

Grouping genetic operators for functional areas delimitation based on spatial interaction

Lucas Martínez-Bernabeu^a, Francisco Flórez-Revuelta^b, José Manuel Casado-Díaz^a

^a*Institute of International Economics, University of Alicante, P.O. Box 99, E-03080, Alicante, Spain*

^b*University Institute for Computing Research, University of Alicante, P.O. Box 99, E-03080, Alicante, Spain*

Abstract

The delineation of functional economic areas, or market areas, is a problem of large practical relevance, since the delineation of functional sets such as Economic Areas in the US, Travel-to-Work Areas in the United Kingdom, and their counterparts in other OECD countries are the basis of many statistical operations and policy making decisions at local levels. This is a combinatorial optimisation problem defined as the partition of a given set of indivisible spatial units (covering a territory) into regions characterised by being (a) self-contained and (b) cohesive, in terms of spatial interaction data (flows, relationships). Usually, each region must reach a minimum size and self-containment level, and must be continuous. Although this type of problems has been typically solved through greedy methods, a recent strand of the literature in this field has been concerned with the use of evolutionary algorithms with *ad hoc* operators. Although these algorithms have proved to be successful in improving the results of some of the more widely applied official procedures, they are so time consuming that cannot be applied directly to solve real-world problems. In this paper we propose a new set of group-based mutation operators featuring general operations over disjoint groups, tailored to tackle with that problem so that all the constraints are respected during the operation to improve efficiency. A comparative analysis of our results with those from previous approaches shows that our algorithm systematically improves them in terms of both quality and processing time, something of crucial relevance since it allows dealing with most large, real-world problems in reasonable time.

Keywords: functional areas, local market, evolutionary algorithm, grouping problem, regionalisation, combinatorial optimisation

1. Introduction

A functional economic area (also termed market area or functional region among others) in Geography and Economics is a geographical space where a given economic market (housing, labour, leisure, transportation, communications, public services, etc.) operates [5]. That is, a set of suppliers and demanders (agents) of a given product are located in that area and trade among them for a certain product, ideally without any interaction with other agents located in different market areas, i.e. functional areas must be highly self-contained in terms of connections between suppliers and demanders. Moreover, in its ideal definition, any supplier should be able to trade with any demander in its market, and viceversa. In other words, the ideal market area is characterised by a high level of internal cohesion among its suppliers and demanders, and a high degree of self containment with regards other areas. As it is obvious both conditions limit each other (i.e. most countries are highly self-contained in terms of housing and labour markets, but a country of average size can hardly be considered as a single market, since many local and regional factors influence housing and labour markets performance). Functional areas delimitations are very relevant in different contexts. They are crucial

for different stakeholders when analysing social and economic phenomena at urban and regional levels, and are used by the authorities to define, implement and monitor many different policies. The examples include planning the amount of land dedicated for housing or different economic activities in a certain place, deciding which unemployment policies apply on each area, or establishing investment priorities and predicting development requirements in passengers and freight transportation. The quality of this wide range of uses, and that of the analyses which stem from them, depends on the accuracy in delineating the boundaries of these market areas.

Particular cases of flow-based functional areas used for official, business and academic purposes in diverse countries are the Economic Areas defined by the US Bureau of Economic Analysis (BEA) [21]; housing market areas [1, 40]; and functional transportation areas of passengers [25] and or freight transport [27]. The larger family of functional areas is however constituted by the different approaches to the concept of Local Labour Market Areas (LLMAs) such as the UK Travel-to-Work Areas (TTWAs), French *zones d'emploi* or Italian *sistemi locali del lavoro*, among others (see [3] for a survey of these procedures).

One particularity of these sets of functional areas is that they are based on the application of different procedures of aggregation over matrices where the spatial interaction between certain basic spatial units (BSUs) is captured. An example of this spatial interaction are travel-to-work commuting flows between the municipalities, wards or counties that constitute one state or

Email addresses: lucas.martinez@ua.es (Lucas Martínez-Bernabeu), fllorez@dtic.ua.es (Francisco Flórez-Revuelta), jmcasado@ua.es (José Manuel Casado-Díaz)

URL: <http://www.iuii.ua.es> (Francisco Flórez-Revuelta), <http://iei.ua.es> (José Manuel Casado-Díaz)

country. Examples of these aggregation procedures are Intramax [28] and the more complex TTWAs method [6]. In [39] the author compares the results from Intramax and TTWAs methods for the case of Australia in 2001, and concludes that none of these methods seems to exhibit robust local properties (at a detail level of basic spatial units) with respect to a criterion of closure (self-containment) as defined by a revised unbiased form of the membership function used by Feng in [11]. These findings lead the author to conclude that other techniques as evolutionary algorithms could be a promising approach for the LLMA's delimitation. This is in fact the path followed by a number of works which have recently approached this class of problems through the use of evolutionary algorithms. This is the case of [14], where the authors propose an evolutionary algorithm in which multiple tailored operators are used to tackle with the LLMA's delimitation problem. Although the results outperform those from traditional methods in quality the procedure cannot be used directly to solve in reasonable time large size problems typical in this context.

