A Parallel Evolutionary Algorithm for the Vehicle Routing Problem with Heterogeneous Fleet

Luiz S. Ochi, Dalessandro S. Vianna
Lúcia M. A. Drummond and André O. Victor

PGCC-UFF, Niterói, RJ-Brazil
e-mail:{satoru,dada,lucia,armis}@pgcc.uff.br

Abstract. Nowadays genetic algorithms stand as a trend to solve NP-complete and NP-hard problems. In this paper, we present a new hybrid metaheuristic which uses Parallel Genetic Algorithms and Scatter Search coupled with a decomposition-into-petals procedure for solving a class of Vehicle Routing and Scheduling Problems. The parallel genetic algorithm presented is based on the island model and was run on a cluster of workstations. Its performance is evaluated for a heterogeneous fleet problem, which is considered a problem much harder to solve than the homogeneous vehicle routing problem.

1 Introduction

Metaheuristics have proven to be very effective approaches to solving various hard problems [5] [12]. Among them, Genetic Algorithms, Tabu Search and Scatter Search have been used successfully in optimization problems.

Genetic algorithms (GA) were introduced by John Holland and researchers from University of Michigan in 1976 [8]. In the last decade they have become widely used, however, they have not presented very good results for several optimization problems in its original model. In order to turn GA more efficient, some proposals have appeared recently, such as the inclusion of notions of scatter search, tabu search and local search in GA ([5], [6], [14]). Particularly, the version of GA with scatter search aims to generate GA "less probabilistic" than the conventional ones [5], [6], [14]. Genetic Algorithms may also require a large amount of time to find solutions, encouraging the use of parallel techniques [13].

There are different ways to parallelize genetic algorithms. The generally used classification divides parallel genetic algorithms in three categories: island, fine grain and panmitic models [13].

In this paper, we propose a new hybrid metaheuristic based on parallel genetic algorithms, scatter search and a decomposition-into-petal procedure to solve a class of vehicle routing and scheduling problems. The parallel genetic algorithm proposed is based on the island model and it was implemented on a heterogeneous cluster of workstations. In order to reduce communication overhead, which is usual in this kind of model [10] [13], and ensure the occupation of all processors during the execution of the algorithm new criteria of migration and termination were proposed.

Vehicle routing problems are generalizations of the classical Traveling Salesman Problem (TSP) and consequently are NP-complete problems. Related literature is vast and it usually presents approximate solutions obtained by heuristics. This present popularity can be attributed to the success many algorithms have achieved for a large variety of vehicle routing and scheduling problems in which real systems constraints were implemented. Among them, we include the following papers: [1]; [2]; [5]; [6]; [10]; [11];[14].

In order to evaluate the proposed algorithm, we apply it to a model with a heterogeneous fleet, it can, however, be adjusted for an additional set of constraints which usually arise in real systems. Partial computational results show the efficiency of applying these hybrid techniques in comparison to the algorithm proposed by Taillard [16].

The remainder of this paper is organized as follows. Section 2 describes the Vehicle Routing Problem. Section 3 presents the algorithm proposed. Experimental results are shown in Section 4. Finally, Section 5 is the conclusion.

2 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) was originally posed by Dantzig and Ramser [2] and may be defined as follows: vehicles with a fixed capacity Q must deliver order quantities q_i $(i = 1, \dots, n)$ of goods to n customers from a single depot $(i = 0)$. Knowing the distance d_{ij} between customers i and j $(i, j = 0, \dots, n)$, the objective of the problem is to minimize the total distance traveled 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 is not larger than Q. Several variants of VRPS exist. In the heterogeneous problems, we have a set $\Psi = \{1, \dots, k\}$ of different vehicle types. A vehicle of type $k \in \Psi$ has a carrying capacity Q_k. The number of vehicles of type k available is n_k. The cost of the travel from customer i to j $(i, j = 0, \dots, n)$ with a vehicle type k is d_{ijk}. The use of one vehicle of type k implies a fixed cost f_k and different vehicle types will reflect different fixed costs.

