Ann Oper Res (2007) 154: 69–92
DOI 10.1007/s10479-007-0181-5

Multiple criteria districting problems

The public transportation network pricing system of the Paris region

Fernando Tavares-Pereira · José Rui Figueira · Vincent Mousseau · Bernard Roy

Published online: 17 May 2007

Abstract Districting problems are of high importance in many different fields. Multiple criteria models seem a more adequate representation of districting problems in real-world situations. Real-life decision situations are by their very nature multidimensional. This paper deals with the problem of partitioning a territory into "homogeneous" zones. Each zone is composed of a set of elementary territorial units. A district map is formed by partitioning the set of elementary units into connected zones without inclusions. When multiple criteria are considered, the problem of enumerating all the efficient solutions for such a model is known as being NP-hard, which is why we decided to avoid using exact methods to solve large-size instances. In this paper, we propose a new method to approximate the Pareto front based on an evolutionary algorithm with local search. The algorithm presents a new solution representation and the *crossover/mutation* operators. Its main features are the following: it

F. Tavares-Pereira (✉)
Departamento de Matemática, Universidade da Beira Interior, Rua Marquês D'Ávila e Bolama,
6201-001 Covilha, Portugal
e-mail: fpereira@mat.ubi.pt

F. Tavares-Pereira
INESC—Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal

F. Tavares-Pereira · J.R. Figueira · V. Mousseau · B. Roy
LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny,
75 775 Paris Cedex 16, France

J.R. Figueira
e-mail: figueira@ist.utl.pt

V. Mousseau
e-mail: mousseau@lamsade.dauphine.fr

B. Roy
e-mail: roy@lamsade.dauphine.fr

F. Tavares-Pereira · J.R. Figueira
CEG-IST, Center for Management Studies, Departamento de Engenharia e Gestão, Instituto
Superior Técnico, Universidade Técnica de Lisboa, Tagus Park, Av. Prof. Cavaco Silva,
2780-990 Porto Salvo, Portugal

deals with multiple criteria; it allows to solve large-size instances in a reasonable CPU time and generates high quality solutions. The algorithm was applied to a real-world problem, that of the Paris region public transportation. Results will be used for a discussion about the reform of its current pricing system.

Keywords Multiple criteria · Districting problems · Evolutionary algorithms · Local search · Combinatorial optimization

1 Introduction

Over the last three decades, many researchers, academics and practitioners from different fields have developed models, built algorithms and implemented solutions concerning the so-called *districting problem*. Elementary units of a territory are grouped into larger clusters or districts, hence producing a district map or partition.

There are many practical questions related to this problem:

- How to define the electoral districts of a country (Bozkaya et al. 2003; Garfinkel and Nemhauser 1970; Hojati 1996; Mehrotra et al. 1998).
- How to establish the different working zones for a travel salesperson team (Easingwood 1973; Hess and Samuels 1971; Shanker et al. 1975; Zoltners and Sinha 1983).
- How to define areas in metropolitan Internet networks for installing hubs (Park et al. 2000).
- How to define the areas for manufactured and consumer goods (Fleischmann and Paraschis 1988).

The same kind of questions also have to be dealt with in police districting (D'Amico et al. 2002); school districting (Ferland and Guénette 1990); districting of salt spreading operations (Muyldermans et al. 2002); defining electrical power zones (Bergey et al. 2003b), and many other domains. These are frequent real-world decision-making questions in territory partition problems.

When carefully observing and analyzing the large scope of possible applications dealing with districting problems we cannot be indifferent to the crucial importance of such a kind of decision-making situations to our societies. Most of the above quoted applications deal with real-life situations that contribute to the development of our societies in a large variety of fields. By their very nature, these problems involve multiple criteria which are frequently incommensurable and conflicting.

On the definition of the problem The partition of a *territory* into different "homogeneous" zones evaluated on the basis of multiple criteria, consists of grouping *elementary units* of a territory in order to form a set of *districts* or *zones*. A territory is thus composed of zones, each zone resulting from a grouping process of elementary units. This problem can be modelled by using graph theory and 0–1 mathematical programming concepts and techniques (Mehrotra 1992). Each elementary unit is associated to a vertex of the graph, while a pair of contiguous elementary units defines an edge of the graph. Some numerical values are also associated to the edges and/or vertices. The zones should fulfill certain, more or less, technical, ethical, ecological, social, and other constraints. Different *maps* of a territory form a set of different *solutions* where each one is evaluated according to a set of consistent *criteria*. Thus, the search for an *optimal* solution, in general, makes no sense and the "best solution" is, frequently, a *compromise* in which the improvement on a given criterion leads to a degradation of the evaluations on at least one of the remaining criteria. This leads to the concept of *non-dominated solution*.

A brief historical view on territory partition problems Historically, among the different types of territory partition problems, it was the so-called electoral districting problem that led the use of scientific methodologies that sought the construction of political districts as close as possible in terms of voting power in order to generate trust and the impartiality in the partition process. The latter is particularly crucial since it is possible to design a partition favoring a certain political, social, or ethnic group. It is well-known in the history of the USA that the Governor of the State of Massachusetts, Albright Gerry (1744–1814), in an attempt to guarantee his re-election, manipulated the division of his state to concentrate his voters so as to elect a representative and to scatter a large number of his opponents in a small number of districts (Mehrotra et al. 1998). Therefore, one of the districts had the shape of a salamander, as stated in the Boston Gazette on March 26, 1812, which led to the expression "gerrymander", the result of the contraction between the words "Gerry" and "salamander".

One of the first works on this very topic appeared in (Vickrey 1961). It described a heuristic process in which a zone is built, at each iteration. A few years later, in 1965, the first mathematical programming model was proposed (Hess et al. 1965), formulating the problem as a location/allocation model.

Apart from the political districting problem, the model that most deserved the attention of researchers is the problem of designing zones for salespersons. Since 1971, different works have been published in which the main objective consists of balancing the workload among the different zones (Easingwood 1973; Hess and Samuels 1971; Shanker et al. 1975; Zoltners and Sinha 1983).

Solving districting problems When solving districting problems, there are three main aspects to be considered, with respect to the modeling features taken into account:

1. The different "techniques" for districting problems can be divided in two big families: one based on the concept of *division* and the second one on the notion of *agglomeration* (Cortona et al. 1999). In *division* based techniques, the territory is considered as a whole and the districting procedure works out by dividing it into pieces (Chance 1965; Forrest 1964). In *agglomerative* based techniques, the territory is composed of a set of elementary units and a district is a subset of units forming a connected piece of land (Deckro 1979; Garfinkel and Nemhauser 1970; Hess et al. 1965; Hojati 1996; Nygreen 1988; Vickrey 1961).
2. We can also classify the districting problems in terms of the number of "criteria". There are models involving only *one criterion*, often, voting potential equality, or workload equality (Garfinkel and Nemhauser 1970; Hess et al. 1965; Hojati 1996). When dealing with *conflicting criteria*, some authors adopted models with more than one criterion (Bergey et al. 2003a; Bourjolly et al. 1981; Bozkaya et al. 2003; Deckro 1979). There are several strategies where the criteria are considered according to a fixed hierarchy reflecting the decision-maker preferences. In other cases, the purpose is to build a mixed objective function combining all the objectives.
3. The different approaches in this field can also be classified in *exact* and *non-exact* algorithms. Exact techniques for single criterion problems are provided in (Mehrotra 1992) where a decomposition and column generation scheme is applied to solve them. A new avenue for dealing with this problem is the use of *meta-heuristics* to approximate the Pareto front. Indeed, this approach had been already proposed by several researchers (Bergey et al. 2003b; Muyldermans et al. 2002).

Most of the papers published in the field deal with a specific problem of districting without establishing an overall framework for the different partitioning territory problems. The applied heuristics are also specific for each problem. Despite the very nature of real-world decision-making situations which are mainly multidimensional (since "always" more than one criterion to be optimized should be considered), there are still several studies applying single criterion models. These amalgamate all the dimensions in the same scale and often provide no meaningful conclusions. In general, we can say that exact techniques only have some interest from a theoretical point of view. They can only be applied to small-size instances, which are frequently unrealistic indeed.