In this paper we aim at adding to this strand of the literature by presenting an evolutionary algorithm able to solve the biggest real-world problems with high standards of quality. We have designed a new set of heuristic grouping operators that (a) features most of the general operations over disjoint sets and (b) takes into consideration the constraints during their process. The generality of the operators is aimed to ensure a high versatility of the search process, and the consideration of the restrictions inside the operators tries to avoid producing invalid solutions to improve efficiency. Specially, the inclusion of the contiguity restriction allows to greatly reduce the possible allocations of each BSU, inducing a strong shrink of the search space without constraining the likelihood of reaching the optimal solution. Our results improve previous approaches both in quality and CPU time, as it is illustrated in the last sections of the paper.

The rest of the paper is structured as follows. Next section puts what we call the functional areas delineation problem (FADP) in the context of constrained combinatorial optimisation problems, analogous to those of graph partitioning problems; then focuses on one particular case of functional areas delineation, that of LLMA's, reviewing both the official methods traditionally used and the recent evolutionary approach; as well as the use of grouping genetic algorithms specifically designed for grouping problems; and the incorporation of knowledge into the operators for improving convergence and performance. Section 3 sets the frame for the formulation of the LLMA's delimitation problem through the specification of the constraints and fitness functions that will be used. In section 4 we describe the evolutionary algorithm and our proposal of multiple grouping operators tailored to tackle with the problem. Section 5 presents a comparison of the outcomes obtained when a real-world case study of the LLMA delimitation problem is approached through different procedures including our proposal. As stated before, the results clearly show that our method outperforms those from recent works. Finally, the last section summarises our conclusions and future work agenda.

2. Background

Finding clean boundaries which isolate one market's constituting BSUs from others' is seldom possible in real-world functional areas delimitation exercises, and any division implies the segregation of some suppliers from some of their actual demanders. Moreover, an *a priori* known ideal number of markets hardly ever exists. This combination of factors implies that the expert conducting a delineation of local markets tries to identify as many markets as possible, so that they are small enough to allow a high cohesion among its elements; but all the efforts have also to be made so that the boundaries cut off as few connections between suppliers and demanders as possible, usually constraining the results by a set of criteria fixed *a priori* and a set of parameter values which are typically adjusted over trial and error.

The delineation of market areas can be undertaken as a combinatorial optimisation problem, which we call Functional Area Delimitation Problem (FADP), where the basic spatial units (wards, districts, counties, municipalities, census areas, etc. which are considered as indivisible) are connected in terms of flows of workers/goods/services with other units and with themselves (very frequently the supplier and demander in one specific exchange are located in the same BSU). The aim is to find the better partition of these basic elements so that each group (functional area) has weak or no connections with other groups (high self-containment) while the elements of the group (the constituent BSUs) are greatly inter-connected between them (high cohesion).

2.1. Related problems

One possible approach to this problem is considering the BSUs as nodes of a graph, and the flows of products among them as its directed weighted edges, where the aim is to group the nodes into K disjoint and non-empty sets in such a way that some given objective functions are optimised subject to a set of given constraints. This formulation resembles that of the graph partitioning problem or GPP. See for example [12] for a survey and [8] for an application of EAs and a short review of different objectives and constraints. Important objectives or constraints are usually edge-cutting (also termed coupling, to be minimised, equivalent to maximising self-containment) as well as size/load balancing (that resembles the minimum size constraint in some FADPs) and subgraphs' shape compactness (somehow similar to cohesion). Therefore we can consider the FADP as a kind of general GPP. Nevertheless, methods designed for GPP are not directly applicable in this context since the fitness functions and restrictions differs, as do the kind of interaction data and the structure of the networks that they form. As it is well known, GPP has been shown to be NP-complete for $K \geq 2$. The number of partitions of a set of n elements into non-empty disjoint groups follows the equation of the Bell's number [34], B_n ; real world problems of delineation of labour or housing markets can have tens of thousands of elements leading to astronomical numbers. Hence, exact methods, like linear and integer programming, are only computationally possible for a very limited number of nodes.