In this paper, we are interested in a special case of the above problem where the travel costs are the same for all vehicle types $(d_{ijk} = d_{ijk^1}, k, k^1 \in \Psi)$ and the number n_k of vehicles of each type is not limited $(n_k = \infty, k \in \Psi)$. The goal of this problem is to determine a fleet of vehicles such that the sum of fixed costs and travel costs is minimized.

Although solution methods for homogeneous vehicle routing problems have substantially improved, the vehicle routing problem with heterogeneous fleet has attracted much less attention. This is mainly due to the increased difficulty in solving such a problem [16].

3 A Parallel Evolutionary Algorithm using GA and Scatter Search with Petal Decomposition

The algorithm proposed PGA-PED (Parallel Genetic Algorithm with Petal Decomposition) is a hybrid procedure which uses parallel genetic algorithm, scatter search and a petal decomposition criterion to build chromosomes. This petal decomposition procedure can be used by several models of routing and scheduling problems, such as: the classical model of vehicle routing, routing with heterogeneous fleet and routing with time constraints.

In island model, the population of chromosomes is partitioned into several subpopulations which evolve in parallel and periodically migrate their individuals (chromosomes) among themselves. Because of the high cost of communication in this model, migration frequency must not be very high. Thus, in the algorithm proposed migration is only executed when subpopulation renewal is necessary.

The criterion of termination of processes is based on a global condition which involves every process which composes the parallel genetic algorithm, in order to prevent a process from being idle while the others are still executing.

The tasks executed by each processor and the communication among them can be visualized in Figure 1, where each pair of tasks m_i and q_i share a single processor, which is switched between them.

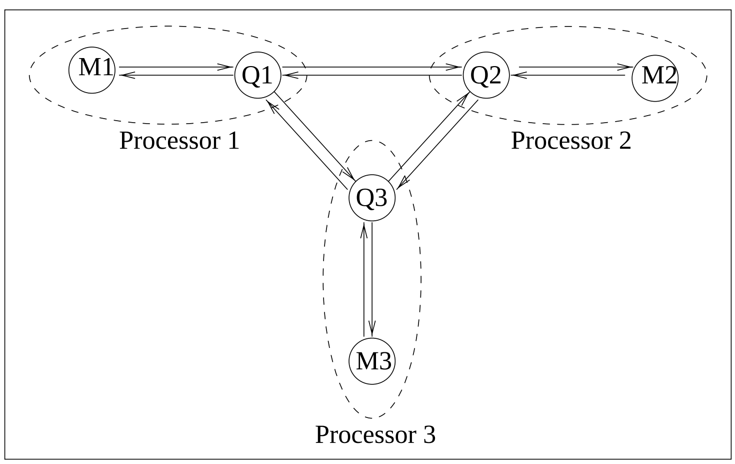

Fig. 1. Processes communication for 3 processors

Each task m_i executes the following steps, described in next subsections: Building of routes/schedules using a petal decomposition procedure; Representation of solutions by chromosomes; Generation of an initial population of chromosomes; Reproduction of chromosomes and their evaluation; Sending of a migration request to q_i if population renewal is required.

The task q_i is responsible for migration and termination of the parallel genetic algorithm.

3.1 Tools for Building of Routes

It has shown in many applications of the vehicle routing problem and its generalizations, that the set of routes has a petal structure, which forms a flower. The problem of establishing the best flower is the focus of several papers [15]. We adopted a node "sweeping" technique described as follows.

We associate the VRP to a complete symmetric planar graph $G(N, A)$ where N is the node set and A is the edge set. A cost c_{ij} is associated with each edge, the depot is located at node S and there are n nodes, each associated with one customer. Geometrically, the petals on the plane can be obtained by tracing a vector $\vec{sx}$ with origin in the node S and parallel to axis x. We rotate $\vec{sx}$ in clockwise direction (or counter clockwise direction) until $\vec{sx}$ intercepts a node n_i with demand q_i. Then, we update the accumulated demand $D := q_i$ and go on the rotation until another node is met. For each node met, its demand is summed to D and D is compared to the capacity of the vehicle (Q). If it is less than Q, the rotation continues. If D exceeds Q, the last node intercepted by $\vec{sx}$ is deleted from the current petal (route) and the petal is closed. The closing of one petal initiates the process of constructing a new petal, beginning with the deleted node. This procedure is repeated until all nodes are incorporated into petals.

