Clustering in a 2d Pareto Front: p-median and p-center are solvable in polynomial time

Nicolas Dupin, El-Ghazali Talbi, Frank Nielsen

December 12, 2018

Abstract

Having many solutions in a Pareto front generated by multi-objective optimization approaches, this paper aims to select and present a small number of representative solutions to decision makers. The p-median and p-center problems are investigated for the 2-dimensional case. This paper proves that these clustering problems can be solved to optimality with a unified dynamic programming algorithm. Furthermore, the algorithm can be adapted to consider cardinality constraints for the clusters. Having $N$ points to partition in $K$ clusters, a complexity in polynomial time is proven in $O(N^3)$ for the p-median problem in $O(N^2 \cdot (K + \log N))$ for the p-center problem. A posteriori, the complexity allows also to consider these algorithms inside multi-objective meta-heuristics to archive diversified non-dominated points along the Pareto front.

Keywords: Dynamic programming ; Clustering algorithms ; p-median problem ; p-center problem ; complexity ; bi-objective optimization ; Pareto front

1 Introduction

This paper is motivated by real-life applications of multi-objective optimization [13, 26]. Multi-objective optimization approaches may generate a large set of non-dominated solutions using Pareto dominance. The problem is here to select for a human decision making only $K$ good compromise solutions from $N \gg K$ non-dominated solutions. It aims to maximize the representativity of these $K$ solutions among the $N$ initial ones. This problem is similar to maximizing the quality of discrete representations of Pareto sets in multiobjective optimization [29], which was studied with the hypervolume measure [2] [19].

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. The p-median and p-center problems can address such problem, minimizing a measure to the most central point belonging in the subset. The p-median and p-center problems are NP-complete in the general case [18, 17] but also for the specific case in $\mathbb{R}^2$ using the Euclidian distance [21]. This paper proves that the special case of
p-median and p-center clustering in a 2-dimensional Pareto front is solvable in polynomial time, using a common dynamic programming algorithm.

This paper is organized as following. In section 2, we describe the considered problems with unified notation. In section 3, we discuss related state-of-the-art elements to appreciate our contributions. In section 4, intermediate results are presented. In section 5, a common dynamic programming algorithm is presented with polynomial complexity thanks to the results of section 4. In section 6, the implications and applications of the results of section 5 are discussed. In section 7, our contributions are summarized, discussing also future directions of research. To ease the readability of the paper, the elementary proofs of some intermediate results are gathered in Appendix A.

![Figure 1: Illustration of Pareto dominance and incomparability quarters minimizing two objectives indexed by $x$ and $y$: zones of $A$ and $D$ are incomparability zones related to $O$](image)

2 Problem statement and notation

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

\begin{align*}
  y \prec z & \iff y^1 < z^1 \text{ and } y^2 > z^2 \\
  y \preceq z & \iff y \prec z \text{ or } y = z \\
  y \preceq z & \iff y \prec z \text{ or } z \prec y
\end{align*}  

These hypotheses on $E$ come from the context of bi-objective optimization. Without loss of generality, transforming objectives to maximize $f$ into $-f$ allows to consider the minimization two objectives. This last assumption impacts the definition of dominance/incomparability zones in the objective space as illustrated in Figure 1 and also the sense of inequalities of $\preceq, \prec$. 

2
The solutions of a bi-objective optimization problem are a set of non-dominated solutions using Pareto dominance to rank the two objectives. The points of \( E \) are the values of the objectives of non-dominated solutions, denoted as Pareto front, as illustrated in Figure 2. If several solutions have to the same costs for all the objective function, it induces a single point in \( E \). For more details and properties of multi-objective optimization, we refer to [11].

A set \( E \) can be the projected costs of the optimal non-dominated solutions using exact approaches to generate the Pareto front [11]. In this paper, the optimality is not required in the further developments. This work applies also for partial Pareto fronts furnished by population meta-heuristics [30, 31].

We consider in this paper the Euclidian distance:

\[
d(y, z) = \|y - z\| = \sqrt{(y^1 - z^1)^2 + (y^2 - z^2)^2}, \quad \forall y = (y^1, y^2), z = (z^1, z^2) \in \mathbb{R}^2
\] (4)

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

\[
\Pi_K(E) = \left\{ P \subset \mathcal{P}(E) \left| \forall p, p' \in P, p \cap p' = \emptyset \text{ and } \bigcup_{p \in P} p = E \text{ and } \text{card}(P) = K \right. \right\}
\] (5)

Clustering problems are combinatorial optimization problems indexed by \( \Pi_K(E) \). Defining a cost function \( f \) for each subset of \( E \) to measure the dissimilarity, the clustering problem is written as following minimization problem:

\[
\min_{\pi \in \Pi_K(E)} \sum_{P \in \pi} f(P)
\] (6)

\( p \)-median clustering minimizes the sum for all the \( K \) clusters of the total distances from the points of the clusters to the most central point in the cluster. The cost function is:

\[
\forall P \subset E, \quad f_{med}(P) = \min_{y \in P} \sum_{x \in P} |x - y|
\] (7)

The (discrete) \( p \)-center clustering minimizes the sum for all the \( K \) clusters of the maximal distances from the points of the clusters to the most central point in the cluster. The cost function is:

\[
\forall P \subset E, \quad f_{ctr}(P) = \min_{y \in P} \max_{x \in P} |x - y|
\] (8)

In these two cases, we define for all subset \( P \subset E \) as centers the points \( p \in P \) furnishing the minimizations of \( f_{med}(P) \), \( f_{ctr}(P) \):

\[
\forall P \subset E, \quad c_{med}(P) = \arg\min_{x \in P} \sum_{p \in P} |x - p|
\] (9)

\[
\forall P \subset E, \quad c_{ctr}(P) = \arg\min_{x \in P} \max_{p \in P} |x - p|
\] (10)

In the selection of \( K \) representative solutions of a Pareto front for a decision maker, the selected solutions are the centers \( c_{med}(P) \) and \( c_{ctr}(P) \) for all \( P \in \pi \).
3 State-of-the-art

This section describes related works to appreciate our contributions, in the state of the art of the p-median and the p-center problems.

3.1 The p-median problem

The p-median problem was originally a logistic problem, having a set of customers and defining the places of depots in order to minimize the total distance for customers to reach the closest depot. We give here the general form of the p-median problem. Let \( N \) be the number of clients, called \( c_1, c_2, \ldots, c_N \), let \( M \) be the number of potential sites or facilities, called \( f_1, f_2, \ldots, f_M \), and let \( d_{i,j} \) be the distance from \( c_i \) to \( f_j \). The p-median problem consists of opening \( p \) facilities and assigning each client to its closest open facility, in order to minimize the total distance. We note that in the general p-median problem, the graph of the possible assignments is not complete, which can be modeled with \( d_{i,j} = +\infty \). In our application, the graph is complete, the points \( f_1, f_2, \ldots, f_M \) are exactly \( c_1, c_2, \ldots, c_N \) and \( d_{i,j} \) is the Euclidean distance in \( \mathbb{R}^2 \).

