A Prize Collecting problem applied to a Real Milk Collection problem in Chile

Elizabeth Montero*, Dario Canales†, Germán Paredes-Belmar‡ and Raúl Soto§

*,‡ Facultad de Ingeniería, Universidad Andres Bello, Viña del Mar, Chile
{elizabeth.montero, german.paredes}@unab.cl

† Departamento de Informática, Universidad Técnica Federico Santa María, Valparaíso, Chile
dario.canales@usm.cl

§ Departamento de Gobierno y Empresa, Universidad de Los Lagos, Puerto Montt, Chile
raul.soto@ulagos.cl

Abstract—In this work, a real world milk collection problem is introduced. A milk processing company, located in south of Chile, needs to collect milk to elaborate their products. To this work, the company has a heterogeneous truck fleet. The farms are scattered in a large geographical area. In this problem, the company buys only the minimum quantity of milk required to its daily operation. Exceeding this minimum quantity of milk generates a cost of over demand. The objective is to find efficient collection routes, satisfying the minimum required quantity and minimizing the over demand and transportation costs. To solve this version of the prize collection problem we propose an integer programming model to solve simpler instances and a GRASP metaheuristic to solve more complex instances in reduced time. Real world problem instances can consider up to 500 farmers. We tested our approaches using small real world cases and possible expansion scenarios. We concluded about the key components of our approaches and their capabilities to solve the problem at hand.

Keywords-Milk collection; prize collecting; vehicle routing problem

I. INTRODUCTION

Logistics plays an important role in the milk supply chain. Its main concern is to decrease the transportation costs and improve the milk collection and distribution process. The Food and Agriculture Organization (FAO) indicates that the transportation costs of milk represents more than 30% of the total milk cost [9].

In this article we address the milk collection process. The milk processing plants are usually located in rural areas, and the milk farmers are scattered in a large geographical area. This is the case of milk plants in south of Chile where the road network is composed mainly by rural roads and unique accesses to farms. The farmers can be associated to a milk cooperative or sell their production to several plants.

Some processing plants –especially the small ones– have a small truck fleet to buy and transport milk according to their daily operation aiming to minimize the transportation costs. In this work we study a milk collection problem for a milk company based on this business model. In this case, the processing plant has a set of suppliers for its daily requirements.

To this task, an efficient set of routes should to be determined to collect the milk. The vehicles start and finish their collection sequences in the precessing plant. No all available farms are collected. Note that the problem is similar to the *Prize-Collecting Vehicle Routing Problem, PCVRP* proposed by [1], but in this case the selection of farmers generates two types of costs: transportation and over demand of milk. A literature review is presented in section II.

The total costs are based on the traveled distances and the loaded volume of milk in each visit. The Figure 1 shows a solution scheme of the problem. In this example, the plant P (diamond) requires 300 liters of milk. The 6 farms/farmers are represented with circles, and their productions are the numbers near to the circles. Using two trucks with a capacity of 200 and 150 liters, two routes can be obtained. The first vehicle (continuous lines) collects the milk from farms 1 and 3, it starts in farm 1 and then visits farm 3. The second truck (dashed lines) collects the milk from farms 5 and 6. It first visits farm 5 and then farm 6. Farms 2 and 4 were no visited in this solution. This scheme prefers the big and near farms around the processing plant.
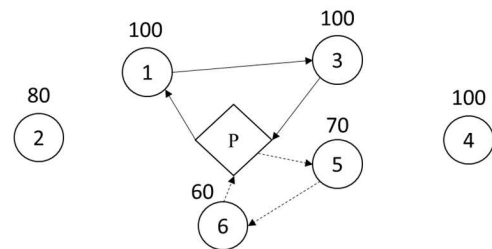

Fig. 1. Solution scheme of the problem

In this work we propose a linear integer programming model to represent all requirements of the problem and solve simple test cases using CPLEX. Problem model is presented in section III.

Moreover, in order to solve more complex instances, that can consider up to 500 farms, we study a metaheuristic approach that allows to find quality solutions in reduced amounts of time. Our GRASP approach is described in section IV.

The experimental setup and our results are are shown in section V. Our final conclusions and future work are presented in section VI.

II. LITERATURE REVIEW

The literature has a broad range of studies related to the PCVRP, where the transportation costs and non-visited client penalizations are both minimized. In this case, also, a minimum profit is required (as a constraint) [1]. However, there is no developed application in milk collection problems that consider the over demand and transportation costs.

Authors in [8] developed a Lagrangian heuristic to solve the PCVRP up to 500 nodes. In [15] the PCVRP was studied considering multiple depots and non-linear costs. In the milk collection problem context, the work presented in [4] develops a collection system based on the periodic vehicle routing problem (PVRP). The authors present an integer programming model, minimizing the deviations (surpluses or deficits) of the collection process.