Figure 2 shows nodes representing customers and their demands, where the capacity of each vehicle is 25. The first petal covers nodes $\{1, 2, 3\}$ whose demands are 8, 13 and 3 respectively. Using the procedure described above, with n nodes, there are $2n$ feasible solutions in petal structures. Of these solutions, n of them are generated by each node in clockwise direction and the remainder is generated in counter clockwise direction. Unfortunately, some solutions of VRP do not have a petal structure. For example, from the original sequence of customers (permutation) $P_1 = (1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\ \ 9\ \ 10\ \ 11\ \ 12)$ if we swap 1 and 2, 3 and 6, and 8 and 12 we obtain a new sequence $P_2 = (2\ \ 1\ \ 6\ \ 4\ \ 5\ \ 3\ \ 7\ \ 12\ \ 9\ \ 10\ \ 11\ \ 8)$. If we initiate the petal construction at node 2 in counter clockwise direction, we obtain the following set of petals $\{\{2, 1\}, \{6, 4, 5, 3\}, \{7, 12, 9\}, \{10, 11\}, \{8\}\}$, as shown in Figure 3. This structure is called artificial flower and it is composed of artificial petals.

We propose the following strategy for the Vehicle Routing Problem with Heterogeneous Fleet. Consider k types of vehicle: $\Psi = \{1, \ldots, k\}$, such that $Q_1 \leq Q_2 \leq \ldots \leq Q_k$ and $f_1 \leq f_2 \leq \ldots \leq f_n$, where Q_i is the capacity of vehicle of type t_i and f_i is its corresponding fixed cost. Each petal is built as follows. The algorithm analyzes the possibilities given by the k types of vehicles and it chooses the type of vehicle t_i which presents the lowest value for $(Q_i - D_i) \times f_i$, where D_i is the accumulated demand of the petal using vehicle t_i. Clearly, the size of each petal will vary according to the type of vehicle associated.

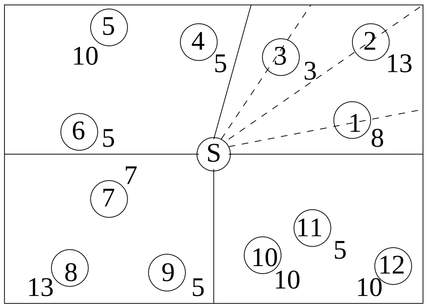

Fig. 2. Building of Petals

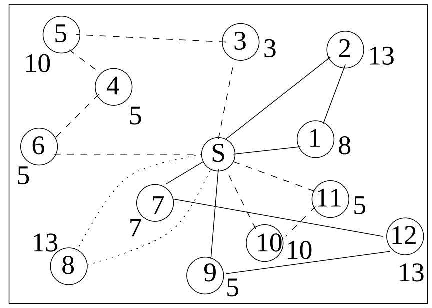

Fig. 3. Artificial Petals

3.2 Coding and Decoding of Chromosomes

In the algorithm PGA-PED applied to a vehicle routing problem with heterogeneous fleet, a chromosome is a list of k genes, where k is the number of customers. The i^{th} position (gene) is occupied by a positive integer which represents the customer.

3.3 Population Generation

The initial population is generated using the following strategy. At first, nodes called as seeds are defined by each task m_i. They are responsible for initiating the construction of flowers. The number of seeds is ⌈ nodes / number of processors ⌉, except for the last task, in case of the quotient is not an integer. For each task, defined the seeds, a flower is built initiating by one of these seeds (see subsection 3.1) and this procedure is repeated by other seeds until a subpopulation is established.