2.2. Local labour market areas

For the rest of the paper, we will focus on a particular case of functional regionalisation, LLMA, where the suppliers are those working or seeking work, the demanders are employers, and the product exchanged is the labour force. This class of geographical entities serve in different countries as the territorial framework for designing, implementing and monitoring effective labour market policies and statistical operations at sub-national levels. As pointed out before, the success of these key policies crucially depends on the adequacy of the LLMA delineation. According to Eurostat [9] an appropriate procedure of delineation of such areas over a given territory should have several features: the resulting LLMA geography should consist of disjoint areas exhaustively covering the territory; each of these LLMA should be characterised by a high degree of self-containment in terms of travel-to-work (i.e. most workers in a specific LLMA must live in that area and most of the LLMA employed residents should also work locally), relatively homogeneous in population size (exceeding a minimum size constraint, for instance) and geographical contiguity. A preference for detail is also a common principle guiding this type of functional delineations [30]. This is related with the desirability of identifying LLMA which are small enough to allow the residents of each LLMA to change from one job to another within that LLMA without this implying a change of the place of residence (i.e. daily travel-to-work should be possible and reasonable between any pair residence-job, following the market definition). All these requisites are coherent with the above stated main requisites of a functional economic area: self-containment and internal cohesion, with the aggregation of minimum size and contiguity constraints, and the fact that none of the basic spatial units should remain unassigned to a LLMA at the end of the process.

One of the more widely and successfully used official procedures for the delineation of LLMA is that of UK Travel-to-Work Areas (TTWAs) (fully described in [6]), that is a sophistication of the older Intramax method [28], that is in turn a greedy hierarchical aggregation procedure that focuses on the relative strength of interactions among spatial units. The TTWAs method has been applied with minor changes in other countries [2, 18, 32, 38]. A new version of the TTWA's method [30] that significantly simplifies the procedure has been used recently to delineate the new set of TTWAs stemming from the last Census data (in [4] this same procedure is applied for the delineation of LLMA in Spain). In its new formulation the number of stages and parameters decreases from five stages and more than ten parameters to only one stage and four parameters respectively (four parameters that fix the thresholds for minimum size and self-containment). This regionalisation method can be defined as a greedy aggregative algorithm that departs from the consideration of total division (one basic spatial unit, one market), and iterates one single step in which the current worst (in terms of minimum self-containment and size) group is dismembered into its constituent BSUs, that are then independently merged with the groups with which each of them shares the maximum interaction, stopping when all the defined LLMA meet both the self-containment and size requirements.

Interaction is measured using an index proposed in [6]. As other greedy aggregative methods this procedure allows reaching adequate solutions with little CPU time, but a real direct optimisation approach is not used (in fact, this type of procedures have finding the first feasible solution as its termination condition). Accordingly, the output (the set of LLMA) of such procedures is just one among many other good solutions that could be expected to be quite far from the optimal one. As mentioned in the Introduction, in [39] the author compares the results from Intramax and TTWAs methods for the case of Australia, and concludes that neither method appear to exhibit robust local properties, with respect to a criterion of closure (self-containment), as defined by the revised specification of the modularity function presented in [11]. The author finally concludes that evolutionary computation could be a promising approach for the LLMA delimitation.

2.3. Evolutionary computation approach

In the last years there have been some contributions aimed at tackling with the delineation of LLMA through evolutionary computation as a combinatorial optimisation problem. An evolutionary algorithm which improved the solutions from the TTWA method was presented in [13]. This procedure included seventeen *ad hoc* operators whose performance was more deeply analysed in [14]. The authors stated that the multiple constraints which are part of the problem cause the number of valid solutions—those that meet the constraints—to be extraordinarily small with regards to total size of the search space, so standard genetic operators hardly ever produce valid solutions, something that led them to design the many tailored operators proposed. Despite the design of *ad hoc* genetic operators, a high percentage of the individuals generated were not valid, causing a poor performance of the algorithm in terms of time required. A version of the same method with the incorporation of memetic techniques (a routine for repairing invalid solutions, and a local optimisation function) was presented in [15]. This proposal improved the results in [13] slightly but involved a significant increase of computational time. Three adaptive version of the same (non memetic) method were introduced in [16], seeking a better use of the multiple operators whose behaviour and performance varies over time, and with the size of the problem. Although this algorithm produced a significant speedup in the early stages of the evolution it does not improve the quality of the final solution.

