A Hybrid Heuristic for the k-medoids Clustering Problem

Mariá C. V. Nascimento*
Instituto de Ciência e Tecnologia - UNIFESP
Rua Talim, 330, CEP: 12230-280, São José dos Campos - SP
mcv.nascimento@unifesp.br

Franklina M. B. Toledo
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo,
Caixa Postal 668, São Carlos-SP, CEP 13560-970, Brazil
fran@icmc.usp.br

André C. P. L. F. de Carvalho
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo,
Caixa Postal 668, São Carlos-SP, CEP 13560-970, Brazil
andre@icmc.usp.br

ABSTRACT

Clustering is an important tool for data analysis, since it allows the exploration of datasets with no or very little prior information. Its main goal is to group a set of data based on their similarity (dissimilarity). A well known mathematical formulation for clustering is the k-medoids problem. Current versions of k-medoids rely on heuristics, with good results reported in the literature. However, few methods that analyze the quality of the partitions found by the heuristics have been proposed. In this paper, we propose a hybrid Lagrangian heuristic for the k-medoids. We compare the performance of the proposed Lagrangian heuristic with other heuristics for the k-medoids problem found in literature. Experimental results presented that the proposed Lagrangian heuristic outperformed the other algorithms.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Heuristic methods; I.5.3 [**Clustering**]: Metrics—*Algorithms*

Keywords

clustering, bioinformatics, heuristic, PAM, integer programming

1 Introduction

Clustering deals with the unsupervised classification of patterns (observations, data items or feature vectors) into groups (clusters) (14). Clustering algorithms can be roughly divided into two main approaches: partitioning clustering algorithms and hierarchical clustering algorithms. Partitioning clustering algorithms look for a partition that optimizes a given clustering criterion (14). Hierarchical clustering algorithms produce a nested series of partitions based on a criterion that either combines (agglomerative algorithms) or divides clusters (divisive algorithms) based on a similarity measure. There are many areas where clustering algorithms have been successfully used, such as pattern recognition, grouping, decision making, data mining and pattern classification (1; 4; 27).

Several clustering algorithms have been proposed in the last decades (19; 9; 21). Good surveys on clustering algorithms can be found, e.g., in (14; 28). Most of these algorithms are either deterministic or based on hill-climbing and can get trapped into local optimal solutions. Some heuristics may reduce the occurrence of this problem, by searching for multiple possible solutions.

Mathematical programming can be also employed in cluster analysis. Earlier studies have explored a mathematical formulation for the clustering problem, like, for example, (23; 19; 10; 24). One advantage of using mathematical programming is its easier validation, since it enables the generation of good bounds for a optimization problem by methods like Lagrangian relaxation. The Lagrangian relaxation is able to find lower (upper) bounds for a minimization (maximization) mathematical model by relaxing some of its constraints. The process works by multiplying relaxed constraints by a constant penalty and adding the result to the objective function. Lagrangian relaxation has been adopted by many researchers with interesting results for different kinds of optimization problems (19; 26).

This paper proposes a hybrid heuristic based on a Lagrangian heuristic (19) and a local search method (Partition Around Medoids, PAM) (15). PAM is a widely used heuristic for solving for the k-medoids problem for data clustering. To evaluate the solutions achieved by the proposed heuristic, called LPAM, we compare it with the Lagrangian heuristic proposed by Mulvey and Crowder (19) and with the original PAM. As a result, the gap relaxation of the proposed heuristic is very low for all partitions, outperforming the other Lagrangian heuristic. The comparison of LPAM with PAM showed that, particularly in datasets with the largest number of objects, LPAM presented much better results. The results from this experiment show a very good efficiency of the heuristic considering the objective function as the parameter of comparison.

This paper is structured as follows. Section 2 presents a

*Corresponding author.

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.

mathematical model for the k-medoids clustering problem. In Section 3, the algorithms and strategies proposed in the literature to solve the studied problem are investigated. Section 4 presents the proposed heuristic to solve the k-medoids problem, LPAM. Section 5 reports the experimental results obtained in the computational tests and Section 6 concludes the paper presenting the final remarks.

2 k medoids problem

This section presents a mathematical model for the k-medoids clustering problem. This formulation is a p-median problem (which consists in locating p facilities in order to minimize the customer demands regarding these facilities) (17; 8) and it can be formulated as:

$$min \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij}$$

subject to:

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad i = 1, \ldots, N \tag{1}$$

$$x_{ij} \leq x_{jj} \qquad i, j = 1, \ldots, N \tag{2}$$

$$\sum_{j=1}^{N} x_{jj} = M \tag{3}$$

$$x_{ij} \in \{0, 1\} \qquad i, j = 1, \ldots, N \tag{4}$$

where N is the number of objects; M is the number of clusters; d_{ij} is the distance between objects i and j; x_{ij} is a binary variable which assumes value 1, if the object i belongs to the cluster whose medoid is the object j and 0, otherwise. The objective function of this formulation aims to minimize the dissimilarity of the objects regarding their medoid. Constraints (1) assure that a object i is associated with one medoid, i.e., it belongs to one cluster. Constraints (2) force the object i to be associated with the medoid j only if j is really a medoid, i.e., if $x_{jj} = 1$. Constraints (3) assure that the number of medoids of the partition is M and Constraints (4) obligate all the variables to be binaries.

