Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 5 - 38123 Povo - Trento (Italy)
http://disi.unitn.it

Pareto Local Search for Alternative Clustering

List of authors:
Duy Tin Truong
Roberto Battiti

February 2013

Technical Report # DISI-13-002
Version 1.0

Published also: …

Pareto Local Search for Alternative Clustering

Duy Tin Truong
University of Trento
truong@disi.unitn.it

Roberto Battiti
University of Trento
battiti@disi.unitn.it

ABSTRACT

Supervised alternative clusterings is the problem of finding a set of clusterings which are of high quality *and* different from a given *negative* clustering. The task is therefore a clear multi-objective optimization problem. Optimizing two conflicting objectives at the same time requires dealing with tradeoffs. Most approaches in the literature optimize these objectives sequentially (one objective after another one) or indirectly (by some heuristic combination of the objectives). Solving a multi-objective optimization problem in these ways can result in solutions which are dominated (and not Pareto-optimal). We develop a direct multi-objective local search algorithm based on Pareto Local Search, called **PLSAC**, which fully acknowledges the multiple objectives, optimizes them directly and simultaneously, and produces solutions approximating the Pareto front. **PLSAC** has no sensitive parameters to be tuned by the user, provides solutions which *dominate* those obtained by other state-of-the-art algorithms, and can accept arbitrary clustering quality and dissimilarity objectives. Besides, it can also be guided by the user to explore specific regions of interest along the Pareto front in an *interactive* manner.

Keywords

clustering, multi-objective optimization, local search

1. INTRODUCTION

Data mining aims at discovering hidden and relevant knowledge from large datasets, and clustering often is one of the most useful approaches. Given a dataset, traditional clustering algorithms often only provide a single set of clusters, a single view of that dataset. On complex datasets, many different ways of clustering can exist and thus asking for *alternative* clusterings to have complementary views is a natural requirement. As an example, clustering flowers depending on colors and aesthetic characteristics can be suitable for a florist, but not for a scientist who prefers different organizations.

Recently, many techniques have been developed for solving the *alternative* clustering problem. They can be split into two groups: unsupervised or supervised. In unsupervised alternative clustering, the algorithm automatically generates a set of alternative clusterings which are of high quality and different from each other [11, 5]. Unsupervised alternative clustering is useful in situations where users do not know what they want and need some initial options. In other cases, users have already known some trivial or *negative* clusterings from the dataset, and they would like to obtain different and more informative clusterings. These algorithms are called supervised because the user is directing the alternative clustering by explicitly labeling some clusterings as undesired, or *negative*. Unsupervised alternative clustering is not suitable in this case, because the alternative clusterings generated in the unsupervised manner can be different from each other but not very different from the negative clustering that the user defines.

This paper focuses on *supervised alternative clustering*[1], the problem of finding new clusterings which are of good quality *and* as different as possible from a given negative clustering. Obviously, supervised alternative clustering is a multi-objective optimization problem with two objectives of clustering quality and clustering dissimilarity, and the goal is to find a representative set of Pareto-optimal solutions. A Pareto-optimal solution is a solution such that there is no solution which improves at least one objective without worsening the other objectives. The Pareto front is the set of all Pareto optimal solutions in the objective space. Most approaches in the literature only optimize these two objectives *sequentially* (optimizing one objective first and then optimizing the second one) [6, 14] or *indirectly*, by some heuristics [1, 4]. Another approach is to model this problem as a constrained optimization problem where the first objective is optimized and the second objective is used to form constraints [9]. Then, applying Lagrange multipliers, the constrained optimization problem is turned into a single objective problem.

Solving a multi-objective optimization problem in the above ways can result in solutions which are not Pareto-optimal, or in a single or a very limited number of solutions on the Pareto front. The user flexibility is thus limited because the tradeoff between the different objectives is decided *a priori*, before knowing the possible range of solutions. The tradeoff

[1]For convenience, in future references we will refer to it as alternative clustering where no confusion can occur.

can be decided in a better way *a posteriori*, by generating a large set of representative solutions along the Pareto front and having the user pick the favourite one among them, or *interactively*, by having the user judge some initial solutions and direct the software to explore the most interesting areas of the objective space.

Some approaches are developed from specific clustering algorithms, and are therefore limited in their application, or require the setting of parameters which influence the preference between clustering quality and dissimilarity. Parameter tuning by the final user can be a difficult task, so that some dominating solutions can be lost because of improper settings.

To deal with the above issues, we propose an explicit multi-objective local search framework, called **PLSAC** (Pareto Local Search for Alternative Clusterings). **PLSAC** is simple, and has the following main advantages:

- Optimizing directly and simultaneously the predefined objectives (clustering quality, and dissimilarity).
- Accepting arbitrary clustering quality and dissimilarity objectives.
- Exploring specific regions according to the user's interest.

In addition, our approach has no sensitive parameters, therefore parameter tuning is not necessary. Another interesting feature of our algorithm is that it can be stopped at *any time* and deliver a more comprehensive coverage of the Pareto front when more CPU time is allocated.

The rest of this paper is organized as follows. We first discuss the related work in Section 2. In Section 3, we recall the Pareto local search framework of Paquete et al.[12] and then detail our multi-objective local search algorithm for alternative clustering. We next describe the experiments done to compare the performance of our algorithm with that of other state-of-the-art algorithms in Section 4. We also explain how our algorithm can be modified to enable a user to explore specific regions and discuss the parameter sensitivity of our algorithm in the same section.