The p-median problem is naturally formulated within the Integer Linear Programming (ILP) framework. A first ILP formulation defines binary variables \( x_{i,j} \in \{0, 1\} \) and \( y_j \in \{0, 1\} \). \( x_{i,j} = 1 \) if and only if the customer \( i \) is assigned to the depot \( j \). \( y_j = 1 \) if and only if the point \( f_j \) is chosen as a depot. Following ILP formulation expresses the p-median problem:

\[
\begin{align*}
\min_{x,y} & \quad \sum_{j=1}^{n} \sum_{i=1}^{n} d_{i,j} x_{i,j} \\
\text{s.t.} & \quad \sum_{j=1}^{n} y_j = p \\
& \quad \sum_{j=1}^{n} x_{i,j} = 1, \quad \forall i \in [1, n] \\
& \quad x_{i,j} \leq y_j, \quad \forall (i, j) \in [1, n]^2, \\
& \quad \forall i, j, x_{i,j}, y_j \in \{0, 1\}
\end{align*}
\]

The p-median problem was proven NP-hard in the general case [18]. The p-median problem in \( \mathbb{R}^2 \) with an Euclidian distance is also NP-hard [21]. With "graded" distances as defined in [15], the p-median problem is NP-complete when the distance matrix is graded up the rows or graded down the rows. The p-median problem is polynomial in a tree structure [32].

The p-median problem can be solved to optimality using ILP techniques. The formulation (11) is tightened in [12] for a more efficient resolution with a Branch & Bound (B&B) solver. For larger sizes of instances, Lagrangian relaxations were investigated in [6, 28] or with column-and-row generation as in [2]. Heuristic algorithms are also widely studied [23].

3.2 The p-center problems

We note firstly that there exist two kinds of p-center problems: the discrete and the continuous p-center problems. Generally, the p-center problem consists in
locating \( p \) facilities among a set of possible locations and assigning \( N \) clients, called \( c_1, c_2, \ldots, c_N \), to the facilities in order to minimize the maximum distance between a client and the facility to which it is allocated. The continuous p-center problem assumes that any place of location can be chosen, whereas the discrete p-center considers a subset of \( M \) potential sites denoted \( f_1, f_2, \ldots, f_M \) similarly with the p-median problem. Our application is a discrete p-center problem, the points \( f_1, f_2, \ldots, f_M \) being exactly \( c_1, c_2, \ldots, c_N \). Similarly with the p-median problem, following ILP formulation models the discrete p-center problem:

\[
\begin{align*}
\min_{x, y, z} & \quad z \\
\text{s.t.} & \quad \sum_{j=1}^{n} \sum_{i=1}^{n} d_{i,j} x_{i,j} \leq z & \forall i \in [1, n] \\
& \quad \sum_{j=1}^{n} y_{j} = p & \forall i \in [1, n] \\
& \quad \sum_{j=1}^{n} x_{i,j} = 1 & \forall i \in [1, n] \\
& \quad x_{i,j} \leq y_{j} & \forall (i, j) \in [1, n]^2, \\
& \quad x_{i,j}, y_{j} \in \{0, 1\} & \forall i, j \\
\end{align*}
\]

(12)

where the binary variables \( x_{i,j} \in \{0, 1\} \) and \( y_{j} \in \{0, 1\} \) are defined with \( x_{i,j} = 1 \) if and only if the customer \( i \) is assigned to the depot \( j \) and \( y_{j} = 1 \) if and only if the point \( f_j \) is chosen as a depot.

The discrete p-center problem with triangle inequality is not only NP-complete but also any \( \alpha \)-approximation for \( \alpha < 2 \) is NP-hard [16, 17]. The discrete p-center problem in \( \mathbb{R}^2 \) with a Euclidean distance is also NP-hard [21]. The k-center problem is polynomial time solvable in a tree structure [22]. To solve to optimality discrete p-center problems, an ILP formulation proposed in [13] is tighter than the formulation in [12]. Similarly to the p-median problem, heuristics are efficient for p-center problems [24].

4 Intermediate results

This section gathers the intermediate result necessary to design a common dynamic programming and to prove the complexity. Some elementary proofs are gathered in Appendix A.

4.1 Indexation of the Pareto front

In this section, we analyze some properties of the relations \( \prec \) and \( \preceq \), before defining an order relation and a new indexation among the points of \( E \). The following lemma is extends trivially the properties of \( \leq \) and \( < \) in \( \mathbb{R} \):

**Lemma 1.** \( \preceq \) is an order relation, and \( \prec \) is a transitive relation:

\[
\forall x, y, z \in \mathbb{R}^2, \quad x \prec y \text{ and } y \prec z \implies x \prec z
\]

(13)

The following proposition implies an order among the points of \( E \), allowing a reindexation. This proposition is proven in the Appendix A, the new order is equivalent to sorting the elements of \( E \) so that the first indexes are decreasing, for a complexity in \( O(N \log N) \):
Figure 2: Illustration of a 2-dimensional Pareto front with 15 points and the indexation implied by Proposition 1

**Proposition 1** (Total order). Points \( (x_i) \) can be indexed such that:

\[ \forall (i_1, i_2) \in [1; N]^2, \quad i_1 < i_2 \implies x_{i_1} \prec x_{i_2} \]  
\[ (14) \]

\[ \forall (i_1, i_2) \in [1; N]^2, \quad i_1 \leq i_2 \implies x_{i_1} \preceq x_{i_2} \]  
\[ (15) \]

This property is stronger than the property that \( \preceq \) induces a total order in \( E \). Furthermore, the complexity of the sorting reindexation is in \( O(N \log N) \)

### 4.2 Optimality properties

In this section, a common characterization of optimal solutions of problems (6) for the p-median and the p-center problems is given and proven in the Appendix A.

**Proposition 2.** We suppose that \( E \) is indexed as in Proposition 1. There exists optimal solutions of the minimization problem (6) with objective defined in (8) using only clusters \( C_{i,i'} = \{ x_j \}_{j \in [i,i']} = \{ x \in E | \exists j \in [i,i'], x = x_j \} \)

**Proposition 3.** We suppose that \( E \) is indexed as in Proposition 1. The optimal solutions of the minimization problem (6) with objective defined in (7) use necessarily only clusters \( C_{i,i'} = \{ x_j \}_{j \in [i,i']} = \{ x \in E | \exists j \in [i,i'], x = x_j \} \)

Propositions 2, 3 imply that optimal solutions of p-median or the p-center clustering problems (6) can be designed using only subsets \( C_{i,i'} \). Enumerating such partitions with a brute force algorithm would lead to \( \Theta(N^K) \) computations of the cost of the partition. This is not enough to guarantee to have a polynomial algorithm, but it is a first step for the clustering algorithm of section 5. The next sections investigate the complexity to compute the cost of the possibly optimal clusters \( C_{i,i'} \) in the both cases.
4.3 Computing the costs for the p-median problem