3.4 Genetic Operator and Chromosome Evaluation

The genetic operator is responsible for the production of new chromosomes from the existing ones. The roulette criterion is applied to choose the parents based on the probability associated with each individual according to its fitness. The operator used is an adaptation of the ERX method [18]. For each customer a list of neighbors is built based on its location in the petals of the parents used for reproduction. For example, in case of two parents and their corresponding set of petals $P_1 = \{\{0,1,3,5,7,2\},\{0,6,4,8,9\}\}$ and $P_2 = \{\{0,3,6,4\},\{0,5,2,1\},\{0,9,7,8\}\}$, the following lists would be obtained for each customer: (1): 3,2; (2): 7,5,1; (3): 1,5,6; (4): 6,8 ;(5): 3,7,2; (6): 4,3; (7): 5,2,8,9; (8): 4,9,7; (9): 8,7.

In order to generate offspring's the following steps are executed:

Step 1 : the closest customer of the depot v is selected as the first gene of the offspring.

Step 2: a customer w, neighbor of v, with the lowest number of neighbors, is inserted in the chromosome, since it has not yet been selected.

Step 3: $v := w$

Step 4: go to Step 2 until all customers are inserted in the chromosome.

If there is not a customer which is neighboring the last customer inserted (v) and that has not yet been inserted in the chromosome, then the closest customer of v (not inserted yet) is selected.

We use the same criterion described in Subsection 3.1 to select the vehicles which will serve the routes in the offspring, which may be composed of petals or artificial petals.

The chromosome evaluation is based on the cost of the set of associated routes, where each route is considered a TSP and is solved through heuristic GENIUS [4].

3.5 Migration of Chromosomes and Termination

The strategy adopted in this algorithm, consists of associating to each task m_i (main module), which takes part of the parallel genetic algorithm of the island model, a task q_i. A task q_i, for $1 \leq i \leq n$, where n is the number of processors, communicate only by message-passing.

A task q_i is activated by the associated task m_i in the following cases:

- When the renewal tax is less than 5 per cent (tax of replacement of parents by offspring's), triggering the migration of individuals.
- When the task m_i is capable of finishing, initiating the termination of the algorithm.

In the first case, q_i executes a broadcast of requests so that all tasks q_i send their best solutions to it. When this occurs, a new population is generated based on the best chromosome received. Thus, a "window" is opened in this best chromosome in a random position and pairs of genes are switched in this window. For example, consider the chromosome $\{1, 3, 5, 7, 2, 6, 4, 8, 9\}$ and the windows: $\{\mathbf{1}, \mathbf{3}, \mathbf{5}, 7, 2, 6, 4, 8, 9\}$; $\{1, 3, \mathbf{5}, \mathbf{7}, \mathbf{2}, 6, 4, 8, 9\}$; $\{1, 3, 5, 7, 2, 6, \mathbf{4}, \mathbf{8}, \mathbf{9}\}$.

After swapping some genes in these windows, we could obtain the chromosomes: $\{\mathbf{3}, \mathbf{1}, \mathbf{5}, 7, 2, 6, 4, 8, 9\}$; $\{1, 3, \mathbf{2}, \mathbf{7}, \mathbf{5}, 6, 4, 8, 9\}$; $\{1, 3, 5, 7, 2, 6, \mathbf{4}, \mathbf{9}, \mathbf{8}\}$.

The vehicles which will serve the routes of these new chromosomes will be chosen using the criterion described in Subsection 3.1.

After y migrations (y is an entry parameter), m_i becomes capable of finishing its execution and consequently it activates q_i, which keeps an array of n elements representing the state of the system. Each position i of the array of state may assume values true or false, indicating that the process m_i is capable of finishing or not. When q_i is activated, it updates the position i in the array of state with true and initiates a broadcast so that the other arrays represent the current state of the system. This strategy of termination is based on a simplification of the algorithm for detection of stable conjunctive predicate presented in [3]. A task m_i will continue its execution, even when capable of finishing, until all processes are able to finish. This prevents a processor from being idle while other processors remain executing and still allows the improvement of the optimal solution while the application does not finish. When all elements of the array of state are true, termination is detected by q_i and m_i is informed to finish its execution.

A task m_i, at each generation of the genetic algorithm, informs to q_i the optimal solution so far.

The strategy described above allows that the process which is responsible for the genetic algorithm execution does not remain occupied with the communication required by migration of individuals, and termination of the parallel genetic algorithm, simplifying its design and implementation.