In [2] the authors study the collection of four different qualities of milk using a truck fleet with compartments. They develop a local search heuristic of two stages: (1) the farms are assigned to vehicle compartments, and (2), the routing problem is solved minimizing the distance traveled by the vehicles. In [10] a milk collection problem is solved using a tabu search heuristic. In [5] and [6] a set of milk collection problems are solved in Canada, using different formulations and solution methods. In [14] authors study a version of the problem where the milk is collected from a set of collection centers. They use trucks with compartments to transport different milk types. In [12] authors proposed a milk collection problem with different milk qualities, allowing the blending of small quantities of milk, whenever this is convenient in terms of the global objective.

Authors in [13] extend the work proposed in [12] adding collection points in the collection sequences, to accumulate milk and reduce the transportation costs. They solve an instance of 500 farms using the ant colony approach, combined with a set-partitioning formulation. Recently, in [3] authors developed a decision support tool to collect sheep milk in Italy. They improve the current routes (in terms of costs and CO_2 emissions) using an ant colony metaheuristic. The results are displayed in a geographical positioning system (GIS).

In this work we propose an integer programing model and a simple heuristic approach to solve the milk collection problem from South of Chile collecting plants. Real world problem instances are used in our experiments in order to evaluate their advantages depending on the features of the problem being solved.

III. THE LINEAR INTEGER PROGRAMMING MODEL

Be $G(N_0, A)$ a complete graph. N_0 is the set of nodes of the network (including the processing plant 0 and farms); A is the arc set, where $A = \{i \in N_0, j \in N_0 : i \neq j\}$. N is the set

of farms; C is the transportation cost per kilometer, d_{ij} is the distance (in kilometers) over the arc $(i, j) \in A$. L_i is the milk cost per liter of the farm $i \in N$ and q_i is the milk quantity produced by farm $i \in N$. Q_k corresponds to the capacity of each truck $k \in K$. Finally, P is the minimum amount of milk required by the processing plant.

The variable $x_{ij}^k = 1$ if the truck k travels from node i to node j; $x_{ij}^k = 0$ otherwise. $y_i^k = 1$ if the vehicle k loads milk from farm i; $y_i^k = 0$ otherwise. T_j^k accounts the milk load from the farm j using vehicle k.

The objective function minimizes both the transportation and milk demand costs.

$$\text{Min} \quad C \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ij}^k + \sum_{k \in K} \sum_{i \in N} L_i q_i y_i^k \quad (1)$$

According to constraints in equation (2) not all farms need to be visited.

$$\sum_{k \in K} y_i^k \leq 1 \quad \forall i \in N \quad (2)$$

Also, if the decision is to collect the milk from a farm, then all the production is loaded, i.e. there is no fractional collection. This constraint is established in order to protect producers to ensure they sell all their production.

Constraints in equation (3) control the compatibility between the loading from farms and routes of the vehicles.

$$\sum_{h \in N_0 : (h,i) \in A} x_{hi}^k = y_i^k \quad \forall i \in N, k \in K \quad (3)$$

Constraints in equation (4) assure the total utilization of the truck fleet. This constraint can be relaxed when the total capacity of trucks is much higher than the plant requirement.

$$\sum_{j \in N : (0,j) \in A} x_{0j}^k = 1 \quad \forall k \in K \quad (4)$$

Constraints in equation (5) control the flow balance to each node and trucks.

$$\sum_{i \in N_0 : (i,j) \in A} x_{ij}^k = \sum_{h \in N_0 : (j,h) \in A} x_{jh}^k \quad \forall k \in K, j \in N_0 \quad (5)$$

Constraints in equation (6) control the satisfaction of the daily minimum requirement of the plant.

$$\sum_{i \in N} \sum_{k \in K} q_i y_i^k \geq P \quad (6)$$

Constraints in equation (7) control the capacity of trucks.

$$\sum_{i \in N} q_i y_i^k \leq Q^k \quad \forall k \in K \quad (7)$$

Constraints in equation (8) (also known as the *Miller-Tucker-Zemlin constraints*) preclude the apparition of subtours in routes. Here, the amount of milk in each step is used to sequence properly the routes.

$$T_j^k \geq T_i^k + q_i - Q^k(1 - x_{ij}^k)$$
$$\forall k \in K, i \in N, j \in N_0 : (i,j) \in A; \quad (8)$$

Finally, constraints in equations (9) and (10) control the domain of decision variables.

$$x_{ij}^k, y_i^k \in \{0,1\} \quad \forall (i,j) \in A, k \in K \tag{9}$$

$$T_j^k \geq 0 \quad \forall j \in N_0, k \in K \tag{10}$$

IV. GRASP APPROACH

In this section we present a Greedy Randomized Adaptive Search Process (GRASP) based approach that includes three main steps: a preprocessing, a construction and a postprocessing phase. The main idea of our approach is to obtain a good quality feasible solution during the construction step that will be then improved using four movements during the postprocessing phase.