We define in this section \( c_{i,i'} \) as the cost of cluster \( C_{i,i'} \) for the p-median clustering. By definition:

\[
\forall i < i', \quad c_{i,i'} = f_{ctr}(C_{i,i'}) = \min_{j \in [i,i']} \sum_{k \in [i,i']} ||x_j - x_k|| \quad (16)
\]

The naive complexity of these computations enumerating all the possibilities for the centers is in \( \Theta(N^4) \). This section will reduce this complexity to \( \Theta(N^3) \). We define as following \( d_{i,c,i'} \), related to the wished \( c_{i,i'} \) with formula (18):

\[
\forall i \leq c \leq i', \quad d_{i,c,i'} = \sum_{k=i}^{i'} ||x_k - x_c|| \quad (17)
\]

\[
\forall i \leq i' \quad c_{i,i'} = \min_{l \in [i,i']} d_{i,l,i'} \quad (18)
\]

\( d \) fulfills the induction formula:

\[
\forall i \leq c \leq i' < N \quad d_{i,c,i'+1} = d_{i,c,i'} + ||x_{i'+1} - x_c|| \quad (19)
\]

Computation of \( d_{i,c,i'+1} \) having already calculated \( d_{i,c,i'} \) is in \( O(1) \). Algorithm 2 uses the relations (19) to compute the matrix \( d_{i,c,i'} \). The different values of \( c_{i,i'} \) are obtained from the computations \( d_{i,c,i'} \) with (18).

Algorithm 2: Computation of matrix \( c_{i,i'} \) for the p-median problem

define matrix \( c \) with \( c_{i,i'} = 0 \) for all \((i,i') \in [1;N]^2 \) with \( i \leq i' \).

define matrix \( d \) with \( d_{i,l,i'} = 0 \) for all \((i,l,i') \in [1;N]^3 \) with \( i \leq l \leq i' \.

for \( l = 1 \) to \( N \) //consider subset of cardinal \( l \)

for \( i = 1 \) to \( N - l \)

for \( k = i \) to \( i + l \)

\[ d_{i,k,i+l} = d_{i,k,i+l-1} + ||x_{i+l} - x_k|| \]

end for

Compute \( c_{i,i+l} = \min_{k \in [i,i+l]} d_{i,k,i+l} \)

end for

end for

return matrix \( c_{i,i'} \)

Proposition 4. Let \( E = \{x_1, \ldots, x_N \} \) a subset of \( N \) points of \( \mathbb{R}^2 \), such that for all \( i \neq j \), \( x_i \neq x_j \). Computing the matrix of costs \( c_{i,i'} \) for the p-median problem for all the possible optimal clusters \( C_{i,i'} = \{x_j\}_{j \in [i,i']} = \{x \in E | \exists j \in [i,i'], x = x_j\} \) has a complexity in \( \Theta(N^3) \).

Proof: The induction formula (19) uses only values \( d_{i,k,i+l} \) with \( l' < l \). In Algorithm 2, it is easy to show by induction that \( d_{i,k,i+l} \), and also \( c_{i,i+l} \), has its final value for all \( l \in [1, N] \) at the end of the for loops from \( k = i \) to \( i + l \). Let us analyze the complexity. Let \( \alpha \) the time to compute \( d_{i,k,i+l} = d_{i,k,i+l-1} + ||x_{i+l} - x_k|| \).
Defining $\beta$ as the time to compute an operation like $\min(d_{i,k,i'}, d_{i,k+1,i'})$ and to store the result, the time to compute $c_{i,i+1} = \min_{k \in [i,i+1]} d_{i,k,i+1}$ is a $\beta l$.

$$T_N = \sum_{i=1}^{N} \sum_{l=1}^{N-l} (\beta l + (l+1)\alpha) = \sum_{i=1}^{N} \sum_{l=1}^{N-l} (\beta l + (l+1)\alpha)$$

$$T_N = \sum_{i=1}^{N} (\beta l(N-l) + \alpha(l+1)(N-l)) = \Theta(N^3) \square$$

### 4.4 Computing the costs for the p-center problem

We define in this section $c_{i,i'}$ as the cost of cluster $C_{i,i'}$ for the p-center clustering. By definition:

$$\forall i < i', \quad c_{i,i'} = f_{ctr}(C_{i,i'}) = \min_{j \in [i,i']} \max_{k \in [i,i']} \|x_j - x_k\| \quad (20)$$

The computation of max ($[x_j - x_i], [x_j - x_i]$) having a complexity in $O(1)$, the minimization over $j \in [i,i']$ induces $O(i' - i)$ operations in $O(1)$, hence a complexity in $O(i' - i)$. Processing such computations of $c_{i,i'}$ for all $i < i'$, this would lead to a complexity in $O(N^3)$. The following developments improves this complexity with a logarithmic search thanks to the Lemma 2 proven in Appendix A. We define:

$$\forall i < i', \forall j \in [i,i'] \quad f_{i,i'}(j) = \max ([x_j - x_i], [x_j - x_i']) \quad (21)$$

**Lemma 2.** Let $(i, i')$ with $i < i'$. $f_{i,i'}: j \in [i,i'] \rightarrow f_{i,i'}(j)$ is strictly decreasing before reaching first a minimum $f_{i,i',1}, f_{i,i',1+1} \geq f_{i,i',1}$, and then is strictly increasing for $j \in [l+1, i']$

**Proposition 5.** Let $E = \{x_1, \ldots, x_N\}$ a subset of $N$ points of $\mathbb{R}^2$, such that for all $i \neq j$, $x_i \neq x_j$. Computing the matrix of costs $c_{i,i'}$ for the p-center problem for all the possible optimal clusters $C_{i,i'} = \{x_j\}_{j \in [i,i']} = \{x \in E \mid \exists j \in [i,i'], x = x_j\}$ has a complexity in $O(N^2 \log N)$.

**Proof:** We prove firstly that Algorithm 1 computes the right values of $c_{i,i'}$ for given $i < i'$, with a complexity in $O(\log N)$. Let $i < i'$. If $i' - i \leq 2$, Algorithm 2 gives the trivial cost of $c_{i,i'}$ in $O(1)$.