4 Partial Experimental Results

Our experiments were run on a heterogeneous cluster of 4 workstations (RS/6000 with different clock speeds) using the programming language C and MPI (Message-Passing Interface) for parallelism [9].

Although the Vehicle Routing Problem with Heterogeneous Fleet is a major optimization problem, it has not attracted much attention due to the fact that it is very difficult to solve [16]. Thus in order to analyze it, we used the results presented by Taillard and the results presented by its sequential version (PeGA) [17].

Problem number	n	Vehicle type											
		A		B		C		D		E		F	
		Q_A	f_A	Q_B	f_B	Q_C	f_C	Q_D	f_D	Q_E	f_E	Q_F	f_F
13	50	20	20	30	35	40	50	70	120	120	225	200	400
14	50	100	120	160	1500	300	3500						
15	50	50	100	100	250	160	450						
16	50	40	100	80	200	140	400						
17	75	50	25	120	80	200	150	350	320				
18	75	20	10	50	35	100	100	150	180	250	400	400	800
19	100	100	500	200	1200	300	2100						
20	100	60	100	140	300	200	500						

Fig. 4. Data for VRP with Heterogeneous Fleet

Our algorithm was analyzed with the values presented in the table of Figure 4. This table provides information about the problem number (as in Golden et al [7]), number of customers (n), vehicle capacities (Q) and their fixed costs (f). An average of results is shown in Figure 5. It presents eight problems where for each of them 10 instances were run. The parameters used were the following: number of parents for reproduction = 4; window size = number of customer / 10; number of pairs of genes swapped in the window = number of customers / 25; population size = 2 $\times$ number of customers; number of migrations = 10.

Figure 5 shows that PGA-PED achieved the best results for the problems 13 and 14, nearly identical results to the Taillard in problems 15, 16, 17, 18 and 20, and presented worst result for the problem 19.

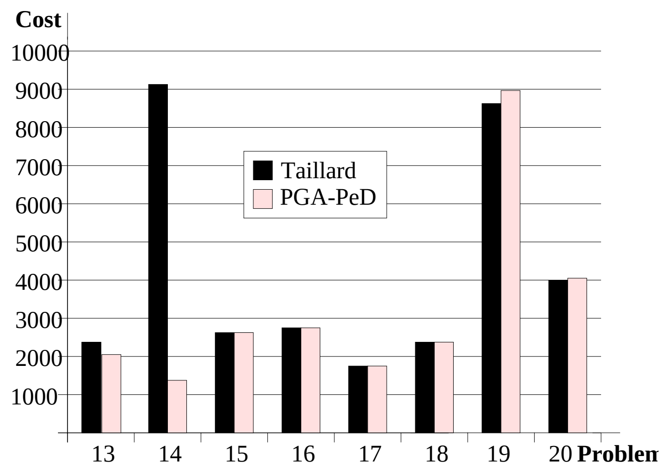

Fig. 5. Experimental Results: PGA-PeD x Taillard

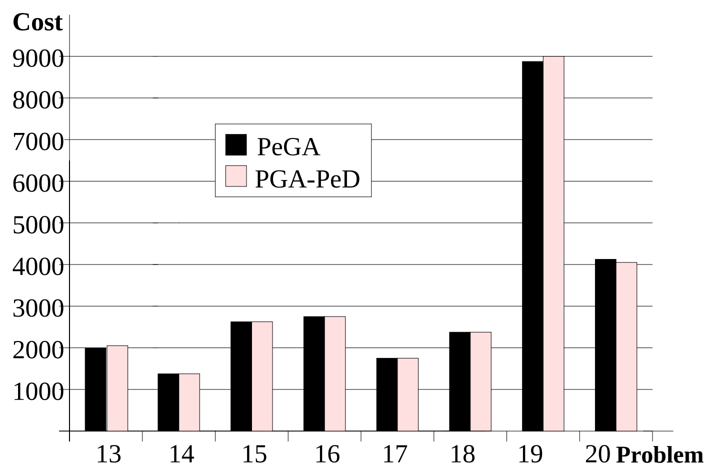

Fig. 6. Experimental Results: PGA-PeD x PeGA

Figure 6 shows that PGA-PED achieved nearly identical results to the sequential version in all problems.