The construction phase selects a tentative set of production farms to be considered in the current solution. These solutions will be then improved using a repair movement, two intensification movements and one diversification movement. These movements are explained in detail below.

Our approach works always with feasible solutions, hence, the construction procedure and the movements generate always feasible solutions that satisfies both, the plant demand and the trucks capacity. First we present the main components of our approach: the representation and evaluation function. Then we detail the procedures performed during the preprocessing, constructive and postprocessing phases.

A. Representation

We represent each solution considering a set of routes, each one is a list of integer values, each value representing the identifier of a visited farm. Fig. 2 represents a solution where Truck 1 visits farmer 1 and farmer 3 while truck 2 visits farmer 5 and then farmer 6.

$Truck_1$	$\to 1$	$\to 3$
$Truck_2$	$\to 5$	$\to 6$

Fig. 2. Solution representation

B. Evaluation function

We consider as evaluation function the minimization of transportation and milk costs as shown in equation 11.

$$\text{Min } C \sum_{k \in K} Transportation(k) + \sum_{k \in K} L_i \sum_{i \in N} Milk(k,i) \tag{11}$$

Transportation costs correspond to kilometers traveled by the trucks in solution multiplied by the cost by kilometer (C). Milk costs correspond to the liters of milk collected by trucks multiplied by the cost per liter of each farmer (L_i).

C. Preprocessing phase

During preprocessing step distances and compatibility between farmers are calculated. The compatibility between farmers are calculated based on their production and the available trucks capacity. Two farmers are compatible if the sum of their productions is lower or equal than the capacity of any truck in the fleet. This, because large differences between farmers productions exist.

D. Construction phase

We use a pseudo-greedy algorithm to construct feasible solutions. The objective of the greedy algorithm is to select the next visit satisfying the capacity constraints of trucks and the processing plant demand. To determine the order in which visits are selected during the construction phase we consider two heuristic criteria:

- **PD**: The pending demand if the node is incorporated to the solution. The difference between the unsatisfied demand of the current solution and the production of the farmer to include is used here. Hence at the beginning of construction process, nodes with the highest production will be preferred because they satisfy better the demand requirements. At the end of construction process farmers with productions best suited to the requirements will be preferred.

- **PDbyD**: The multiplication of the pending demand (PD) and the distance to farmer node. This, in order to consider not only how well the demand is satisfied by the farmer but also how far is he located.

Next node to visit is uniformly randomly selected from the best $\alpha\%$ possible nodes, with $0 < \alpha \leq 100.0$. Visits to farmers are allocated until the processing plant demand is satisfied.

E. Postprocessing phase

During our postprocessing phase we use four movements:

- Repair – **node deletion** performs the best improving node deletion. The idea here is to reduce the over demand that could have been induced during the construction step. Moreover, it can determine to remove any node, even when it could have been included in early steps by the construction method. This, because, the movement has a global view of a complete solution. The focus of this operator is the selection of farmers. The movement is repeated until no possible node deletions are detected.

- Intensification – **intra swap** performs the best improving consecutive swap in a randomly selected route. The focus of this process is to find the best path for routes. Improvements in the visits order can have an important effect on the selection of remaining farmers. Swaps are performed only if they produce improvements on the quality of the current solution.

- Intensification/Diversification – **external swap** performs the best improving swap between a selected node in

current solution and any other external node. External nodes are those that have not been visited in current solution. This operator is focused on changing the set of farmers selected. Swaps are performed only if they improve the quality of current solution.

- Diversification – **node adding** adds the node that generates the lower quality deterioration in current solution. For this, it adds to a truck with available capacity the farmer that produces the lower increase of cost in terms of both, transportation and milk production. The focus of this operator is to change the set of nodes selected. The objective of this movement is to incorporate diversification during the search process.

Algorithm 1 shows the pseudocode of our GRASP approach. At the beginning it performs the preprocessing phase. Then, at each iteration, a new solution is constructed and submitted to an iterative combination of repair, intensification and diversification steps. Each movement checks the quality of solutions and updates $x*$ when high quality solutions are found.