2. RELATED WORK

One of the first algorithms in supervised alternative clustering is Conditional Information Bottleneck (CIB) [9] proposed by Gondek et al. and CIB is based on the information bottleneck (IB) method [15]. Gondek et al.'s approach in modelling the clustering problem is similar to that of data compression. Given two variables X representing objects and Y representing features, and a negative clustering Z, Gondek's CIB algorithm finds an alternative clustering C which is different from Z but still good in quality by maximizing the objective $I(C;Y|Z)$ under the constraint that the information rate $I(C;X)$ between C and X is less than a threshold R. However, this approach requires an explicit joint distribution between the objects and the features which is difficult to estimate.

Although CIB can find alternative clusterings which are different from the negative clusterings, Bae et al.[1] have shown that the clusterings found by CIB are not of high quality compared with their **COALA** algorithm which is an extension of the agglomerative hierarchical clustering algorithm. Let d_1 be the smallest distance between two arbitrary clusters and d_2 be the smallest distance between two clusters where merging them does not violate the constraints (generated by the negative clustering). If the ratio $\frac{d_1}{d_2}$ is less than a threshold, then two nearest clusters are merged to preserve the clustering quality. Otherwise, two clusters with the distance of d_2 are merged to find dissimilar clusterings. A drawback of this method is that it only considers cannot-link constraints, hence useful information which can be obtained through must-link constraints is lost. In addition, the application cope of the method is limited because it was developed particularly for the agglomerative clustering algorithms.

To overcome the scope limitation of **COALA**, Davidson and Qi [6] propose a method, called **AFDT** which transforms the dataset into a different space where the negative clustering is difficult to be detected and then uses an arbitrary clustering algorithm to partition the transformed dataset. However, transforming the dataset into a different space can destroy the characteristics of the dataset. Davidson and Qi subsequently[14] fix this problem by finding a transformation which minimizes the Kullback-Leibler divergence between the probability distribution of the dataset in the original space and the transformation space under the constraint that the negative clustering should not be detected. This approach requires specifying user preference on the clustering quality and the clustering dissimilarity. In this paper, we refer to this modified algorithm of Davidson and Qi as **AFDT2**.

Alternative clustering can also be discovered by two orthogonalization algorithms proposed by Cui et al. [4]. The algorithms first project the dataset into an orthogonal subspace and then apply an arbitrary clustering algorithm on the transformed dataset. However, when the objects of the dataset are in low dimensional spaces, the orthogonal subspace may not exist and therefore projecting all objects in the dataset to the orthogonal subspace can map all objects to the same point [6]. In addition, a requirement of Cui et al.'s algorithms which is the number of clusters must be smaller than the number of dimensions in the original dataset may not be satisfied in many practical datasets. In fact, Davidson and Qi [6] have shown that their algorithm **AFDT** outperforms Cui et al.'s algorithms. Therefore, in this paper, we will only compare our algorithm against **COALA**, **AFDT**, **AFDT2**.

3. MULTI-OBJECTIVE LOCAL SEARCH FOR ALTERNATIVE CLUSTERING

In this section, we first formally define the problem of alternative clustering in Section 3.1 and then detail our multi-objective local search algorithm to address such problem in Section 3.2.

3.1 Alternative Clustering Problem

Given a dataset $X = \{x_i\}_{i=1}^{N}$ of N objects, the traditional clustering problem is to partition this dataset into K disjoint clusters such that the clustering quality is as high as possible. Let s be a clustering solution where $s(i)$ presents the index of the cluster that x_i belongs to, $D(s, \bar{s})$ be the dissimilarity between two clusterings s and $\bar{s}$, and $Q(s)$ be the inner quality of a clustering s. We defer the definitions of $D(s, \bar{s})$ and $Q(s)$ to Section 4 where we also present the experimental results and define in this section the dissimilarity between a clustering and a set of clusterings, the overall quality of a clustering and the dominance relation between two clusterings.

Definition 1. (Dissimilarity) The *dissimilarity* betweens a clustering s and a set of clusterings $\bar{S}$ is the average dissimilarity between s and all clusterings $\bar{s} \in \bar{S}$:

$$D(s, \bar{S}) = \frac{1}{|\bar{S}|} \sum_{\bar{s} \in \bar{S}} D(s, \bar{s}) \tag{1}$$

Definition 2. (Overall Clustering Quality) The *overall quality* of a clustering s is characterized by the following bi-objective function $F(s, \bar{s})$:

$$F(s, \bar{s}) = [Q(s), D(s, \bar{s})] \tag{2}$$

$$\text{where } \begin{cases} \bar{s} & \text{is a given negative clustering.} \\ Q(s) & \text{is the quality of a clustering } s. \\ D(s, \bar{s}) & \text{is the dissimilarity between } s \text{ and } \bar{s}. \end{cases}$$

Definition 3. (Clustering Dominance) Given a negative clustering $\bar{s}$, a clustering s *dominates* another clustering s' w.r.t $\bar{s}$, written $s' \prec_{\bar{s}} s$ if and only if the following conditions are hold:

$$(s \neq s') \wedge \big[\big(Q(s) > Q(s') \wedge D(s, \bar{s}) \geq D(s', \bar{s})\big) \vee \big(Q(s) \geq Q(s') \wedge D(s, \bar{s}) > D(s', \bar{s})\big)\big] \tag{3}$$

Finally, the alternative clustering problem is defined as follows:

Definition 4. (Alternative Clustering) Given a negative clustering $\bar{s}$, *alternative clustering* is the problem of finding clusterings s such that there is no other clustering s' that dominates s w.r.t $\bar{s}$.