We suppose thus $i' - i > 2$, and having values $i \leq m, M \leq i'$ such that the center of the cluster $C_{i,i'}$ is between $m$ and $M$. The initialization $m = i$, $M = i'$ is valid. Algorithm 2 computes $l = \lceil \frac{m+M}{2} \rceil$ and the values $f_{i,i'}(l)$ and $f_{i,i'}(l+1)$, operations in $O(1)$. If $f_{i,i'}(l) < f_{i,i'}(l+1)$, the Lemma 2 ensures that the center of $C_{i,i'}$ is before $l$, so that it can be reactualized $M = l$. If $f_{i,i'}(l) > f_{i,i'}(l+1)$, the Lemma 2 ensures that the center of $C_{i,i'}$ is after $l+1$; so that it can be reactualized $m = l+1$. The case $f_{i,i'}(l) = f_{i,i'}(l+1)$ is special, the Lemma 2 ensures that the center of $C_{i,i'}$ can be either $l$ or $l+1$. Iterating this procedure till $M - m < 2$ finds the center of $C_{i,i'}$ using at most $\log(i' - i)$ operations in $O(1)$. This proves the termination and the complexity in $O(\log N)$ to compute costs $c_{i,i'}$ for given $i < i'$. Hence, the complexity of all the independent computation $c_{i,i'}$ for all $i < i'$ is in $O(N^2 \log N)$. □
Algorithm 1: Computation of $c_{i,i'}$ for the p-center problem

**input:** indexes $i < i'$

**output:** the cost $c_{i,i'} = f_{str}(G_{i,i'})$

**Initialization:**
- define $idInf = i$, $valInf = \|x_i - x_{i'}\|$
- define $idSup = i'$, $valSup = \|x_i - x_{i'}\|$

while $idSup - idInf > 2$ //Dichotomic search

Compute $idMid = \left\lfloor \frac{i + i'}{2} \right\rfloor$, $valTemp = f_{i,i',idMid}$, $valTemp2 = f_{i,i',idMid+1}$

if $valTemp = valTemp2$

$idInf = idMid$, $valInf = valTemp$
$idSup = 1+idMid$, $valSup = valTemp2$

if $valTemp < valTemp2$ // increasing phase

$idSup = idMid$, $valSup = valTemp$

if $valTemp > valTemp2$

$idInf = 1+idMid$, $valInf = valTemp2$

end while

return $\min(valInf, valSup)$

5 Dynamic Programming algorithm

The additive propriety (6) and the Propositions [2, 3] allow to derive a common dynamic programming algorithm. Only the initialization differs, computing all the $c_{i,i'}$ with $i < i'$ with the specific algorithm presented in the section 4. 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:

$$
\forall i \in [1, N], \forall k \in [2, K], \quad C_{i,k} = \min_{j \in [1,i]} C_{j-1,k-1} + c_{j,i} \quad (22)
$$

with the convention $C_{0,k} = 0$ for all $k \geq 0$. The case $k = 1$ is directly given by:

$$
\forall i \in [1, N], \quad C_{i,1} = c_{1,i} \quad (23)
$$

These relations allow to compute the optimal values of $C_{i,k}$ by dynamic programming in the Algorithm 3. $C_{N,K}$ is the optimal solution of the k-means problem, a backtracking algorithm on the matrix $(C_{i,k})_{i,k}$ allows to compute the optimal partitioning clusters.

**Theorem 1.** Let $E = \{x_1, \ldots, x_N\}$ a subset of $N$ points of $\mathbb{R}^2$, such that for all $i \neq j$, $x_i \not\in x_j$. The p-medians and p-center problems are solvable to optimality in polynomial time with Algorithm 3. The complexity of Algorithm 3 is in $O(N^3)$ for the p-median problem, in $O(N^2(K + \log N))$ for the p-center problem.

**Proof:** The induction formula (22) uses only values $C_{i,j}$ with $j < k$ in Algorithm 3. $C_{N,K}$ is thus at the end of these loops the optimal value of the p-median or p-center clustering among the $N$ points of $E$. Induction proves
Algorithm 3: p-median or p-center clustering in a 2d-Pareto Front

Input:
- \( Pblm \) a p-center or p-median problem;
- \( N \) points of \( \mathbb{R}^2 \), \( E = \{x_1, \ldots, x_N\} \) such that for all \( i \neq j \), \( x_i \not\equiv x_j \);
- \( K \in \mathbb{N} \) the number of clusters

\( \text{Cluster2dPareto}(E, K, Pblm) \)
- initialize matrix \( c \) with \( c_{i,j} = 0 \) for all \( (i, j) \in [1; N]^2 \)
- initialize matrix \( C \) with \( C_{i,k} = 0 \) for all \( i \in [0; N], k \in [1; K] \)
- initialize \( P = \text{null} \), a set of sub-intervals of \([1; N]\).
- sort \( E \) following the order of Proposition 1
- compute \( c_{i,j} \) for all \( (i, j) \in [1; N]^2 \)
- for \( i = 1 \) to \( N \) //Construction of the matrix \( C \)
  - set \( C_{i,k} = c_{1,i} \)
  - 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
- \( i = N \) //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 \( P \)
  - \( i = j - 1 \)
- end for

return \( C_{N,K} \) the optimal cost and the partition \( P \) giving the cost \( C_{N,K} \)

that \( C_{i,k} \) has its final value for all \( i \in [1, N] \) at the end of the for loops from \( k = 2 \) to \( K \). The reason is that The backtracking phase searches for the equal-}
ities in \( C_{i,k} = C_{j-1,k-1} + c_{j,i} = \min_{j \in [1,i]} C_{j-1,k-1} + c_{j,i} \), proving that such cluster \( C_{j,i} \) allows to give an optimal solution.

Let us analyze the complexity. Sorting and indexing the elements of \( E \)

Proposition 1 has a complexity in \( O(N \log N) \). The computation of the matrix \( c_{i,i'} \) has a complexity in \( O(N^3) \) for the p-median problem, in \( O(N^2 \cdot \log N) \) for the p-center problem. The construction of the matrix \( C_{i,k} \) requires \( N \times K \) computations of \( \min_{j \in [1,i]} C_{j-1,k-1} + c_{j,i} \), which are in \( O(N) \), the complexity of this phase is in \( O(K \cdot N^2) \). The final backtracking phase requires \( K \) computations having a complexity in \( O(N) \), the complexity is in \( O(K \cdot N) \).

The bottleneck for the p-median problem is the computation of the matrix \( c_{i,i'} \) as \( K < N \), the complexity of Algorithm 3 is in \( O(N^3) \) for the p-median problem. For the p-center problem, the complexity is in \( O(N^2 \cdot (K + \log N)) \) with the computation of the the matrices \( c_{i,i'} \) and \( C_{i,k} \). \( \Box \)
6 Discussions

This section discusses some hypotheses and the implications and applications of Theorem 1 and Algorithm 3.

6.1 Importance of hypotheses

The p-median and the p-center problems were proven NP-hard even in a Euclidean space of dimension 2 since [21]. This emphasizes that the additional hypothesis to have non-dominated solutions with Pareto dominance is crucial to have a polynomial algorithm to solve these clustering problems to optimality.

The non-dominance hypothesis in $\mathbb{R}^2$ induces a 1-dimensional structure thanks to Proposition 1. It is a basis for the dynamic programming algorithm. The non-linearity of the Pareto front induced more complications in the proofs, with no additivity of distance but a triangular inequality.

Each instance of the general case in $\mathbb{R}^2$ can be injected in an instance in a Pareto frontier in $\mathbb{R}^3$, using hyperplanes. It ensures that the case of a Pareto frontier in $\mathbb{R}^3$ is also NP-complete thanks to [21].