Algorithm 1: GRASP algorithm

Input: α, hc: heuristic criterion;
Output: $x*$: Best solution found;

1 PREPROCESSING: Construct the distance matrices and filter compatible farmers
2 **while** *maximum time not reached* **do**
3   CONSTRUCTION: Generate solution x using heuristic criterion hc and α;
4   POSTPROCESSING: Improve x;
5   **while** *stopping criterion is not satisfied* **do**
6   x = NodeDeletion();
7   x = IntraSwap();
8   x = ExternalSwap();
9   x = AddNode();
10   **end**
11   **if** $cost(x) < cost(x*)$ **then**
12         $x* = x$
13   **end**
14 **end**
15 **return** $x*$;

It is important to notice that, from the number of available trucks, our GRASP approach can generate solutions with at most these number of routes. Construction and postprocessing steps allow GRASP to adaptively change the number of routes as the search process progresses. This feature, useful during the search process, does not produce final solutions that do not use all available trucks.

V. EXPERIMENTS AND RESULTS

In this section we evaluate our approach using some case studies obtained from a milk processing plant located in south of Chile. Our main objectives here are:

- Evaluate the performance of our GRASP approach analyzing the relevance of their main components.

- Compare the performance of the solutions obtained by our integer programming model formulation and our GRASP approach.

A. Problem Instances

In our case study, we have considered a processing plant with a heterogeneous truck fleet. The capacity of these trucks varies between $14,000$ and $30,000$ liters. The processor plant has 40 available farmers producing all the same quality of milk, hence the milk cost per liter will be the same for all farmers. The milk production of farms varies between 400 and $20,000$ liters. Fig. 3 shows the location of milk suppliers of the company. As can be observed they are scattered around a large geographical area.
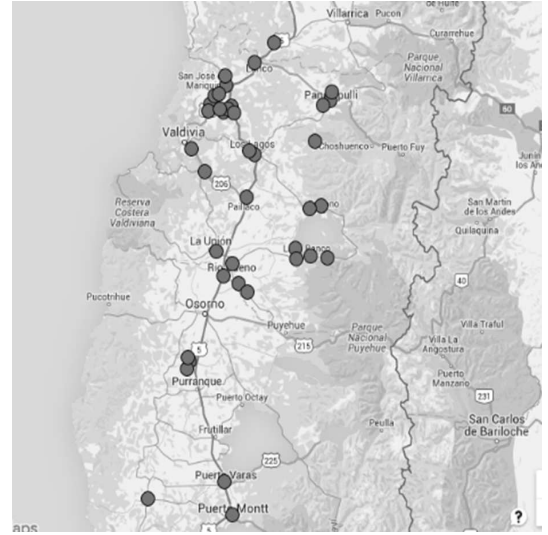
Fig. 3.   Network of milk production in south of Chile

Currently, the plant collects $35,000$ liters daily. The processor plant wants to expand its operation, collecting between $60,000$ and $100,000$ liters.

From this, we have designed 7 additional case studies. For these additional case studies we have increased the processor plant demand and the number and capacity of trucks. All the case studies features are listed in table I. In our experiments we set the distance cost to \$220 chilean peso per kilometer and the milk cost was set to the same value for all farmers (\$1.99 chilean peso per liter).

Our GRASP algorithm was implement in C++ and its source code is available in our website[1]. The experiments were executed on a Power Edge R630 server with 2 Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 64 GB of RAM under Ubuntu x64 16.10 distribution. Problem instances and raw data of our experiments are also available in our website.

In this section we analyze the performance results obtained from our experiments. The performance in each case is measured as the total cost of the best solution found.

---

[1]https://labcomp.cl/~emontero/

TABLE I
FEATURES OF CASE STUDIES

Case	# of Trucks	Trucks capacity	Demand
1	3	22,14,14	35,000
2	3	30,30,30	60,000
3	4	30,30,30,15	100,000
4	4	20,20,20,20	60,000
5	5	15,15,15,15,15	60,000
6	5	30,30,15,15,15	100,000
7	6	30,15,15,15,15,15	100,000
8	7	22,22,14,14,14,14,14	100,000

B. MIP results

The MIP model was coded in AMPL and solved using CPLEX 12.7. The time limit was set to 1 hour.

1) Base Case: Here, we compare the results obtained for the current base case with the manually determined solution (Case 1 in table I). Note that current manual routes are determined using intuitive criteria, based on the dispatcher experience. The current manual costs are indicated in Table II, using one vehicle of 22,000 liters and two vehicles of 14,000 liters. Clearly, the MIP solution improves the costs of current manual solution in 33%, mainly because of the reduction on the transportation costs.

TABLE II
MANUAL AND MIP SOLUTION

Feature	Manual	MIP
Transportation costs [$]	96960	42460
Milk volume costs [$]	68539	68611
Total costs [$]	165499	111071
Collected Milk (CM) [l]	34,472	34,478
Integrality GAP (IG) [%]	-	0
Routes	35-31-30	30-32-3-2-31
	28-26-25-24-23	29
	39	28-27

For each case study we show the transport costs (TRC), milk volume costs (MVC) and total costs (TC). Costs are shown in Chilean peso. If a problem can not be solved optimally during the fixed time limit, the integrality GAP is also reported.