Main features of the proposed approach Our approach is a local search evolutionary based algorithm. The representation we adopted for the individuals/solutions is as close as possible to the solutions themselves. Each solution is a set of subsets where each subset represents a zone. This makes it possible to guide the operators (*crossover* and *mutation*) according to the specific criteria used, regardless of the kind of problem we are dealing with. The main features of the algorithm are the following:

1. It deals with multiple criteria.
2. It allows to consider certain specific constraints according to each type of problem.
3. It allows to solve large-size instances in a reasonable CPU time and for different kinds of instances.
4. It generates high quality solutions.

The Paris region case study The public transportation tickets pricing in the Paris region, ("Carte Orange"), is defined on the basis of a partition of the Paris region into concentric zones. The only criterion considered is the distance from the center. The price increases according to this distance, but it is the same in each concentric zone. A study undertaken by the "Syndicat des Transports Parisiens" (STP) showed that this district map does not correspond any longer to the needs of the users. With the current tendency of moving services from the center of the big cities to the suburbs, many users of public transportation services only use them inside suburban zones without having the need to move downtown. Such observations led researchers at STP and LAMSADE[1] to study a reform of the current pricing ticket system (Mousseau et al. 2001).

The outline of the paper This paper is organized as follows. Section 2 describes how the problem can be modelled by using graph theory notions and presents the main concepts, definitions, and notation. The local search evolutionary algorithm is described in Sect. 3. This is followed by an example for checking its effectiveness in Sect. 4. Section 5 is devoted to the study of a real-world problem. Finally, in Sect. 6, conclusions are provided together with some suggestions for future research work.

2 Problem statement

This section is entirely devoted to modelling issues, mainly to the formulation of the problem based on graph theory concepts and some elementary multiple criteria background.

[1] Laboratoire d'Analyse et Modélisation de Systèmes pour à l'Aide la Décision (http://www.lamsade.dauphine.fr).

2.1 Modelling issues

Given a territory composed of indivisible elementary territorial units, we define a *contiguity graph* as an undirected and connected planar graph $\mathcal{G} = (V, E)$, where $V = \{1, 2, \ldots, i, \ldots, n\}$ denotes the set of *vertices* representing territorial units and $E = \{e_1, e_2, \ldots, e_k, \ldots, e_m\} \subset V \times V$ denotes the set of *edges*, where $e_k = \{i, j\}$, represents two adjacent elementary units i and j.

Once the contiguity graph $\mathcal{G} = (V, E)$ is defined, a *district map* can be considered as a partition of V into connected subsets of vertices. Furthermore, all the values associated to the territorial units (population, surface, etc.) can be associated with the corresponding vertices. Similarly, the values associated to a pair of contiguous territorial units (length of the common frontier, for example) can be associated to the corresponding edge.

In this context, a solution Y will be represented as a partition of V, composed of K subsets, as follows,

$$Y = \{y_1, y_2, \ldots, y_K\},$$

where

$$y_u \cap y_v = \emptyset, \; u \neq v \quad \text{and} \quad \bigcup_{1 \leq u \leq K} y_u = V.$$

It is not easy to distinguish between criteria and constraints. Some criteria are considered as constraints in different models. Without going into the details of such a distinction, let us consider that all the partition problems we are concerned with can be modelled through an undirected, connected, and planar graph, where the vertices represent the elementary units and the edges the adjacency relation between elementary units. Our problem is a particular graph partition problem where each partition is obviously composed of a connected subgraph. The following *constraints* should be considered:

1. *Integrity*. A vertex cannot belong to several subgraphs at the same time. It belongs to one and only one subgraph of the partition.
2. *Contiguity*. It is possible to define a path from two pairs of vertices in a certain zone without passing through a different zone.
3. *Absence of holes*. It is not possible to obtain embedded connected subgraphs.

The mathematical formulation of these constraints in the traditional mathematical programming approaches has very diverse degrees of difficulty. As for each integrity constraint, a binary variable is defined for each pair composed of a vertex and a zone. Hence, it becomes easy to impose that all the vertices belong to one and only one zone. Enforcing contiguity requires an exponential number of constraints in the model with respect to the number of vertices in the graph (Mehrotra et al. 1998). In an attempt to model the constraints, we found a number of mathematical expressions with a *rate of growth* of $Kn!$. Concerning the absence of holes, as far as we know, there is no model for such a purpose. As for the contiguity, its nature is exponential.
The most common criteria are the following:

1. Criteria aiming at homogenizing the surface, population, etc.
2. Criteria aiming at defining districts with a compact shape.
3. Criteria aiming at defining districts "compatible" with or similar to an existing district map.

In (Tavares-Pereira et al. 2004) there is a mathematical representation of each one of these type of criteria. Section 4.1 gives an example of an implementation of homogeneity criteria.

2.2 Multiple criteria background: Concepts, definitions and notation

A multiple criteria mathematical program may be written as follows:

$$\begin{aligned}
&\max\{f_1(x) = z_1\}\\
&\max\{f_2(x) = z_2\}\\
&\vdots\\
&\max\{f_l(x) = z_l\}\\
&\text{s.t.:}\quad x \in X
\end{aligned}$$

or

$$\text{“max”}\quad Z = \{G(x) = z \in \mathbb{R}^l \mid x \in X\}$$

where l, is the number of criteria; x, is the vector with M decision variables; f_i, is a real function defined on $\mathbb{R}^M$ representing the ith criterion; z_i, is the criterion value (criterion function value) of the ith criterion; X, is the *feasible region* in the *decision space*; “max”, means that the purpose is to maximize all the criteria simultaneously; G, is a vectorial function composed of l, f_i, functions, $i = 1, 2, \ldots, l$; z, is the criterion function vector; and Z, is the *feasible region* in the *criterion space*.

Let $z, z' \in \mathbb{R}^l$ be two criteria vectors. Then, z *dominates* z' if and only if $z \geq z'$ and $z \neq z'$, i.e., $z_i \geq z'_i$ for all i and $z_i > z'_i$ for at least one i. A vector $\overline{z} \in Z$ is *non-dominated* if and only if it does not exist another vector $z \in Z$ such that $z \geq \overline{z}$ and $z \neq \overline{z}$. Otherwise, $\overline{z}$ is a *dominated* criteria vector. The set, $Z^{\mathrm{nd}} \subseteq Z$, composed of all non-dominated criteria vectors is called *non-dominated set* or *Pareto front*. A solution $\overline{x} \in X$ is *efficient* or *Pareto-optimal* if the corresponding criteria vector, $z = G(\overline{x})$, is non-dominated. The set of all efficient solutions is called *efficient set* and is represented by X^{eff}.

In general, for large-size instances, it is not possible to enumerate all the vectors belonging to the set Z^{nd}, therefore we must approximate it using non-exact methods. Let $\hat{Z}^{\mathrm{nd}}$ denote the approximation of Z^{nd}. This brings us to the concept of *potential non-dominated solution*. A point $\hat{z}$ is a *potential non-dominated* with respect to a feasible subset $\hat{Z} \subseteq Z$ if and only if it does not exist another $z \in \hat{Z}$ such that $z \geq \hat{z}$ and $z \neq \hat{z}$. Otherwise, $\hat{z}$ is a *dominated* criteria vector. When the subset $\hat{Z}$ is omitted, it is perceived by the context. Very often the subset $\hat{Z}$ represents the solutions determined by an algorithm. The set $\hat{Z}^{\mathrm{nd}} \subseteq Z$ of all potential non-dominated criteria vectors with respect to $\hat{Z}$ is called *potential non-dominated set*.

3 A local search evolutionary algorithm (LSEA)

A local search evolutionary algorithm (LSEA) results from the combination of an evolutionary algorithm with local search. The expression *hybrid evolutionary algorithm* is also used in this context. There are no rules about the way these combinations are done. Normally, each researcher applies his/her skills when he/she chooses a particular technique.

Over the recent years, combinations of different heuristics have been proposed, thus opening the research path of hybrid algorithms (Preux and Talbi 1999). They have shown their ability to provide high quality local optima. In general, genetic operators are not adjusted to find better solutions close to another one (Ross 1997). Therefore, the combination between evolutionary algorithms and local search seems to be very promising, powerful, and profitable (Reeves 1997). For multiple criteria optimization problems, evolutionary algorithms seem also particularly adequate because they deal simultaneously with a set of potential solutions which makes it possible to approximate several solutions of the efficient set in a single run.

