Making Heuristics Faster with Data Mining

Daniel Martins

[1] Departamento de Ciência da Computação – Universidade Federal Fluminense (UFF)
Niterói, Rio de Janeiro, Brazil
dmartins@id.uff.br

, **Advisers: Isabel Rosseti, Simone L. Martins and Alexandre Plastino**

Departamento de Ciência da Computação – Universidade Federal Fluminense (UFF)
Niterói, Rio de Janeiro, Brazil
{rosseti,simone,plastino}@ic.uff.br

Resumo. *O desafio desse trabalho é introduzir um procedimento de mineração de dados a uma heurística que é considerada o estado da arte para um problema específico, de forma a obter evidências que, quando esta abordagem é capaz de atingir a solução ótima, ou uma solução sub-ótima com pouca chance de melhora, os padrões minerados podem ser usados para guiar a busca pela solução ótima ou sub-ótima em menor tempo computacional.*

Palavras Chave: Metaheurísticas Híbridas, Problema das p-Medianas, Mineração de Dados.

Área Principal: Metaheurísticas

Abstract. *The challenge of this work is to introduce a data mining procedure into a state-of-the-art heuristic for a specific problem in order to give evidences that, when a technique is able to reach the optimal solution, or a near-optimal solution with little chance of improvements, the mined patterns could be used to guide the search for the optimal or near optimal solutions in less computational time.*

Keywords: Hybrid Metaheuristics, p-Median Problem, Data Mining.

1. Introduction

Metaheuristics represent an important class of techniques for obtaining good solutions, in reasonable time, for hard combinatorial optimization problems. They are general purpose high-level procedures that can be instantiated to explore efficiently the solution space of a specific optimization problem [Osman and Laporte (1996)]. Tabu search, genetic algorithms, simulated annealing, ant systems and GRASP are examples of metaheuristics and have been applied to real-life problems of several areas of science over the last decades.

An important topic in metaheuristics research is the development of hybrid metaheuristics [Talbi (2002)]. Such methods result from the combination of concepts and procedures of different metaheuristics or from the combination of metaheuristics with concepts and processes from other research areas responsible for performing specific tasks that can improve the original technique. An instance of the latter, and subject of this work, is the hybrid version of a multistart heuristic that incorporates a data mining process.

Data mining refers to the automatic extraction of knowledge from datasets [Han and Kamber (2006), Witten and Frank (2005)]. The extracted knowledge, expressed in terms of patterns or rules, represents important features of the dataset at hand. Hence, data mining provides a means to better understand features implicit in raw data, which is fundamental in a decision-making process.

The challenge of this work is to introduce a data mining procedure into a state-of-the-art heuristic for a specific problem in order to give evidences that, when a technique is able to reach the optimal solution, or a near-optimal solution with little chance of improvements, the mined patterns could be used to guide the search for the optimal or near optimal solutions in less computational time. We chose, as the state-of-the-art algorithm to be the base of our study, the heuristic proposed in [Resende and Werneck (2004)] for the classical p-median problem, referred as Hybrid Heuristic (HH).

We then developed the Data Mining Hybrid Heuristic (DM-HH), introducing a data mining procedure. Computational experiments, comparing the HH and DM-HH strategies showed that DM-HH was able to reach optimal and near-optimal solutions, on average, 27.32% faster than the original strategy.

The remaining of this paper is organized as follows. In Section 2, we present the p-median problem and review the main concepts and the structure of the state-of-the-art Hybrid Heuristic for this combinatorial problem. The Data Mining Hybrid Heuristic, proposed in this work, is presented in Section 3. The computational experiments conducted to compare both strategies are reported and discussed in Section 4. In Section 5, we illustrate and justify the behavior of both strategies with some additional analysis. Finally, concluding remarks and some future works are pointed out in Section 6.

2. Multistart Hybrid Heuristic

Given a set F of m potential facilities, a set U of n customers, a distance function $d : U \times F \rightarrow \mathbb{R}$, and a constant $p \leq m$, the p-median problem consists of determining which p facilities to open so as to minimize the sum of the distances from each costumer to its closest open facility. It is a well-known NP-hard problem [Kariv and Hakimi (1979)].

