8th International Conference of Modeling and Simulation - MOSIM'10 - May 10-12, 2010 - Hammamet - Tunisia
"Evaluation and optimization of innovative production systems of goods and services"

AN EXACT ALGORITHM FOR THE SINGLE-VEHICLE CYCLIC INVENTORY ROUTING PROBLEM

El-Houssaine AGHEZZAF[1], Yiqing ZHONG[1], Birger RAA[2], Manel MATEO[3]

[1]Department of Industrial Management, Faculty of Engineering, Ghent University, Belgium
[2]Departement of Information & Operations Management Faculty of Economics, Ghent University, Belgium
[3]Department of Management, Technological University of Catalonia, UPC, Spain

ABSTRACT: *The single-vehicle cyclic inventory routing problem ($SV-CIRP$) consists of a repetitive distribution of a product from a single depot to a selected subset of customers. For each customer that is selected for replenishments, the supplier collects a corresponding fixed reward. The objective is to determine the subset of customers to replenish, the quantity of the product to be delivered to each, and to design the vehicle route so that the resulting profit (difference between the total reward and the total logistical cost) is maximized while preventing stockouts at each of the selected customers. In this paper, the $SV-CIRP$ is formulated as a mixed-integer program with a nonlinear objective function. After an efficient analysis of the problem, an exact algorithm for its solution is proposed. This exact algorithm requires only solutions of linear mixed-integer programs. Values of an insertion-based heuristic for this problem are compared to the optimal values obtained for a set of some test problems. In general the gap may get as large as 25%, which justifies the effort to continue exploring and developing exact and approximation algorithms for the $SV-CIRP$.*

KEYWORDS: *Inventory-Routing, Nonlinear Mixed Integer Programming, Exact Algorithms.*

1 INTRODUCTION

The single-vehicle cyclic inventory routing problem ($SV-CIRP$) is an optimization problem underlying the vendor managed inventory policy when demand rates are stable. Under this policy, the supplier is granted full authority to manage inventories at his customers. It is thus the supplier who decides when and how much to deliver to each one of his customers, making sure that they never run out of stock. The purpose of the $SV-CIRP$ is to design an optimal cyclic distribution strategy in which the inventory at each one of the selected customers is replenished on a cyclical basis. The quantity of the product to be delivered to each customer and the vehicle route must be determined so that the resulting logistical costs are minimized, while preventing stockouts at each one of the selected customers.

The single-vehicle cyclic inventory routing problem arises naturally as a subproblem, when branch-and-price or column generation based approaches are used to solve the cyclic inventory routing problem ($CIRP$) (see (Aghezzaf et al., 2006) and (Raa and Aghezzaf, 2009)). The $CIRP$ differs from the $SV-CIRP$ in that a fleet of vehicle is available and each customer must be assigned to one of the vehicle. The non cyclical version of the $SV-CIRP$ appears also as a daily subproblem in the general inventory routing problem (IRP) (see for example (Anily and Federgruen, 1991), (Bard et al., 1998), and (Dror and Ball, 1987)).

This paper is organized as follows. In section 2, the single-vehicle cyclic inventory routing problem is formulated as mixed-integer program with a nonlinear objective function. In section 3, an exact algorithm for its solution is proposed. In section 4, an efficient heuristic is presented and its results on some problems found in the literature are compared with the results of the exact algorithm on the same problems. The gaps and the computational times of both approaches are also reported. Finally, some concluding remarks on possible approximation algorithms are given in section 5.

2 SINGLE-VEHICLE CYCLIC INVENTORY ROUTING PROBLEM (SV-CIRP)

As already mentioned above, the single-vehicle cyclic inventory routing problem ($SV-CIRP$), discussed in this paper, consists of a single distribution center r using a single vehicle to distribute a single product to a selected subset of customers from a set of potential customers S. If a customer is selected by the supplier, this later collects the corresponding reward. It is assumed that customer-demand rates and travel times are stable over time. Thus, the objective of the $SV-CIRP$ is to select a subset of customers $C \subset S$ to be replenished, in a repetitive manner, by the single available vehicle and to design a vehicle route that maximizes the difference between the total reward and the total logistical cost (total average distribution and inventory costs), without causing any stockout at any of the selected customers during the planning horizon.