The main feature of the proposed approach is to find, in a short CPU time, a high quality set of potential non-dominated solutions that approximate the Pareto front. In each iteration, after attributing a *fitness* value to each individual, two different solutions are "randomly" selected in order to apply the *crossover* and the *mutation operators* and thus form a new generation. At the same time a list of potential non-dominated solutions is built. When a new solution is successfully inserted in this list, a local search is then applied to it.

3.1 General framework

This section is devoted to a comprehensive presentation of the proposed LSEA algorithm.

3.1.1 Assigning a fitness value to each individual

The selected strategy for evaluating the fitness value of each solution makes use of a well-known technique suggested by Srinivas and Deb (1995) called Non-dominated Sorting Genetic Algorithm (NSGA).

This technique is based on the Pareto ranking where the individuals from the entire population $\mathcal{P} = \{Y_1, Y_2, \ldots, Y_N\}$, composed of N individuals, are classified into several levels according to the concept of dominance. The potential non-dominated individuals belonging to the population are identified at first. These individuals form the first Pareto front of the potential non-dominated frontier. Afterwards, we assign a large dummy fitness value F to each one of them. In order to preserve the diversity of the population, these individuals, assigned to different levels, are then shared according to their dummy fitness values.

The sharing value of each individual, $\mathcal{F}_i$, is determined by dividing its original fitness value by the quantity,

$$\alpha_i = \sum_{j \in \hat{Z}^{\text{nd}}} \text{sh}(d_{ij}),$$

where

$$\text{sh}(d_{ij}) = \begin{cases} 1 - (\frac{d_{ij}}{\sigma_{\text{share}}})^2, & \text{if } d_{ij} < \sigma_{\text{share}}, \\ 0, & \text{otherwise.} \end{cases}$$

This quantity, α_i, is proportional to the number of individuals around it. Thus,

$$\mathcal{F}_i = \frac{F}{\alpha_i}.$$

The value d_{ij} is the Euclidian distance between two solutions Y_i and Y_j and σ_{share} is the maximum distance allowed between any two solutions to become members of a *niche* (i.e., a set of solutions having common features). For each individual, Y_i, the value α_i represents the density of individuals around it, and, $\text{sh}(d_{ij})$ is a sharing function value between two individuals in the same front that makes it possible to implement the sharing process. Afterwards, this front composed of potential non-dominated individuals is temporarily ignored to process the remaining members of the population in order to identify the second front. The new potential non-dominated individuals are then assigned to a new dummy fitness value which is kept smaller than the minimum shared dummy fitness value of the previous front. This method continues until the entire population is classified into several fronts, and no more fronts can be identified. Such a technique is presented in Algorithm 1, where $p \in]0, 1[$ is a parameter. The main idea is to penalize or decrease the fitness value of the solutions which are too close, in the criteria space, with respect to the remaining solutions in the current front.

Algorithm 1 Evaluation

Input: $\mathcal{P} = \{Y_1, Y_2, \ldots, Y_N\}$
Output: $\mathcal{F}_i$ for each Y_i

```
P_aux ← P
F ← a large dummy fitness value
while P_aux ≠ ∅ do
  Ẑ^nd ← all potential non-dominated individuals in P_aux
  for all Y_i ∈ Ẑ^nd do
    Calculate α_i
    F_i ← F / α_i
  end for
  P_aux ← P_aux − Ẑ^nd
  F ← min{F_i : Y_i ∈ Ẑ^nd} × p
end while
```

3.1.2 The crossover operator

The *crossover* operator identifies two parents (solutions) and makes an exchange of sections of their chromosomes. In order to generate an offspring, the crossover operator works as follows. Consider the two parents

$$Y = \{y_1, y_2, \ldots, y_K\}$$

and

$$Y' = \{y'_1, y'_2, \ldots, y'_{K'}\}$$

composed of K and K' zones, respectively. First we choose a subset $S_Y = \{\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_k\} \subseteq Y$ composed of $k \le K$ zones. These k zones will belong to the offspring. Then, we will define an equivalence relation $\backsim$ on the set $V \backslash S_Y$ as follows:

1. $v_1,\ v_2 \in V \backslash S_Y$.
2. $v_1 \backsim v_2 \Leftrightarrow \exists\, j \in \{1, \ldots, K'\}$ such that there is a path between v_1 and v_2 in $\overline{y}_j \backslash S_Y$.

In other words, we say that, v_1 is equivalent to v_2 if and only if, for some zone, $\overline{y}_j$, belonging to Y' there is a path between v_1 and v_2 in $\overline{y}_j$ without the vertices belonging to S_Y.

The offspring candidate solution is composed of the zones $\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_k$ along with the equivalence classes of the equivalence relation $\backsim$ defined on $V \backslash S_Y$. Generally, at this point, the number of zones has increased. Consequently a `Merging` procedure is applied to group together zones as described in Algorithm 2. As can seen in Fig. 1, one chooses a zone from

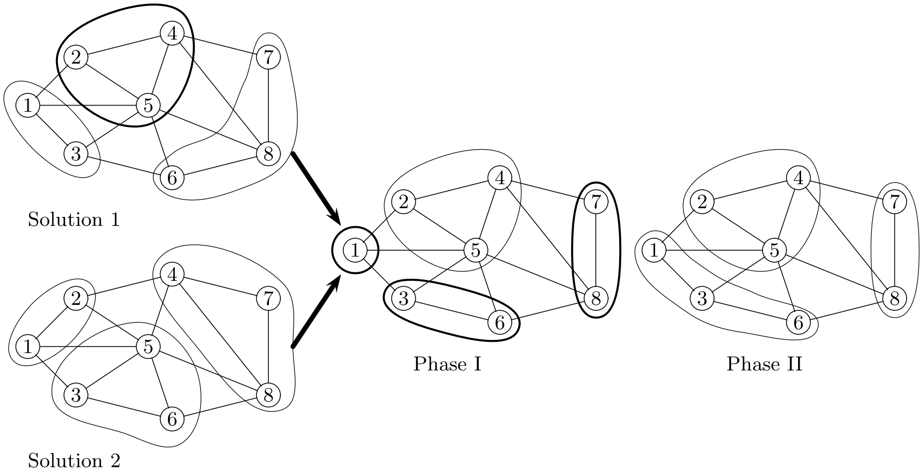


Fig. 1 Crossover operator

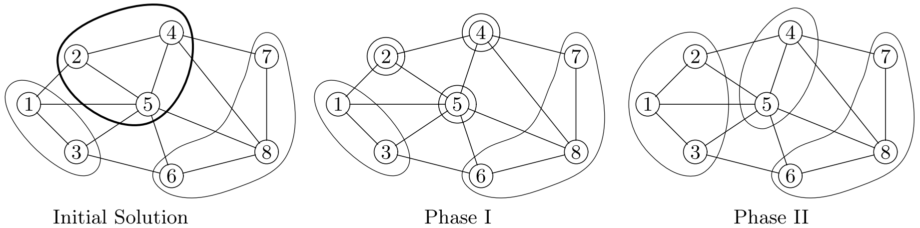


Fig. 2 Mutation operator

solution 1, composed of vertices 2, 4 and 5. In Phase I, the equivalence classes defined by the relation $\backsim$ in $\{1, 3, 6, 7, 8\}$ are $\{1\}$, $\{3, 6\}$ and $\{7, 8\}$. Afterwards, zones $\{1\}$ and $\{3, 6\}$ are merged in Phase II (see Fig. 1).

3.1.3 The mutation operator

The *mutation* operator takes one solution and randomly modifies the chromosome. Typically, there is a probability associated to this operator. In our case this probability is 1 since the results of the implemented operator are, frequently, feasible solutions and some times, better solutions.

The operator starts for breaking up a set of zones in such a way that each vertex constitutes a zone. After this operation the number of zones is greater than the fixed one. Then the procedure `Merging` is applied as described in Algorithm 2. Figure 2 shows the different phases of the mutation. The zone with vertices 2, 4 and 5 is chosen and broken up into three zones (Phase I). In Phase II the zone with vertex 4 is clustered to the zone containing vertex 5 and the zone with vertex 2 is clustered to the zone comprising vertices 1 and 3.