In [Resende and Werneck (2004)], Resende and Werneck proposed a state-of-art multistart hybrid heuristic for the p-median problem, that combined elements of several

traditional metaheuristics to find near-optimal solutions to this problem. Each iteration of this algorithm starts with a randomized construction of a solution. The construction method starts with an empty solution and adds facilities, one at time, choosing the most profitable, among $\lceil log_2(m/p) \rceil$ possible insertions, chosen uniformly at random. This solution then is submitted to local search. After, a solution chosen from the pool of elite solutions, made with some of the best solutions found in previous iterations, is combined with the solution obtained by the local search through a process called path-relinking [Glover et al. (2000)]. After all iterations are completed, the second phase called post-optimization is executed, in which elite solutions are combined with each other, and the best solution found after the post-optimization phase execution is taken as result.

3. Data Mining Hybrid Heuristic

In this section, we propose a new version of the hybrid heuristic (HH) which incorporates a data mining process, called DM-HH, to solve the p-median problem. The basic concept of incorporating a data mining process is that patterns found in high quality solutions obtained in earlier iterations can be used to conduct and improve the search process.

The DM-HH procedure is composed of two phases. The first one consists of executing n pure HH iterations to obtain a set of different solutions. The d best solutions from this set of solutions compose the elite set for mining.

After this first phase, the data mining process is applied. It is responsible for extracting a set of patterns from the elite set. The patterns to be mined are sets of elements that frequently appear in solutions from the elite set. This extraction of patterns characterizes a frequent itemset mining application [Han and Kamber (2006)]. A frequent itemset mined with support $s\%$ represents a set of elements that occur in $s\%$ of the elite solutions.

Next, the second phase is performed. In this part, another n slightly different HH iterations are executed. In these n iterations, an adapted construction phase starts building a solution guided by a mined pattern selected from the set of mined patterns. Initially, all elements of the selected pattern are inserted into the partial solution, from which a complete solution will be built executing the standard construction procedure. This way, all constructed solutions will contain the elements of the selected pattern.

The extraction of patterns from the elite set corresponds to the well-known frequent itemset mining (FIM) task. The FIM problem can be defined as follows.

Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items. A transaction is a subset of I and a dataset D is a set of transactions. A frequent itemset F, with support s, is a subset of I which occurs in at least $s\%$ of the transactions in D. FIM consists of extracting all frequent itemset from a dataset D with a minimum support specified as a parameter.

In this work, the patterns to be mined are sets of elements that commonly appear in sub-optimal solutions of the p-median problem. Each transaction of the dataset represents a sub-optimal solution of the elite set. A frequent itemset mined with support $s\%$ represents a set of locations that occur in $s\%$ of the elite solutions. To execute this task, we adopted the FPmax* algorithm [Grahne and Zhu (2003)].

4. Computational Experiments

In this section, the computational results obtained for HH and DM-HH are presented. The strategies were evaluated using three classes of instances. The first class, named ORLIB, consists of 40 instances and was taken from the OR-Library [Beasley (1985)], identified by *pmed01* to *pmed40*. The number of nodes (customers) in each instance varies from 100 to 900, and the value of p ranges from 5 to 200.

Instances of the second class, named TSP and available at the TSPLIB [Reinelt (1991)], are sets of points on the plane. Every point is considered both a potential facility and a customer, and the cost of assigning customer c to facility f is simply the Euclidean distance between the points representing c and f (e.g. the costs are real values). From the TSP class, we considered the $FL1400$ instances, with 1400 nodes and with several different values for p (number of facilities to open).

The third class we study is named RW. Originally proposed in [Resende and Werneck (2003)], it corresponds to random distance matrices. In every case, the number of potential facilities (m) is equal to the number of customers (n). The distance between each facility and each customer has an integer value taken uniformly at random from the interval $[1, n]$. Six different values of n were considered: 100, 250, 500, 1000, 1500, and 2000. In each case, several values of p were tested.

The algorithms were implemented in C++ and compiled with g++ (GCC) 4.2.3. The tests were performed on a 2.4 GHz Intel Core 2 Quad CPU Q6600 with 3 Gbytes of RAM, running Linux Kernel 2.6.24.

Both HH and DM-HH were run 9 times with different random seeds. Each strategy executed 500 iterations. The size of the elite set for mining and the size of the set of patterns were set to 10, a set of facilities was considered a pattern if it was present in at least two of the elite solutions, values set after tuning experiments.