In (Aghezzaf et al., 2006), a heuristic algorithm based on column generation is employed to solve the complete cyclic inventory routing problem ($CIRP$). The $SV - CIRP$ arises as a sub-problem in the column generation process. This $SV - CIRP$ is solved in (Aghezzaf et al., 2006) using a combined insertion and savings-based heuristic. This paper is mainly interested in developing an exact algorithm for the solution of the mixed integer nonlinear formulation of this sub-problem. As a first step in this process, the formulation of the $SV - CIRP$ presented below removes the nonlinearity in the constraints of the original sub-problem formulation given in (Aghezzaf et al., 2006). However the objective function is inherently nonlinear and remains nonlinear in the new proposed formulation. The following paragraphs describe the necessary parameters and variables to reformulate the $SV - CIRP$:

- Parameters of the model:

  t_{ij}: Travel time from customer $i \in S^+ = S \cup \{r\}$ to customer $j \in S^+$ (in hours); it is assumed that the necessary loading and unloading times of the vehicle are included in these travel times;

  d_j: Demand rate at customer j (in ton per hour); it is assumed to be stable (constant average with a small standard deviation);

  ψ: Fixed operating cost of vehicle (in euro per vehicle);

  δ: Travel cost of the vehicle (in euro per km);

  ν: Vehicle speed (in km per hour); an average speed is assumed throughout the trip;

  φ_j: Fixed ordering and delivery cost at customer j (in euro per order or per cycle);

  η_j: Holding cost at customer j (in euro per ton per hour);

  λ_j: Reward (dual price) if customer j is selected for replenishment; (these prices are usually obtained from the master problem in the column generation framework);

  κ_j: Holding capacity at customer j (in ton);

  κ^v: Capacity of the vehicle (in ton);

- Variables of the model:

  x_{ij}: A binary variable set to 1 if customer $j \in S^+$ is replenished immediately after customer $i \in S^+$ by the vehicle, and 0 otherwise;

  Q_{ij}: Quantity of the product remaining in the vehicle when it travels to customer $j \in S^+$ immediately after it has replenished customer $i \in S^+$; this quantity equals zero if the link (i, j) is not on the vehicle's trip;

  q_j: Quantity of the product delivered to customer $j \in S$ in each cycle;

  T: The cycle time of the trip made by the vehicle (in hours).

The nonlinear mixed integer formulation for $SV - CIRP$:

Minimize

$$RC(T) = \sum_{i \in S^+} \sum_{j \in S^+} \left((\delta \nu t_{ij} + \varphi_j) \frac{1}{T} + \frac{1}{2} \eta_j d_j T \right) x_{ij} - \sum_{i \in S^+} \sum_{j \in S^+} \lambda_j x_{ij} + \psi$$

Subject to:

$$\sum_{i \in S^+} x_{ij} \leq 1, \text{ for all } j \in S, \quad (1)$$

$$\sum_{i \in S^+} x_{ij} - \sum_{k \in S^+} x_{jk} = 0, \text{ for all } j \in S^+, \quad (2)$$

$$\sum_{i \in S^+} \sum_{j \in S^+} t_{ij} x_{ij} - T \leq 0, \quad (3)$$

$$\sum_{i \in S^+} Q_{ij} - \sum_{k \in S^+} Q_{jk} = q_j, \text{ for all } j \in S, \quad (4)$$

$$0 \leq d_j \cdot T - q_j \leq Q_j^{max} \cdot \left(1 - \sum_{i \in S^+} x_{ij} \right), \text{ for all } j \in S, \quad (5)$$

$$Q_{ij} \leq \kappa^v \cdot x_{ij} \text{ for all } i, j \in S^+, \quad (6)$$

$$q_j \leq \kappa^j \sum_{i \in S^+} x_{ij}, \text{ for all } j \in S, \quad (7)$$

$$x_{ij} \in \{0, 1\}, Q_{ij} \geq 0, q_j \geq 0, T \geq 0, \text{ for all } i, j \in S^+$$

The first part of the objective function provides, for each visited customer, the total transportation and total average inventory costs. The inventory cost component has the same form as the EOQ-based inventory cost function.

Constraints (1) indicate that each customer is visited in at most one of the tours made by the vehicle. Constraints (2) are the usual flow conservation constraints assuring that if the vehicle arrives at a customer, it must leave after it served this customer to a next customer or to depot. Constraints (3) guarantee that the total travel time of the vehicle is smaller than the cycle time T, so that the vehicle can return to each visited customer before the end of the cycle. Constraints (4) and (5) assure that each visited customer acquires a quantity q_j that is sufficient to cover its demand during the cycle T. In constraints (5), Q_j^{max} is given by $Q_j^{max} = d_j \cdot T_L$, where T_L is the largest possible value for the cycle time. These constraints indicate that the quantity delivered to customer j in a cycle should be greater than its demand during the cycle time. Constraints (6) guarantee that the quantity carried by a vehicle doesn't exceed the vehicle's maximum capacity. Constraints (7) guarantee that the quantity delivered to each customer doesn't exceed the customer's maximum holding capacity.