According to Cornuejols et al. (3), the problem of finding the optimal solution for this problem is NP-hard. In the next section we present some algorithms to solve heuristically the k-medoids formulation.

3 Related work

One of the most popular heuristics for the investigated problem is the Partition Around Medoids, PAM (15). PAM is a deterministic algorithm composed of two stages. While the first stage (known as BUILD) defines a set of initial medoids, the second stage (known as SWAP) fine tunes the medoids by swapping objects between the clusters. Kaufman and Rousseeuw (15) also proposed a variation of PAM, Clustering large Applications named CLARA, to make the algorithm manageable for large datasets. Large datasets mean datasets of order of thousands of objects. In this strategy, subsamples of the original dataset are clustered by PAM, instead of the whole original data. The best solution found through these subsample strategy is kept, producing the final partition.

A modified version of CLARA was proposed by Ng and Han (22), named CLARANS. This strategy uses a graph to represent the sets of medoids, indicated in its nodes. A local search with restricted neighbors is performed to find the nodes with the best medoids. The main difference between CLARA and CLARANS is that, the former limits the search for the medoids to the objects from the sorted subsample, whereas the latter, restricts the neighborhood of the local search for the medoids, which can be any object of the dataset.

Another method to solve the k-medoids formulation for large datasets was presented in (25). It consists in a hybrid genetic algorithm (GA) that does not require the definition of the number of clusters a priori.

A solution method to solve the k-medoids formulation, strongly related to the proposed methodology, is the Lagrangian relaxation presented in (19). This methodology is detailed in Section 4, together with the proposed hybrid heuristic.

4 Lagrangian Relaxation

The Lagrangian relaxation is a technique of relaxing some constraints of hard optimization problems. This relaxation enables the study of an easier (relaxed) problem, for which it is possible to propose exact solution methods. Moreover, it is a powerful technique to provide bounds for the original problem.

A Lagrangian relaxation of a minimization problem yields lower bounds for the original problem. The lower bound enables the assessment of the quality of the feasible solution found by some strategy. The quality of these lower bounds depends on the trait of the relaxation. The Lagrangian relaxation employed in this paper aims to dualize Constraints (1) of the previous model, resulting in a trivial problem, as proposed in (19). This relaxation is carried out in the following way:

Let μ be a N-vector of real numbers of Lagrangian multipliers and let $L(\mu)$ be the Lagrangian function defined as:

$$L(\mu) = \{\min \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij} + \sum_{i=1}^{N} \mu_i (1 - \sum_{j=1}^{N} x_{ij}) : s.t. (2-4)\}. \tag{5}$$

For our purpose, it is necessary to get the best lower bound, which is obtained by maximizing the $L(\mu)$ function. Therefore, consider the following problem:

$$\max_{\mu} \{\min \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij} - \mu_i) x_{ij} + \sum_{i=1}^{N} \mu_i : s.t.\ (2-4)\}. \tag{6}$$

Notice that the function (5) was rearranged in order to make its solution simpler. The solution of the maximization problem (6) provides the best lower bound for the proposed mathematical model. In order to get the best μ that maximizes $L(\mu)$, we use the subgradient algorithm. This algorithm is an iterative method that produces a μ for each iteration k, μ^k, guided by the subgradient of the relaxed constraints, with $k = 1, \ldots, Max\_iterations$, where $Max\_iterations$ is a fixed number of iterations. Therefore, for each Lagrangian iteration k, a vector μ^k is supplied for the calculation of a new lower bound. Generally, the initial value μ^0 is a null vector. When $k > 0$, μ^k is calculated by the following equation:

$$\mu^k = \mu_{k-1} + \alpha_k * g^k, \tag{7}$$

where α_k indicates the step size in the subgradient direction in the iteration k and the N-dimensional vector g^k is the subgradient of the model, being given by: $g_i^k = 1 - \sum_{j=1}^{N} x_{ij} + \theta g_i^{k-1}$, for $i = 1, \ldots, N$.

