On Improving Evolutionary Algorithms by Using Data Mining for the Oil Collector Vehicle Routing Problem

Fábio Dalboni, Lúcia M. A. Drummond
and Luiz Satoru Ochi
Department of Computer Science - Fluminense Federal University
Rua Passo da Pátria, 156, Bloco E, Niterói, Rio de Janeiro, 24210-240, BRAZIL
e-mail: {lucia,satoru}@dcc.ic.uff.br

April 22, 2003

Abstract

This paper presents some proposals to improve the performance of an evolutive algorithm applied to a problem known as the "Oil Collector Vehicle Routing Problem" (OCVRP) used in a Brazilian oil company to collect oil in artificial lift wells. This paper focus on the analysis of evolutive algorithms added with procedures of local search and data mining. Three algorithms were developed: a basic model of genetic algorithm (BGA), another using additional procedures of local search (GA1) and a third that employed procedures of local search and data mining (GA2). Computational results show that the algorithm GA2 outperforms the other versions concerning the quality of solutions, indicating that the inclusion of a data mining procedure can significantly improve the performance of this kind of metaheuristic.

Keywords: Evolutionary Algorithms, Data Mining, Vehicle Routing Problem.

1 Introduction

Concerning the oil exploitation of onshore wells there is a class of them called artificial lift wells where the use of artificial elevation methods to exploit oil is necessary. In this case a fixed system of beam pumping is used when the well has a high productivity. Because the product is not renewable, the quantity of oil diminishes until it becomes economically unfeasible to keep the equipment allocated to that location permanently. The oil exploitation of wells of low productivity can be done by mobile equipment coupled to a truck, that visits these wells to exploit oil. Usually, the collector mobile is not able to visit all these wells in a unique day. In this way this problem, called the Oil Collector Vehicle Routing Problem (OCVRP), can be described as a vehicle routing problem beginning and finishing at a local origin, known as the oil treatment station, visiting all wells of a subset J of N, where N is the set of wells, aiming the maximization of the quantity of collected oil without violating the constraint of time to cover the tour [2].

The OCVRP can be associated with a symmetric graph $G(N, E)$, where each vertex represents a well and the edges correspond to the roads existing between each pair of wells. Each vertex $j \in N$ has a level of oil q_j and each edge $\{i, j\}$ of E has an associated value t_{ij} that represents the time to cover this edge.

The OCVRP can be considered as a generalization of the Travelling Salesman Problem (TSP) and is classified as a NP-hard problem. In literature, to our best knowledge there is only one paper about the OCVRP, in which the authors present a basic genetic algorithm for solving this problem

[2]. There are similar problems to the OCVRP such as the travelling purchaser problem [6], the prize collecting problem [3] and the orienteering problem [8]. Because of the high computational complexity of the OCVRP, approximate methods or heuristics are good alternatives to find solutions of good qualities for this problem. The literature has shown that this kind of problem can be solved efficiently through evolutive metaheuristics [5] [10]. In this work, three versions of evolutive algorithms are proposed to solve the OCVRP. Initially a basic GA is developed (BGA) and then other versions including procedures of local search (GA1), and local search with data mining (GA2) are proposed. In Section 2 the proposed algorithms are presented. Section 3 presents the computational results. Finally, Section 4 concludes the paper.

2 Evolutive Algorithms

Evolutive algorithms and more specifically its most popular model, genetic algorithms, are iterative heuristic procedures where at each iteration a population of solutions is generated. In these algorithms, the process of generation of new solutions are usually done through combinations of existing solutions. In case of GAs, the usual reproduction operators are known as mutation and crossover.

GAs in particular are techniques already very known in literature and are used to solve problems considered hard in several areas, although they have not shown to be much efficient to solve combinatorial optimization problems of high computational complexity in their basic form.

In order to improve the performance of GAs, researchers have proposed variations of GAs such as memetic algorithms [9], scatter search [8], and population heuristics [4]. Many papers propose the use of procedures of local search to improve the performance of GAs.

In this paper we propose not only a basic genetic algorithm for solving the OCVRP but versions incorporated with procedures of local search and data mining as well.

