# TER - Clustering Algorithms in a 2-dimensional Pareto Front

HE Yuru and XU Yuetian

Université Paris-Sud, Informatique Master1
yuru.he@u-psud.fr,yuetian.xu@u-psud.fr

**Abstract.** This paper is motivated by a real life application of multi-objective optimization without preference. Having many incomparable solutions with Pareto optimality, the motivation is to select a small number of representative solutions for decision makers.This paper proves that these clustering problems can be solved to optimality with a unified dynamic programming algorithm. The clustering measures is investigated in this paper for the 2-dimensional case using the specific property that the points to cluster are Pareto optimal in $\mathbb{R}^2$.

**Keywords**: Clustering algorithms; Pareto frontier; dynamic programming; matheuristics

## 1 Problem Description

Clustering is a statistical analysis method used to organize raw data into homogeneous silos. Within each cluster, the data is grouped according to a common characteristic. The scheduling tool is an algorithm that measures the proximity between each element based on defined criteria.

In this paper, the representativity measure comes from clustering algorithms, partitioning the N elements into K subsets with a maximal similarity, and giving a representative (i.e. central) element of the optimal clusters.

There are five clustering measures mentioned in this paper: $k$-means, $k$-medoids, $k$-median, discrete $k$-center,continuous $k$-center.

$k$-means is a simple and effective measure.In each cluster, there is an averaged center called centroids .It clustering minimizes the sum of squared distance from each item to its nearest averaged center.

For $k$-medoids,in each cluster, there is a medoid, which is a real data item from the data set.$k$-medoids clustering minimizes the sum of squared distance from each item to its nearest medoids.

For $k$-median ,in each cluster, there is a median.$k$-median clustering minimizes the sum of distance from each item to its nearest median.

For $k$-center, in each cluster, there is a cluster center.$k$-center clustering minimizes the maximum distance from each item to its nearest cluster centers.

## 2 Notations

We suppose in this paper having a set $E = \{x_1, \dots, x_N\}$ of $N$ elements of $\mathbb{R}^2$, such that for all $i \neq j$, $x_i \ \mathcal{I} \ x_j$ defining the binary relations $\mathcal{I}, \prec$ for all $y = (y^1, y^2), z = (z^1, z^2) \in \mathbb{R}^2$ with:

$$y \ \mathcal{I} \ z \Longleftrightarrow y \prec z \text{ or } z \prec y \tag{1}$$

$$y \prec z \Longleftrightarrow y^1 < z^1 \text{ and } y^2 > z^2 \tag{2}$$

This property is verified in the applicative context, $E$ being the solution of a bi-objective optimization problem without preference. This applies for exact approaches or population meta-heuristics like evolutionary algorithms and others [**?**].

We consider in this paper the Euclidian distance, defining for all $y = (y^1, y^2), z = (z^1, z^2) \in \mathbb{R}^2$:

$$d(y, z) = \|y - z\| = \sqrt{(y^1 - z^1)^2 + (y^2 - z^2)^2} \tag{3}$$

We define $\Pi_K(E)$, as the set of all the possible partitions of $E$ in $K$ subsets:

$$\Pi_K(E) = \left\{ P \subset \mathcal{P}(E) \,\middle|\, \forall p, p' \in P,\;\; p \cap p' = \emptyset \text{ and } \bigcup_{p \in P} = E \text{ and } \mathrm{card}(P) = K \right\} \tag{4}$$

Defining a cost function $f$ for each subset of E to measure the dissimilarity, the clustering problem is written as following optimization problem:

$$\min_{\pi \in \Pi_K(E)} \sum_{p \in \pi} f(P) \tag{5}$$

K-means clustering is a combinatorial optimization problems indexed by $\Pi_K(E)$. K-means clustering minimizes the sum for all the $K$ clusters of the average distances from the points of the clusters to the centroid. Mathematically, this can be written as:

$$f_{means}(P) = \frac{1}{\mathrm{card}(p)} \sum_{x \in p} \left\| x - \frac{1}{\mathrm{card}(p)} \sum_{y \in p} y \right\|^2 \tag{6}$$

K-medoids clustering minimizes the sum of squared distance from each item to its nearest medoids.Mathematically, this can be written as:

$$f_{medoids}(P) = \frac{1}{\mathrm{card}(p)} \min_{y \in p} \sum_{x \in p} \|x - y\|^2 \tag{7}$$