Initially, g^0 is a null vector, g_i^k is the i-th component of g^k, and $\theta \in [0, 1]$. The α_k parameter can assume many distinct values and be updated by different rules. In this paper, we adopted the step rule from (11). We also tested two other rules for updating α_k: keeping it constant or randomly changing it at each iteration using a value in the range $[0, 1]$. Its formulation is given by: $\alpha_k = \alpha_0 \rho^k$, where α_0 is the initial value of α and $\rho < 1$. Different parameter values were tested with this formulation, but their results were not as good and stable as those obtained using the step rule proposed in (11). In (11), the step size rule, α_k, is updated at each Lagrangian iteration as follows.

$$\alpha_k = \lambda_k \frac{(best_{upper} - best_{lower})}{\sum_{i=1}^{N} g_i^k g_i^k}, \quad (8)$$

where $best_{lower}$ and $best_{upper}$ are, respectively, the best lower bound and the best upper bound found until iteration k. The λ_k parameter is generated by a decreasing sequence that depends on other parameters: n_{it}, r and MAX. The calculation of λ_k is given by the following steps:

Calculating $\lambda_k(k)$

Step 1: Define $(\lambda_0, r, n_{it}, MAX)$.

Step 2: For each n_{it} iterations do:

- If $k < MAX$, then $n_{it} = \frac{n_{it}}{r}$ and $\lambda_k = \frac{\lambda_k}{r}$.
- If $k \geq MAX$, then the parameters n_{it} and λ_k will be equal to n_{it} in its previous iteration and λ_{MAX-1}, respectively.

Therefore, the solution of maximizing $L(\mu^k)$ is determined by the following steps.

Maximization of $L(\mu^k)$

Step 1: (Definition of the medoids) Make $x_{jj} = 1$ for every $j \in S$, where S is the set of the M indexes $1 \leq j \leq N$ that provides the smallest values among $\sum_{i=1}^{N} \min(d_{ij} - \mu_i^k, 0)$.

Step 2: (Assignment of the objects) For every $i \in \{1, \ldots, N\}$, such that $d_{ij} - \mu_i^k < 0$ for $j \in S$, make $x_{ij} = 1$; otherwise, $x_{ij} = 0$.

As can be observed, the relaxation of some model constraints turns the model easy to solve. Nevertheless, this solution is just a lower bound for the original problem and its feasibility is not guaranteed.

Meanwhile, the knowledge of an effective lower bound has the advantage of estimating how acceptable is the upper bound that is a feasible solution for the original model. To find upper bounds, this paper proposes a novel Lagrangian heuristic (LPAM) based on the Lagrangian relaxation found in (19). In this heuristic, the medoids provided by each step of the Lagrangian relaxation are used as initial medoids for PAM. Giving the initial medoids for PAM, one of its two phases, the BUILD, is not used. Each step of the proposed Lagrangian heuristic is detailed as follows.

Procedure LPAM(x,d,μ)

Step 1: $k \leftarrow 0$, $\mu_0 \leftarrow 0$.

Step 2: (Lagrangian Evaluation) Find the solution of $L(\mu^k)$, the k-th lower bound, according to the routine *Maximization of* $L(\mu^k)$. Replace the best lower bound by the k-th lower bound if $k = 0$ or the best lower bound is lower than this found in iteration k.

Step 3: (Find the upper bound by LPAM) Perform PAM, defining its initial medoids those found in the Lagrangian step to produce lower bound of the current iteration.

Step 4: (Solution Update) Replace the best solution by the solution found in iteration k if it is lower than the current best solution or if $k = 0$.

Step 5: (Lagrangian Parameters Update) Update the Lagrangian parameters according to the routine *Calculating* $\lambda_k(k)$. Calculate α_k and μ_k using, respectively, Equations 8 and 7.

Step 6: $k \leftarrow k+1$. If $k > Max\_iterations$ or $gap = 0$, where gap is calculated according to Equation 10, then return the best solution and lower bound. Else, go to *Step 2*.

In the Lagrangian heuristic proposed by (19), named LH, the *Step 2* is replaced by:

Step 2: (Find the upper bound by LH) Define as medoids of the upper bound those from the relaxed solution of the actual iteration. To produce the upper bound, assign each object from the dataset to its nearest medoid.

Next section reports the computational experiments performed using some biological datasets.

5 Computational Experiments

The first experiment compares the proposed LPAM with the other k-medoids algorithms, LH and PAM. In this experiment, all parameters of the Lagrangian heuristics, LPAM and LH, are set according to their performance in the tests. In another experiment, we evaluate the clustering partitions found by LPAM according to the real classification of the data sets using an external validation index. In this experiment, we compare LPAM with three other clustering algorithms: k-means (15), k-medians (15) and PAM.

5.1 Data sets

Progresses in biology research have lead to the production of a large amount of data, which needs to be analyzed. Clustering algorithms have become an important tool in the analysis of biological data, because they can discover useful patterns to support the understanding of biological processes. One of these processes is observed in gene expression data, where tumor tissues can be grouped by the identification of their underline patterns.

To evaluate the use of LPAM for clustering, we carried out experiments using: six cancer tissues data sets, another biological data set and one artificial data set. The real classification of these data sets is known. In some cases, there is more than one classification structure for the data set. Each classification is defined as a structure. Moreover, each one of these structures may have a different number of groups or classes. The main characteristics of these eight data sets can be seen in Table 1, which illustrates, for each data set, the number of objects (♯Obj.), the number of different structures (♯Str.) and the number of groups for each different structure between parenthesis (♯Groups), the number of attributes (♯Attrib.) and the main paper reference (Reference).