6.2 Relations with other results from the literature

A posteriori, the problem and the dynamic programming algorithm can be connected with other results from the literature.

6.2.1 K-means clustering

k-means clustering is one of the most famous unsupervised learning problems, and is widely studied in the literature. k-means is also a problem of the type \( f_{\text{means}}(P) \), defining the objective function with:

\[
f_{\text{means}}(P) = \frac{1}{\text{card}(P)} \sum_{x \in P} \| x - \frac{1}{\text{card}(P)} \sum_{y \in P} y \|^2
\]

k-means was proven to be NP-hard in [9]. Being in a general Euclidean space, K-means can be solved by a Polynomial Time approximation Schemes (PTAS), i.e, algorithms allowing to have a $1 + \varepsilon$ approximation solvable in polynomial time for all $\varepsilon > 0$, as developed in [4]. Special cases of k-means are also been proven NP-hard in a general Euclidean space: the problem is still NP-hard when the number of clusters k is at least 2 [1], or when the dimensionality is 2 [20].

The case $K = 1$ is trivially polynomial. The 1-dimensional case was proven solvable in polynomial time thanks to a dynamic programming algorithm in [33]. This algorithm has similarities with Algorithm 3, and has a complexity in $O(KN^2)$ like in the construction of the matrix $C_{i,k}$ in Algorithm 3. Clustering a linear Pareto Front with k-means is equivalent to the 1-dimensional case of k-means, it is solvable polynomial time, whereas superior dimensions leads to NP-completeness similarly to the p-median and p-center problems.
6.2.2 Hypervolume subset selection

We note also similar results for the hypervolume subset selection, selecting K-points while maximizing the volume of the dominated region by the selected points \([2, 29]\). The problem is NP-hard in 3 dimensions and provided an exact algorithm in \(n^{O(\sqrt{k})}\) and a polynomial-time approximation scheme for any constant dimension \(d\) \([7]\). The 2-dimensional case is solvable in polynomial time thanks to a dynamic programming algorithm \([2]\). This algorithm has similarities with Algorithm 3, and has a complexity in \(O(KN^2)\) like in the construction of the matrix \(C_{i,k}\) in Algorithm 3.

6.2.3 Perspectives to improve the complexity for the Algorithm 3?

Both previous cases are solved by dynamic programming with a complexity in \(O(KN^2)\) in \([2, 33]\). In both cases, the complexity of the dynamic programming was improved. The complexity of 1-dimension k-means was improved in \([14]\), for a dynamic programming algorithm with a complexity in \(O(KN)\) using memory space in \(O(N)\). The complexity of the dynamic programming for the hypervolume subset selection was improved in \(O(KN + N \log N)\) by \([3]\) and in \(O(K(N - K) + N \log N)\) by \([19]\). Some similar improvements may be still valid for the Algorithm 3, to accelerate the construction of the matrix \(C_{i,k}\) in Algorithm 3. However, the complexity in Algorithm 3 is mainly due to the initialization phase, which reduce the impact of improving the complexity of the construction of the matrix \(C_{i,k}\).

6.3 Adding cardinality constraints

Similarly to \([25]\), Algorithm 3 allows to incorporate cardinality constraints, considering only clusters \(C_{i,i'}\) with specific cardinality of \(i' - i\) and setting values \(c_{i,i'} = +\infty\). Such cardinality constraints can have a positive impact on the complexity of Algorithm 3. Computations \(C_{i,k} = \min_{j \in [1,i]} C_{j-1,k-1} + c_{j,i}\) are easier with less possibilities in \(j\) to enumerate. For the p-center problem, the computations of \(c_{i,i'}\) are independent, the useless computations of \(c_{i,i'}\) can be ignored. Allowing for each \(i \in [1,N]\) only \(\alpha K\) definite values of \(c_{i,i'}\), it improves the final complexity. Considering only the subsets \([i, i']\) with \([N/K] - \alpha K < |i' - i| < [N/K] + \alpha K\) for a given \(\alpha \in \mathbb{N}\), the construction of the matrix \(C_{i,k}\) is in \(O(K^2.N)\) with operations \(C_{i,k} = \min_{j \in [1,i]} C_{j-1,k-1} + c_{j,i}\) having a complexity in \(O(K)\). The initialization for the p-center is in \(O(K.N \log N)\), the complexity would be in \(O(K.N.(K + \log N))\) for the p-center problem.

6.4 Towards a parallel implementation

The computations of Algorithm 3 can be accelerated with a parallel implementation. The construction of the matrix \(C_{i,k}\) requires independent computations for a given \(k\), using the final values in \(k - 1\). For a parallel implementation, it requires to wait that all the coefficient \(C_{i,k-1}\) are computed to start the computations of \(C_{i,k-1}\). With a distributed implementation with Message Passing
Interface (MPI), this requires to broadcast the results of $C_{i,k}$ computations to the other threads.

The initial computation of $c_{i,i'}$ are independent for all $i < i'$ for the p-center problem and are thus easily parallelized. For the p-median problem, Algorithm 2 can be parallelized similarly with the $C_{i,k}$ computations, the waiting criterion to synchronize the computations is that all the $d_{i,k,i+l}$ are finished before the computations of $d_{i,k,i+l+1}$. The computations of $c_{i,i+l} = \min_{k \in [i,i+l]} d_{i,k,i+l}$ are easily parallelized with independent sub-computations.

6.5 Applications to bi-objective meta-heuristics

The hypothesis defining $E$ is verified for non-dominated points of bi-objective optimization. The initial motivation of this work was to aid the decision makers when a multi-objective optimization approach without preference furnishes a large set of non dominated solutions. In this application, the value of $K$ is small, for human analyses to give some preferences.

A posteriori, the complexity allows to use Algorithm 3 inside bi-objective optimization meta-heuristics. Archiving Pareto fronts is a common issue of population meta-heuristics facing multi-objective optimization problems [5, 30]. A key issue is to have diversified points of the Pareto front in the archive, to compute diversified solutions along the current Pareto front. Algorithm 3 can be used to address this issue, embedded in multi-objective optimization approaches, similarly with [2]. Archiving diversified solutions of Pareto sets has an application for the diversification of genetic algorithms to select diversified solutions for cross-over and mutation phases [34], but also for swarm particle optimization heuristics [27] or multi-objective simulated annealing [5].

Embedded in bi-objective meta-heuristics, Algorithm 3 is called iteratively. Having a dynamic programming algorithm, this makes easier online optimization where several points are not changing from an iteration to another. In this case, the computation of matrices $c, C$ can reuse the previous values that are still valid, which accelerates the computations of Algorithm 3.