Notice that the problem is presented in a minimization form. When $SV - CIRP$ appears as a subproblem in a column generation process, the objective function RC is actually the reduced cost of the to be generated column.

3 AN EXACT ALGORITHM FOR THE SV-CIRP

In order to provide a complete description of the exact algorithm, this section reviews some important concepts already introduced in (Aghezzaf et al., 2006) such as minimal and maximal cycle times. An example is also given to show how the optimal cost as a function of the cycle time behaves. An exact algorithm for the solution of the $SV - IRP$ is then described.

3.1 Review of some important concepts

Consider a vehicle replenishing a set of customers C. Assume that the vehicle makes a trip visiting these customers in a set of consecutive disjoint tours P. Assume also that each tour $p \in P$ goes through a subset $S_p \cup \{r\}$ of customers such that $C = \bigcup_{p \in P} S_p$ and $S_p \cap S_q = \emptyset$ for any p and $q \neq p$ in P. Under this pattern, the most effective way to supply customers in C is to travel along the traveling salesman tour in each subset $S_p \cup \{r\}$. Let $T_{TSP}(S_p \cup \{r\})$ denotes the travel time of each TSP tour on $S_p \cup \{r\}$. Now, if we define the time between two consecutive iterations of the trip as the 'cycle time' and we denote it by $T(C)$. Clearly, this cycle time is bounded from below by the sum of the TSP-tours' travel times through the subsets $S_p \cup \{r\}$ for $p \in P$. This lower bound is called the 'minimal cycle time' and is denoted by $T_{min}(C)$. It is given by:

$$T_{min}(C) = \sum_{p \in P} T_{TSP}(S_p \cup \{r\}) \tag{8}$$

There is also an upper bound on the cycle time $T(C)$. It is called 'maximal cycle time' and is denoted by $T_{max}(C)$. This upper bound results from the limited capacity of the vehicle, and is given by:

$$T_{max}(C) = min_{p \in P} \left\{ \frac{\kappa}{\sum_{j \in S_p} d_j} \right\} \tag{9}$$

where d_j is the demand rate of customer $j \in S_p$ and κ is the capacity of the vehicle. For a trip going through customers in $C = \bigcup_{p \in P} S_p$ to be feasible, it is necessary that $T_{min}(C)$ be smaller or equal to $T_{max}(S_p) = \kappa / \sum_{j \in S_p} d_j$ for each $p \in P$. This upper bound results from the fact that the vehicle cannot carry more than its capacity in every tour that it makes.

Finally, there is a theoretical optimal cycle time which can be obtained, as explained in (Aghezzaf et al., 2006), as the value of T that minimizes the cost function $RC(T)$ for a given trip C:

$$T_{EOQ}(C) = \left(\frac{\delta \nu T_{min}(C) + \sum_{j \in C} \varphi_j}{\sum_{j \in C} \eta_j d_j / 2} \right)^{1/2} \tag{10}$$

This is an extension of the EOQ-formula and is denoted by the 'EOQ cycle time' $T_{EOQ}(C)$. This last value may turn out to be greater than the maximal cycle time or smaller than the minimal cycle time. In these cases, the actual optimal cycle time $T^*(C)$ is equal to the maximal cycle time or minimal cycle time respectively. For a more detailed discussion of these concepts we refer the reader to (Aghezzaf et al., 2006) or (Raa and Aghezzaf, 2009).

3.2 Illustrative examples of the objective value $RC(T)$

A small example with two customers is build and the value of each feasible solution as a function of the cycle time T is graphically depicted. The following two graph (fig. 4 and fig. 6) show the behavior of the objective function $RC(T)$ as a function of T. The solid line shows de epigraph giving for a fixed value of T the corresponding optimal value of the problem $SV - CIRP$. As one can see this curve is nonconvex, nondifferentiable, and there are some subintervals in the domain of T in which there is no feasible solution.