3.2 Pareto Local Search for Alternative Clustering

Alternative clusterings is a multi-objective optimization problem with two objectives of clustering quality and clustering dissimilarity, and our goal is to find a representative set of Pareto-optimal solutions. A Pareto-optimal solution is a solution such that there is no solution which improves at least one objective without worsening the other objectives. In general, the problem of finding the Pareto global optimum set consisting of all non-dominated solutions of a combinatorial optimization problem is NP-hard [7]. Therefore Paquete et al.[12] approximate the exact global optimum set by searching for the *Pareto local optimum* sets. A solution s is Pareto local optimal with respect to a neighbourhood Nb if and only if there is no $s' \in Nb(s)$ such that s' dominates s. Similarly, a set of solutions S is Pareto local optimal with respect to a neighbourhood Nb if and only it contains only Pareto local optimum solutions with respect to Nb. Paquete et al. also propose a Pareto Local Search (PLS) framework [12] for finding the Pareto local optimum sets of the bi-objective Travelling Salesman Problem (TSP). Starting from Paquete et al.'s framework, we have derived our Pareto local search algorithm for solving the alternative clustering problem.

3.2.1 Pareto Local Search framework

The PLS framework is presented in Algorithm 1. At the beginning, the archive A consists of a random initial solution s_0. Then, an unvisited solution s in the archive is picked randomly from the archive. For every solution s' in the neighbourhood of s, its objectives $F(s')$ are computed. If this new solution is not dominated by any solution $s'' \in A$ then it is added to the archive, and all solutions $s'' \in A$ which are dominated by s' are removed from A. This process is repeated until all solutions in A are visited. This Pareto Local Search framework has been proven to terminate and return the local optimum sets [12, 13]. Two main advantages of this framework are: no aggregation function of objectives is required in solving the multi-objective optimization problem, and no parameter settings are necessary. Its disadvantage is that running PLS until termination leads to excessive CPU time for non-trivial problems. Thus, its practical success depends on the speed with which reasonable approximations of the Pareto front are built. Although the PLS framework is simple, it has been shown to produce very good solutions for the bi-objective TSP compared to other meta-heuristics [12].

Algorithm 1: Pareto Local Search framework

```
Input  : Multi-objective function F, neighbourhood
         function Nb, an initial solution s_0
Output: Pareto Local Minimum set A
1  begin
2      A = {s_0}
3      repeat
4          Pick an unvisited solution s from A.
5          Mark s as visited.
6          for all s' ∈ Nb(s) do
7              Evaluate F(s').
8              if s' is not nominated by any s'' ∈ A then
9                  Add s' to A.
10                 Remove all clusterings s'' ∈ A which are
                   dominated by s'.
11     until all solutions in the archive A are visited ;
12     return A
13 end
```

3.2.2 Pareto Local Search for Alternative Clustering

Based on the PLS framework, we propose our algorithm for finding the Pareto local optimal clusterings for the bi-objective function in Equation 2 as in Algorithm 2. In the original PLS framework, the archive is a set but in our

PLSAC algorithm, it is a list storing *diverse* potential solutions discovered from the solution space in the *breath first search* manner.

Algorithm 2: Pareto Local Search for Alternative Clusterings (PLSAC)

Input : Dataset X of N objects, the number of clusters K, a negative clustering $\bar{s}$
Output: A list of alternative clusterings A

```
1  begin
2      Generate an initial clustering s_0.
3      A = {s_0}
4      repeat
5          Pick the first unvisited clustering s from A.
6          Mark s as visited.
7          for i = 1 to N do
8              for j = 1 to K do
9                  if s(i) ≠ j then
10                     s' = s
11                     s'(i) = j
12                     if s' is not dominated by any solution
                       s'' ∈ A then
13                         UpdateArchive(A, s')
14     until all solutions in the archive A are visited or the
       maximum number of iterations is reached ;
15     return A
16 end
```

The steps of our algorithm are as follows. First, an initial clustering s_0 is generated and added to the archive A. The initial clustering can be a high quality clustering that the user knows or a random clustering. Then, the first unvisited clustering $s \in A$ is picked. For each clustering s, a neighbour s' is formed by moving of an object x_i to a cluster c_j where $c_j \neq s(i)$. If a neighbour s' is not dominated by any clusterings $s'' \in A$, then all clusterings $s'' \in A$ dominated by s' are removed from A and s' is added to the last position of A. In the PLS framework, the archive size is unbounded. However, for the clustering problem, the archive size can grow exponentially, therefore in our algorithm the archive size is restricted to M. Because of this reason, if the archive size grows larger than M after the insertion step, then the clustering $s'' \in A$ with the smallest dissimilarity to the clustering set $A \setminus \{s''\}$ is removed from A. Such operation aims at preserving the diversity for the solution list. In all experiments that we have performed, the archive size M is set to 10. The above process is repeated until all solutions in A are visited or a number of iterations is reached.

The complexity of the algorithm in each iteration is $O(NKF)$ where F is the cost of computing the objectives. In addition, a neighbour s' of a clustering s is only different from s on two clusters, namely the cluster where x_i is removed and the cluster where x_i is added. As a result, the objective values can be updated incrementally which in turn reduces the cost of computing the objectives F.