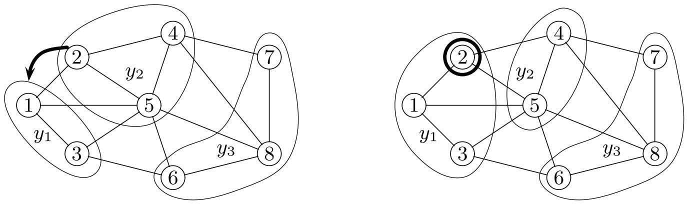


Fig. 3 Neighborhood structure

3.1.4 A local search procedure

In the local search procedure, the concept of *neighborhood structure* was used. From a current solution, all the neighbor solutions can be found by moving at most one vertex from a certain zone in the current solution to one of its neighbors zones, as shown in Fig. 3.

We check whether it is possible to move a vertex to a neighbor zone; if it is, the move is made, which will lead to a different solution that will be checked in order to find out if the solution is a potential non-dominated one.

As shown in Fig. 3, when moving vertex 2 from zone 2 to zone 1, two new zones are obtained. This move is feasible, but others may not. For example, vertex 8 cannot be moved from zone 3.

As we can see the neighborhood structure can be searched by using an $\mathcal{O}(n)$ polynomial time algorithm.

3.2 Implementation issues

The description of the operators raises some issues that should be addressed by those who implement such an algorithm. This section presents these issues and those related to the implementation of the algorithm. In some parts only the bi-criteria case is considered.

3.2.1 Representing a solution

A *solution*, Y, is represented as a *partition*, $Y = \{y_1, y_2, \ldots, y_K\}$, of the set of vertices, V. Each subset y_u is the set of vertices of a *connected subgraph* of $\mathcal{G}$. Thus, a partition is implemented by a list of lists where each list represents an element of Y. The following partition,

$$Y = \{y_1 = \{1, 3\}, y_2 = \{2, 5, 4\}, y_3 = \{2, 7, 8\}\}$$

is represented in Fig. 4. Each element, y_u, of Y is called *zone*.

3.2.2 Defining the initial population by using a merging procedure

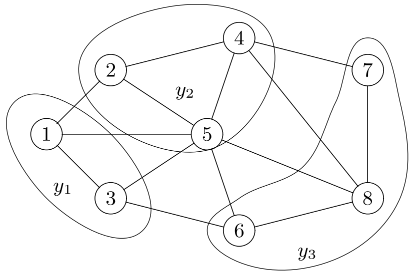
When generating new solutions, we have to deal with two main concerns:

1. To reach feasible solutions.
2. To obtain the best possible solution according to the criteria considered.

The initial population, $\mathcal{P}_0$, is composed of a set of individuals or solutions,

$$\mathcal{P}_0 = \{Y_1, Y_2, \ldots, Y_N\}.$$

Fig. 4 Graph



To generate an individual, the algorithm starts by the trivial solution where each subset of the partition is composed of a unique vertex. For each individual the position of each zone in the list is generated randomly. Then, the procedure `Merging` is applied, that consists of grouping two neighboring zones until the number of zones previously fixed is reached (see Algorithm 2). In this procedure, a heuristic rule can be implemented for choosing two neighboring zones, y_i and y_j that will be merged. If we have constraints on the number of vertices in each zone, the one comprising the lowest number of vertices, y_i, is merged with one of its neighbors, y_j. The strategy of choosing the neighboring zone generally depends on the criteria and/or the constraints of each problem. Normally, we use a greedy heuristic that depends on a weighted-sum of the two criteria. In Sect. 4.2 we exhibit an implementation of this heuristic. This procedure is also used elsewhere in the *crossover* and *mutation* operators. In order to promote diversity in the set of initial solutions, the order of the incident edges to each vertex is modified at random.

Algorithm 2 `Merging`

Input: $Y = \{y_1, y_2, \ldots, y_K\}$
Output: $Y' = \{y'_1, y'_2, \ldots, y'_{K'}\}$
  while ($K \geq$ the fixed number of zones) **do**
    Let y_i, y_j be two neighbors zones, chosen according to a heuristic rule
        defined for each type of criteria;
    Merge y_j and y_i;
    $K \leftarrow K - 1$;
  end while
  $Y' \leftarrow Y$;

3.2.3 Selecting individuals

To select each pair of individuals we apply the *roulette wheel* method. It guarantees that the fitness value of any individual is just kept smaller than the minimum fitness value of the elements belonging to the previous potential Pareto front. Therefore, the elements in the first fronts have higher probability of being selected in the next generation than the remaining solutions. Although we do not implement any elitism mechanism, there is a great probability of the best solution remains in the population.

3.2.4 Choosing the specific features of the operators

To implement the genetic operators, defined according to Sects. 3.1.2 and 3.1.3, some choices have to be made on how to cope with the above mentioned issues. Let Y and Y' be two partitions as in Sect. 3.1.2.

- As for the crossover operator, the k zones of partition Y in order to built the offspring solution can be randomly chosen or selected according to the criteria of the problem. In our case, it is possible to select the "k-best" zones according to a weighted-sum of the criteria. The number k is randomly selected within the range $[\frac{1}{4}K, K - \frac{1}{4}K]$.
- As for the mutation operator the zones that are broken up are chosen randomly, according to a uniform distribution or they can also be selected according to the criteria of the problem.

Each new potential non-dominated solution, resulting either from the *crossover* and *mutation* operators or the application of local search, is placed in a queue data structure. Local search will be applied to it, later on.

3.3 Outline of the algorithm

The pseudo-code for the LSEA procedure is outlined in Algorithm 3.

Algorithm 3 Outline of the algorithm

```
t ← 0
Generate P_0
Compute the Fitness value for each individual in P_0
Initialize the set of potential non-dominated solutions, Ẑ^nd, from P_0
while (stop condition is false) do
    Children ← Crossover(P_t)
    Mutant ← Mutation(Children)
    Update Ẑ^nd with Children ∪ Mutant
    Local_Search(Ẑ^nd)
    Compute the Fitness value for each individual in P_t ∪ Children ∪ Mutant
    P_{t+1} ← the best of P_t ∪ Children ∪ Mutant
    t ← t + 1
end while
```

3.4 Particular implementation issues

When designing and implementing of the proposed algorithm, some issues were treated in a very particular way. We emphasize tree main points:

1. *Assigning a fitness value*. One important rule in assigning a fitness value is to guarantee that the fitness value of any individual is just kept smaller than the minimum fitness value of the previous front. When we have to choose the next dummy fitness value, it is not sufficient to keep the minimum fitness value of the previous front. A pathological

case can occur: after the computation of the fitness, the whole population have the same final fitness value. To avoid this situation, the next dummy fitness value will be fixed at 90% of the minimum fitness value of the previous front. This percentage was chosen after an empirical study. With this value, we get a very good distribution of the fitness values of the whole population.

2. *Selecting individuals*. The *crossover* operator is applied to a previously selected pair of solutions. The first is chosen in the population while the second one is selected from the set $\hat{Z}^{\mathrm{nd}}$ that contains the potentially efficient solutions. The main reason is that the new potentially efficient solutions resulting from local search are kept in set $\hat{Z}^{\mathrm{nd}}$. They are not directly included into the population. Thus, there is an opportunity to apply the genetic operator to themselves.
3. *Applying local search*. Each new potentially efficient solution resulting from genetic operators and local search is stored in a queue structure. Therefore, when the process of applying the genetic operators stops, until the queue structure is not empty, a local search is applied to the first one, and so on, i.e., until we reach the last of the remaining solutions in that queue. This process stops before emptying the queue structure when an upper bound of iterations regarding the local search procedure is achieved.

4 Evaluating the quality of the solutions

Our main concern in this section is to test how the algorithm behaves in practice. It is frequent to use instances with known results to evaluate the performances of the meta-heuristics (see Van Veldhuizen 1999). However, in our case there are no available instances. Consequently, we have to build some examples.

In this section, we will propose a heuristic for each kind of criterion to be tested (it should be noted that each criterion requires a specific heuristic). The main feature which makes it possible to distinguish each heuristic is related to the rule used to choose two neighboring zones that will be merged later.

The algorithm was first tested with a small-size instance (22 vertices and 49 edges) for which all the exact non-dominated solutions were calculated by using the ϵ-constraint method (Steuer 1986) along with the Mehrotra model (Mehrotra 1992). This instance will not be presented here; the reader can find it in (Tavares-Pereira et al. 2004). All the solutions were generated in the initial population. In order to test the quality of the algorithm for larger instances when knowing the set of exact solutions, the original graph was multiplied by 4 (with a four-time copy-paste operations). When applying the LSEA, the results were quite satisfactory. On average, we got about 70% of the exact solutions and for the remaining solutions, a good approximation was obtained too.