The objective function of the k-medoids problem requires a dissimilarity measure between pairs of objects from the data sets. To supply this dissimilarity matrix, $D = [d_{ij}]_{N \times N}$, this paper evaluates the Manhattan and Euclidean distances.

Table 1: Data sets main characteristics. This table shows the main characteristics of each data set used in the experiments.

Data set	Main characteristics			
	♯Obj.	♯Str. (♯groups)	♯Attrib.	Reference
Golub	72	4 (2,3,4)	3571	(7)
Leukemia	327	2 (3,7)	271	(29)
Novartis	103	1 (4)	1000	(12)
MultiA	103	1 (4)	5565	(12)
MiRNA	218	6 (2,3,4,9, 20)	217	(16)
Breast	699	1 (8)	9	(2)
Simulated6	60	1 (6)	600	(18)
Ecoli	336	1 (8)	7	(20)

Both are special cases of the *p-distance* metric and can be represented by equation (9):

$$d_{ij} = (\sum_{k=1}^{N_a} |a_{ik} - a_{jk}|^p)^{\frac{1}{p}}, \quad (9)$$

where a_{ik} and N_a represent, respectively, the k-th attribute of the object i and the number of attributes of the objects, and p is an integer number that depends on the studied metric. If $p = 1$, the distance is known as the Manhattan distance; if $p = 2$, the distance is defined as the Euclidean distance, which is the most frequently used dissimilarity metric.

5.2 Experiment I

We implemented LH and LPAM using the R language from the R-project v.2.8.1. The implementation of PAM used in the LPAM was obtained from the R-project package *cluster*. This implementation enables the user to define the initial medoids for the PAM algorithm.

To check and compare the performance of the Lagrangian heuristics, a quantitative measure is adopted: the gap between the lower bound and upper bound. This gap indicates how far is the feasible solution obtained by LPAM and by LH from the best lower bound found by the relaxation of the mathematical model. It is calculated in the following way:

$$gap = \frac{(z_{upper} - z_{lower})}{z_{lower}} * 100 \quad (10)$$

where

z_{upper} is the upper bound, i.e., the feasible solution;

z_{lower} is the Lagrangian lower bound.

Given that the subgradient method has a profound impact on the performance of the Lagrangian heuristics, we investigated the effect of different parameter values in them. For such, we considered the following range values for the parameters: $MAX \in \{5, 10, 15\}$; $n_{it} \in \{20, 25, 30\}$; $r \in \{1, 2, 3\}$; and $\lambda_0 \in [0.70, 2.0]$. The step size used for the interval of the last parameter, λ_0, was 0.05. For the data sets used in this paper, we found the best values adopting the following parameter values: $MAX = 15$, $n_{it} = 25$, $r = 2$ and $\lambda_0 = 1.75$. Other values were tested, but the gaps outside the previous parameter ranges achieved very high values. We also adopted $\alpha_0 = 0$ and a null value for the vector μ_0. The number of maximum Lagrangian iterations, $Max\_iterations$ parameter, was set to 100 for LPAM and 500 for LH. The number of maximum iterations fixed to LH was set according to the elapsed time that LPAM took to perform its 100 iterations. The number of iterations for LPAM was set after tests that showed that the gap reduction is not so significant after 100 iterations for most of the data sets. The value for the parameter θ was defined 0.0 after preliminary tests. In these tests, we took into account the performance of both Lagrangian heuristics, LPAM and LH.

Table 2: Mean and standard deviation of the gaps in percentage of LPAM considering the Euclidean distance metric.

Data set	LPAM		LH	
	Mean	SD	Mean	SD
Simulated6	0.046	0.102	0.149	0.232
Golub	0.039	0.045	0.335	0.288
MultiA	0.057	0.080	0.445	0.567
Novartis	0.059	0.104	0.411	0.518
MiRNA	0.126	0.064	1.591	1.819
Leukemia	0.040	0.028	0.314	0.412
Ecoli	0.281	0.146	4.966	4.141
Breast	2.869	1.585	25.387	13.278

Table 3: Mean and standard deviation of the gaps in percentage of LPAM considering the Manhattan distance metric.