7 Conclusion and perspectives

This paper examined properties of the p-median and the p-center problems in the special case of a discrete set of non-dominated points in a two dimensional Euclidian space. A common characterization of optimal clusters is proven both problems. It allows to solve these problems to optimality with a unified dynamic programming algorithm. A polynomial complexity is proven in $O(N^3)$ for the p-median problem and in $O(N^2(K + \log N))$ for the p-center problem, whereas both problems are NP-hard without the property of non-dominance in $\mathbb{R}^2$. Furthermore, the algorithm can be adapted to consider cardinality constraints for the clusters. The presented algorithm can also be parallelized for a distributed implementation to speed up the computational time.
The implications and applications of these results are discussed. The initial motivation was the selection a small number of representative solutions for a human decision making. Furthermore, the complexity of the algorithms allows to consider these clustering algorithms inside a multi-objective. This last point offers implementation perspective to use the dynamic programming algorithm in an online optimization context. Other research perspectives are to extend the result of this paper to other clustering algorithms like k-means, k-medoids and weighted clustering algorithms. In these cases, the critical point is to able to prove that the optimal properties of Propositions 2-3 are still valid.

References

[1] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

[2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences. In *Proceedings of GECCO 2009*, pages 563–570. ACM, 2009.

[3] P. Avella, A. Sassano, and I. Vasiliev. Computational study of large-scale p-median problems. *Mathematical Programming*, 109(1):89–114, 2007.

[4] P. Awasthi, A. Blum, and O. Sheffet. Stability yields a PTAS for k-median and k-means clustering. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 309–323, 2010.

[5] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE transactions on evolutionary computation*, 12(3):269–283, 2008.

[6] C. Beltran, C. Tadonki, and J. P. Vial. Solving the p-median problem with a semi-lagrangian relaxation. *Computational Optimization and Applications*, 35(2):239–260, 2006.

[7] K. Bringmann, S. Cabello, and M. Emmerich. Maximum volume subset selection for anchored boxes. *arXiv preprint arXiv:1803.00849*, 2018.

[8] K. Bringmann, T. Friedrich, and P. Klitzke. Two-dimensional subset selection for hypervolume and epsilon-indicator. In *Annual Conference on Genetic and Evolutionary Computation*, pages 589–596. ACM, 2014.

[9] S. Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, San Diego, 2008.

[10] N. Dupin. *Modélisation et résolution de grands problèmes stochastiques combinatoires: application à la gestion de production d’électricité*. PhD thesis, Univ. Lille 1, 2015.

[11] M. Ehrgott and X. Gandibleux. Multiobjective combinatorial optimization - theory, methodology, and applications. In *Multiple criteria optimization: State of the art annotated bibliographic surveys*, pages 369–444. Springer, 2003.

[12] S. Elloumi. A tighter formulation of the p-median problem. *Journal of combinatorial optimization*, 19(1):69–83, 2010.

[13] S. Elloumi, M. Labbé, and Y. Pochet. A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing*, 16(1):84–94, 2004.

[14] A. Grundlund, K. Larsen, A. Mathiasen, J. Nielsen, S. Schneider, and M. Song. Fast exact k-means, k-mediants and Bregman divergence clustering. In *Id*. *arXiv preprint arXiv:1701.07324*, 2017.

[15] L. Guo-Hui and G. Xue. K-center and k-median problems in graded distances. *Theoretical computer science*, 207(1):181–192, 1998.
[16] D.S. Hochbaum. When are NP-hard location problems easy? Annals of Operations Research, 1(3):201–214, 1984.

[17] W. Hsu and G. Nemhauser. Easy and hard bottleneck location problems. Discrete Applied Mathematics, 1(3):209–215, 1979.

[18] O. Kariv and S. Hakimi. An algorithmic approach to network location problems. ii: The p-median. SIAM Journal on Applied Mathematics, 37(3):539–560, 1979.

[19] T. Kuhn, C. Fonseca, L. Paquete, S. Ruzika, M. Duarte, and J. Figueira. Hypervolume subset selection in two dimensions: Formulations and algorithms. Evolutionary Computation, 24(3):411–425, 2016.

[20] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is NP-hard. Theoretical Computer Science, 442:13–21, 2012.

[21] N. Megiddo and K. Supowit. On the complexity of some common geometric location problems. SIAM Journal on Computing, 13(1):182–196, 1984.

[22] N. Megiddo and A. Tamir. New results on the complexity of p-centre problems. SIAM Journal on Computing, 12(4):751–758, 1983.

[23] N. Mladenović, J. Brimberg, P. Hansen, and J. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research, 179(3):927–939, 2007.

[24] N. Mladenović, M. Labbé, and P. Hansen. Solving the p-center problem with tabu search and variable neighborhood search. Networks, 42(1):48–64, 2003.

[25] F. Nielsen and R. Nock. Optimal interval clustering: Application to Bregman clustering and statistical mixture learning. IEEE Signal Processing Letters, 21(10):1289–1292, 2014.

[26] T. Peugeot, N. Dupin, M-J Sembely, and C. Dubecq. MBSE, PLM, MIP and Robust Optimization for System of Systems Management, Application to SCCOA French Air Defense Program. In Complex Systems Design & Management, pages 29–40. Springer, 2017.

[27] G. Pulido and C. Coello. Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer. In Genetic and Evolutionary Computation Conference, pages 225–237. Springer, 2004.

[28] A. Santos. Solving large p-median problems using a lagrangean heuristic. 2009.

[29] S. Saym. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Mathematical Programming, 87(3):543–560, 2000.

[30] E. Talbi. Metaheuristics: from design to implementation, volume 74. John Wiley & Sons, 2009.

[31] E-G. Talbi, M. Basseur, A. Nebro, and E. Alba. Multi-objective optimization using metaheuristics: non-standard algorithms. International Transactions in Operational Research, 19(1-2):283–305, 2012.

[32] A. Tamir, D. Pérez-Brito, and J. Moreno-Pérez. A polynomial algorithm for the p-centdian problem on a tree. Networks: An International Journal, 32(4):255–262, 1998.

[33] H. Wang and M. Song. Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. The R Journal, 3(2):29, 2011.

[34] E. Zio and R. Bazzo. A clustering procedure for reducing the number of representative solutions in the pareto front of multiobjective optimization problems. European Journal of Operational Research, 210(3):624–634, 2011.
Appendix: Proof of the intermediary lemmas

This section gives the elementary proofs and results necessary for the results of section 4.

Proof of the Proposition 1

We prove (14) by induction on $N \in \mathbb{N}$. (15) is an immediate corollary of (14).

For $N = 1$, the property (14) is trivially verified. Let us suppose $N > 1$ and the Induction Hypothesis (IH) that (14) is true for $N - 1$.

Let $A = \{a \in \mathbb{R} \mid \exists x_i \in E, \exists b \in \mathbb{R}, x_i = (a, b)\}$. $A$ is a finite subset of $\mathbb{R}$, it has a maximum. Let $m$ such that $x_m = (x^1_m, x^2_m)$ and $x^1_m = \max A$.

Let $m' \neq m$, $x_m \neq x_{m'}$ with the definition of $E$, it implies $x_{m'} < x_m$ or $x_m < x_{m'}$.