Procedure `UpdateArchive`

Input : Archive A, non-dominated solution s'
Output: Updated archive

```
1  begin
2      A = {s''|s'' ∈ A : s'' is not dominated by s'}
3      Insert s' to the last position in A.
4      if size(A) > M then
5          s'' = argmin_{s∈A} D(s, A \ {s})
6          A = A \ {s''}
7  end
```

We have performed different experiments on different datasets and under diverse settings of our algorithm for comparing our algorithm's performance against state-of-the-art alternative clustering algorithms including **COALA**[1], **AFDT**[6], **AFDT2**[14]. We present in this section the experimental results of ten such experiments. The performance comparison is based on evaluating the dissimilarity and quality of the clusterings found by the algorithms in a quantitative manner. The structure of this section is as follows. We first describe the evaluation methods on the clustering dissimilarity and quality in Section 4.1. We then present our experimental results in Section 4.2 and illustrate the ability of exploring specific regions of our algorithm in Section 4.3. We finish this section with a discussion on the parameter sensitivity of our algorithm in Section 4.4.

4. EXPERIMENTS

4.1 Evaluation Methods

We advocate the Dunn-Index[3] and Rand-Index[18] to evaluate the clustering dissimilarity (D_{RI}) and quality (Q_{DI}), respectively. The larger the values of D_{RI} and Q_{DI} are, the larger the dissimilarity between two clusterings is and the better the clustering quality is.

Definition 5. (Clustering Quality) The *quality* of a clustering measured by the generalized Dunn-Index is defined as:

$$Q_{DI}(s) = \frac{\min_{i \neq j} \delta(c_i, c_j)}{\max_{1 \leq l \leq K} \Delta(c_l)} \tag{4}$$

$$\delta(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{x_i \in c_i, x_j \in c_j} dist(x_i, x_j) \tag{5}$$

$$\Delta(c_l) = \frac{1}{|c_l|} \sum_{x_l \in c_l} dist(x_l, \mu_l) \tag{6}$$

$$\mu_l = \frac{1}{|c_l|} \sum_{x_l \in c_l} x_l$$

where $\delta(c_i, c_j)$ is the distance between two clusters c_i and c_j formed from the clustering s; $\Delta(c_l)$ is the diameter of a cluster c_l; $dist(x_i, x_j)$ is the Euclidean distance between two objects x_i and x_j.

Definition 6. (Clustering Dissimilarity) The *dissimilarity*

Dataset	Cardinality	# of dimensions	# of classes
glass	214	9	6
ionosphere	351	34	2
iris	150	4	3
protein	116	20	6
soybean	47	35	4
sonar	208	60	2

Table 1: UCI dataset characteristics.

between two clusterings s_1 and s_2 is defined as:

$$D_{RI}(s_1, s_2) = \frac{N_{10} + N_{01}}{N_{11} + N_{00} + N_{10} + N_{01}} \quad (7)$$

where

$$N_{11} = |\{(x_i, x_j) : (s_1(i) = s_1(j)) \wedge (s_2(i) = s_2(j))\}|$$
$$N_{10} = |\{(x_i, x_j) : (s_1(i) = s_1(j)) \wedge (s_2(i) \neq s_2(j))\}|$$
$$N_{01} = |\{(x_i, x_j) : (s_1(i) \neq s_1(j)) \wedge (s_2(i) = s_2(j))\}|$$
$$N_{00} = |\{(x_i, x_j) : (s_1(i) \neq s_1(j)) \wedge (s_2(i) \neq s_2(j))\}|$$

N_{11} is the number of object pairs that are in the same cluster in both two clusterings; N_{00} is the number of pairs that are in different clusters in both clusterings; N_{10} is the number of pairs that are assigned in the same cluster by the clustering s_1 and in different clusters by the clustering s_2; N_{01} is the number of pairs that are assigned in different clusters by the clustering s_1 and in the same cluster by the clustering s_2.

4.2 Performance Comparison

We compare the performance of our algorithm with that of three other state-of-the-art algorithms, namely **COALA**[1], **AFDT**[6], **AFDT2**[14] on six UCI datasets[2]. We have used two different initialization schemes in our algorithms to test the impact of the initial clusterings on the algorithm performance. This is because different initial clusterings in general can make a local search algorithm return different results. More in details, we have specified a random clustering in the first scheme and the negative clustering in the second scheme as the initial clustering to our algorithm. We refer to our algorithm initialized with the first scheme as **PLSAC-R** and the second as **PLSAC-Q**. The latter is expected to return clustering solutions with high clustering quality because it starts from a high quality solution, i.e. the negative clustering while the former is supposed to produce a more diverse set of solutions due to its random initialization.

4.2.1 Experimental Setup

We have chosen six UCI datasets (*glass*, *ionosphere*, *iris*, *protein*, *soybean*, *sonar*) as benchmarks whose characteristics are summarized in Table 1. We have also used the clustering returned by the standard agglomerative hierarchical clustering with the average-linkage method [17] as the negative clustering. On each dataset, **PLSAC-R** is restarted ten times and non-dominated solutions are recorded for later performance evaluations. **PLSAC-R** and **PLSAC-Q** are stopped when all solutions in the archive are visited or when the number of iterations is equal to 1000.

In addition, because **COALA** is an extension of the agglomerative hierarchical clustering algorithm, therefore the base clustering algorithms of **AFDT** and **AFDT2**[3] are replaced by the agglomerative clustering algorithm with the average linkage method [17] in all experiments.