To develop an evaluation of the performance of the algorithm, we decided to use a large-size instance with 1300 vertices and 3719 edges, which corresponds indeed to the real-world graph concerning the Paris region transportation problem (see Fig. 5).

4.1 Criteria and constraints

To have an idea of how the algorithm behaves with this data, the following strategy was defined. It allows to know two exact solutions of the efficient set: the optimal solution for each criterion.

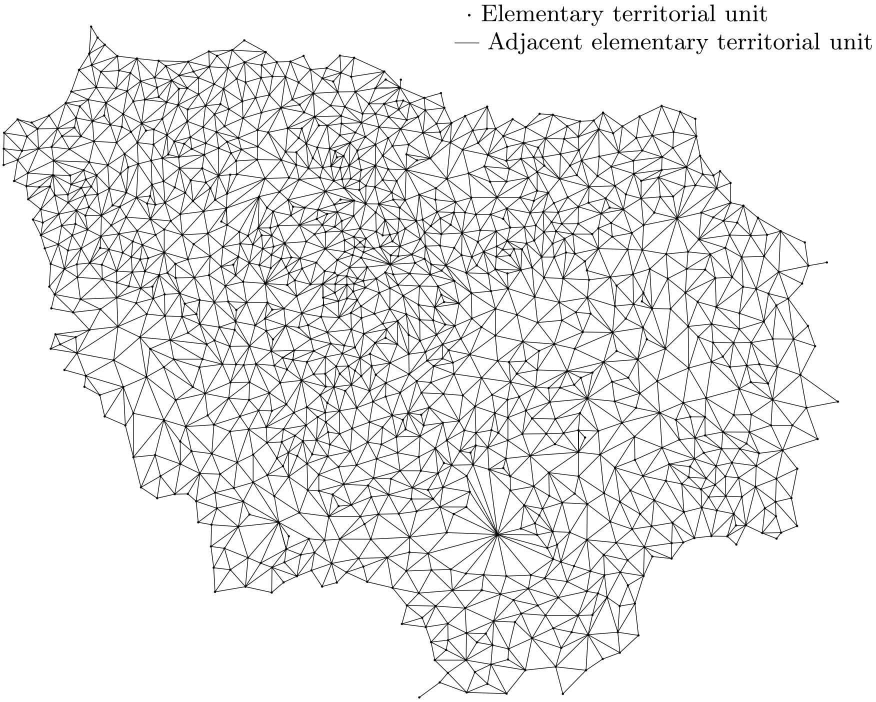


Fig. 5 Paris region contiguity graph

An instance, with two values c_i and c'_i for each vertex $i \in V$, was built as follows:

1. From a large set of solutions, previously generated, select the two most distant ones $Y^* = \{y_1^*, y_2^*, \ldots, y_K^*\}$ and $Y'^* = \{y_1'^*, y_2'^*, \ldots, y_K'^*\}$ according to a distance in the decision space defined a priori.
2. Choose the solution $Y^*(Y'^*)$ for set values to $c_i(c'_i)$. For each zone assign the same value to all of its vertices, although different values are assigned to each zone.
3. Consider two homogeneity criteria,

$$f_1(Y) = \sum_{u=1}^{K} (\max_{i \in y_u}\{c_i\} - \min_{i \in y_u}\{c_i\})$$

and

$$f_2(Y) = \sum_{u=1}^{K} (\max_{i \in y_u}\{c'_i\} - \min_{i \in y_u}\{c'_i\}),$$

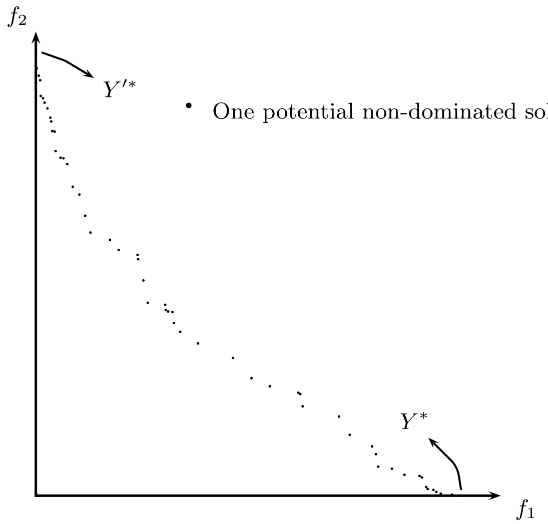
both to be minimized.

Therefore, we have the guarantee that $Y^*(Y'^*)$ is the optimal solution for the first (second) criterion with value equal to 0, i.e., $f_1(Y^*) = f_2(Y'^*) = 0$.

The constraints are the following:

1. Number of zones: 30.
2. Number of vertices *per* zone: between 20 and 70.

Fig. 6 Initial potential non-dominated solutions



4.2 Heuristic

For this type of criteria, the heuristic implemented can be summarized as follows:

1. Choose the zone with the lowest number of vertices, $y_{\min}$.
2. For each neighbor, y_k, determine the difference δ_1 (δ_2) between the maximum and the minimum weight c_i (c_i') after a possible merging with $y_{\min}$. A weighted-sum $\lambda_1\delta_1 + \lambda_2\delta_2$ is used to rank all the neighbors according to an increasing order.
3. Among its neighbors, choose the first zone y_{Fn} for which the constraint associated with the number of vertices per zone is not violated.
4. Merge zones $y_{\min}$ and y_{Fn}.

4.3 Results

The parameters for the LSEA were fixed as follows:

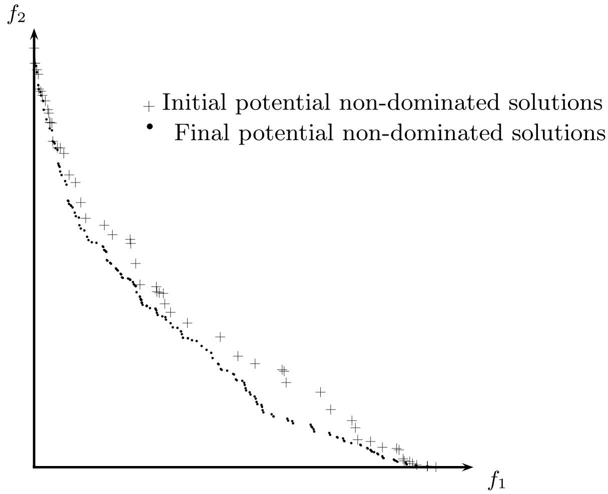
1. Population size $pop\_size = 300$.
2. Crossover probability $cross\_prob = 0.3$.
3. Maximum generations $max\_gen = 50$.

When the weighting factors (λ_1, λ_2) are set to $(1, 0)$ and $(0, 1)$, the heuristic applied for generating a solution determines, almost always, the two efficient solutions Y^* and Y'^*. Figure 6 shows the 59 potential non-dominated solutions extracted from the initial population. As it can be seen Y^* and Y'^* were found.

Figure 7 shows the potential non-dominated solutions identified after 50 generations. 190 potentially efficient solutions were found. However, a large proportion of them have the same value in the criteria space. It shows that the results obtained concerning the genetic operators produce good solutions.

Nevertheless, it will not be possible to determine the set Z^{nd} for this instance that would make it possible to evaluate the performance of our algorithm in a "rigorous" way. Our intuition led us to assert that we obtained good results and it is based on two main points:

Fig. 7 Final potential non-dominated solutions



1. The heuristic that chooses the two neighboring zones to be merged appeared to have a very good behavior: in the generation process of new solutions, and when requested, this heuristic was able to determine, with high frequency, solutions Y^* and Y'^*;
2. Since this type of criteria has a "smooth" variation (the weights belonging thus to a small range), it is likely that the curve of the potential non-dominated solutions will present a certain "smoothness" as shown in the Fig. 7.

5 Case study: the public transportation network pricing system

The observation of social and economic trends (falling population in the inner city of Paris, increased commuter flows between the center and the suburbs, and demand for local ticket prices) led STP, (the Paris transportation authority) to re-examine the current ticket pricing system.

The pricing system is grounded on the definition of geographical zones. The current district map is defined by concentric zones, which does not correspond to the use of the transportation network. Therefore, one of the first goals of the reform is to modify the map on which the ticket prices are based. Such a problem involves approximately 1300 elementary units (the municipalities in the Paris region) and each zone of the new map is supposed to represent autonomous units as far as public transportation is concerned. This autonomy of zones is modelled by using several criteria (see Mousseau et al. 2001). Hence this problem involves multiple criteria and is of a combinatorial structure. The proposed algorithm was applied to this real-world problem.