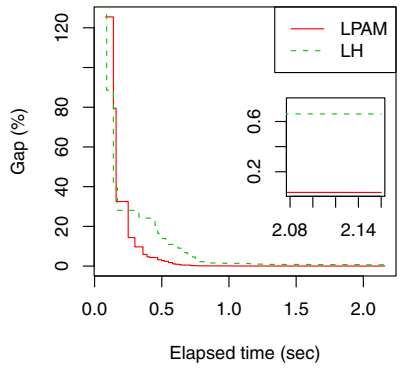
Data set	LPAM		LH	
	Mean	SD	Mean	SD
Simulated6	0.034	0.120	0.160	0.295
Golub	0.150	0.269	0.608	0.562
MultiA	0.104	0.122	0.501	0.521
Novartis	0.153	0.276	0.679	0.766
MiRNA	0.089	0.074	1.304	2.182
Leukemia	0.150	0.085	0.905	0.641
Ecoli	0.448	0.167	5.405	4.213
Breast	4.435	2.203	30.860	17.570

Initially, we present, in Tables 2 and 3 the mean gap values of the proposed Lagrangian heuristic, LPAM, and the Lagrangian heuristic found in the literature, LH. These values are the average gap results for LPAM and LH considering the Euclidean and Manhattan distances. This table also presents the standard deviation of the gaps. For each data set, we consider $M \in \{2, \ldots, 30\}$. We display these results in Figures 1 and 2 for every data set. These results refer to $M = 15$, considering the Euclidean distance as the dissimilarity between the pairs of objects. We used $M = 15$ because we wanted the heuristics found partitions with a reasonable number of medoids. The adopted time interval refers to 0 and the least elapsed time between LPAM and LH to achieve the last best gap, in the considered number of iterations.

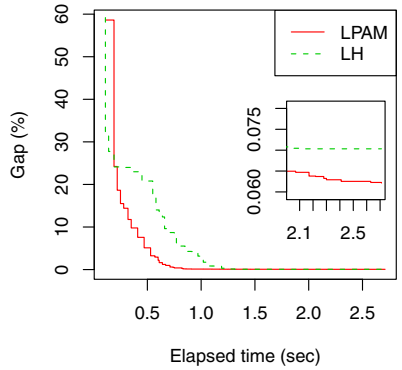
Figure 1 presents the behavior of the data sets with the lowest number of objects. It can be noticed that, except for the MultiA, Figure1(d), LPAM achieved much better results than LH in the considered time interval. The inset subplots present the behavior of the gaps in the last period, discriminating their differences in such period.

Figure 2 clearly shows the superior performance of LPAM over LH. Although for MiRNA, Leukemia and Breast there were some intervals that LH was superior to LPAM, LPAM was more robust than LH, always achieving better final gaps for all data sets. It can also be observed that the differences between the final gaps of LPAM and LH were very significant in the last two data sets.

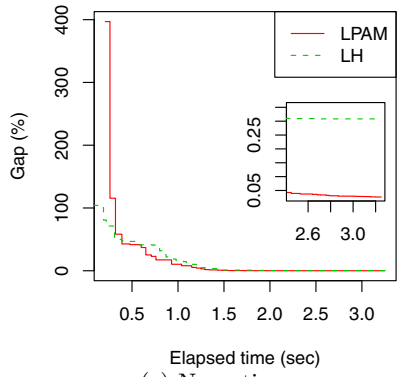
It can be observed from Tables 2 and 3 that, in all data sets, the mean gaps of LPAM outperformed the mean gaps of LH. Moreover, the standard deviations of the gaps of LPAM



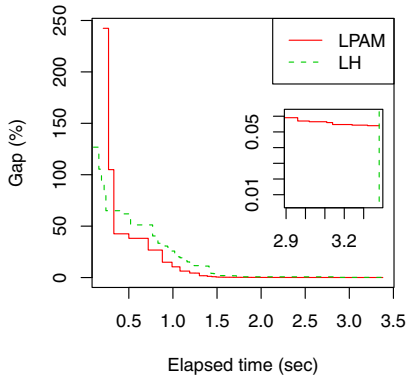
(a) Simulated6



(b) Golub

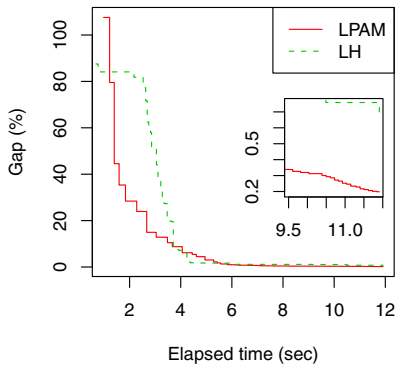


(c) Novartis

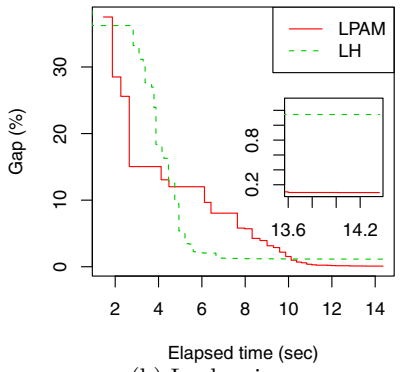


(d) MultiA

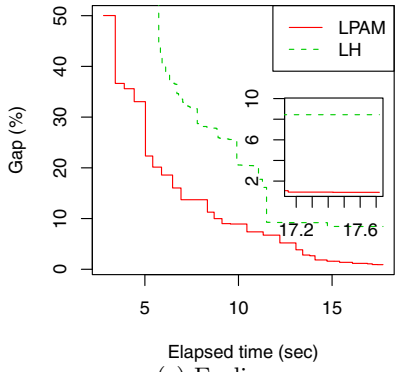
Figure 1: Relation gap versus elapsed time of LPAM and LH for, respectively, Simulated6, Golub, Novartis and MultiA data sets.