For a specific configuration, **COALA**, and **AFDT2** can only produce one solution. In order to generate a set of different solutions from **COALA**, the parameter declaring user reference on clustering quality and dissimilarity w of **COALA** is iterated 10 times from 0.1 to 1.0 with step of 0.1 (the maximum possible value of w is 1.0) . The larger the value of w is, the better the clustering quality is and the smaller the clustering dissimilarity is. Similarly, the trade-off parameter on the clustering quality and the clustering dissimilarity a of **AFDT2** is iterated 10 times from 1.0 to 2.8 with step of 0.2. Increasing a makes the clustering dissimilarity improved and the clustering quality decreased. The default values for w and a are originally set to 0.6[1] and 2.0[14] respectively. Regarding **AFDT**, it has no parameters, and can only produce one solution, so no parameter settings are required.

4.2.2 Experimental Results

Figure 1 shows the performance of five algorithms on six UCI datasets. In most datasets, our algorithm provide diverse sets of high quality (in both clustering quality and clustering dissimilarity) solutions. The Pareto stair in Figure 1 shows the lines connecting all non-dominated solutions returned by our algorithm. Therefore, the solutions below the Pareto stair are dominated by the solutions of our algorithm.

As can be seen from the figure, the solutions of other algorithms are all below the Pareto stair of **PLSAC-R** and **PLSAC-Q**. In other words, for each clustering solution produced by the other algorithms, there is always a solution produced by our algorithm (either **PLSAC-R** or **PLSAC-Q**) of better quality in both two objectives. In addition, it is noticeable that on some datasets like *ionosphere* and *sonar*, **COALA** and **AFDT2** produce solutions which are very similar to the negative clustering, while **PLSAC-R** and **PLSAC-Q** can reveal more diverse sets of solutions.

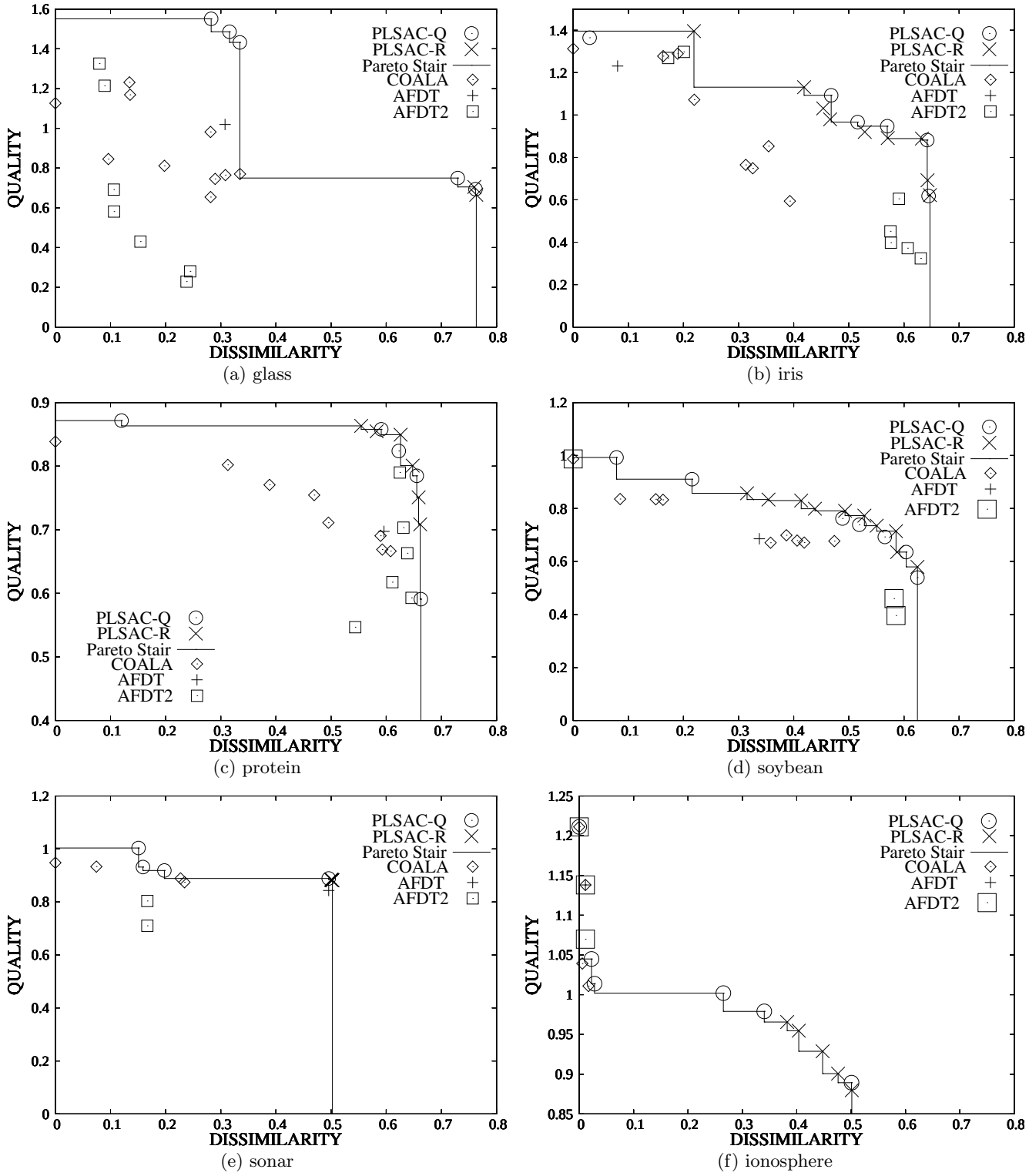
In comparing the performance of our algorithm under two initialization schemes, we have observed that on most datasets **PLSAC-Q** returns solutions in high clustering quality while **PLSAC-R** can produce more diverse sets of solutions than **PLSAC-Q**. This is because **PLSAC-Q** starts from a high clustering quality solution and **PLSAC-R** starts from a random clustering. In addition, because of the local minima property, on datasets *glass* and *sonar*, **PLSAC-R** produces only few different solutions. To obtain solutions in regions which have been missed during the search phase of our algorithm, the users need to guide it as described in the next section.