5.1 Data set

The descriptors were defined to highlight the acceptability of a zone in a district map and were validated by the "stakeholders". They are grouped according to the type of concern they refer to. Thus, for each territorial unit we have some real data that enables us to build the criteria from the following descriptors:

1. Location of the zone with respect to the network
   - Number of stations in rail network (rs_i).
   - Number of buses on road service.
   - Density of the internal offer.
   - Density of the external offer on the rail network.
   - Density of the bus external offer.
   - Location of the stations in rail network.
2. Mobility structure within a zone
   - Access to the rail network.
   - Commuting.
   - Presence of public services.
3. Zone corresponding to administrative structures
   - Conformity to the current "département" boundaries.
   - Conformity to the current urban community boundaries.
4. Centers of attraction in the zone
   - Location of shopping centers and malls.
   - Location of healthcare centers.
5. Social nature
   - Population (pop_i);
   - Active population (act_pop_i).
   - Homes without cars ($h0c_i$).
   - Homes with one car ($h1c_i$).
   - Homes with two or more cars ($h2c_i$).
6. Geographical nature
   - Surface ($surf_i$).

   From the descriptors $h0c_i$, $h1c_i$ and $h2c_i$ on each unit i, two new descriptors were built:

1. The proportion of homes with two or more cars,

$$ph2c_i = \frac{h2c_i}{h0c_i + h1c_i + h2c_i}.$$

2. The proportion of homes with one or more cars,

$$ph1c_i = \frac{h2c_i + h1c_i}{h0c_i + h1c_i + h2c_i}.$$

5.2 Criteria and constraints

The need to create criteria for comparing zones is mainly due to the fact that the choice of several district maps have to be made by stakeholders. Thy must, therefore, be able to compare their choices of district maps using criteria which have been accepted by *consensus* as a basis for comparison. The criteria were chosen by all and were deemed suitable for this task. Some of them are presented below.

A set of criteria was built with the available data. Each criterion is defined through two stages:

Table 1 The criteria

Data	Evaluation of y_u	Evaluation of Y	Max\Min
$surf_i$	$\mathcal{S}(y_u)=\sum_{i\in y_u} suf_i$	$f_1(Y)=\max_{y_u\in Y}\mathcal{S}(y_u) - \min_{y_u\in Y}\mathcal{S}(y_u)$	Min
pop_i	$P(y_u)=\sum_{i\in y_u} pop_i$	$f_2(Y)=\max_{y_u\in Y}P(y_u) - \min_{y_u\in Y}P(y_u)$	Min
act_pop_i	$AP(y_u)=\sum_{i\in y_u} act\_pop_i$	$f_3(Y)=\max_{y_u\in Y}AP(y_u) - \min_{y_u\in Y}AP(y_u)$	Min
rs_i	$RS(y_u)=\sum_{i\in y_u} rs_i$	$f_4(Y)=\min_{y_u\in Y}RS(y_u)$	Max
$ph2c_i$	$H2(y_u)=\max_{i\in y_u} ph2c_i - \min_{i\in y_u} ph2c_i$	$f_5(Y)=\max_{y_u\in Y}H2(y_u)$	Min
$ph1c_i$	$H1(y_u)=\max_{i\in y_u} ph1c_i - \min_{i\in y_u} ph1c_i$	$f_6(Y)=\max_{y_u\in Y}H1(y_u)$	Min

1. A value for each zone y_u is determined.
2. The set of values is aggregated in a unique number representing the value of the criterion for a solution $Y=\{y_1, y_2, \dots, y_K\}$.

Table 1 shows the criteria considered. All of them are (inter or intra-zone) homogeneity criteria. The inter-zone homogeneity criteria (aiming at partitions where the attributes are uniformly distributed over all zones) are the following:

– f_1, surface homogenization.
– f_2, population homogenization.
– f_3, active population homogenization.
– f_4, rail station homogenization.

The intra-zone homogeneity criteria (aiming at partitions where each zone is as uniform as possible according to the descriptor attributes) are the following:

– f_5, homogenization of the proportion of homes with 2 or more cars.
– f_6, homogenization of the proportion of homes with 1 or more cars.

To build up bi-criteria problems we have coupled only the relevant pairs. These pairs are the following: (f_1, f_2), (f_1, f_3), (f_1, f_4), (f_1, f_5) and (f_1, f_6). Although we are able to apply the algorithm with all the criteria, only pairs of criteria were chosen. In this way, it is easy to visualize the set of the potential non-dominated solutions and the "stakeholders" found it a very interesting starting point. We also tested the algorithm with the pair (f_1, f_3) when the remaining criteria were treated as constraints. For such a propose we found acceptable bounds for each of these criteria and we penalize the solutions that did not respect these bounds.

The constraints are related to the compactness, the number of zones and the number of units *per* zone. The degree of compactness $\mathcal{C}(Y)$ of a district map, $Y=\{y_1, y_2, \dots, y_K\}$, is equal to the degree of the worse one of its zones $\mathrm{Comp}(y_u)$ as regards the compactness, i.e.

$$\mathcal{C}(Y)=\min_{y_u\in Y}\mathrm{Comp}(y_u).$$

Table 2 Number of potential efficient solutions

Pair	$K=20$	$K=25$	$K=30$	$K\in[20,30]$
(f_1,f_2)	199	329	167	250
(f_1,f_3)	82	307	160	158
(f_1,f_4)	37	16	51	57
(f_1,f_5)	72	35	334	97
(f_1,f_6)	347	567	125	988

Table 3 Number of potential efficient solutions with constraints

Pair	$K=20$	$K=25$	$K=30$	$K\in[20,30]$
(f_1,f_3)	550	268	211	97

The degree of compactness of a zone results of the quotient between its surface, $\mathcal{S}(y_u)$ and the surface of the smallest circumference that will enclose it, $\mathcal{S}(\mathring{y}_u)$, i.e.

$$\mathrm{Comp}(y_u)=\frac{\mathcal{S}(y_u)}{\mathcal{S}(\mathring{y}_u)}.$$

We decided to choose an acceptable limit for compactness empirically and to make tests with the following characteristics:

1. A fixed number of zones: 20, 25 and 30.
2. A variable number of zones: between 20 and 30.
3. A number of units *per* zone: between 20 and 110.

5.3 Results

In this case, the parameters for the LSEA were the following:

1. Population size, $pop\_size=700$.
2. Crossover probability, $cross\_prob=0.5$.
3. Maximum generations, $max\_gen=50$.

Table 2 presents the number of potentially efficient solutions found for each pair of criteria. In many cases the number of the corresponding solutions in the criteria space is very small. For example, pair (f_1,f_6) when $K\in[20,30]$ has 988 potential efficient solutions, but in the criteria space these solutions correspond only to 7 points.

Figures 8, 9, 10 and 11 represent the graphics with the initial and final non-dominated solutions for the pair (f_1,f_3) (the homogenization of surface and active population) when $K=20$, $K=25$, $K=30$ and $K\in[20,30]$. For all of them, the progress made by the LSEA is clear. In some cases, the value of f_1 and f_3 improved more than 50%. Figure 13 represents the best district map concerning the surface homogenization, criterion f_1, when $K=25$.

Table 3 presents the number of potentially efficient solutions found for the pair (f_1,f_3) that fulfill constraints concerning the remaining criteria. Figure 12 shows the initial and final potential non-dominated solutions for $K=20$.

Figure 14 represents a compact district map. This partition was reached when we tested the compactness criterion described in the previous section.

