Prize-collecting Arc Routing Problems
and Extensions

Julián Aráoz       Elena Fernández
Carles Franquesa

Department of Statistics and Operation Research
Technical University of Catalonia
Campus Nord, C5, Jordi Girona 1-3, 08034 Barcelona, Spain
fax: +34 – 93 – 401 58 55, e.mail: e.fernandez@upc.edu

Oscar Meza
Department of Computation and Information Technology
Simon Bolivar University

Abstract

In this work we present the Prize-collecting Arc Routing Problem and some extensions. We derive properties that allow determining linear defining systems for the problems and solution approaches based on upper bounds obtained from relaxations reinforced with cuts, and lower bounds derived from heuristics. We also present numerical results from computational experiments that show that in many cases optimality of the obtained solutions could be proven.

1 Introduction

We study the Prize-collecting Arc Routing Problem (PARP) and some extensions. Problems of this type were defined in Aráoz, Fernández and Zoltan [1], and they are a generalization of Arc Routing Problems (ARPs), which usually aim to determine a least-cost traversal of a specified arc subset of a graph, with or without constraints. Typical ARPs constraints require the routes to begin and finish at a given point (the depot), and guarantee the connectivity of the solutions with the depot. A recent and comprehensive state on the art of such problems can be found in Dror [3].

PARPs can be seen as the arc-routing counterpart of Prize-Collecting Traveling Salesman Problems (see [2]). PARPs can also be stated in directed and mixed graphs. However, in this work we only consider Prize-collecting Edge Routing Problems.
2 Prize-collecting Arc Routing Problems

The basic definition is the following:

**Definition 2.1** Given a graph $G(V,E)$ with a distinguished vertex $d$, the Depot, and two real functions on $E$, the profit function $b$ and the cost function $c$, the Prize-collecting Arc Routing Problem is to find one cycle $C^*$ which maximizes the value of

$$\sum_{e \in C} (b_e - t_e c_e)$$

where $C$ is a cycle in $G$ passing trough $d$, and not necessarily simple, and $t_e$ is the number of times that edge $e$ is traversed in $C$.

Like in other ARPs, in the PARPs we assume that the demand for service is placed at the edges of a graph. However, there is no specific edge subset to be traversed. Instead, we assume that giving service to an edge incurs both a cost (associated with displacement), and a profit (associated with servicing edges). The displacement cost to an edge will certainly account for the cost of all the edges that are traversed in that same route, as many times as they are traversed. Similarly, the profit associated with servicing an edge, must also take into account the profit of the additional edges that are serviced in that same route. However, the profit of each edge serviced in the route will be collected only once, independently of the number of times the edge is traversed.

In a PARP we look for traversals that maximize the total servicing profit minus the displacement costs. They constitute a generalization of most ARPs and of most Traveling Salesman Problems.

In this work, in addition to the basic PARP, we consider other problems of the same family that can be defined by imposing additional conditions. The considered problems are:

- The basic PARP, which is the Prize-collecting version of the Rural Postman Problem, that we denote Price-collecting Rural Postman Problem (PRPP). In a PRPP, we look for the most Profitable Subtour passing through the depot.

- The Weighted PARP (WPARP). Here there is a weight associated with servicing each edge, as well as a limit on the total weight that one route can service. Two different versions of this problem are considered; they differ on whether or not a limit on the number of routes to be performed is imposed.

- The Clustered PARP (CPARP). Here some subsets of edges are grouped into clusters. The additional requirement with respect to a basic PARP is that for each cluster either all the edges are serviced or none of them are serviced.
3 Solutions and Heuristics

In [1] it was proven that the PRPP is NP–Hard. Here we obtain Dominance Relations that allow to formulate PRPP by means of a system of linear inequalities with binary variables. The specific characteristics of WARPs and CPARPs are taken into account to define appropriate systems for these other problems.

One common characteristic of all these systems is that they have an exponential number of inequalities. Hence, we propose algorithms for these problems that approximate the optimal solutions by generating both upper and lower bounds. The upper bounds are derived from the solution to a series of LP relaxations of the linear systems that define the problems. While possible we reinforce the relaxation by generating cuts violated by the current solution. We present separation algorithms, both for the family of inequalities that guarantee the parity of the nodes, and for the family of inequalities that guarantee connectivity of the solutions with the depot.