The results of table III indicate that as the minimum required milk increases, the total costs also do. The lowest cost solution when collecting 60,000 liters was found using 3 vehicles of 30,000 liters. The setting that fits better to the minimum amount required of 100,000 liters was found using 3 vehicles of 30,000 liters and one of 15,000 liters. Clearly, the inclusion of vehicles of higher capacity, helps to reduce the transportation costs. Moreover, from table III we observe that as the problem size grows the integrality GAP also grows, making more difficult to find quality solutions.

C. GRASP results

Here, we study the relevance of our main GRASP components. For this, we evaluated the performance obtained by our approach in the 8 case studies described in section V. To analyze the relevance of each movement we executed the GRASP approach removing the component studied in each scenario. Hence we studied 5 different scenarios:

- **ALL:** All the movements detailed in section IV were used.
- **NoRepair (NR):** Repair movement – node deletion – was removed from the GRASP approach. This implies no reductions on the number of farmers of initial solutions.
- **NoIntraSwap (NIS):** Intra swap movement was removed from the GRASP approach. This implies that routes can not be reordered.
- **NoExternalSwap (NES):** External swap movement was removed from the GRASP approach. In this cases the set of nodes can not be changed.
- **NoDiversification (ND):** Diversification movement –node adding – was removed from the GRASP approach. This implies no increment on the number of farmers of initial solutions.

Fig. 4 and 5 show the boxplots of the performance obtained by all the case studies in each of these 5 scenarios using the two heuristic criteria in the construction phase. A boxplot is a simple way of representing statistical data on a plot in which a rectangle represents the second and third quartiles, a line inside to indicate the median value, lower and upper quartiles are shown as vertical lines at the borders of boxes and outliers are drawn as dots outside the boxes.

Performance is always measured as the total cost of the best solution found. White boxes show the performance obtained using the pending demand (PD) criterion while gray bars use the pending demand and distance product (PDbyD) during the construction phase. Each pair of boxes correspond to one of the 5 scenarios studied: ALL, NRepair (NR), NoIntraSwap (NIS), NoExternalSwap (NES) or NoDiversification (ND).

From these graphs we can clearly notice that the performance of both construction heuristics PD and PDbyD look similar. Table IV shows, for each pair of boxes, the result of their Wilcoxon test comparisons. In this table only gray cells indicate a statistical difference in performance. Statistical differences were tested with a 95% confidence level. From table IV we can conclude that, in most cases, there is no difference on the performance of our approach when varying the construction heuristic. The only two test cases where statistical differences were found are case 2 and 6, and in both of them, the best heuristic was pending demand and distance factor (PDbyD).

For case 1 the selection criterion becomes important for NoR and NoIS scenarios. In these cases the pending demand and distance factor (PDbyD) performs better. In most cases (4-8) differences were found for NoES and NoAdd scenarios. In all cases the best heuristic was the pending demand (PD). External swap and node adding movements are in charge of adding diversity to solutions because they both add new

TABLE III
MIP RESULTS

Case	TRC [$]	MVC [$]	Total costs [$]	IG [%]
1	42460	68611	111071	-
2	71940	119684	191624	10.60
3	159280	200303	359583	19.09
4	93940	120148	214088	2.90
5	123420	119543	242963	5.73
6	234520	201236	435756	32.60
7	240680	199616	440296	30.07
8	254320	199119	453439	31.60