All these works approach the delineation problem through the maximisation of markets' internal interaction in terms of travel-to-work flows, using a generalisation of the index proposed in [6], subject to the restrictions of minimum self-containment and size thresholds, with the aim of identifying as many independent markets as possible. Moreover, following the established practice, the procedures do not take into account the information about spatial adjacency among the spatial units during their operation, although according to the needs of the final users and the recommendations by Eurostat [9] the final set of LLMA made available to potential users and notably to public authorities must consist of LLMA which are internally continuous in geographical terms. It is not then guaranteed that

the final set of LLMA is composed of markets whose constituent BSUs are geographically connected (in [6] and [30], for example, contiguity is achieved in a posterior step and not included in the two versions of the TTWAs algorithm). The reason why contiguity is not considered during these algorithms operation is somehow inherited from the greedy methods used in LLMA's official procedures of delineation. As pointed out in [3], methods which require an explicit contiguity constraint at every stage of the analysis will inevitably tend to fall well short of providing an optimal form of boundary definition. In [4] a greedy procedure aimed at fixing the discontinuities that could arise when applying the TTWAs method (or any other method not including the contiguity constraint) is presented showing that the final (continuous) solution is frequently substantially different from the (not always continuous) algorithm output. Thus, if the final map of LLMA must consist of geographically continuous markets then it is worth considering the inclusion of such restriction during the optimisation procedure, so that it can be assessed against other alternatives, instead of applying a final (ex-algorithm) procedure to solve them, a strategy which sometimes can offer aggregations which are farther from the optimum. This is in any case especially true when an evolutionary computation alternative is considered, since the inclusion of a contiguity constraint is obviously less harmful for the quality of the results when the number of alternative groupings considered is enormous. Moreover, the potential restriction associated to the consideration of contiguity during the regionalisation procedure is substantially reduced when the method employed is a search technique and the focus is moved from individual BSU interactions to groups of them. This issue will be discussed later with more detail.

2.3.1. Grouping problem approach

In [10] is extensively discussed the nature of the grouping problems (reviewing the GPP, quite similar to the FADP as have been said previously). Among negative interactions that the representation of the solution and the dynamics of the operators employed can exert over the evolutionary process in a grouping problem, [10] lists (a) redundancy in the representation, i.e. the existence of many genetic representations for the same solution, that enlarges the search space compared to the solution space—and therefore reduces the performance of the algorithm—; (b) context insensitivity in the reproduction operators, what causes the loss of the meaning of the schemata with respect to the problem in the process of recombination; and (c) the schema disruption, i.e. the destruction of good schemata (groups), more frequent as groups' size increases, when applying standard crossover operators over object-oriented (or item-based) representations. Falkenauer proposed a group-based representation where the crossover is applied over the groups instead of the elements, with a final problem-dependent repair procedure for fixing the feasibility of broken groups. He also suggested three possible mutation operators for grouping problems, all of them problem dependant: creation of a new group, deletion of a group, and shuffle of some elements among their groups. This kind of evolutionary computation techniques are called grouping genetic algorithms (GGAs).

Several partitioning problems have been addressed with GGAs based on the proposal of Falkenauer, achieving good results compared with item-based genetic algorithms (GA) and other search procedures. In most of these applications only one mutation operator is used: In [26] and [31] random groups are removed and a reconstruction subroutine (repair) is applied to reassign their elements to the remaining groups and to create new groups if necessary, as it is done by Falkenauer in [10] solving a bin packing problem. In [36] random elements are removed from their groups and reassigned to the remaining groups, or to new groups if necessary. Similarly, in [29] a group-based differential mutation is used to remove the elements of an individual that have the same encoding as other given individual from its groups, and then reassign the free elements through a repair heuristic. In [19] and [20] there are no mutation operators, instead local-search operators are used (that can be seen as a mutation operator embedded in a search technique), respectively a hill climbing technique and a Tabu Search, where in both of them an informed operator tries to swap elements between groups.

Other partitioning problems have been solved employing evolutionary algorithms with group-based operators working over standard encodings different from that of the GGAs, that can also be considered as GGAs since they are solving grouping problems with group-oriented operators. In previously cited [8] a generic GPP is solved with group-based crossover and mutation (in this case, random swaps of elements from a group to another).

This has motivated the present work to exploit the potentiality of the GGA approach to solve the FADP.

2.3.2. Knowledge incorporation into the operators

Standard genetic operators are random and blind to the problem (the phenotype), i.e. they just change alleles of the genotype to random values among a set or interval of possible values. It is the natural selection what contributes intelligence to the search process. If we know or suppose that some possible values for certain allele are better than the others, then we can limit the possible choices of the genetic operators and expect improved results. Limiting too much the (random) behaviour of the search operators could lead to the inability to reach the optimal solution, but producing as few invalid solutions as possible would be a reasonable strategy to increase the efficiency and efficacy of an evolutionary algorithm for hard combinatorial problems as FADP. Therefore, our aim is to reduce the average number of infeasible solutions per generation while still allowing a robust exploration of the search space. Other authors have done this before, as in [35], where knowledge-based operators are used to drastically reduce the number of infeasible solutions produce by mutation. If we also incorporate knowledge about the objective functions into the operators so that the more promising changes (in terms of quality apart from feasibility) have more probabilities to happen, it is expectable to improve the convergence rate to the optimal solution [41]. In [33] several random mutations are performed over a surrogate model of the fitness function and the best one is chosen for effective application, greatly improving results over pure random

mutations.