Table 1 presents the results related to execution time of both strategies. In this table, the first column presents the class of the working instances. Second to fifth columns show the percentual difference between the HH and DM-HH average times in relation to the HH average time, for each instance class, in average, standard deviation, best value and worst value, respectively. In Table 2, the results related to the quality of the obtained solutions are shown. The first column presents the class name of the working instances, the second and fourth columns present the deviation value of the best cost obtained by HH and DM-HH related to the best known value, in average for each class. The third and fifth columns present the deviation value of the average cost obtained by both strategy in average for each class. And the sixth column presents the number of instances that DM-HH reached better solution costs in average, for each class.

The deviation value is computed as follows:

$$dev = \frac{(HeuristicCost - BestCost)}{BestCost} \times 100, \tag{1}$$

where $HeuristicCost$ is the (best or average) cost obtained by the heuristic technique and the $BestCost$ is the optimal or best known value for the working instance.

When executed for the 40 instances from the ORLIB class, both HH and DM-HH reached the optimal solution in all 9 runs. We can observe that DM-HH was always

faster than HH and that the standard deviations are quite small. On average, DM-HH was 25.06% faster than the HH strategy for the ORLIB instances.

Table 1. Time reduction by instance class

Class	Avg	Std Dev	Best	Worst
ORLIB	25.06	9.65	36.86	1.08
RW	28.26	6.71	38.17	12.66
FL1400	30.03	5.78	38.13	18.12

When executed for the 45 instances from the RW class, both HH and DM-HH reached the best known solutions in all 9 runs for 23 instances. For the other 22 instances, they obtained slightly different solutions.

Table 2. Quality deviation for the RW and FL1400 classes

	HH		DM-HH		
Class	Best	Avg	Best	Avg	# of wins
RW	0.051	0.152	0.002	0.107	15
FL1400	0.001	0.018	0.006	0.024	4

Out of the 22 instances for which HH and DM-HH presented different results, in 10 instances, HH reached the best know value in all 9 runs and the DM-HH reached this result in 18 instances. The DM-HH strategy found 11 better results for best values and 4 were found by HH. Considering the average results, DM-HH found 15 better values and HH found 7. These results show that the DM-HH strategy was able to improve slightly the results obtained by HH for the RW class. In terms of computational time, again, the DM-HH strategy was faster than HH. On average, DH-MM was 28.26% faster.

As of the results related to the quality of the solutions obtained by HH and DM-HH when evaluated for the 18 instances from the FL1400 class, both strategy reached the best known solutions in all 9 runs for just 3 instances. For the other 15, they obtained slightly different solutions. The HH strategy found 6 better results for best values and just one was found by DM-HH. Considering the average results, HH found 7 better values and HH found 4. Differently from ORLIB and RW classes, these results show that the HH strategy, for the FL1400 class, was able to obtain slightly better results than DM-HH.

However, for the time analysis, DM-HH always improved the HH performance, as we observe that, once more, the DM-HH strategy was faster than HH, this time, 30.03%.

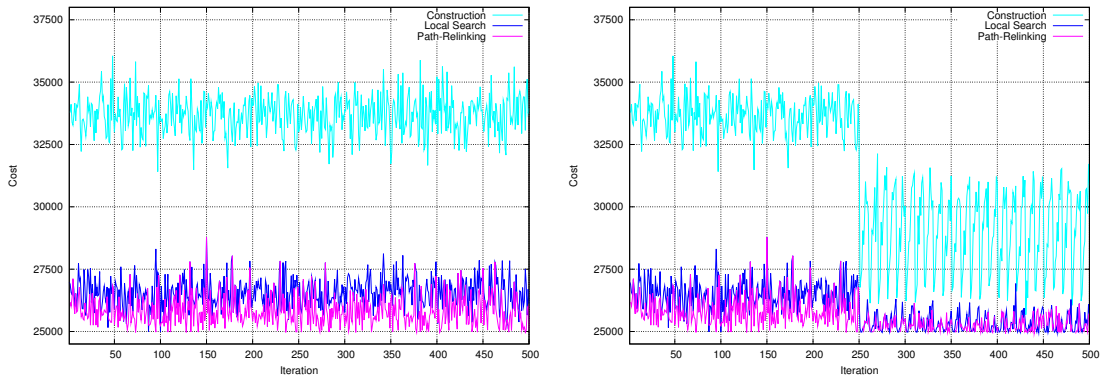
5. Strategies Behavior Analysis

In this section, we present some additional analysis of computational experiments performed to illustrate the behavior of both strategies.