(a) Case 1

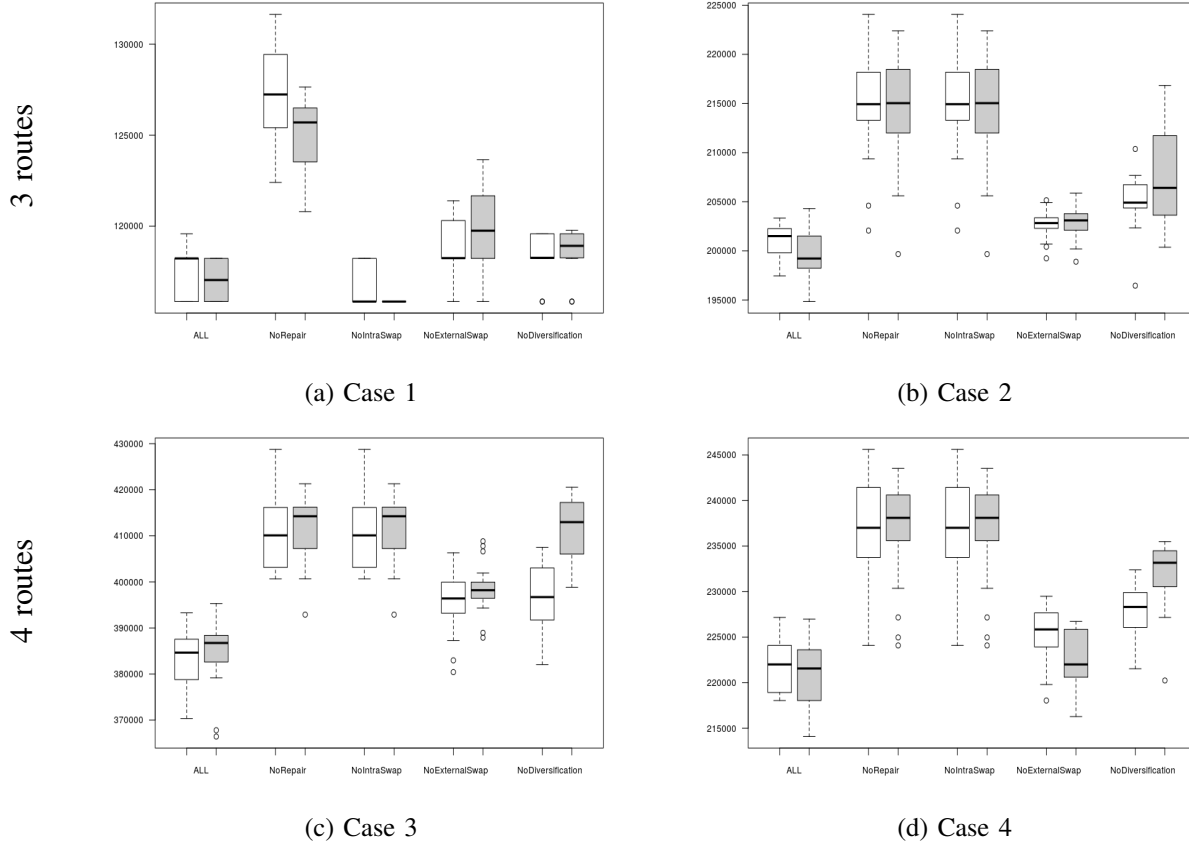(b) Case 2

(c) Case 3

(d) Case 4

Fig. 4. Cases 1 - 4: Boxplots of Total cost obtained by the 5 scenarios studied. White boxes show the performance obtained using the pending demand (PD) heuristic criterion while gray bars use the pending demand by distance ratio (PDbyD).

farmer nodes to solutions. When these operators are removed from our approach, solutions require to use more flexible heuristic criteria during construction phase, this additional diversification is probably obtained using the PD criterion.

Table V shows the Wilcoxon comparison between the performance of our GRASP approach and each alternative scenario. Gray cells highlight the cases where a statistical difference with a 95% confidence level was found. From these results, it is clear that all the proposed movements are crucial on GRASP performance, there is only one case where the performance obtained by excluding the external swap was able to obtain high quality solutions.

TABLE IV
STATISTICAL COMPARISON OF HEURISTICS CRITERIA

Case	ALL	NoR	NoIS	NoES	NoD
1	==	<>	<>	==	==
2	<>	==	==	==	==
3	==	==	==	==	<>
4	==	==	==	<>	<>
5	==	==	==	<>	<>
6	<>	==	==	<>	==
7	==	==	==	<>	<>
8	==	==	==	<>	<>

As can be seen in boxplots in figures 4 and 5, in most cases the performance of alternative scenarios shows a