K-median clustering minimizes the sum of distance from each item to its nearest median.Mathematically, this can be written as:

$$f_{median}(P) = \min_{y \in p} \sum_{x \in p} \|x - y\| \tag{8}$$

Discrete K-center can be written as:

$$f_{ctr}^{D}(P) = \min_{y \in p} \max_{x \in p} \|x - y\| \tag{9}$$

Continous K-center can be writtern as:

$$f_{ctr}^{C}(P) = \min_{y \in \mathbb{R}^2} \max_{x \in p} \|x - y\| \tag{10}$$

Industrial applications of Pareto Front in multi-objective optimization: system engineering, design of industrial systems. In the case of dimension 2, minimizing 2 objectives, we can define Pareto fronts more easily:

A discrete 2-dimensional Pareto Front is defined here as a set E of N points in $\mathbb{R}^2$ , indexed with $E = (x_k, y_k)_{k \in [\![1,N]\!]}$ such that $k \in [\![1,N]\!] \mapsto x_k$ is increasing and $k \in [\![1,N]\!] \mapsto y_k$ is decreasing.

## 3 Dynamic Programming Algorithm

Conjecture and the polynomial computations of all the $c_{i,i'}$ with $i < i'$ allows to derive a dynamic programming algorithm. Defining $C_{i,k}$ as the optimal cost of the $k$-means clustering with $k$ cluster among points $[\![1,i]\!]$ for all $i \in [\![1,N]\!]$ and $k \in [\![1,K]\!]$, we have following induction relation:

$$\forall i \in [\![1,N]\!],\; \forall k \in [\![2,K]\!],\quad C_{i,k} = \min_{j \in [\![1,i]\!]} C_{j-1,k-1} + c_{j,i} \tag{11}$$

This last relation use the convention that $C_{0,k} = 0$ for all $k \geqslant 0$. These relations allow to compute the optimal values of $C_{i,k}$ for all $i \in [\![1,N]\!]$ and $k \in [\![1,K]\!]$. The optimal partition is computed with backtracking.

## 4 Lloyd Algorithm

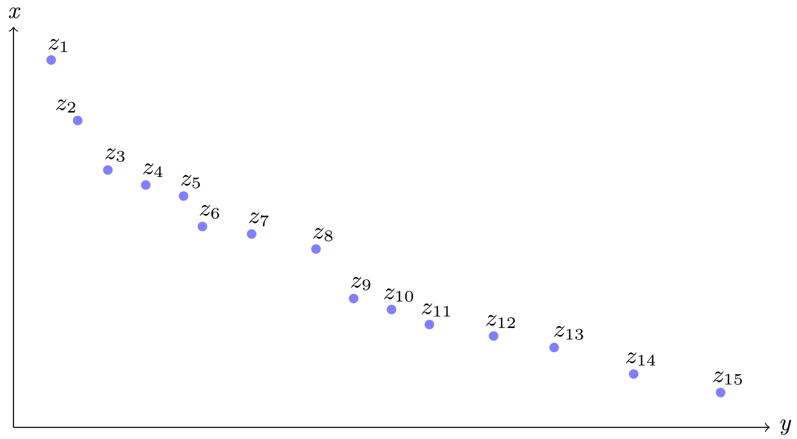

**Fig. 1.** Illustration of the indexation implied by Proposition in a 2-d Pareto front

**Algorithm 1: DP interval clustering in a 2d-Pareto Front**

```
compute cost matrix $c_{i,j}$ for all $(i,j) \in [\![1,N]\!]^2$

  for $i = 1$ to $N$ //Construction of the matrix $C$
    set $C_{i,k} = c_{1,i}$ // case $k = 1$ treated separately
    for $k = 2$ to $K$
      set $C_{i,k} = \min_{j \in [\![1,i]\!]} C_{j-1,k-1} + c_{j,i}$
    end for
  end for
return $C_{N,K}$ the optimal cost

  initialize $i = N$ and $P = nil$ //Backtrack phase
  for $k = K$ to 1 with increment $k \leftarrow k - 1$
    find $j \in [\![1,i]\!]$ such that $C_{i,k} = C_{j-1,k-1} + c_{j,i}$
    add $[\![j,i]\!]$ in $\mathcal{P}$
    $i = j - 1$
  end for

return the partition $\mathcal{P}$ giving the cost $C_{N,K}$
```