The results reported in this section reveal more or less how the decision-making process have evolved since we started the analysis of the case study, in particular, the elements that

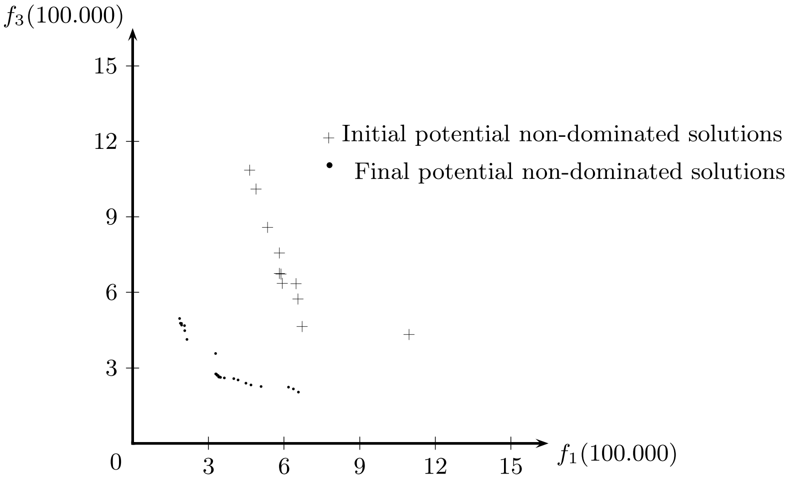


Fig. 8 Potential non-dominated solutions: $K = 20$

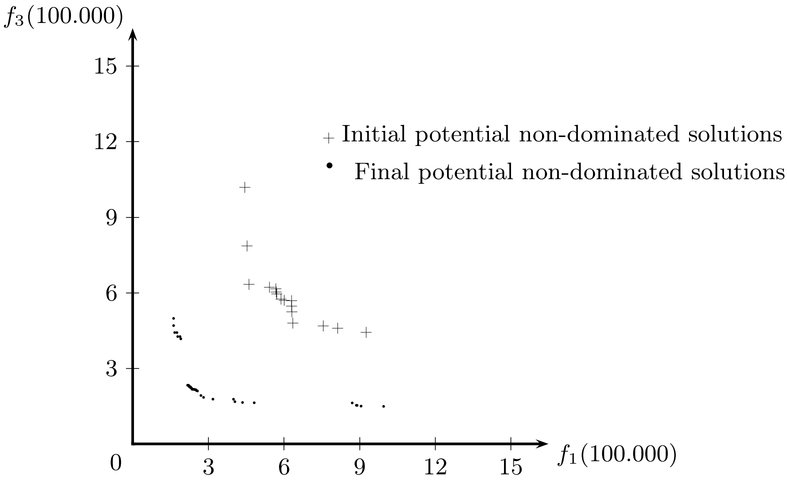


Fig. 9 Potential non-dominated solutions: $K = 25$

concern the "resolution" of the "problem". Several aspects should be taken into account for a better understanding of the decision-making process:

1. The model comprises 6 criteria built according to the descriptors presented in Sect. 5.1.
2. The algorithm is able to deal with all the criteria simultaneously.
3. But, from a practical point of view, "stakeholders" at STP were unable to apprehend all the different aspects of the problems and proposed to start the analysis by an elementary level, making it easier to understand the real-world decision-making process.
4. According to their suggestion, we decided to analyze the problem taking into account only pairs of criteria and to observe what happens when looking at the criteria space.

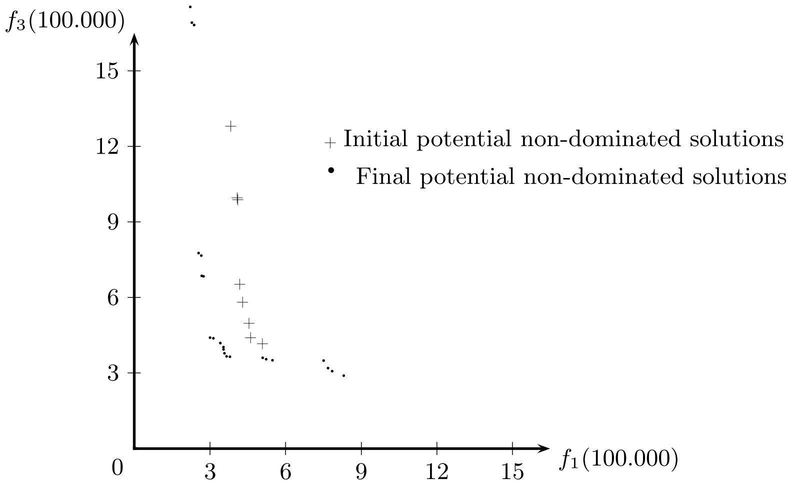


Fig. 10 Potential non-dominated solutions: $K = 30$

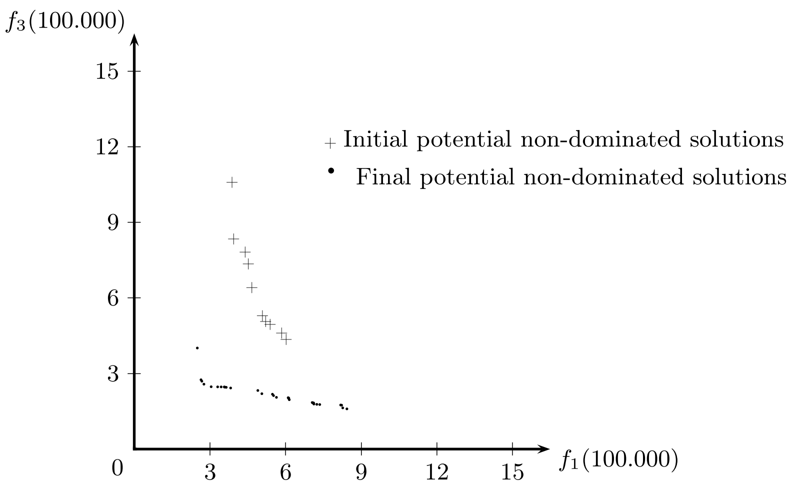


Fig. 11 Potential non-dominated solutions: $K \in [20, 30]$

5. The generation of the potential non-dominated frontier was well accepted by the "stakeholders", but they wanted still to fix their study in a particular region of such a frontier.
6. After locating that region some solutions were chosen up and the corresponding maps were built.

The computational experiments performed and the obtained district maps made it possible to elaborate a list of contrasted district maps (that have various criteria values, number of zones, etc.) that can be proposed to the decision-maker for a final decision. Such a decision is however highly political and strategic and the decision-making process is still ongoing.

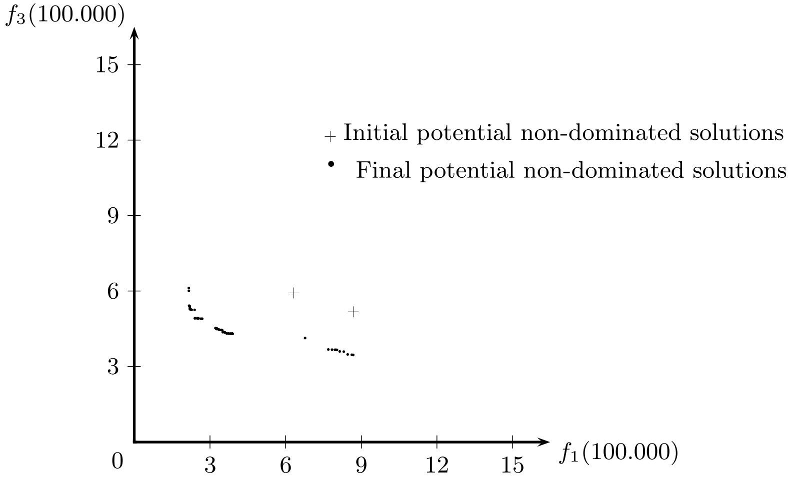


Fig. 12 Potential non-dominated solutions constrained: $K = 20$

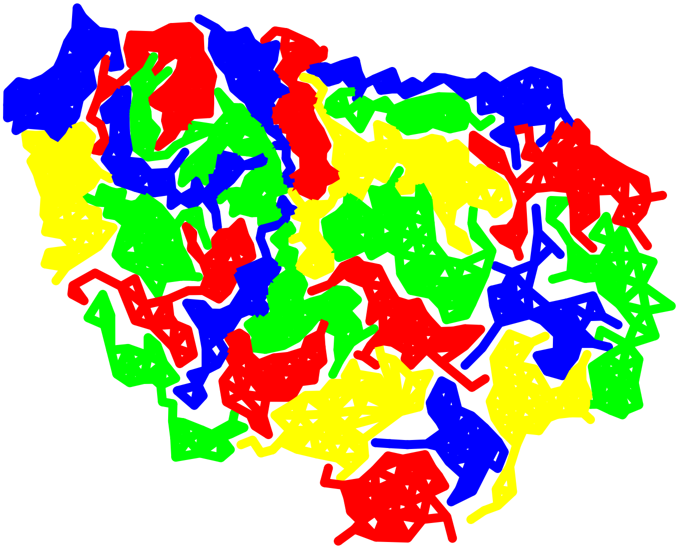


Fig. 13 $(f_1, f_3) = (162382, 498071)$: The best surface homogenization when $K = 25$

6 Conclusions and future research

In this paper, we presented a new local search evolutionary algorithm. The algorithm is a hybridization of recombination operators with local search that allows the use of local heuristics taking into account the nature of each criterion. We used a solution coding that allowed to guide the genetic operators towards the criteria. The combinations with the local search allow to check all efficient solutions near the new obtained solutions.