5 Concluding Remarks

Our results so far show some advantages for PGA-PED when compared to the algorithms presented by Taillard.

Experimental tests will continue and time analysis will be conducted on the parallel computer IBM SP/2 in order to evaluate the new criteria of migration and termination proposed. We will also conduct deeper analysis for optimization of the parameters presented in the section above.

References

1. BALL, M.O.; MAGNANTI, T.L.; MONNA, C.L. and NENHAUSER, G.L., 1995, Network Routing, *Handbook in Op. Res. and Manag. Sc.*, Vol 8, Elsevier Science.
2. BODIN, L.D.; GOLDEN, L.; ASSAD, A.A. and BALL, M., 1983, Routing and Scheduling of Vehicles and Crews, The State of the Art,*Comp. and Op. Res.*, Vol 10, 63-211.
3. DRUMMOND, L. M. A. and BARBOSA, V. C., 1996, Distributed Breakpoint Detection in Message-passing Programs, *J. of Par. and Dist. Comp.*, 39, 153-167.
4. GENDRAU, M.; HERTZ, A. and LAPORTE, G., 1992, New Insertion Post-Optimization Procedures for the Traveling Salesman Problem, *Oper. Res.* 40, 1086-1094.
5. GLOVER, F., 1995, Scatter Search and Star-Paths: Beyond the Genetic Metaphor, *OR SPECTRUM*, Vol 17, Issue 2/3.
6. GLOVER, F., 1997, Tabu Search and Adaptive Memory Programming: Advances, Applications and Challenges. *Interfaces in Comp. Sc. and Oper. Res.*, 1-76, Kluwer Academic Publishers.
7. GOLDEN, B.; ASSAD, A.; LEVY, L. and GHEYSENS, F.G., 1984, The Fleet Size and Mix Vehicle Routing Problem, *Comp. and Op. Res.*, Vol 11, 49-66.
8. HOLLAND, J.H., 1976, Adaptation in Natural and Artificial Systems, *University of Michigan Press, Ann Arbor.*
9. SNIR, M.; OTTO, S. W.; HUSS-LEDERMAN, S.; WALKER, D. W. and DONGARRA, J., 1996, MPI: The Complete Reference, *The MIT Press.*
10. OCHI, L.S.; DRUMMOND, L.M.A. and FIGUEIREDO, R.M., 1997, Design and Implementation of a Parallel Genetic Algorithm for the Traveling Purchaser Problem, *ACM Symp. on Applied Computing*, 257-263.
11. OCHI, L.S.; RABELO, P.G. and MACULAN, N., 1997, A New Genetic Metaheuristic for the Clustered Traveling Salesman Problem, *II Metaheuristic International Conf.*, 59-64.
12. OCHI, L. S., VIANNA, D. S.; DRUMMOND, L. M. A. and VICTOR, A. O., 1998, An Evolutionary Hybrid Metaheuristic for Solving the Vehicle Routing Problem with Heterogeneous Fleet, to appear in Euro. Workshop on Genetic Programming.
13. REEVES, R., 1993, Modern Heuristic Technics for Combinatorial Problems, *Blackwell Scientific Publications.*
14. RIBEIRO, J. L., 1995, An Object-oriented Programming Environment for Parallel Genetic Algorithm, Ph.D. Thesis, *Dep. of Comp. Sc., Univ. College London.*
15. ROCHAT, Y. and TAILLARD, E.D., 1995, Probabilistic Diversification and Intensification on Local Search for Vehicle Routing, *J. of Heuristics*, 147-167.
16. RYAN, D.M.; HJORRING and GLOVER, F., 1993, Extensions of the Petal Method for Vehicle Routing, *J.Op.Res.*, Vol 44(3), 289-296.
17. TAILLARD, E.D., 1996, A Heuristic Column Generation Method for the Heterogeneous Fleet, *Publication CRT-03-96, Université de Montreal.*
18. WHITLEY, D.; STARKWEATHER, T. and SHANER, D., 1991, The traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, NY.

This article was processed using the LaTeX macro package with LLNCS style