A Parallel Hybrid Evolutionary Metaheuristic for the Period Vehicle Routing Problem

Dalessandro Soares Vianna, Luiz S. Ochi and Lúcia M. A. Drummond

Universidade Federal Fluminense - Departamento de Ciência da Computação
Travessa Capitão Zeferino 56/808
Niterói - Rio de Janeiro - Brazil - 24220-230
e-mail: dada@pgcc.uff.br, {satoru,lucia}@dcc.uff.br

Abstract. This paper presents a Parallel Hybrid Evolutionary Metaheuristic for the Period Vehicle Routing Problem (PVRP). The PVRP generalizes the classical Vehicle Routing Problem by extending the planning period from a single day to M days. The algorithm proposed is based on concepts used in Parallel Genetic Algorithms and Local Search Heuristics. The algorithm employs the island model in which the migration frequency must not be very high. The results of computational experiments carried out on problems taken from the literature indicate that the proposed approach outperforms existing heuristics in most cases.

1 Introduction

The classical Vehicle Routing Problem (VRP) is defined as follows: vehicles with a fixed capacity Q must deliver order quantities \( q_i \) (\( i = 1, \ldots, n \)) of goods to \( n \) customers from a single depot (\( i = 0 \)). Knowing the distance \( d_{ij} \) between customers \( i \) and \( j \) (\( i,j = 0, \ldots, n \)), the objective of the problem is to minimize the total distance travelled by the vehicles in such a way that only one vehicle handles the deliveries for a given customer and the total quantity of goods that a single vehicle delivers do not be larger than Q.

In classical VRPs, typically the planning period is a single day. The period Vehicle Routing Problem (PVRP) generalizes the classical VRP by extending the planning period to M days. Over the M-day period, each customer must be visited at least once during the considered period. The classical PVRP consists of a homogeneous vehicle fleet (vehicles with same capacities) which must visit a group of customers from a depot where the vehicles must start and return to at the end of their journeys. Each vehicle has a fixed capacity that cannot be exceeded and each customer has a known daily demand that must be completely satisfied in only one visit by exactly one vehicle. The planning period is M days. If \( M = 1 \), then PVRP becomes an instance of the classical VRP. Each customer in PVRP must be visited \( k \) times, where \( 1 \leq k \leq M \). In the classical model of PVRP, the daily demand of a customer is always fixed. The PVRP can be seen as a problem of generating a group of routes for each day so that the constraints involved are satisfied and the global costs are minimized.
PVRP can also be seen as a multi-level combinatorial optimization problem. In the first level, the objective is to generate a group of feasible alternatives (combinations) for each customer. For example, if the planning period has \( t = 3 \) days \( \{d_1, d_2, d_3\} \) then the possible combinations are: 0 \( \rightarrow \) 000; 1 \( \rightarrow \) 001; 2 \( \rightarrow \) 010; 3 \( \rightarrow \) 011; 4 \( \rightarrow \) 100; 5 \( \rightarrow \) 101; 6 \( \rightarrow \) 110 and 7 \( \rightarrow \) 111. If a customer requests two visits, then this customer has the following visiting alternatives: \( \{d_1, d_2\} \), \( \{d_1, d_3\} \), and \( \{d_2, d_3\} \) (or the options: 3, 5 and 6 of Table 1). In the second level, we must select one of the alternatives for each customer, so that the daily constraints are satisfied. Thus we must select the customers to be visited in each day. In the third level, we solve the vehicle routing problem for each day.

In this paper, our algorithm is applied to the basic model of PVRP with an additional constraint: the number of vehicles is limited, although this limit is not necessarily the same every day. The technique proposed here can be applied to various models of PVRP.

<table>
<thead>
<tr>
<th>Customer</th>
<th>Diary Demand</th>
<th># of Visits</th>
<th># of Combinations</th>
<th>Possible Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>1</td>
<td>3</td>
<td>1, 2 e 4</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>2</td>
<td>3</td>
<td>3, 5 e 6</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>2</td>
<td>3</td>
<td>3, 5 e 6</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>1</td>
<td>3</td>
<td>1, 2 e 4</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 1. A PVRP with \( t=3 \) days

In the last years, Evolutionary Metaheuristics and in particular Genetic Algorithms (GA) have been used successfully in solution of NP-Complete and NP-HARD problems of high dimensions. Hard problems require large search spaces resulting in high computational costs. In this context, Evolutionary Metaheuristics including GA may require a large amount of time to find good feasible solutions, encouraging the use of parallel techniques [10]. Although the main goal of a Parallel Evolutionary Metaheuristic is the reduction of the execution time necessary to find an acceptable solution, sometimes it can also be used to improve the results obtained by sequential versions.

Most of Evolutionary Metaheuristics, such as GA, Scatter Search, Ant Systems and Neural Nets, are easy to parallelize because of their intrinsic parallelism. There are different ways to parallelize GA. The generally used classification divides parallel GAs in three categories: Island or stepping stone, Fine Grain and Panmitic Models [10].

In this paper, we propose a parallel hybrid evolutionary metaheuristic based on parallel GA, scatter search and local search methods. The algorithm is based on the Island Model and it was implemented on a cluster of workstations with 4 RISC/6000 processors.

The remainder of this paper is organized as follows. Section 2 presents the