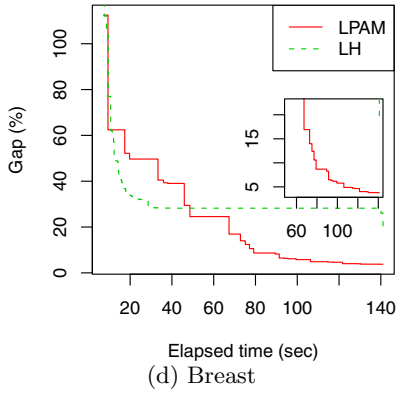
(a) MiRNA



(b) Leukemia



(c) Ecoli



(d) Breast

Figure 2: Relation gap versus elapsed time of LPAM and LH for, respectively, MiRNA, Leukemia, Ecoli and Breast data sets.

were much lower than those from LH. These results evidence the best performance of LPAM over LH. It is worth mentioning that, although the number of data sets is 8, 29 runs corresponding to different numbers of clusters were carried out. Moreover, two different dissimilarity metrics were used. Therefore, the total number of test cases was 464.

In order to compare the results from the original PAM with LPAM, we confronted their objective function solutions. For such, we calculated how far are the solutions found by PAM and LPAM using the following equation.

$$gap = \frac{z_{PAM} - z_{LPAM}}{z_{LPAM}} \quad (11)$$

In this equation, z_{PAM} and z_{LPAM} correspond to, respectively, the PAM solution and the LPAM solution. This gap is presented in percentage in the experiments. It is worth mentioning that the LPAM solutions were always better than or equal to the PAM solutions. Figure 3 presents the results of this comparison. Table 4 present the results of these comparisons.

Table 4: Mean and standard deviation of the gaps in percentage for LPAM.

Dataset	Euclidean		Manhattan	
	Mean	SD	Mean	SD
Simulated6	0.029	0.027	0.006	0.014
Golub	0.038	0.039	0.075	0.080
MultiA	0.002	0.005	0.002	0.011
Novartis	0.041	0.049	0.048	0.107
MiRNA	0.106	0.192	0.242	0.212
Leukemia	0.022	0.030	0.028	0.057
Ecoli	0.251	0.229	0.282	0.228
Breast	0.287	0.476	0.283	0.310

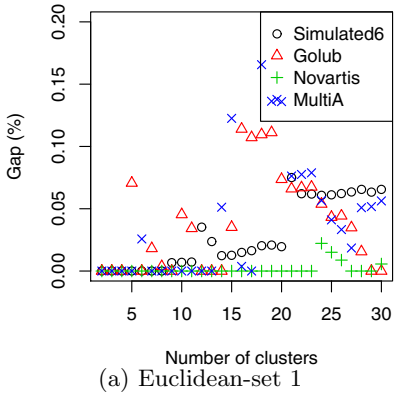
Note that, in Table 4, the higher the number of objects, the better the solutions found by LPAM over PAM. This is an indicative of a good performance of the proposed heuristic, since the higher the dimension of the matrices, the more difficult is to find good quality solutions for them.

In Figure 3, the data sets were divided into two sets. The set 1 refers to the data sets with the lowest number of objects, whereas the set 2 has the 4 data sets with the highest number of objects. Figures 3(a) and 3(c) show the results for the set 1, when the Euclidean and Manhattan distances between objects were used, respectively. It can be noticed that both Golub and MultiA presented the highest gaps. Moreover, the gaps were more significant when the Manhattan distance was used.

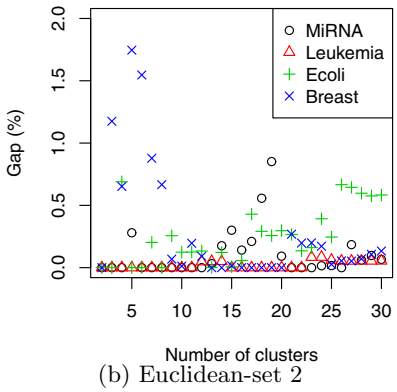
For the set 2, Figure 3(b) and 3(d) indicates that Breast, the data set with the highest number of objects, achieved the highest gaps. Moreover, considering the Figure 3(d), it can be observed that all data sets presented gaps higher than the previous analyzed data. Therefore, all these results demonstrate the superiority of LPAM over LH and PAM for the k-medoids problem.