Figures 1a and 1b present the behavior of the construction, local search and path-relinking phases, in terms of the cost values obtained, by HH and DM-HH through the execution of 500 iterations, for the specific instance rw1000-p25.

In Figure 1a, we observe that the behavior of the construction, local search and path-relinking phases performed in HH looks the same through all iterations.

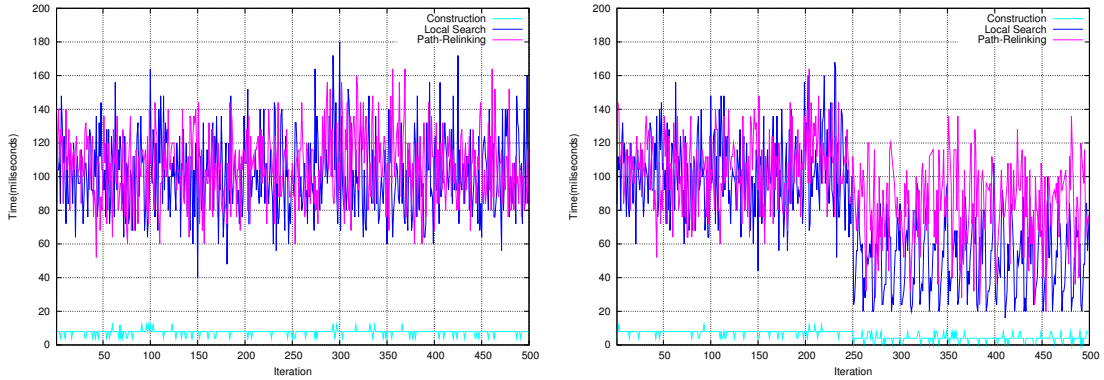
Figure 1b shows that the DM-HH strategy provides an improvement in the quality of the solutions reached by the construction, local search and path-relinking phases after iteration 250, where DM-HH starts to use the patterns found by the data mining procedure.



(a) HH (b) DM-HH

Figure 1. Cost analysis for one execution of rw1000-p25

Figures 2a and 2b show the behavior of the construction, local search and path-relinking phases, for both strategies HH and DM-HH in terms of the computational time, through the execution of 500 iterations, for the same instance rw1000-p25.



(a) HH (b) DM-HH

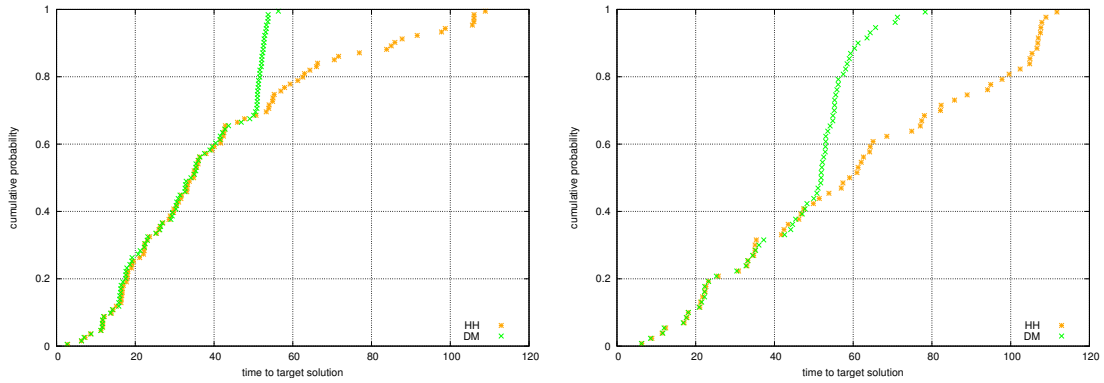
Figure 2. Time analysis for one execution of rw1000-p25

We can clearly see that computational times of all phases dropped substantially after starting to use the patterns generated by the data mining procedure. The construction phase demands less computational time because it starts from a solution partially built using the obtained patterns. The necessary effort required by the local search procedure to find a local optimum decreases due to the better solutions provided by the construction phase. As the solutions generated after the local search procedure present better cost in the iterations which use the data mining patterns, they are more similar to the solutions in the path-relinking pool and the path-relinking procedure takes less time to execute.