2.1 Basic Genetic Algorithm (BGA)

In this algorithm, concepts of the basic version of GAs were employed. The steps of representation of a solution, generation of a initial population of solutions, evaluation of solutions (fitness function), reproduction of new solutions and stopping criterion are described as follows.

In order to represent a solution, a list of integer numbers of variable size, whose maximum size is n, where $n = |N|$ represents the number of wells existing with enough oil level to justify a visiting of the collector vehicle, is used. The list is initiated with zero (origin $i = 0$) and each integer number of the list is associated with a well. The position of a well in this list represents its order in the visiting. Thus, a solution composed of five wells to be visited from 0, could be 0-6-3-4-5-2-0. In this solution the visiting would occur in the following order: 6,3,4,5,2.

To generate a set of feasible initial solutions, a criterion based on the priority of visiting of each well j not yet visited is used. This priority is represented by the quotient $priority(i, j) = (level(j)/time(j))$ where $level(j)$ and $time(i, j)$ represent respectively the level of oil available in j and the time spent to travel from the well i included more recently to the well j in the partial solution. Thus, let i be the most recently included well in the solution (that initiates with vertex zero), a candidate list (CL) composed of all wells not yet selected is created and from it the k best candidates are selected, where k is an entry parameter, what constitutes a restricted candidate list(RCL). From the RCL a vertex is chosen randomly.

This procedure is repeated generating new lists RCL until a complete solution is reached. This random choice on RCL allows to generate different initial solutions of OCRL.

The goal of this problem is the maximization of the oil volume collected in the visited wells respecting the time constraint to conclude the route. The fitness of a solution is measured by the objective function of the problem. For each reproduction phase, the classical operators of mutation and crossover were employed.

2.2 Genetic algorithms with local search (GA1)

In basic genetic algorithms such as the BGA, the inefficiency of operators such as crossover to generate local optimal solutions of good quality in combinatorial problems is well known.

One possibility to reduce or eliminate this limitation would be the replacement of the operator crossover by a more efficient one or the addition of procedures of local search in BGA.

In this context, we propose a new version of GA, called GA1, using a nearest neighborhood heuristic (NNH) instead of the operator crossover and an additional procedure of local search.

The algorithm NNH generates a solution (offspring) from p solutions (parents) of the current population, where $p \geq 2$. Chosen p parents, for each vertex (well) belonging to at least one parent, a list of its adjacent vertexes (neighbors) in the p vertexes is built. To generate an offspring from an origin $i = 0$ the following steps are executed. Considering the vertex most recently allocated in the solution, called *current*, the nearest vertex of its adjacent list is selected. If it is already present in the partial solution, the next nearest is chosen. If the list empties, a vertex not yet allocated is selected randomly. To generate more than one offspring, we can adopt a restricted candidate list (RCL) composed of the s nearest neighbors of *current*, from which a vertex is chosen randomly. This random choice allows p parents to generate different offspring's.

In the local search procedure here proposed, for a feasible solution of the problem (base solution), for each vertex k of this solution a list l_k of the r closest neighbors not present in the current solution is created. Then, from the base solution, a replacement of a vertex k is executed one at a time, exchanging it with one of the list l_k. At each exchanging, the fitness is figured out and the best solution may be updated. In order to accomplish the next exchange the base solution is re-considered again. The values p, s and r are entry parameters.

The local search procedure is executed whenever the GA updates the best solution.

The stopping criterion usually adopted is the maximum number of iterations of GA and/or the number of consecutive iterations without improvement of the best generated solution so far.

2.3 Genetic algorithm with local search and data mining (GA2)

Concerning the improvement of GAs, one possibility is to take advantage of the information of the best feasible solutions generated so far (elite solutions).

To implement this idea, schemata notions could be used, where part of the genes of a chromosome of the GA would be fixed. However, few practical contributions exist because of the difficulty of definition of the way and the moment a gene should be fixed to a constant value to improve the performance of the algorithm.

In this paper, the use of data mining concepts is proposed in order to obtain relevant information found in the best generated solutions by the GA.