4.3 Interactive Exploration of Specific Regions

After obtaining the approximate Pareto front found by an alternative clustering algorithm, a user often wants to pick a solution region of his interest and explore it in more depth in order to make his analysis on the clustering solutions more comprehensive. With a small modification in our algorithm structure, it can allow the user to guide its search phase so

[2] `http://archive.ics.uci.edu/ml/datasets.html`

[3] `http://wwwcsif.cs.ucdavis.edu/~qiz/code/code.html`



(a) glass (b) iris

(c) protein (d) soybean

(e) sonar (f) ionosphere

Figure 1: Performance comparison

that he can explore such region. The exploration of some specific region under the user's guide provided by our algorithm is a very useful property that has not been considered before in other algorithms. This property enables the user to pick an arbitrary point of his interest and then ask the algorithm to search for clustering solutions in the surrounding regions of such point. For sake of simplicity, in this section we only show how our algorithm can be modified to enable the choice of one dimension in specifying the surrounding areas. Similar modifications can then be done

to allow the specification of the other dimension. Figure 2 illustrates the ideas of our modified algorithm which we refer to as **PLSAC-S**. The user is able to pick a target dissimilarity value D_{Target} and its surrounding interval Δ_D, then run **PLSAC-S** to find clustering solutions in the area where the dissimilarity can vary from $D_{Target} - \Delta_D/2$ to $D_{Target} + \Delta_D/2$.

Algorithm 4: PLSAC for Specific Regions (PLSAC-S)

Input : Dataset X of N objects, the number of clusters K, a negative clustering $\bar{s}$, the target dissimilarity D_{Target}, the surrounding interval Δ_D

Output: A set of alternative clusterings A

```
1  begin
2      s_0 = s̄
3      rids = randperm(N)
4      for i = 1 to N do
5          s_0(rids(i)) = randsample(K)
6          if D(s_0, s̄) >= D_Target then
7              break
8      A = {s_0}
9      repeat
10         Pick the first unvisited clustering s from A.
11         for i = 1 to N do
12             for j = 1 to K do
13                 if s(i) ≠ j then
14                     s' = s
15                     s'(i) = j
16                     if D_Target − Δ_D/2 ≤ D(s', s̄) and
                       D(s', s̄) ≤ D_Target + Δ_D/2 then
17                         if s' is not dominated by any solution
                           s'' ∈ A then
18                             UpdateArchive(A, s')
19         Mark s as visited.
20     until all solutions in the archive A are visited ;
21     return A
22 end
```

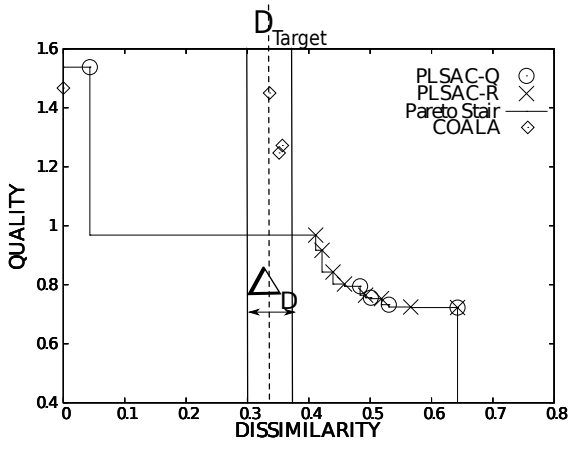


Figure 2: Exploration of Specific Regions

4.3.1 PLSAC for Region Exploration

The modified **PLSAC** enabling the ability of searching in specific regions is presented in Algorithm 4. The algorithm has two phases: the initialization phase (line 1-10) and the search phase (line 11-29). In the initialization phase, the algorithm generates a solution such that the dissimilarity between this solution and the negative clustering is around the target dissimilarity. At the beginning of this phase, the initial solution s_0 is assigned to be the negative clustering $\bar{s}$, meaning that its dissimilarity at this time is 0. Then, the initial solution is perturbated to increase the dissimilarity until the dissimilarity of s_0 and the negative clustering $\bar{s}$ is greater than or equal to the target dissimilarity. The function $randperm(N)$ returns a random permutation of N values from 1 to N and the function $randsample(K)$ returns a number sampled uniformly from 1 to K. In the search phase, only solutions with the dissimilarity in the surrounding interval of the target dissimilarity are considered. The other steps of the algorithm remain the same as in its original version shown in Algorithm 2.