Figure 2 shows the parameters for a two-clients problem. In order to analyze the behavior of the objective

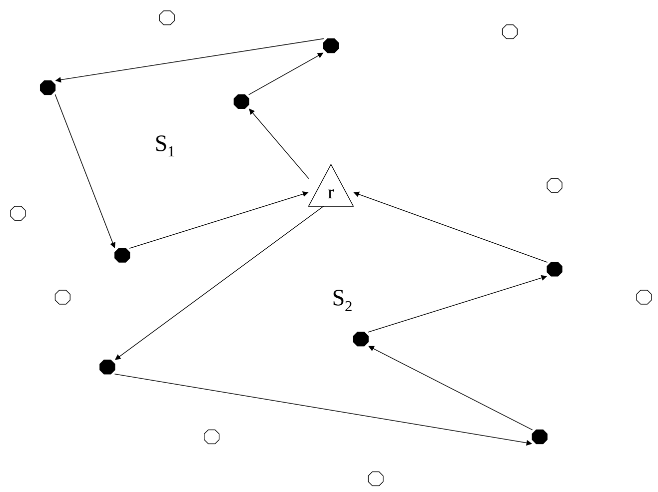

Figure 1: S is the set of all nodes, C is the set of filled nodes and the vehicle's trip is made out of two tours, going through the disjoint subsets S_1 and S_2.

Clients	1	2	
Reward:	250	200	
Fixed Ordering Cost:	100	5	
Holding cost:	1	4	
Vehicle			
Travel cost/km:	10		
Speed in km/h:	90		
Vehicle Capacity:	90		
Links	(0,1)	(0,2)	(1,2)
Travel Time:	6	4	3

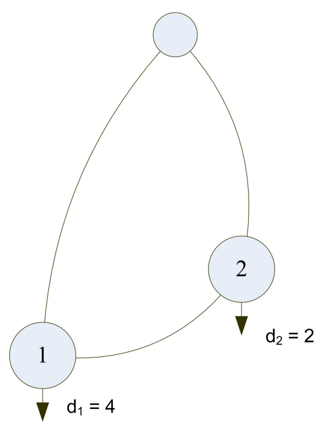

Figure 2: Data for a two-clients $SV-CIRP$

function $RC(T)$, the four possible solutions (shown in the figure 3 below) are generated and the corresponding cost function of each is determined as a function of T. Figure 4 shows the values (y-axis) of these four possible solutions as a function of T (x-axis). The feasible domain $[T^l(s_i), T^u(s_i)]$ for each solution s_i is also shown.

Figure 5 shows another set of values for the problem parameters and figure 6 shows the values (y-axis) of the above four possible solutions as a function of T (x-axis). This example shows that there are interval in which no feasible solution exists.

3.3 An exact Algorithm

As one can observe from the above example (fig. 2 and fig. 4), the $SV-CIRP$ is inherently a nonlinear and complex problem. We managed however to develop an exact algorithm to solve the problem to optimality, using linear-mixed integer programming methods. The basic structure of the algorithm consists of two major steps; the first solves the resulting linear mixed integer program for a fixed cycle time and a second renews the cycle time making sure that no possible optimal solution is overlooked.

More specifically, reconsider again the set of all potential customers S. The domain for the cycle time variable T is determined by the interval $[T_s, T_l]$, where T_s is the smallest value assumed by T and given by $T_s = \kappa / \sum_{j \in S} d_j$; and T_l is the largest value assumed