The inclusion of the Data Mining procedure aims to improve the performance of the GA. This procedure was based on the Apriori algorithm proposed by Rakesh Agrawal [1], but with some adaptations that turned it more efficient to our proposals.

The adaptations allowed for the data mining algorithm to recognize the most usual sequences of vertexes (wells) belonging to the elite set considering the different orders of theses wells in these sequences [1].

Each generated sequence of vertexes (wells) that belongs to the elite set and that satisfies the support, given as an entry parameter of the program, corresponds to an *itemset* [1].

The data mining procedure is executed whenever the *best* solution is updated by the iterations of the GA.

In the following step, after the execution of the data mining procedure, each solution of the current population is analyzed and the feasibility of inclusion of each *itemset* generated is verified. If the included *itemset* does not result in an improvement in the fitness of the solution, the previous state is recovered and the following *itemset* is verified. An *itemset* is included in the best position of the current route, i.e., all pairs of vertexes of the solution are analyzed, and the *itemset* is included between the pair that offers the best value. Eventually, some vertexes will have to be discarded with the inclusion of an *itemset* in order not to violate the time constraint.

3 Computational Results

To our best knowledge, there are not public sites with instances of this problem. Thus, to evaluate the algorithms here proposed a set of test problems of different dimensions was generated. Ten instances were created for each dimension: 100, 120, 300, 500 and 1000 vertexes. Each instance from 100 to 500 vertexes was executed 10 times and the instances of 1000 vertexes once for each algorithm. In all instances there is an origin vertex at which every solution must begin and finish. The level of oil of each well and the time to cover the distance between each pair of wells were generated randomly.

The average results are shown in tables 1 and 2. We used the following parameters in the algorithms: number of genes of a chromosome is the number of wells of OCVRP, the stopping criterion is 500 iterations, size of population is 200, size of the elite set is 5 solutions, maximum time to cover the route is 8 hours.

In tables 1 and 2, the first column presents the algorithms and the dimension of each instance, in the other columns the value of each cell represents the average error concerning the best solution and the best time obtained by each instance varying from 1 to 10, considering all proposed algorithms.

Table 1 shows that the version with data mining always obtains the best solution and that this superiority increases with the dimension of the problem. This fact demonstrates the high potential of this technique for solving real problems of high dimensions. Concerning the required computational time, BGA, as already expected, presented the best times and GA2 the worst (see Table 2).

Although the execution times required by GA2 are much worse than those obtained by the other versions, we could observe in our empirical tests that in GA2, the best solution has always been obtained before the iteration 300 while in the other versions, in many cases, the best solutions were obtained in a number of iterations close to 500.

4 Conclusions

Based on tables 1 and 2, we can conclude that the proposal of incorporation of procedures of data mining in an evolutive algorithm can improve its performance significantly concerning the quality of generated solutions, mainly in instances with high dimensions. This improvement can be explained by the fact that the procedure of data mining reduces the search space in highly combinatorial problems. Consequently, best quality local optima are explored, increasing the chances of reaching a global optimum in such problems.

Algorithm-dimension *instances*	1	2	3	4	5	6	7	8	9	10
BGA100	31.9	30.68	31.02	29.5	26.53	48.81	47.02	19.04	48.85	45.24
GA1-100	**0.00**	**0.00**	**0.00**	0.19	1.25	7.15	10.57	9.18	13.98	12.2
GA2-100	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
BGA-120	31.94	33.58	31.1	21.13	21.66	52.49	56.58	50.64	47.28	49.08
GA1-120	**0.00**	0.16	0.89	3.2	2.05	13.14	21.76	16.64	16.64	13.71
GA2-120	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
BGA-300	22.07	22.25	21.89	24.51	27.79	49.15	48.62	54.48	49.1	51.61
GA1-300	12.79	15.34	14.07	17.51	17.83	24.21	26.13	26.96	27.75	26.39
GA2-300	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
BGA-500	55.86	43.27	51.04	44.53	47.24	50.61	44.57	46.1	46.95	47.93
GA1-500	30.57	27.87	30.46	22.97	28.7	28.68	32.27	23.54	28.07	30.74
GA2-500	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
BGA-1000	49.88	41.48	42.15	48.71	45.85	46.37	43.94	43.89	46.28	56.35
GA1-1000	33.57	24.48	29.24	29.96	32.38	30.67	29.89	27.07	28.94	33.77
GA2-1000	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table 1: Average performance of the proposed algorithms concerning the quality of solutions

In future work, we intend to develop more economical versions of the algorithm GA2 where the selected *itemsets* by the data mining procedure are inserted only in a small percentage of the best individuals of the population. We also intend to adjust the maximum number of iterations of GA2. Thus, we expect to obtain a similar algorithm to GA2 concerning the quality of the generated solutions with a drastical reduction of the execution times.

