solution of several classes of inventory-routing problems". *Computers & Operations Research*. Vol. 40, pp. 558-565. 2012.
- [17] L. Bertazzi, G. Paletta and M.G. Speranza. "Deterministic order-up-to level policies in an inventory routing problem". *Transportation Science*. Vol. 36, pp. 119-132. 2002.