$x^1_m < x^1_{m'}$ implies $x^1_{m'} > x^1_m$ which is in contradiction with $x^1_m = \max A$, thus $x_{m'} < x_m$.

Hence:
\[\forall i \in [1; N] - \{m\}, x_i < x_m \tag{24}\]

Applying (IH) to $[1; N] - \{m\}$ allows to index $[1; N] - \{m\}$ as $i_1 < \cdots < i_{N-1}$ with property (14). Defining $i_N = m$, the missing inequalities are furnished by (24) to have the result true for $N$. It proves by induction that (14) is true for all $N \in \mathbb{N}$. The new order is equivalent to sorting the elements of $E$ so that the first indexes are decreasing, for a complexity in $O(N \log N)$. \(\square\)

Lemma 3. We suppose that points $(x_i)$ are sorted following Proposition 4. Let $(i_1, i_2, i_3) \in \{1; n\}^3$.

\[i_1 \leq i_2 < i_3 \implies d(x_{i_1}, x_{i_2}) < d(x_{i_1}, x_{i_3}) \tag{25}\]

\[i_1 < i_2 \leq i_3 \implies d(x_{i_2}, x_{i_3}) < d(x_{i_1}, x_{i_3}) \tag{26}\]

Proof of Lemma 3. We note firstly that the equality cases are trivial, so that we can suppose $i_1 < i_2 < i_3$ in the following proof. We prove the property (26), the proof of (25) is analogous.

Let $i_1 < i_2 < i_3$. We note $x_{i_1} = (x^1_{i_1}, x^2_{i_1}), x_{i_2} = (x^1_{i_2}, x^2_{i_2})$ and $x_{i_3} = (x^1_{i_3}, x^2_{i_3})$.

Proposition 1 ordering ensures $x^1_{i_1} < x^1_{i_2} < x^1_{i_3}$ and $x^2_{i_1} > x^2_{i_2} > x^2_{i_3}$.

\[d(x_{i_1}, x_{i_2})^2 = (x^1_{i_1} - x^1_{i_2})^2 + (x^2_{i_1} - x^2_{i_2})^2\]

With $x^1_{i_1} > x^1_{i_2} > x^1_{i_3} > 0$, $(x^1_{i_1} - x^1_{i_2})^2 < (x^1_{i_1} - x^1_{i_3})^2$

Thus $d(x_{i_1}, x_{i_2})^2 < (x^1_{i_1} - x^1_{i_3})^2 + (x^2_{i_1} - x^2_{i_3})^2 = d(x_{i_1}, x_{i_3})^2$. \(\square\)

Proof of Proposition 6. We suppose having an optimal clustering $C_1, \ldots, C_K$ of the $K$-median problem, the center of $C_k$ is denoted $c_k$ for all $k \in [1; K]$. Let $x_i \in E$, and $j$ such that $x_i \in C_j$. Then $c_j$ is one of the closest center from $x_i$:

\[\forall k \in [1; K], \quad d(x_i, c_j) \leq d(x_i, c_k) \tag{27}\]

Proof of Proposition 6. We prove the result by contradiction and we suppose that it exists $j' \in [1; K]$ such that $d(x_i, c_{j'}) < d(x_i, c_j)$. We define the clusters $(C'')_{k \in [1; K]}$ with $C''_k = C_k$ if $k \notin \{j, j'\}, C''_j = C_j \setminus \{x_i\}$ and $C''_{j'} = C_{j'} \cup \{x_i\}$. 16
\[ f_{med}(C_j') = \sum_{x \in C_j - \{x_1\}} |x - c_j| = \sum_{x \in C_j} |x - c_j| - d(x_i, c_j) = f_{med}(C_j) - d(x_i, c_j) \]

\[ f_{med}(C_{j'}) = \sum_{x \in C_{j'} \cup \{x_1\}} \|x - c_j\| = \sum_{x \in C_{j'}} \|x - c_j\| - d(x_i, c_{j'}) = f_{med}(C_{j'}) + d(x_i, c_{j'}) \]

\[ \forall k \in [1:K] - \{j, j'\}, \quad f_{med}(C_k') = f_{med}(C_k) \]

Adding these three types of equalities:

\[ \sum_{k=1}^{K} f_{med}(C_k') = \sum_{k=1}^{K} f_{med}(C_k) - d(x_i, c_j) + d(x_i, c_{j'}) \]

\[ d(x_i, c_{j'}) < d(x_i, c_j) \text{ implies that: } \sum_{k=1}^{K} f_{med}(C_k') < \sum_{k=1}^{K} f_{med}(C_k). \] This is in contradiction with the optimality of \( C_1, \ldots, C_K \) in (8). \( \Box \)

**Proof of Proposition 2:**

Let \( \pi \in \Pi_K(E) \) an optimal solution of problem (8). We suppose the existence of \( C_1, C_2 \) two "nested" clusters: it exists \( j < k < j' \) such that \( x_j, x_j' \in C_1 \) and \( x_k \in C_2 \). Denoting \( I = \{j \in [1,N] \mid x_j \in C_1\} \), we can suppose that \( j = \min I \) and \( j' = \max I \). The center of \( C_1 \) is denoted \( x_{c_1}, c_1 \in [j, j'] \). Proposition 3 ensures that \( d(x_{c_1}, x_k) \leq d(x_{c_1}, x_j) \) and \( d(x_{c_1}, x_k) \leq d(x_{c_1}, x_{j'}) \). We define clusters \( C'_k = C_2 - \{x_k\} \) and \( C'_1 = C_1 \cup \{x_k\} \).\( f_{ctr}(C'_2) \leq f_{ctr}(C'_1) \), removing element \( x_k \). \( f_{ctr}(C_1) = \max(d(x_{c_1}, x_j), d(x_{c_1}, x_{j'})) \geq d(x_{c_1}, x_k) \) thus \( f_{ctr}(C'_1) = f_{ctr}(C_1) \). The optimality of \( \pi \) imposes \( f_{ctr}(C_2) = f_{ctr}(C'_2) \), a strict inequality would lead with \( C'_1, C'_2 \) to a strictly better solution of (8) than the optimal one. This can be iterated for all \( k \in [j, j'] \) such that \( x_k \in C_2 \), so that we obtain equivalent solutions in the minimization problem (8) with clusters \( C_1 \) and \( C_2 \) not nested. Iterating this procedure till there exist nested clusters, we have in the last optimal solutions only clusters \( C_{i,i'} = \{x_j\}_{j \in [i,i']} = \{x \in E \mid \exists j \in [i,i'], x = x_j\} \). \( \Box \)

**Proof of Proposition 3:**

We prove the result by induction on \( K \in \mathbb{N} \). For \( K = 1 \), the optimal cluster is \( E = \{x_j\}_{j \in [1,N]} \). We suppose now \( K > 1 \) and the Induction Hypothesis (IH) that Proposition 3 is true for \( K-1 \)-means clustering. We suppose having an optimal solution of the \( K \)-means clustering on \( E \). We denote with \( C \) the cluster of \( x_N \) and \( x_c \) the center of the cluster \( C \). Let \( A = \{i \in [1,N] \mid \forall k \in [i,N], x_k \in C\} \). \( A \) is a subset of \( \mathbb{N} \), non empty as \( N \in A \), it has a minimum. Let \( j = \min \{i \in [1,N] \mid \forall k \in [i,N], x_k \in C\} \).