4.3.2 An Example

Consider an example where the user wants to explore some parts of the Pareto front found by **PLSAC-R** and **PLSAC-Q**. We use a synthetic dataset of six Gaussian sub-clusters as in Figure 3 and set the desired number of clusters to 3. To simplify the discussion, we use the same experimental setup as in the previous section and compare the performance of **PLSAC-R** and **PLSAC-Q** with **COALA** only. Figure 4(a) compares the clustering solutions obtained from the three algorithms. It also shows that most solutions returned by **COALA** are in the region with dissimilarity in the interval [0.35-0.4] and are of high clustering quality. This is because hierarchical clustering algorithms like **COALA** can work very well on these separated datasets. In this example, although **PLSAC-R** and **PLSAC-Q** are able to find solutions with dissimilarity ranging from 0 to 0.8, the actual dominating region produced by them does not cover all solutions of **COALA**. By allowing the users to be involved in the search phase, our enhanced algorithm can find high quality solutions in regions specified for them. Figure 4(b) shows the solutions obtained by our **PLSAC-S** in specific regions of dissimilarity $D_{Target} = 0.3$ and 0.35. The surrounding interval Δ_D in both cases is 0.1. The new solutions found by **PLSAC-S** are shown to dominate those found by **COALA** in the figure.

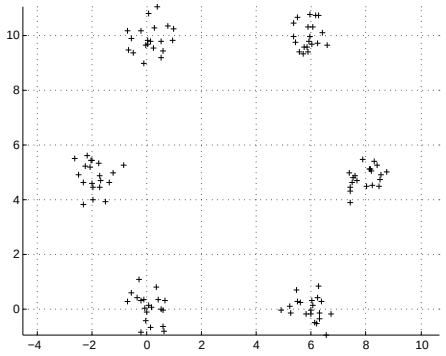
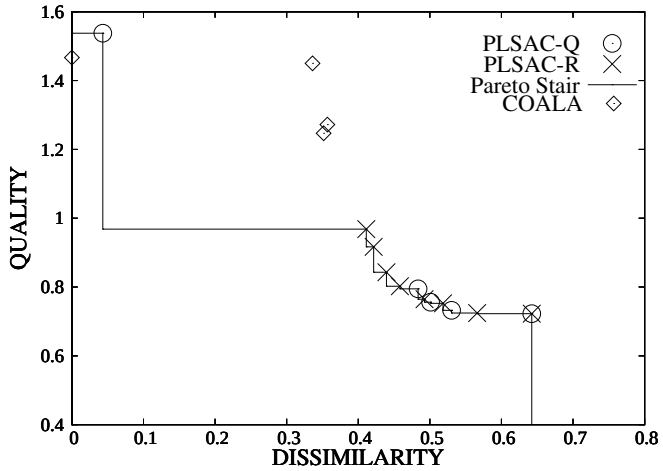
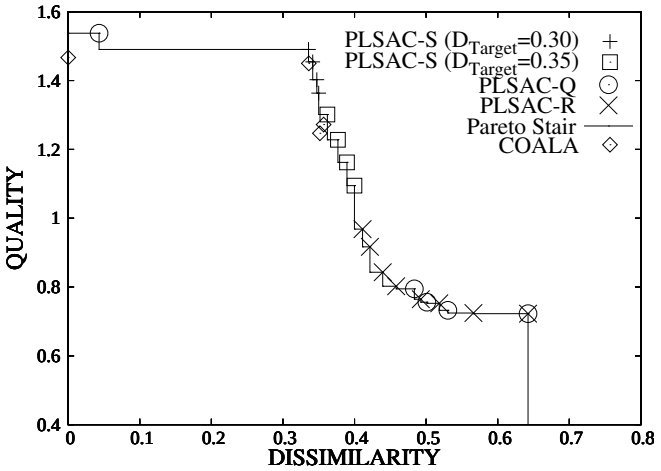


Figure 3: Six Gaussians dataset



(a) Without specific regions

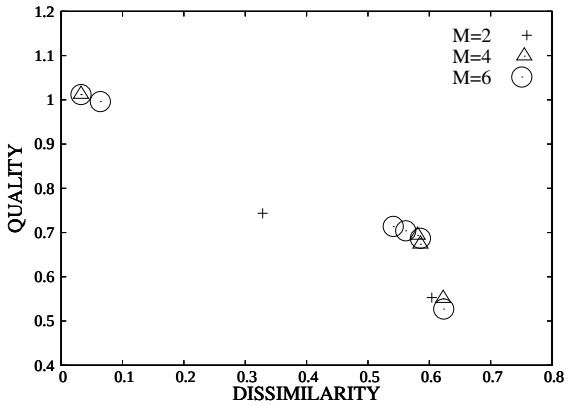


(b) Specific regions with target dissimilarity $D_{Target} = 0.3/0.35$ and surrounding interval $\Delta_D = 0.1$