Figures 3a and 3b show another comparison between HH and DM-HH strategies, based on $Time\text{-}to\text{-}target$ (TTT) plots [Aiex et al. (2007)], which are used to analyze the behavior of randomized algorithms.

A TTT plot is generated, initially, by executing an algorithm several times and measuring the time required to reach a solution at least as good as a target solution. We executed each strategy a hundred times. Then, the i-th sorted running time t_i is associated with a probability $p_i = (i - 1/2)/100$ and the points $z_i = (t_i, p_i)$, for $i = 1, ..., 100$ are plotted. Each plotted point indicates the probability (vertical axis) for the strategy to achieve the target solution in the indicated time (horizontal axis). The plots presented in Figures 3a and 3b were generated by the execution of HH and DM-HH, for instance rw1000-p25, using an easy target (value 24964), and a more difficult one (value 24923).

For the easy target, we observe in Figure 3a that HH and DM-HH present similar behaviors until about 50 seconds when the probability for DM-HH to find the target value starts to be greater than for HH. This happens because, until the data mining procedure is executed in DM-HH, both strategies obtain the same solution in each iteration, but DM-HH starts to find the target value faster when the patterns are used.



(a) Easy target (b) Difficult target

Figure 3. Time-to-target plots for rw1000-p25

For the difficult target, Figure 3b, we observe that DM-HH behaves better than HH. This plot indicates that DM-HH is able to reach difficult solutions faster than HH.

This analysis shows that DM-HH was able to reach good quality solutions much faster than the original strategy. It also demonstrates that a sophisticated heuristic like HH can benefit from the incorporation of a data mining procedure.

6. Conclusions

In this work we developed the DM-HH, a data mining version of a hybrid and state-of-the-art multistart heuristic to solve the p-median problem. Computational experiments, conducted on a set of instances from the literature, showed that the new version of the hybrid heuristic was able to reach optimal and near-optimal solutions, on average, 27.32% faster than the original strategy, which represents significant savings on execution times.

A secondary contribution of this work was to show that sophisticated heuristics,

improved with a memory-based intensification mechanism, like the path-relinking technique, could benefit from the incorporation of a data mining procedure.

These encouraging results motivate us, as future work, to try to introduce into other metaheuristics the idea of extracting patterns from sub-optimal solutions using data mining techniques and exploring them in search procedures.

7. Comments

This work is part of a research project on hybrid metaheuristics with data mining. The student worked on the development of the data mining version (DM-HH) of the hybrid heuristic (HH), developed in [Resende and Werneck (2004)], and conducted the computational experiments and analisys which compare these strategies. He is the first author of an extended version of this paper which has been submitted to the Annals of Operations Research Journal.

References

R. Aiex, M. G. C. Resende, and C. Ribeiro, *TTT plots: a perl program to create time-to-target plots*, Optimization Letters 4, pp. 355–366 (2007)

J. E. Beasley, *A note on solving large p-median problems*, European Journal of Operational Research 21, pp. 270–273 (1985).

F. Glover, M. Laguna, and R. Martí, *Fundamentals of scatter search and path-relinking* Control and Cybernetics 39, pp. 653–684 (2000).

G. Grahne and J. Zhu, *Efficiently using prefix-trees in mining frequent item-sets*, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (2003).

J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2^{nd} Ed., Morgan Kaufmann Publishers (2006).

O. Kariv and L. Hakimi, *An algorithmic approach to network location problems, Part II: The p-medians*, SIAM Journal of Applied Mathematics 37, pp. 539–560 (1979).

I. Osman and G. Laporte, *Metaheuristics: A bibliography*, Annals of Operations Research 63, pp. 513–623 (1996).

G. Reinelt, *TSPLIB: A traveling salesman problem library*, ORSA Journal on Computing 3, pp. 376–384 (1991), *http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/*.

M. G. C. Resende and R. F. Werneck, *A hybrid heuristic for the p-median problem*, Journal of Heuristics 10, pp. 59–88 (2004).

M. G. C. Resende and R .F . Werneck, *On the implementation of a swap-based local search procedure for the p-median problem*, Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments – ALENEX03, pp. 119–127 (2003).

E. G. Talbi, *A taxonomy of hybrid metaheuristics*, Journal of Heuristics 8, pp. 541–564 (2002).

I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. 2^{nd} Ed., Morgan Kaufmann Publishers (2005).