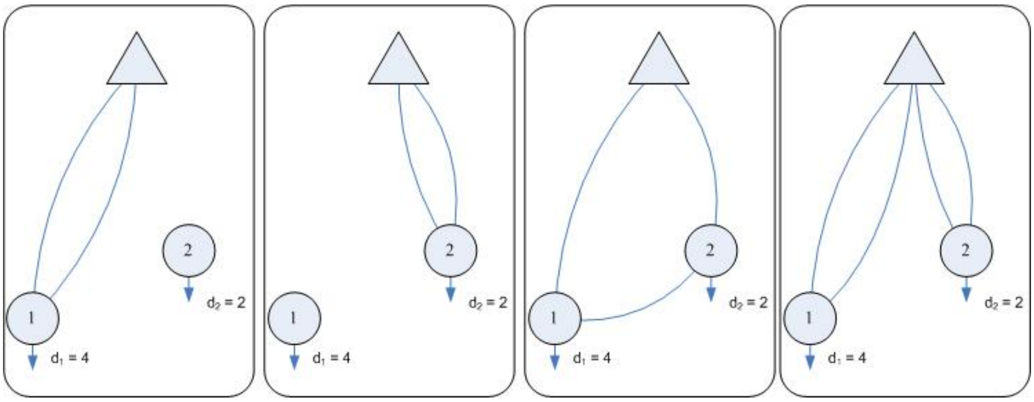

Figure 3: The four possible feasible solutions for the $SV - CIRP$

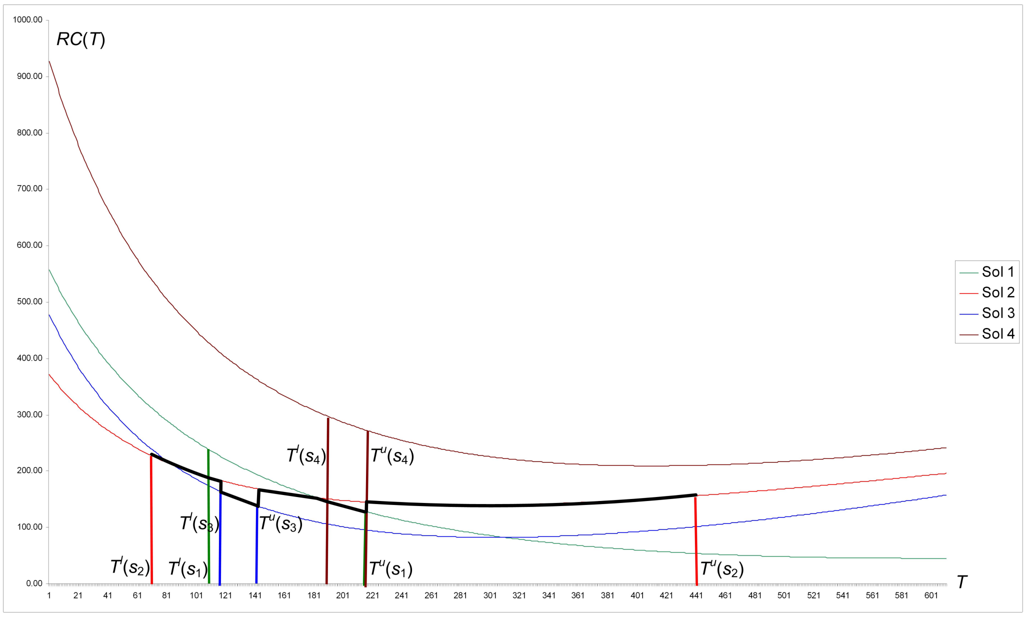

Figure 4: $RC(T)$ is a nonconvex nondifferentiable function of the cycle time

Clients	1	2	
Reward:	250	200	
Fixed Ordering Cost:	100	5	
Holding cost:	1	4	
Vehicle			
Travel cost/km:	10		
Speed in km/h:	90		
Vehicle Capacity:	90		
Links	(0,1)	(0,2)	(1,2)
Travel Time:	6	10	3