Other proposals include the development of genetic algorithms with procedures of data mining to solve other optimization problems and the implementation of parallel versions of GA2 to reduce the computational time.

References

[1] Agrawal, R., Imielinski, T., and Swami, A. (1993), Mining association rules between sets of items in large databases. Proc. of the ACM SIGMOD Conf. on Management of Data, Washington, DC, USA, 207-216.

[2] Aloise, D. J., Barros, J. A, and Souza, M. (2000), A genetic algorithm for a Oil Retrieval System (In Portuguese). Proc. of the XXXII Brazilian Symposium on Operations Research, São Paulo, Brazil.

[3] Balas, E. (1989), The Prize Collecting Traveling Salesman Problem. Networks 19,621-636.

[4] Beasley, J. (2002), Population Heuristics. In Handbook of Applied Optimization, Pardalos, P. M. and Resende, M.G.C. (eds), Oxford University Press, Oxford, 138-157.

[5] Drummond, L. M. A., Vianna, D. S., and Ochi, L. S. (1998), An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet. Lecture Notes in Computer Science 1391, 187-195.

[6] Drummond, L. M. A., Vianna, L. S., Silva, M. B., and Ochi, L. S. (2002), Distributed Parallel Metaheuristic based on GRASP and VNS for solving The Traveling Purchaser Problem. Proc. of the 2002 Int. Conf. on Parallel and Distributed Systems, Taiwan, China, 257-263.

Algorithm-dimension *instances*	1	2	3	4	5	6	7	8	9	10
BGA-100	10	9	9	9	10	5	6	5	5	5
GA1-100	35	35	35	33	30	18	8	17	18	26
GA2-100	69	73	64	171	152	82	53	61	75	87
BGA-120	15	15	14	15	14	9	9	9	9	9
GA1-120	52	49	47	41	41	26	26	26	27	27
GA2-120	96	128	279	189	175	106	110	93	131	120
BGA-300	342	341	341	342	343	342	342	342	346	374
GA1-300	468	466	467	463	460	447	447	450	449	448
GA2-300	8836	10082	9906	10173	10380	1743	2522	2369	2316	2462
BGA-500	3298	3371	3187	3107	3111	3123	3135	3130	3146	3119
GA1-500	3800	3825	3842	3840	3833	3816	3857	3905	3866	3888
GA2-500	16711	11961	16079	8891	16651	15434	15811	11914	16884	13768
BGA-1000	36358	35722	33020	33941	32912	33816	33415	33165	33818	35376
GA1-1000	48685	46720	46667	48717	49109	47327	47050	47136	47469	47063
GA2-1000	181215	141257	119212	104198	113472	112071	101349	76072	62944	82133

Table 2: Average performance of the proposed algorithms concerning the execution time in seconds

[7] Glover, F., Laguna, M., and Marti, R. (2000), Fundamentals of Scatter Search and Path Relinking. Control and Cybernetics 39(3), 653-684.

[8] Golden, B.L., Levy, L. & Vohra, R. (1987). The Orienteering Problem. Naval Research Logistics 24, 307-318.

[9] Moscato, P. (1989), On Evolution, Search, Optimization Algorithms and Martial Arts: Towards Memetic Algorithms. Report 826, Caltech Concurrent Computation Program, California Institute of Technology.

[10] Ochi, L. S., Vianna, D. S., and Drummond, L. M. A. (2001), An Asynchronous Parallel Metaheuristic for the Period Vehicle Routing Problem. Future Generation Computer Systems 17, 379-386.