(a) Case 5

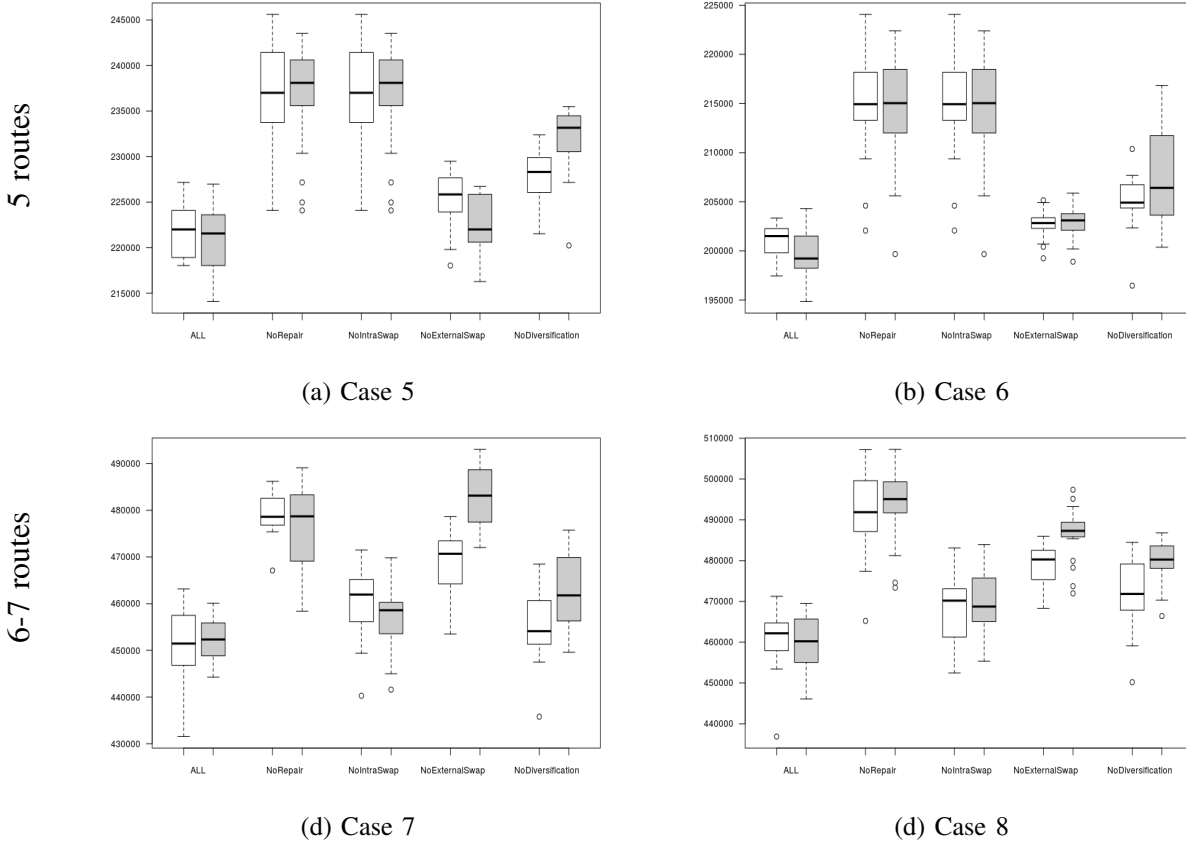(b) Case 6

(d) Case 7

(d) Case 8

Fig. 5. Cases 5-8: Boxplots of Total cost obtained by the 5 scenarios studied. White boxes show the performance obtained using the pending demand (PD) heuristic criterion while gray bars use the pending demand by distance ratio (PDbyD).

clear deterioration compared to the whole GRASP proposal. From this, we observe that the most critical operator is the Repair process (node deletion). This because node deletion allows the GRASP approach to rapidly found promising areas of the search space – that selects a proper set of farmer nodes.

Second most critical operator is the intra swap movement. Except for case 1, its removal from GRASP approach generated clear deteriorations of solutions quality. Intra swap movement is in charge of perform intensification during the search process, hence, its participation in GRASP can be crucial.

Diversification performed by node add movement also plays an important role in our proposal, but after deletion node and intra swap operators. External swaps show being important only for small instances (cases 1-6), but as the problem grows they become less useful.

D. MIP and GRASP comparison

In this section we compare the performance of our integer programing model and our GRASP proposal. Table VI shows the performance comparison between these two approaches. For each study case we show the quality of the best solution

TABLE V
STATISTICAL COMPARISON OF RELEVANCE OF GRASP MOVEMENTS

Case	NoR	NoIS	NoES	NoD
1	<>	<>	<>	<>
2	<>	<>	<>	<>
3	<>	<>	<>	<>
4	<>	<>	==	<>
5	<>	<>	<>	<>
6	<>	<>	<>	<>
7	<>	<>	<>	<>
8	<>	<>	<>	<>

found by CPLEX, the average quality and best solution found by our GRASP and the GAP of these GRASP solutions respect to the corresponding MIP solution. The best solution found by CPLEX was obtained after 1 hour considering the integer programming model proposed using four cores. The best solution found by GRASP was obtained after 20 runs using different seeds. The GAP is calculated as $100 * \frac{(GRASP-MIP)}{MIP}$. Each GRASP execution time was limited to 60 seconds. Total time spent on 20 GRASP executions is listed on table VI.