Also memetic techniques where the evolutionary algorithm (EA) is hybridised with repair and local-search techniques can be seen as a way of incorporating knowledge into the evolutionary process. For example, [24] explores the use of self-adapting local search heuristics in a memetic algorithm outperforming the equivalent GA. In [37] a GA that also includes a local-search (downhill simplex) operator and a knowledge-based operator is proposed.

With that motivation, in this paper we present a new set of grouping genetic operators featuring general unary, binary and n -ary grouping operations over disjoint sets, provided with simple heuristics designed under three premises: (a) being able to produce any possible solution to ensure exploration, (b) giving preference to promising solutions so as to improve exploitation and (c) avoiding to produce invalid solutions (in terms of minimum self-containment, minimum size and geographical contiguity) to increase efficiency.

3. Local labour market delimitation problem statement

As stated in the Introduction, the problem consists of the grouping of basic spatial units (BSU) —such as districts, municipalities or counties— into functional areas (in this case LL-MAs) so that the proportion of workers that cross their boundaries in their travel to work is low, while the number of defined areas is maximised.

Let $S = \{s_1, s_2, \dots, s_N\}$ be a set of $N = |S|$ BSUs (the territory to be divided into LL-MAs) and W_{s_i, s_j} the number of commuters from BSU s_i to BSU s_j , that is, the number of residents in s_i that work in s_j (thus, W_{s_i, s_i} is the amount of people who simultaneously live and work in BSU s_i). The objective is to obtain the set of markets $M = \{M_1, M_2, \dots, M_K\}$ (where K is unknown *a priori*), so as $M_i \neq \emptyset, \forall M_i \in M; \bigcup_{i=1}^m M_i = S$ and $M_i \cap M_j = \emptyset, \forall i, j \in [1, K], i \neq j, 1 \leq K \leq N$, that maximises fitness function $f(M)$, which is fully explained in section 3.2.

3.1. Interaction index

To assess the degree of interaction between a pair of markets (that can consist of a single BSU) we use the interaction index proposed in [14], a generalisation of the index used in the TTWA's method [6]. Let the interaction index between two markets $II(M_i, M_j)$ be defined as:

$$II(M_i, M_j) = \underbrace{\frac{W_{M_i, M_j}}{R_i}}_{PE_{M_i, M_j}} \times \underbrace{\frac{W_{M_i, M_j}}{J_j}}_{PJ_{M_i, M_j}} + \underbrace{\frac{W_{M_j, M_i}}{R_j}}_{PE_{M_j, M_i}} \times \underbrace{\frac{W_{M_j, M_i}}{J_i}}_{PJ_{M_j, M_i}} \quad (1)$$

where

$$W_{M_s, M_t} = \sum_{\forall s_i \in M_s} \sum_{\forall s_j \in M_t} W_{s_i, s_j} \quad (2)$$

is the number of commuters residing in the set of BSUs belonging to M_s that works in any of the BSUs belonging to M_t ; $R_k = W_{\{M_k\}, S}$ the number of workers residing in M_k ; and $J_k = W_{S, \{M_k\}}$ the number of jobs in M_k .

Factor PE_{M_i, M_j} is the fraction of the employed population residing in M_i and working in M_j ; and PJ_{M_i, M_j} is the portion of jobs in M_j that are held by workers residing in (coming from) M_i .

This function is similar to a gravitational measure of attraction, that takes into account the relative size of the markets to weight the flows between them, so that interaction between small interdependent markets don't get eclipsed by the relations between large ones, that tend to have high absolute values but relatively low importance.

3.2. Fitness function

This interaction index can be the base for different fitness functions. For comparison against previous works in this exercise we have tested our method with:

$$f(M) = \text{card}(M) \times \sum_{\forall s_i \in S} II(\{s_i\}, M^i - \{s_i\}) \quad (3)$$

Where M^i is the market to which s_i belongs. For each BSU, the interaction index between that BSU —considered as a mono-BSU market— and the rest of its market (the result of subtracting that BSU from the market to which it belongs). The inclusion of the number of markets as a factor seeks to reaching the highest possible number of independent markets —this is one of the criteria usually applied in practical exercises [3].