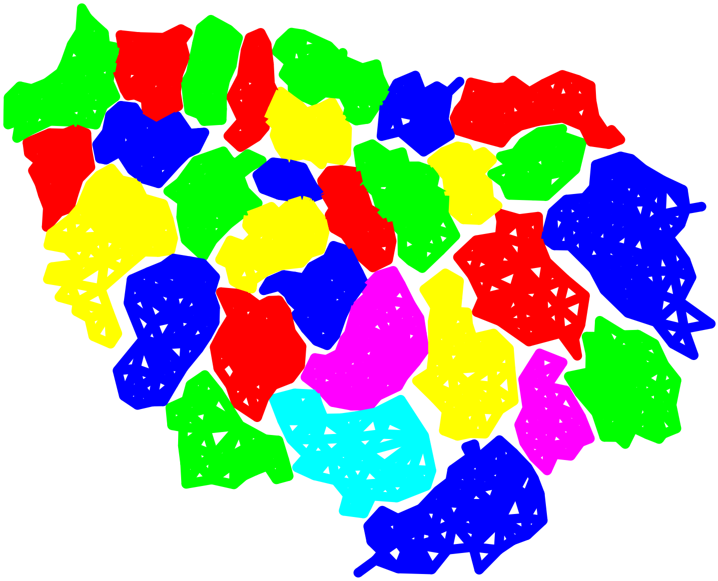


Fig. 14 The best compact solution found

Computational experiments and results showed that the new algorithm generates high quality potentially efficient solutions from all regions of the non-dominated set. The experiments performed exhibit the excellent combination between evolutionary algorithms and local search. The possibility of guiding the genetic operators also showed the high convergence increment that made it possible to find good solutions very quickly. On the whole, the research work not only deals with districting problems but also exhibits the potential power of LSEAs for combinatorial problems with multiple criteria.

On the one hand, we believe that our work has a great potential of development. Its great flexibility allows to adapt it to many kinds of problems of a different nature. On the other hand, the LSEA can be improved in some aspects: the search in the neighborhood structure can be improved because there is a high level of intersections between neighborhoods. In the future, a more or less "automatically" interactive procedure should be implemented in order to find the "best" solution according to the "stakeholders" preferences.

Acknowledgement This work has benefited from the Luso-French grant PESSOA 2004, No. 07863YE (GRICES/Ambassade de France au Portugal). The first two authors would like to acknowledge the partial support from MONET research project (POCTI/GES/37707/2001) and the second author also acknowledges the support of the grant SFRH/BDP/6800/2001 (Fundação para a Ciência e a Tecnologia, Portugal). The authors acknowledge Nicole Chenik for helping us with the English checking of the manuscript.

References

Bergey, P. K., Ragsdale, C. T., & Hoskote, M. (2003a). A decision support system for the electrical power districting problem. *Decision Support Systems*, *36*, 1–17.

Bergey, P. K., Ragsdale, C. T., & Hoskote, M. (2003b). A simulated annealing genetic algorithm for the electrical power districting problem. *Annals of Operations Research*, *121*, 33–55.

Bourjolly, J. M., Laporte, G., & Rousseau, J. M. (1981). Découpage electoral automatisé: application á l'Île de Montréal. *INFOR: Information Systems and Operational Research*, *19*, 113–124.

Bozkaya, B., Erkut, E., & Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, *144*, 12–26.

Chance, C. W. (1965). *Representation and reapportionment*. Political studies: number 2, Dept. of Political Science, Ohio State University, Columbus.

Cortona, P. G., Manzi, C., Pennisi, A., Ricca, F., & Simeone, B. (1999). *Evaluation and optimization of electoral systems. SIAM monographs on discrete mathematics and applications*. Philadelphia: SIAM.

D'Amico, S. J., Wang, S. J., Batta, R., & Rump, C. M. (2002). A simulated annealing approach to police district design. *Computers & Operations Research*, *29*, 667–684.

Deckro, R. F. (1979). Multiple objective districting: a general heuristic approach using multiple criteria. *Operational Research Quarterly*, *28*, 953–961.

Easingwood, C. (1973). A heuristic approach to selecting sales regions and territories. *Operational Research Quarterly*, *24*(4), 527–534.

Ferland, J. A., & Guénette, G. (1990). Decision support system for the school districting problem. *Operations Research*, *38*, 15–21.

Fleischmann, B., & Paraschis, J. N. (1988). Solving a large scale districting problem: a case report. *Computers & Operations Research*, *15*(6), 521–533.

Forrest, E. (1964). Apportionment by computer. *American Behavioral Scientist*, *7*, 23–25.

Garfinkel, R. S., & Nemhauser, G. L. (1970). Optimal political districting by implicit enumeration techniques. *Management Science*, *16*(8), 495–508.

Hess, S. W., & Samuels, S. A. (1971). Experiences with a sales districting model: criteria and implementation. *Management Science*, *18*(4), 41–54.

Hess, S. W., Siegfeldt, J. B., Whelan, J. N., & Zitlau, P. A. (1965). Nonpartisan political redistricting by computer. *Operations Research*, *13*(6), 998–1006.

Hojati, M. (1996). Optimal political districting. *Computers & Operations Research*, *23*(12), 1147–1161.

Mehrotra, A. (1992). *Constrained graph*. PhD thesis, Georgia Institute of Technology.

Mehrotra, A., Johnson, E. L., & Nemhauser, G. L. (1998). An optimization based heuristic for political districting. *Management Science*, *44*(8), 1100–1114.

Mousseau, V., Roy, B., & Sommerlatt, I. (2001). Development of a decision aiding tool for the evolution of public transport ticket pricing in the Paris region. In M. Paruccini, A. Colorni & B. Roy (Eds.), *A-MCD-A aide multicritère à la décision—multiple criteria decision aiding* (pp. 213–230). Luxembourg: Joint Research Center, European Commission.

Muyldermans, L., Cattrysse, D., Oudheusden, D. V., & Lotan, T. (2002). Districting for salt spreading operations. *European Journal of Operational Research*, *139*, 521–532.

Nygreen, B. (1988). European assembly constituencies for wales. Comparing of methods for solving a political districting problem. *Mathematical Programming*, *42*, 159–169.

Park, K., Lee, K., Park, S., & Lee, H. (2000). Telecommunication node clustering with node compatibility and network survivability requirements. *Management Science*, *46*(3), 363–374.

Preux, P., & Talbi, E. G. (1999). Towards hybrid evolutionary algorithms. *International Transactions in Operational Research*, *6*, 557–570.

Reeves, C. R. (1997). Genetic algorithms for the operations researcher. *INFORMS Journal on Computing*, *9*(3), 231–250.

Ross, P. (1997). What are genetic algorithms good at? *INFORMS Journal on Computing*, *9*(3), 260–262.

Shanker, R. J., Turner, R. E., & Zoltners, A. A. (1975). Sales territory design: an integrated approach. *Management Science*, *22*(3), 309–320.

Srinivas, N., & Deb, K. (1995). Multiobjective optimization using nondominated sorting genetic algorithms. *Evolutionary Computation*, *2/3*, 221–248.

Steuer, R. E. (1986). *Multiple criteria optimization: theory, computation and application*. New York: Wiley.

Tavares-Pereira, F., Figueira, J., Mousseau, V., & Roy, B. (2004). *Multiple criteria districting problems, models, algorithms, and applications: the public transportation Paris region pricing system*. Research report no. 21/2004, INESC—Coimbra, Portugal.

Van Veldhuizen, D. A. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

Vickrey, W. (1961). On the preventions of gerrymandering. *Political Science Quarterly*, *76*(1), 105–110.

Zoltners, A. A., & Sinha, P. (1983). Sales territory alignment: a review and model. *Management Science*, *29*(3), 1237–1256.