5.3 Experiment II

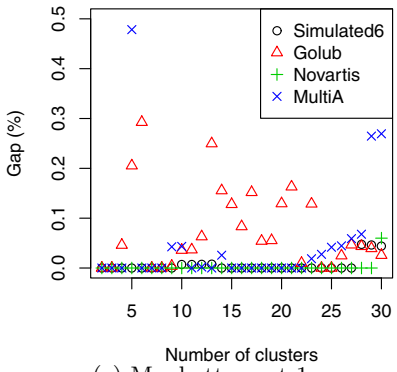
In this experiment, we compare LPAM with three well known clustering algorithms from literature: k-means (15), k-medians (15) and PAM. The software implementation of the two first algorithms was obtained from (5). The use of the k-means and the k-medians for comparison is due to, as well as the k-medoids problem, they produce cluster with spherical shape.



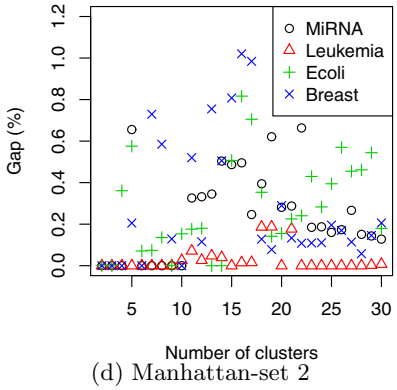
(a) Euclidean-set 1



(b) Euclidean-set 2



(c) Manhattan-set 1



(d) Manhattan-set 2

Figure 3: These graphs present the gaps between LPAM and PAM regarding their number of clusters.

We evaluated the partitions produced by the four algorithms using the CRand measure (13), which is based on the real data classification. Thus, CRand provides an external evaluation criterion that estimates how much the partitions determined match the real classification. The CRand varies from -1 to 1, and the larger the CRand, the more accurate is the partition regarding the true classification. It is worth to remind that this is a classification index and it is just a validation measure for evaluating partitions found by a clustering algorithm. It indicates the agreement between a partition found and the real classification.

The experimental results present the CRand values with respect to the Euclidean and the Manhattan distances. Alike (9), we performed tests by firstly generating partitions with cluster numbers in the set $\{2, \ldots, 30\}$ for each data set. The partitions with the highest CRand are reported. This procedure was followed for every clustering algorithm used in this experiment.

A performance profile analysis proposed by Dolan and Moré (6) was employed to compare the CRand values of the four different algorithms. Let S and P be, respectively, the set of n_s algorithms to be analyzed and the set of n_p problems (data sets) into consideration. A factor, known as *performance ratio*, evaluates the performance of the algorithm s with regard to the problem (or instance) p. It is given by the following equation:

$$r_{ps} = \frac{t_{ps}}{min\{t_{ps} | s \in S\}} \tag{12}$$

The value of t_{ps}, used in the comparison of the algorithms, gives the performance of the algorithm s in the problem p. According to Dolan and Moré (6), the lower this value, the better. Regarding Equation 12, it must be observed that the best r_{ps} is 1, which occurs when t_{ps} has the minimum value among all algorithms.

Another factor considered in (6) is the ratio: the number of problems that an algorithm s presented a *performance ratio* equal to or better than an coefficient τ divided by the total number of problems ($\rho_s(\tau) = \frac{1}{n_p} size\{p \in P | r_{ps} \leq \tau\}$).

This ratio $\rho_s(\tau)$ represents the probability that an algorithm s has a *performance ratio* within a factor τ. In this performance analysis graph, the curves of the graph are plotted according to the values of τ and $\rho_s(\tau)$.

In this paper, S is composed by the four algorithms analyzed according to the CRand value: LPAM, k-means, k-medians and PAM. The distance matrix associated with each data set was considered an instance. Every CRand was taken into account for analysis, which means that a total of 34 instances composes the set P.

Figure 4 displays the results from this analysis. $\tau = 1$ shows the percentage that an algorithm s achieved results better than the other algorithms. Both LPAM and PAM achieved the best results in more than 60% of the problems. However, LPAM presented a slightly higher percentage than PAM, as can be observed in the subplot of Figure 4. Each one of the other algorithms achieved the best results in a little more than 20% of the instances. The robustness of the algorithms in the instances analysed can be observed when $\tau > 1$. For example, for $\tau = 1.2$, y-axis indicates the percentage of instances that an algorithm s achieved results at most 20% worse than the best results. It can be noticed that LPAM achieved the best results for almost every value of τ.

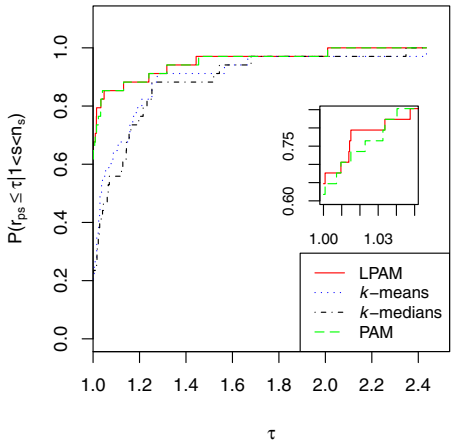


Figure 4: This graph shows the performance profile of all distance metrics, algorithms and data sets relating to the CRand index.