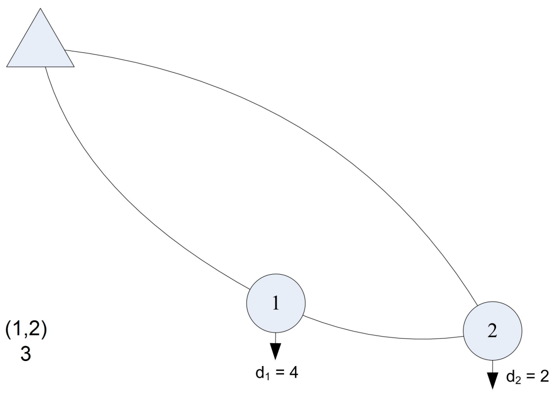

Figure 5: Data for a second two-clients $SV - CIRP$ example

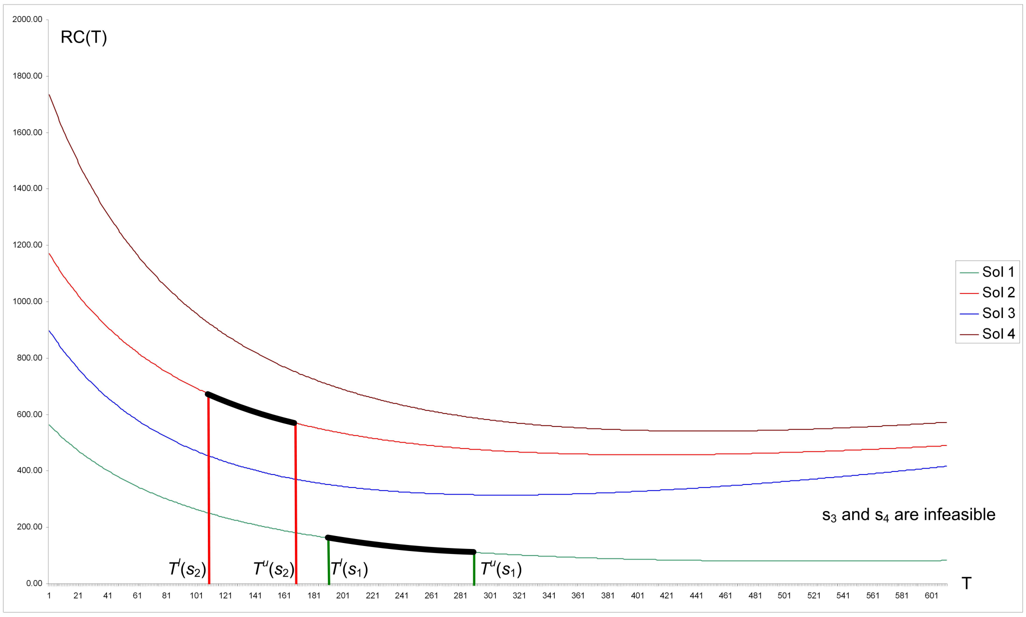

Figure 6: Some subintervals in the domain of T contain no feasible trips

by T and given by $T_l = \kappa/\mathrm{Min}_{j\in S}\{d_j\}$. Within the domain of T, a starting value T_k $(k = 0)$ (*i.e.* normally a small value of T for which a feasible solution x exists) is selected. The resulting linear mixed integer problem $SV - CIRP(T_k)$, obtained from the problem $SV - CIRP$ by fixing the variable $T = T_k$, is solved to determine its optimal solution x_k. The value of T_k is then updated by increasing it as follows:

$$T_{k+1} = T_{max}^{x_k} + \delta_k \tag{11}$$

where δ_k is selected in each iteration so as to avoid overlooking any possible optimal solution and jumping over subintervals of $[T_s, T_l]$ in which no feasible solution x exists. A summary of the proposed exact algorithm is described below. In the description of the algorithm, we let x^*_{opt} and v^*_{opt} denote, respectively, the optimal solution and its value.

The algorithm determines the optimal solution of the $SV - CIRP$ by investigating all possible values of the cycle time T, for which a feasible trip exists. The way the step δ_k is determined is crucial for the correctness of the algorithm. Solving the problem $Step_k$, guarantees that none of the feasible solutions of $SV - CIRP$ is overlooked. It allows also to jump over the subintervals in $[T_s, T_l]$ in which no feasible trip exists.

To make sure no feasible solution is overlooked during the search progress, the value of ϵ must be carefully chosen. One possible way, is to decrease the value of ϵ by a half until the same value for T^* is obtained in two consecutive solutions of the problem $Step_k$. Choosing this last value of T^*, guarantees that no feasible solution is overlooked. Clearly the algorithm, given in this form, requires a huge amount of computational times. However, it can be a source of many approximation algorithms which can be used to solve the problem. Some specific values of T, carefully chosen, may be used to determine good approximate solutions to the $SV - CIRP$.

4 COMPARISON WITH A HEURISTIC APPROACH

In this section we outline some efficient variants of an insertion heuristic that is used to solve the problem. We then provide a table for the results of both the exact algorithm and the heuristic obtained for some problems found in the literature. The table provides the gaps as well as computational times of both approaches.

4.1 Heuristic solution approach

The proposed heuristic approaches, based on the heuristic described in (Aghezzaf et al., 2006), are some variants of a basic insertion heuristic. In the first variant, customers are inserted into a solution one by one, according to a given order. In the second variant, the customer selected for insertion is the one that leads to the greatest reduction in the reduced cost rate of the solution. Both variants are followed by a local-search improvement phase. The pseudo-code is given below.