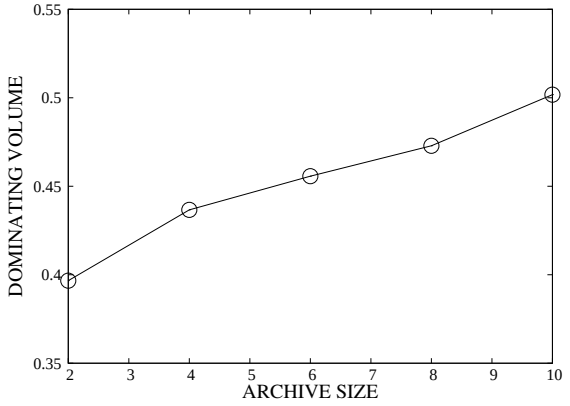
Figure 4: Example on Region Exploration

4.4 Parameter Sensitivity

The performance of our algorithm can be influenced by changing the archive size. To analyse the sensitivity of such parameter, we have carried out several experiments on our **PLSAC-Q** algorithm which uses the negative clustering as the initial clustering. Following the same experimental setup as in Section 4.2, **PLSAC-Q** is run on six UCI datasets with different archive sizes from 2 to 10. We have observed that the impact of changing the archive size on **PLSAC-Q** is similar for all datasets. That is the algorithm produces a more diverse solution set as the archive size increases. Therefore, we only present here the result on the **soybean** dataset. Figure 5(a) shows the solutions produced by **PLSAC-Q** with different archive size of $2, 4, 6$. We have witnessed a similar behaviour on the larger values of the archive size. However, in order to make the figure readable, only small values of the archive size are plotted. In addition, a multi-objective algorithm is often evaluated by measuring the volume of the dominating region of the Pareto front produced by that algorithm [19]. The bigger the volume is, the better the algorithm behaviour is. Figure 5(b) displays the volume of the dominating region below the Pareto stair of **PLSAC-Q** on different archive sizes. As expected, **PLSAC-Q** enlarges the dominating region and produces better solution sets when the archive size increases.



(a) Solutions returned by PLSAC-Q with different archive sizes.



(b) Dominating Volume of PLSAC-Q with different archive sizes.

Figure 5: PLSAC-Q with different archive sizes on the *soybean* dataset.

5. CONCLUSION

In this paper, we propose an explicit multi-objective local search algorithm for alternative clustering, called **PLSAC**. **PLSAC** not only provides good solutions and outperforms those produced by other state-of-the-art algorithms, but it also provides attractive features as follows. The first feature is that it is very simple and can accept arbitrary objectives, therefore it can be used as a baseline in comparing with different alternative clustering algorithms. In addition, because of its flexibility, it can also be used as a general framework to study the performance of different clustering quality and clustering dissimilarity objectives. Another appealing feature is that our algorithm does not have any sensitive parameter so no parameter tuning is required. Lastly, the user's interaction with **PLSAC** is the very useful feature provided by our algorithm, allowing the user to directly interact with the algorithm to explore specific regions of his

interest along the Pareto front.

In the paper, we have used the standard local search technique in our algorithm. Therefore, it can get stuck at locally optimal configurations. In the future, we plan to integrate diversification mechanisms to escape from local optima by deploying more complex global search techniques for multi-objective optimization like Multi-objective Simulated Annealing [16, 2], or Multi-objective Tabu Search [8, 10]. Besides, the running time of our algorithm is still larger than that of other alternative clustering algorithms, thus a careful selection of a subset of promising neighbours to reduce the search space (a "candidate list" strategy) is also another part of the future work.

6. REFERENCES

[1] E. Bae and J. Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 53–62. IEEE, 2006.

[2] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *Evolutionary Computation, IEEE Transactions on*, 12(3):269–283, 2008.

[3] JC Bezdek, WQ Li, Y. Attikiouzel, and M. Windham. A geometric approach to cluster validity for normal mixtures. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 1(4):166–179, 1997.

[4] Y. Cui, X.Z. Fern, and J.G. Dy. Non-redundant multi-view clustering via orthogonalization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 133–142. IEEE, 2007.

[5] X.H. Dang and J. Bailey. Generation of alternative clusterings using the cami approach. In *Proceedings of the SIAM International Conference on Data Mining*, pages 118–129, 2010.

[6] I. Davidson and Z. Qi. Finding alternative clusterings using constraints. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 773–778. IEEE, 2008.

[7] M. Ehrgott. Approximation algorithms for combinatorial multicriteria optimization problems. *International Transactions in Operational Research*, 7(1):5–31, 2000.

[8] X. Gandibleux, N. Mezdaoui, and A. Fréville. A tabu search procedure to solve multiobjective combinatorial optimization problem. *Lecture notes in economics and mathematical systems*, pages 291–300, 1997.

[9] D. Gondek and T. Hofmann. Conditional information bottleneck clustering. In *3rd ieee international conference on data mining, workshop on clustering large data sets*, pages 36–42. Citeseer, 2003.

[10] M.P. Hansen. Tabu search for multiobjective optimization: Mots. In *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM 97), Cape Town, South Africa*, pages 574–586. Citeseer, 1997.

[11] P. Jain, R. Meka, and I.S. Dhillon. Simultaneous unsupervised learning of disparate clusterings. *Statistical Analysis and Data Mining*, 1(3):195–210, 2008.

[12] L. Paquete, M. Chiarandini, and T. Stützle. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. *Metaheuristics for Multiobjective Optimisation*, pages 177–199, 2004.

[13] L. Paquete, T. Schiavinotto, and T. Stützle. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research*, 156(1):83–97, 2007.

[14] Z.J. Qi and I. Davidson. A principled and flexible framework for finding alternative clusterings. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726. ACM, 2009.

[15] N. Tishby, F.C. Pereira, and W. Bialek. The information bottleneck method. *Arxiv preprint physics/0004057*, 2000.

[16] EL Ulungu, J. Teghem, PH Fortemps, and D. Tuyttens. Mosa method: a tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999.

[17] E.M. Voorhees. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing & Management*, 22(6):465–476, 1986.

[18] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. *Advances in neural information processing systems*, 15:505–512, 2002.

[19] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4):257–271, 1999.