The subplot in Figure 4 highlights the superior performance of LPAM over PAM for small values of τ.

6 Final Remarks

This paper investigated a hybridization of a Lagrangian based heuristic with a known local search clustering method, PAM for the k-medoids problem. The first experiment investigated the efficiency of LPAM, comparing it with another k-medoids heuristics found in literature for the same problem. According to the experimental results, LPAM outperformed the other heuristics, mainly in the data sets with high number of objects. In the second experiment, the proposed heuristic was compared with the real partitions using the CRand index. In this experiment, LPAM was compared with the k-means, k-medians and PAM algorithms. The proposed algorithm performed better in most of the partitions, considering the Euclidean and Manhattan distances. Furthermore, LPAM is deterministic, in contrast to k-means and k-medians methods, known for the variation of the partitions obtained in different runs. PAM achieved, for some cases, the same clustering partitions as LPAM, however, the medoids were different, presenting worse solutions than LPAM. Regarding the computational time, LPAM took just some few seconds to find its solutions, even for medium sized data sets.

Acknowledgments

The authors would like to thank FAPESP and CNPq for the research funding.

References

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96(12):6745–6750, 1999.

[2] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[3] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optmize float: an analytic study of exact and approximate algorithms. *Management Science*, 23:789–810, 1977.

[4] F. Corpet. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research*, 22:10881–10890, 1988.

[5] M. J. L. de Hoon, S. Imoto, J. Nolan, and S. Miyano. Open source clustering software. *Bioinformatics*, 20(9):1453–1454, 2004.

[6] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program., Ser. A*, 91:201–213, 2002.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[8] S. L. Hakimi. Optimum distributions of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13:462–475, 1965.

[9] J. Handl and J. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions On Evolutionary Computation*, 11:56–76, 2007.

[10] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79:191–215, 1997.

[11] M. Held, P. Wolfe, and P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.

[12] Y. Hoshida, J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesiro. Subclass mapping: Identifying common subtypes in independent disease data sets. *PLOS one*, 2(11):e1195, 2007.

[13] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[14] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, 1999.

[15] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. New York: Wiley, 1990.

[16] J. Lu, G. Getz, E. A. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B. L. Ebert, R. H. Mak, A. A. Ferrando, J. R. Downing, T. Jacks, R. R. Horvitz, and T. R. Golub. Microrna expression profiles classify human cancers. *Nature*, 435(7043):834–838, 2005.

[17] F. Maranzana. On the location of supply points to minimize transport costs. *Operational Research Quarterly*, 15:261–270, 1964.

[18] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Technical report, Broad Institute/MIT - Kluwer Academic Publishers, 2003.

[19] J. M. Mulvey and H. P. Crowder. Cluster analysis: an application of lagrangian relaxation. *Management Science*, 25:329–340, 1979.

[20] K. Nakai and M. Kanehisa. Expert system for predicting protein localization sites in gram-negative bacteria. *PROTEINS: Structure, Function, and Genetics*, 11:95–110, 1991.

[21] M. C. V. Nascimento, F. M. B. Toledo, and A. C. P. L. F. Carvalho. Investigation of a new GRASP-based clustering algorithm applied to biological data. *Computers & Operations Research*, 37:1381–1388, 2010.

[22] R. Ng and J. Han. CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions Knowledge of Data Engineering*, 14:1003–1016, 2002.

[23] M. R. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626, 1971.

[24] B. Sağlam, F. S. Salman, S.l Sayin, and M. Türkay. A mixed-integer programming approach to the clustering problem with an application in customer segmentation. *European Journal of Operational Research*, 173:866–879, 2006.

[25] W. Sheng and X. Liu. A genetic k-medoids clustering algorithm. *J Heuristics*, 12:447–466, 2006.

[26] W. W. Trigeiro, L. J. Thomas, and J. O. McClain. Capacitated lot sizing with setup times. *Management Science*, 35:353–366, 1989.

[27] A. Ushioda and J. Kawasaki. Hierarchical clustering of words and application to nlp tasks. In E. Ejerhed and I. Dagan, editors, *Fourth Workshop on Very Large Corpora*, pages 28–41, Somerset, New Jersey, 1996. Association for Computational Linguistics.

[28] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16:645–678, 2005.

[29] E.-J. Yeoh, M. E. Ross, S. A. Shurtleff, W. K. Williams, D.Patel, R. Mahfouz, F.G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.-H. Pui, W. E. Evans, C. Naeve, L. Wong, and J.R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer cell*, 1:133–143, 2002.