When inserting a customer into a solution, the following alternatives are evaluated and the best one selected: (a) inserting the customer in a new, separate tour; and (b) inserting the customer into one of the already existing tours. Option (b) is often the best alternative, but repeatedly selecting option (b)

		Exact Algorithm		Heuritics Algorithm		
Problems	Clients	Optimal Value	CPU Time (s)	Best Value	CPU Time (s)	Gap
S 10-0	10	-2053.540	1329.750	-2053.540	0.386	0.000
S 10-1	10	-1948.190	3061.580	-1856.520	0.463	4.938
S 10-2	10	-1865.840	1637.270	-1646.490	0.422	13.322
S 10-3	10	-2304.000	3098.980	-2304.000	0.458	0.000
S 10-4	10	-1961.820	2947.120	-1911.250	0.386	2.646
S 10-5	10	-1809.950	1586.550	-1558.530	0.404	16.132
S 10-6	10	-2171.020	2339.440	-2070.410	0.417	4.859
S 10-7	10	-1890.720	2420.390	-1890.720	0.386	0.000
S 10-8	10	-1592.620	6140.270	-1592.620	0.468	0.000
S 10-9	10	-2025.650	4232.480	-1807.350	0.421	12.078
A 15-0	15	-328.510	116.094	-307.519	0.638	6.826
A 15-1	15	-295.206	308.391	-278.636	0.717	5.947
A 15-2	15	-277.382	173.906	-250.847	0.735	10.578
A 15-3	15	-386.908	811.516	-386.908	0.677	0.000
A 15-4	15	-360.925	363.953	-360.925	0.694	0.000
A 15-5	15	-339.352	723.781	-291.448	0.682	16.437
A 15-6	15	-399.845	271.094	-319.990	0.709	24.955
A 15-7	15	-347.068	290.531	-340.270	0.708	1.998
A 15-8	15	-380.829	573.703	-332.613	0.642	14.496
A 15-9	15	-308.583	165.812	-277.040	0.709	11.386

Table 1: Computational results of some problems cases

rapidly decreases the maximal cycle time of the multi-tour, making it harder to insert other customers afterwards. Therefore, this insertion procedure is sometimes adjusted such that when option (a) is feasible, option (b) is no longer evaluated. Thus, the maximal cycle time does not decrease that rapidly, and more customers (bearing more profit) can eventually be inserted. In the improvement phase, option (b) of the insertion procedure is never discarded because it then no longer makes sense to limit the decrease of the maximal cycle time. The following is the pseudo-code for the proposed heuristics

- Construction phase:

    1. Customers are sorted according to their priority (given by the customer reward).
    2. Initialize the solution with a single-customer tour to the customer with highest priority.
    3. V1: Insert the next-priority customer into the solution. If this decreases the reduced cost, keep this new solution. Otherwise, go back to the previous solution (without the new customer).

       V2: For all remaining customers, create a new solution by inserting the customer into the existing solution. Keep the solution with the smallest reduced cost rate.
    4. Repeat Step 3 as long as the reduced cost rate is decreased and as long as there are remaining customers.

- Improvement phase:

    1. Remove the next-priority customer from the solution and then reinsert it into the solution. (If the customer ends up in a different position after the re-insertion, the solution has been improved. If not, the solution is restored.)
    2. Do the re-insertion for all customers.
    3. Restart the improvement phase as long as improvements are being found.

These variants of the two-phase (construction + improvement) heuristic are embedded in a multi-start framework. In each iteration, both variants are applied and the best solution is kept. To obtain different solutions in different iterations, new customer priorities are generated for each iteration. In the first iteration, priorities are given by the rewards (as in the pseudo-code), but in subsequent iterations, prioritizing is done by dividing the costs of the last found solution over all customers in that solution. However, if in a certain iteration a solution is found that is the same as a solution from an earlier iteration, priorities are randomly generated. Otherwise, the same sequence of solutions would be regenerated over and

Step 0. {Initialization}

Set $k = 0$ and choose $T_k \in [T_s, T_l]$ to be a smallest value for which a feasible solution exists. If the linear mixed-integer problem obtained from $SV-CIRP$ by fixing T and setting it to T_s has a feasible solution, then let $T_k = T_s$ and start the solution procedure.

Step 1. { Solving for the linear problem for T_k}

Solve the linear mixed-integer program $SV-CIRP(T_k)$:

$$\text{minimize } \{RC(T_k) : \text{ subject to } (1)-(7)\}$$

obtained from $SV-CIRP$ by fixing T and setting it to T_k and where the linear objective function $RC(T_k)$ is given by:

$$RC(T_k) = \sum_{i \in S^+} \sum_{j \in S^+} \left((\delta \nu t_{ij} + \varphi_j) \frac{1}{T_k} + \frac{1}{2} \eta_j d_j T_k \right) x_{ij} - \sum_{i \in S^+} \sum_{j \in S^+} \lambda_j x_{ij} + \psi$$