We must notice that markets consisting of only one BSU get a null interaction index, but these markets should have a positive value if they are cohesive (and we can assume that any municipality is cohesive) and self-contained. Therefore that value is approximated by assigning each singleton market the accumulated interaction index of all the multi-BSU markets divided by the number of BSUs they embrace, prior to the product by $\text{card}(M)$.

3.3. Constraints

In order to consider a given solution M as valid, each market $M_i \in M$ must fulfil two requirements of minimum size in terms of employed population ($\beta_4 \geq 1$) and minimum self-containment ($0 \leq \beta_1 \leq 1$).

$$W_{M_i, S} \geq \beta_4 \quad (4)$$

$$\text{msc}(M_i) \geq \beta_1 \quad (5)$$

where

$$\text{msc}(M_i) = \min \left(\frac{W_{M_i, M_i}}{W_{M_i, S}}, \frac{W_{M_i, M_i}}{W_{S, M_i}} \right) \quad (6)$$

is the minimum between supply-side and demand-side self-containment levels. That is, both the proportion of the occupied working locally, and the proportion of jobs filled by local workers must exceed a given threshold.

Very urbanised environments in real world are characterised by the intensity and complexity of the network of commuting flows, something which makes it difficult to identify isolated groups of BSU in such environments, resulting in huge markets that despite being very self-contained are too big to allow

daily trips to work between any pair of BSUs in the market (that is, the market has a low cohesion and could be too far from the definition of the ideal market, where any supplier can trade with any demander). To facilitate the identification of more and smaller separate LLMA in such environments a trade-off between both constraints (self-containment and minimum size) was introduced similarly to [6], but using the formulation proposed by Casado-Díaz [3]. According to this proposal, the minimum self-containment requirement is linearly relaxed from β_2 to β_1 ($0 \leq \beta_1 \leq \beta_2 \leq 1$) for population's sizes from β_4 to β_1 ($1 \leq \beta_4 \leq \beta_3$). For each market in a given solution, this trade-off is evaluated as follows:

$$msc(M_i) \geq \beta_2 + \frac{\beta_2 - \beta_1}{\beta_4 - \beta_3} (W_{M_i, S} - \beta_4) \quad (7)$$

So, when a market does not fulfil all this three constraints —eqs. 4, 5 and 7—, the solution containing it is considered invalid.

We have also included the requisite of geographic contiguity of the LLMA over an adjacency matrix A ($N \times N$, where cell A_{ij} is 1 if BSUs i and j are geographically adjacent, and 0 otherwise), so that any BSU of a valid LLMA should be reachable from any other BSU of that LLMA without crossing its geographic boundaries, that is, for each pair of BSUs in a given LLMA there must be at least one path of finite length through the edges of the adjacency matrix that connects those BSUs.

In the formulation presented in [16] the authors included a similar requisite expressed in terms of functional connectivity or functional neighbourhood (instead of geographic contiguity) through the largest outgoing/incoming commuting flows of each BSU (so that two BSUs are contiguous if they are connected by one of those flows). The aim was to focus the search on the more promising solutions, and to induce some degree of contiguity in the solutions produced without explicitly incorporating that information. Although that constraint is either considered in the cited widely-used TTWAs method, it is enforced in the final solution by means of expert guided or automatic adjustments that repair the discontinuous markets trying to maintain the feasibility of the rest of the markets. As we mentioned in section 2, we consider that in the context of an optimisation-based procedure all the hard constraints should be taken into account during the main optimisation process in order to improve the results. Therefore, we haven't included the functional neighbourhood requisite in this work, since the geographic contiguity implies some degree of functional neighbourhood and focuses the search on what should be good solutions as well. Producing solutions that always meet the geographic contiguity let us to skip the final correction that usually means a loss in fitness [4]. The results shown later support this decision.

4. Grouping evolutionary algorithm with multiple group-based operators

As stated in section 1, we have decided to use a grouping evolutionary algorithm (GEA), an evolutionary approach with group based operators working over an encoding able to represent the partitions of a set, in this case LLMA delimitations.