Lower bounds for PRPP are obtained with an adaptation of the 3T heuristic of Fernández, Meza, Garfinkel and Ortega [4] for the RPP. For CPARP the lower bounds are obtained with a heuristic based on the one for cactus presented in [1] for the Rural Postman Problem.

4 Computational Results

Next we report on the results of some preliminary computational experiments. Programs have been coded in C using CPLEX 7.0. All tests were run on a Sun ULTRA 10, model 440, 1x440 Mhz, 1Gb RAM. Since there are no available benchmark instances for PARPs, we have generated PARP instances from well known sets of RPP benchmark instances. The considered instances are divided in five groups labeled ALBAIDA (2 instances), P (24 instances), DEGREE (36 instances), GRID (36 instances) and RANDOM (24 instances), respectively. A description of the characteristics of each group can be found, for instance, in [4]. In all cases the depot has been taken as vertex 1 and we have kept the cost function $c$. Profits $b$ have been assigned to the edges as follows:

- $b_e \in \mathbb{U}[c_e, 3c_e]$, if $e$ is a required edge of the RPP instance.
- $b_e \in \mathbb{U}[0, c_e]$, if $e$ is a non-required edge of the RPP instance.

A summary of the results for PRPP is given in Table 1. The first two columns depict the number of instances in each group and the number of instances in each group for which optimality could be proven, respectively. As can be seen we have proven optimality for 42 of the 118 instances. The remaining columns give average results over all the instances in the group. In general the percent gaps between the upper and the lower bounds are quite small, although in some cases they are too large. This is particularly true for two instances, one in group P and one in group GRID, with percent gaps over 50%. Since the optimal values of the instances are unknown, we cannot assess...
Table 1: Results for PRPP

<table>
<thead>
<tr>
<th></th>
<th>no.</th>
<th>no.</th>
<th>Gap</th>
<th>cpu LP</th>
<th>cpu heur</th>
<th>no. its.</th>
<th>Connect. (set)</th>
<th>parity (set)</th>
<th>parity (vertex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBAIDA</td>
<td>2</td>
<td>0</td>
<td>4.132</td>
<td>71.954</td>
<td>0.447</td>
<td>41.500</td>
<td>4028.000</td>
<td>74.000</td>
<td>42.500</td>
</tr>
<tr>
<td>P</td>
<td>24</td>
<td>8</td>
<td>8.493</td>
<td>0.396</td>
<td>0.076</td>
<td>4.583</td>
<td>61.833</td>
<td>9.125</td>
<td>11.875</td>
</tr>
<tr>
<td>DEGREE</td>
<td>36</td>
<td>9</td>
<td>7.493</td>
<td>22.609</td>
<td>0.669</td>
<td>11.500</td>
<td>574.167</td>
<td>107.389</td>
<td>21.278</td>
</tr>
<tr>
<td>GRID</td>
<td>36</td>
<td>12</td>
<td>10.542</td>
<td>66.294</td>
<td>0.786</td>
<td>29.639</td>
<td>884.667</td>
<td>173.722</td>
<td>10.917</td>
</tr>
<tr>
<td>RANDOM</td>
<td>20</td>
<td>13</td>
<td>2.951</td>
<td>0.729</td>
<td>0.014</td>
<td>15.700</td>
<td>292.600</td>
<td>58.550</td>
<td>7.800</td>
</tr>
</tbody>
</table>

the optimality of the heuristic solutions unless the percent gap is zero. However, observing some of the obtained solutions, we conjecture that there are more instances for which the solution provided by the heuristic is in fact optimal. In terms of the number of iterations and the number of cuts added, we observe that these numbers tend to increase for the instances that are harder to solve and that the number of connectivity constraints exceeds that of parity constraints. As for the cpu times they are very small, both for obtaining the lower bound and for the heuristic.

For the moment, we have only solved 24 CPARP instances generated from group P of RPP instances. For CPARPs profits have only been assigned to the required edges of the RPP instances, and the clusters have been taken as the connected components induced by these required edges. The optimal LP solution was feasible, and therefore optimal, for 9 out of the 24 instances. The average percent gap between the LP solution and the optimal solution is 2.59, whereas the percent gap between the heuristic solution and the optimal solution is 10.18. These preliminary results are quite satisfactory, although we believe that they could be improved, especially the ones of the heuristic.

References