For the obtained optimal solution x_k compute $T^{x_k}_{min}$, $T^{x_k}_{max}$, and $T^{x_k}_{EOQ}$ the solution's corresponding minimal, maximal, and EOQ cycle times respectively. If $T^{x_k}_{min} \leq T^{x_k}_{EOQ} \leq T^{x_k}_{max}$, let $v_{opt} = RC(T^{x_k}_{EOQ})$, if $T^{x_k}_{EOQ} < T^{x_k}_{min}$, let $v_{opt} = RC(T^{x_k}_{min})$, and if $T^{x_k}_{max} < T^{x_k}_{EOQ}$, let $v_{opt} = RC(T^{x_k}_{max})$. Finally, if $v_{opt} < v^*_{opt}$ then let $v^*_{opt} := v_{opt}$ and $x^*_{opt} := x_k$.

Step 2. { Computing the step δ_k}

Solve the problem $Step_k$:

$$\text{minimize } \{T : \text{ subject to } T_k + \epsilon \leq T, (1)-(7)\}$$

where ϵ is a positive smallest value selected in function of the elementary unit of time that is used. If, for example, data is given in hours and the elementary unit of time is one minute, then ϵ may be set to 0.01. If T^* is the optimal solution of the problem $Step_k$, then let $\delta_k = T^* - T_k$ and $T_k := T_k + \delta_k$.

Step 3. { Stopping rule }

If $(T_k > T_l)$ stop, otherwise go to step 1.

over. The adjustment of the insertion procedure (discarding option (b) when option (a) is feasible) is only used in the even iterations of the multi-start.

4.2 Computational results

To compare the exact and heuristic solution approaches, both algorithms were applied to some test instances, of small size, from (Sindhuchao et al., 2005) (denoted by S 10-x) and from (Aghezzaf et al., 2006) (denoted by A 15-x). The obtained results are displayed in Table 1.

Table 1 shows that the average gap falls round 8% and it can be as large as 25% in some cases. This justifies the fact that exact and approximate algorithms are worth investigating, in particular when the $SV-CIRP$ appears as a sub problem in a column generation process. Of course in terms of computational times we cannot compete with well constructed heuristics. However, in many real world applications the computational time may not be an issue, especially when the proposed solution is to be implemented for few months. This is the case for the CIRP, where the solution is implemented and revised each month and sometime for even each three months.

5 CONCLUDING REMARKS

This paper discusses the particular single-vehicle cyclic inventory routing problem $SV-CIRP$. The main objective of this analysis is to investigate the model's properties and involvedness that must be tackled if one wants to solve the problem it to optimality. A "steepest-descend" like exact algorithm is proposed for the solution of the problem. The striking and encouraging result of this analysis is the fact that there are many cases in which even a well and carefully programmed heuristic provides results that are as much as 25% worst than the optimal results. This means that exact and approximate algorithms are worth investigating, in particular when the SV-CIRP appears as a subproblem in a column generation procedure used, for example, to solve $CIRP$. The less positive outcome is that the current exact algorithm requires a huge amount of computational time, and needs to be accelerated. The discussion in this paper is meant to be a starting point for a thorough investigation of this problem since it appears as a subproblem in a very large number of logistical problems. This issue is the objective of our current research.

REFERENCES

Aghezzaf, E.H., B. Raa, and H. Van Landeghem, 2006. Modeling Inventory Routing Problem in Supply Chain of High Consumption Products.

European Journal of Operation Research, Vol. 169, p. 1048-1063.

Anily, S. and A. Federgruen, 1991. Rejoinder to one warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 37(11), p. 1497-1499.

Bard, J., L. Huang, P. Jaillet, and M. Dror, 1998. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32, p. 189-203.

Campbell, A. M., and M.W.P. Savelsbergh, 2004. A Decomposition Approach for the Inventory-Routing Problem. *Transportation Science*, Vol. 38, No. 4, p. 488-502.

Dror, M., and M. Ball, 1987. Inventory-routing: Reduction from an annual to a short period problem. *Naval Research Logistics Quarterly*, 34(6), p. 891-905.

Raa B., and E.H. Aghezzaf, 2009. A practical solution approach for the cyclic inventory routing problem. *European Journal of Operation Research*, 192, Issue 2, p. 429-441.

Sindhuchao, S., H.E. Romeijn, E. Akcali, and R. Boondiskulchok, 2005. An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. *Journal of Global Optimization*, Vol. 32, p. 93-118.