Falkenauer proposed to employ one crossover and three mutation operators (creation, destruction and alteration of groups), to be tailored for the particular problem to be solved. In most GGA applications cited in section 2 only one mutation operator is used. In [14] the authors stated that eliminating any of the many operators proposed from the problem results in a reduction of quality of the solutions or the algorithms' performance, something that is in contradiction with Falkenauer's proposal of using three general operators. Keeping that in mind, we have designed one crossover and nine different group-based mutation operations, in an attempt to cover all the more general operations in one-to-one, one-to-many and many-to-many relations between disjoint groups:

1. partition through a greedy heuristic (see section 4.2) of a union of two related groups;
2. creation of a new group by taking a border element of a random group and assigning elements to it from the adjacent groups, until the criteria is met or it is not possible to continue (creates a new group and modifies one or more);
3. dismembering of a group and assignation of its elements to related, adjacent groups (eliminates one group and modifies one or more);
4. exclusion of elements from a group and reassignment to related, adjacent groups (modifies two or more groups);
5. inclusion in a group of related, adjacent elements from surrounding groups, the inverse of expulsion, (modifies two or more groups);
6. segregation of a subset of elements from a group and reassignment to a related, adjacent group (modifies two groups);
7. annexation by one group of a subset of elements from another related, adjacent group, the inverse of segregation (modifies two groups);
8. random reassignments of border elements in random groups to related, adjacent groups (modifies many groups); and
9. exchange of elements between two related, adjacent groups (modifies two groups).

In the previous and next descriptions, two markets/BSUs are considered related if they have a value higher than 0 in eq.1 (i.e. there is some interaction between them), and two markets are adjacent if any of the BSUs in one market is adjacent to any of the BSUs in the other. Also border BSUs are those in a market that are adjacent to different markets. In the rest of the paper we refer to groups as markets (general case of LLMA) and to elements as BSUs.

4.1. Chromosome representation

The chromosomes of the individuals in the population represent feasible solutions, that is, partitions of S that constitute a regionalisation of the territory into LLMA. Contrarily to the strategy followed in several GGA applications cited in section 2, we have decided to use a variation of group-number encoding [22], where each individual is represented by a vector of N components, each of which corresponds to a BSU of S and

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Chromosome:	1	2	1	3	2	2	1	4	4	2
$M_1=\{s_1, s_3, s_7\}$	$M_2=\{s_2, s_5, s_6, s_{10}\}$			$M_3=\{s_4\}$		$M_4=\{s_8, s_9\}$				

Figure 1: Representation of individuals

takes the value of the identifier of the market the BSU belongs to (Fig. 1). This representation ensures the absence of overlapping and therefore this constraint has not to be evaluated. As Falkenauer explains in [10], the standard group-number encoding suffers from very high redundancy (see section 2.3.1) while GGA representation suffers from low redundancy (in the group-part, as stated later in this section). We solve the redundancy issue by re-codifying any new individual so that the market identifiers increase sequentially in the vector of BSUs, in such a way that the group-identifier of the first BSU in the vector (s_1) is always 1, the next BSU in the vector not belonging to market 1 receives the group identifier 2, the market of the next BSU in the vector not belonging to an already numbered market will receive the next group number (last+1). Therefore, vectors 12346, 13121 or 32123 would be recoded to the equivalent representations 12345, 12131 and 12321, respectively. This way we ensure that equal solutions (phenotype) will have equal representations (chromosome), having no redundancy at all (for a given sequence of the BSUs), thus allowing to easily identify new solutions already present in the population, so that they can be discarded without evaluating them, and assuring that the search space is not bigger than the solution's space.

It must be noticed that the maximum group-identifier in a chromosome is also the number of markets in that solution. This implies that our representation is easily transformed into that of the GGAs (e.g. 12131 would be 12131|123), and that both representations gain the benefits of the group-oriented encoding of the GGAs. Besides, our representation suffers from less redundancy than that of GGAs, as what would be our group-part of the chromosome in the GGA (12131|123) is unique for a given partition. Moreover, our encoding in its basic form has a fixed length (the number of BSUs), what simplifies the programming of the operators. But in the implementation some values of the markets (such as the number of constituting BSUs, the sums of incoming and outgoing flows, the contributed fitness value, and so on) can be included in the data structure containing the chromosome in order to accelerate the execution of the algorithm —mainly the evaluation of solutions which have been modified only partially. Since the number of markets in a solution changes during the evolution this would imply a variable-length representation, but in practice it can be implemented transparently to the operators.

4.2. Stochastic hierarchical agglomeration

For the creation of the initial population —a collection of randomly distributed start points for the search process— we use a stochastic hierarchical agglomerative algorithm (SHA) that, starting from a given partition of a set, iteratively chooses a market of low degree of validity and another related, adjacent

market and merges them, until all the markets considered meet the constraints of the problem. It is used also as a subroutine in three of the mutation operators as well as repair procedure in the crossover operator. The procedure is as follows:

1. Calculate the degree of validity (eq.8) of all the markets.
2. Terminate successfully if all the markets are valid.
3. Select, by 3-way tournament over the degree of validity, a market G with low validity.
4. Select a market H adjacent to G, with probability proportional to the relation (eq.1) to G.
5. Merge markets G and H and go to step 2.