From these results we can observe that the smallest instances are better solved by the linear integer programing

TABLE VI
COMPARISON OF FINAL SOLUTIONS

Case	MIP					GRASP		
	min	time [s]	ave	ave GAP%	time [s]	min	min GAP%	time [s]
1	111071	3600	117587	5.87	60	115860	4.31	1200
2	191624	3600	201089	4.94	60	197456	3.04	1200
3	359583	3600	393284	9.37	60	370343	2.99	1200
4	214088	3600	222123	3.75	60	218048	1.85	1200
5	242963	3600	243750	0.32	60	243625	0.27	1200
6	435756	3600	398537	-8.54	60	388048	-10.95	1200
7	440296	3600	451423	2.53	60	431530	-1.99	1200
8	453439	3600	460880	1.64	60	436864	-3.66	1200

model approach. Smallest cases are those that consider a low demand and at most 4 routes. For the largest case studies, those where MIP gets the highest integrality gaps, GRASP approach becomes useful. In this case for instances with the highest number of routes, more than 4 routes, GRASP solutions outperform MIP solutions and required lower computational resources to be computed. Moreover, from these cases, the maximum difference in terms of quality was obtained for case 6. This difference is clear for both, the average and best solution. Case 6 can be considered the more complex problem instance. This, because it has the same demand than cases 7 and 8, but in this case the demand must be satisfied using the lower number (and in some cases total capacity) of trucks.

VI. CONCLUSIONS & FUTURE WORK

In this work we have proposed an application of the prize collecting vehicle routing problem in a real optimization problem of milk collection in a processing plant from south of Chile. A linear integer programming model was developed to solve small case studies of the problem and an ad-hoc GRASP metaheuristic to solve the largest problem instances. The base case study was solved considering the current scenario, where the company only uses three trucks. Using both, the MIP model and th GRASP algorithm, we improved the current manual solutions. Moreover, we studied the performance of these proposals on some possible expansion scenarios of the processing plant. For this, we increased both, the demand of milk collected and the trucks fleet. All these scenarios were studied using the integer programming model and the GRASP metaheuristic proposed. Smallest instances were properly solved using the MIP approach, while largest instances were better solved using the GRASP proposed approach. Moreover, we studied the main components of the GRASP proposal and concluded that all the designed components are crucial on our GRASP proposal performance.

As future work we plan to study more real world problem instances, specially those from our milk production region that posses complex distribution of production nodes and, hence, become a challenge for our research. Proposal should also be analyzed to determine crucial components and broader conclusions on the problem.

REFERENCES

[1] Balas, E. 1989. The prize collecting traveling salesman problem Networks, vol. 19,pp. 621-636

[2] Caramia, M., Guerriero, F., 2010. A milk collection problem with incompatibility constraints. Interfaces 40 (2), 130-143.

[3] Caria M., Todde G., Pazzona A. 2018. Modeling the Collection and Delivery of Sheep Milk: A Tool to Optimise the Logistics Costs of Cheese Factories. Agriculture 8, 1-11.

[4] Claassen, G.D., Hendriks, T.H., 2007. An application of special ordered sets to a periodic milk collection problem. Eur. J. Oper. Res. 180 (2), 754-769.

[5] Dayarian, I., Crainic, T.G., Gendreau, M., Rei, W., 2013. An adaptive large neighborhood search heuristic for a multi-period vehicle routing problem. Technical Report CIRRELT-2013-60, Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT).

[6] Dayarian, I., Crainic, T.G., Gendreau, M., Rei, W., 2015. A column generation approach for a multi-attribute vehicle routing problem. Eur. J. Oper. Res. 241 (3), 888-906.

[7] Dayarian, I., Crainic, T.G., Gendreau, M., Rei, W., 2015. A branch-and-price approach for a multi-period vehicle routing problem. Comput. Oper. Res. 55,167-184.

[8] Dell'Amico, M., Maffioli, F., Sciomachen A., A lagrangian heuristic for the prize collecting traveling salesman problem Annals of Operations Research, 81 (0) (1998), pp. 289-306

[9] Food and Agriculture Organization 2012. Dairy production and products: Collection and transport.

[10] Lahrichi, N., Crainic, T.G., Gendreau, M., Rei, W., Rousseau, L.M., 2013. Strategic analysis of the dairy transportation problem. J. Oper. Res. Soc.

[11] Masson, R., Lahrichi, N., Rousseau, L.M., 2015. A two-stage solution method for the annual dairy transportation problem. Eur. J. Oper. Res.

[12] Paredes-Belmar G., Marianov V., Bronfman A., Obreque C., Lüer-Villagra A. 2016. A milk collection problem with blending. Transp. Res. Part E 94, 26-43.

[13] Paredes-Belmar G., Lüer-Villagra A., Marianov V., Cortés C.E., Bronfman A. 2017. The milk collection problem with blending and collection points. Computers and Electronics in Agriculture 134, 108-123.

[14] Sethanan, K., Pitakaso, R., 2016. Differential evolution algorithms for scheduling raw milk transportation. Comput. Electr. Agric. 121, 245-259.

[15] Stenger, A., Vigo, D., Enz, S., and Schwind, M., A variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. Transportation Science, (to appear), doi:10.1287/trsc.1110.0396