If \( j = 1 \), \( E = C = \{x_j\}_{j \in [1,N]} \) and the result is proven. We suppose now \( j > 1 \). \( j-1 \notin A \), \( j-1 \in A \) would have been a contradiction to \( j = \min A \).

For all \( k \in [i,j], x_k \in C \), thus the only possibility to have \( j-1 \notin A \) is that \( x_{j-1} \notin C \). We denote with \( C' \) the cluster of \( j-1 \), and \( x_c' \) the center of the cluster \( C' \). Necessarily \( c' < j \) as \( x_c' \notin C \).

We prove by contradiction that \( j-1 < c \). We suppose \( c < j \). Proposition 6 implies \( d(x_{c'}, x_{j-1}) \leq d(x_{c'}, x_{j-1}) \) (applied to \( x_{j-1} \)) and its optimal cluster \( C' \).
With proposition\(^3\) this is possible only if \(c < c'\). We would have thus with Proposition\(^4\) \(x_c < x_{c'} \leq x_{j-1} < x_j\). Applying Proposition\(^5\) we would have \(d(x_{c'}, x_j) < d(x_c, x_j)\) which is in contradiction with Proposition\(^6\) applied to \(x_j\) and its optimal cluster \(C\).

Proposition\(^1\) ensures that \(x_{c'} < x_{j-1} < x_j \leq x_c\), and also \(x_{c'} < x_{j-1} < x_j\). We prove now that for all \(j' < j - 1, x_{j'} \notin C\). Let \(j' < j - 1\).

If \(j' < c\), Proposition\(^5\) with \(x_{j'} < x_{c'} < x_c\) implies directly \(d(x_{c'}, x_{j'}) < d(x_c, x_{j'})\) and thus \(x_{j'} \in C\) would be in contradiction with Proposition\(^6\).

If \(j' > c\), we have following inequalities \(x_{c'} < x_{j'} < x_{j-1} < x_c\). Proposition\(^6\) implies \(d(x_{c'}, x_{j-1}) < d(x_c, x_{j-1})\) applied to \(x_{j-1}\) and its optimal cluster \(C'\). Proposition\(^3\) implies \(d(x_{c'}, x_{j'}) < d(x_{c'}, x_{j-1})\) and also \(d(x_c, x_{j'}) < d(x_c, x_{j-1})\). By transitivity \(d(x_{c'}, x_{j'}) < d(x_{c'}, x_{j-1}) < d(x_c, x_{j-1}) < d(x_c, x_{j'})\). Thus \(d(x_{c'}, x_{j'}) < d(x_c, x_{j'})\), and \(x_{j'} \notin C\) with Proposition\(^6\). This showed that for all \(j' < j, x_{j'} \notin C\). A first consequence is that the cluster \(C\) is exactly \(\{x_i\}_{i \in [i, N]}\), fulfilling the result of Proposition\(^3\). A second consequence is that the other clusters are optimal clustering for \(E' = E - C\) with \(K - 1\)-means. Applying IH to the \(K - 1\) clustering to points \(\{x_i\}_{i \in [1, j-1]}\) prove that the optimal clusters are on the form \(C_{i,i'} = \{x_j\}_{j \in [i, i']}\). It proves by induction that Proposition\(^3\) is true for all \(K \in \mathbb{N}\). \(\square\)

**Proof of Lemma\(^2\)**

We define \(g_{i,i'}, h_{i,i', j}\) with:

\[
g_{i,i'} : j \in [i, i'] \rightarrow |x_j - x_i| \quad \text{and} \quad h_{i,i'} : j \in [i, i'] \rightarrow |x_j - x_{i'}|
\]

Let \(i < i'\). Proposition\(^3\) applied to \(i\) and any \(j, j + 1\) with \(j \geq i\) and \(j < i'\) assures that \(g\) is strictly decreasing. Similarly, Proposition\(^3\) applied to \(i'\) and any \(j, j + 1\) ensures that \(h\) is strictly increasing.

Let \(A = \{j \in [i, i']| \forall m \in [i, j], g_{i,i'}(m) < h_{i,i'}(m)\}. g_{i,i'}(i) = 0\) and \(h_{i,i'}(i) = |x_{i'} - x_i| > 0\) so that \(i \in A\). \(A\) is a non empty and bounded subset of \(\mathbb{N}\), so that \(A\) has a maximum. We note \(l = \max A\). \(h_{i,i'}(i') = 0\) and \(g_{i,i'}(i') = |x_{i'} - x_i| > 0\) so that \(i' \notin A\) and \(l < i'\).

Let \(j \in [i, l - 1]\), \(g_{i,i'}(j) < g_{i,i'}(j + 1)\) and \(h_{i,i',j}(j + 1) < h_{i,i',j}(j)\) using monotony of \(g_{i,i'}\) and \(h_{i,i',j}\), \(f_{i,i'}(j + 1) = \max (g_{i,i'}(j + 1), h_{i,i',j}(j + 1)) = h_{i,i',j}(j + 1)\) and \(f_{i,i'}(j) = \max (g_{i,i'}(j), h_{i,i',j}(j)) = h_{i,i',j}(j)\) as \(j, j + 1 \in A\). Hence, \(f_{i,i'}(j + 1) = h_{i,i',j}(j + 1) < h_{i,i',j}(j) = f_{i,i'}(j)\). It proves that \(f_{i,i'}\) is strictly decreasing in \([i, l].\)

\(l + 1 \notin A\) and \(g_{i,i',l}(l+1) < h_{i,i',l}(l+1)\) to be coherent with the fact that \(l = \max A\).

Let \(j \in [l + 1, i' - 1]\), \(j + 1 > j \geq l + 1\) so \(g_{i,i'}(j + 1) > g_{i,i'}(j)\) and \(h_{i,i',l}(l + 1) > h_{i,i',l}(j + 1)\). \(h_{i,i',l}(l + 1) > h_{i,i',l}(j + 1)\) using monotony of \(g_{i,i'}\) and \(h_{i,i',l}\).

It proves that \(f_{i,i'}\) is strictly increasing in \([l + 1, i']\).

Lastly, the minimum of \(f\) can be reached in \(l\) or in \(l + 1\), depending on the sign of \(f_{i,i'}(l + 1) - f_{i,i'}(l)\). If \(f_{i,i'}(l + 1) = f_{i,i'}(l)\) there are two minimums \(l, l + 1\). Otherwise, there exist a unique minimum \(l_0 \in \{l, l + 1\}\), \(f_{i,i'}\) decreasing strictly before increasing strictly. \(\square\)