The degree of validity of each market ranges from 0.0^+ to 1 (to be valid it must score 1.0), and it is calculated as follows:

$$validity(M_i) = \min\left(\frac{msc(M_i)}{\beta_2}; 1\right) \times \min\left(\frac{W_{M_i,S}}{\beta_4}; 1\right) \quad (8)$$

4.3. Evolutionary algorithm

The structure of the multi-operator evolutionary algorithm for the FADP follows the next steps:

1. Initialise population with N_p individuals with positive evaluation, generated by applying SHA over the whole set of BSUs of the problem taking them as mono-BSU markets.
2. Order population by fitness.
3. Apply genetic operators (see section 4.4) until N_o new valid individuals have been produced.
4. Order population by fitness.
5. Select next generation's population by ranking with elitism for the best.
6. Adapt the probabilities of application of the operators (optional, not used in this work).
7. Go to step 3 if termination condition is not met.

The parameter N_o controls how many new individuals are generated in each generation.

4.4. Application of operators

For clarity, we describe separately the way the operators are applied to generate new individuals.

1. Select by fitness-ranking a parent from the current population.
2. Select an operator.
3. If the operator is a crossover, select by fitness ranking a second different parent.
4. Create a new individual as a copy of the (first) parent.
5. Apply the selected operator to the new individual.
6. If the operator terminates successfully, the resulting individual is evaluated; otherwise its fitness will be set to 0 (invalid).

Notice that, contrarily to genetic algorithms, the crossover and mutation operators are treated equally in a single stage, so that N_o mutations and no crossovers (or viceversa) could be applied in a given generation.

4.5. Group-based operators

We have tried to allow the algorithm to perform any possible unary and binary operation that could be applied over sets of non overlapping markets, such as division of a market, union of two markets, simultaneous interchange of BSUs between two markets, etc., covering also 1-to-many and many-to-many relations. This has led us to design nine mutation operators. We have also designed a grouping crossover to work over the group-number encoding representation, following the common structure of a crossover in a GGA, as described in [10] and used in many GGA applications [8, 19, 20, 26, 29, 31]. This operator chooses markets from one parent to copy over the other parent and then uses a repair heuristic (in our case the SHA routine).

As stated in section 2, in order to accelerate the convergence to the optimum we have designed all the operators so that they never try changes that violate any hard constraint—for example, the union operator will never merge non-adjacent markets, or contiguity constraint would not be respected except with major, disruptive fixing operations—, unless it is possible to solve that violation with low computational effort before finishing the operation—for example, the creation of a new market from a BSU acting as a seed that absorbs adjacent BSUs from other markets could imply making some steps through invalid solutions before getting to a valid (and probably better) state, although it could also lead to invalid solutions if the process gets stuck—. Therefore, not all invalid changes are forbidden, as some *risky* changes could lead to valid and valuable solutions, and some operations can still produce invalid individuals. In our implementation, when an operator can not find a valid change, it returns an unsuccessful state and the result is not evaluated. Moreover, some operators always return successful state when producing repeated individuals (not allowed in the population), but these are detected in a stage previous to the evaluation and considered as invalid.

For further improvement of the performance, we also provide the operators with stochastic heuristics of the form of “select by 3-way tournament a market (or BSU) of high (or low) <attribute>”, “select a market (or BSU) of high (or low) <relationship> with market/BSU” or combinations of these, that try to bias the search process towards the more promising areas while still allowing any possible change. In the current problem, <attribute> can be minimum self-containment (eq.5), degree of validity (eq.8) and size (in terms of flows), and <relationship> is interaction (eq.1). We have used (a) 3-way tournament selection when choosing among all the markets or BSUs because it is fast when dealing with large numbers, and (b) selection with relationship-proportional probability when trying to identify markets or BSUs which are related to another, since the number of BSUs or markets related to a given one will often be small, even less than three, and in such cases 3-way tournament would always select the best option where we prefer stochastic behaviour. Obviously, these heuristics are problem-dependant, and other attributes or relationships would be of interest in other problems.

The descriptions and motivations of the proposed operators follows. Fig. 2 depicts the possible effects of some of the operators.

4.5.1. Mutation R: re-partition of the BSUs of two related, adjacent markets.

This operator takes two related, adjacent markets, merges them, and then tries to find a new partition of the BSUs of the BSUs that form the merged market using a greedy algorithm, in this case SHA (see 4.2). The description is as follows:

1. Select by 3-way tournament a market M_i with low self-containment.
2. Select a market M_j adjacent to M_i , with probability proportional to the attraction to M_i . If not found, terminate unsuccessfully.
3. Merge markets M_i and M_j into M_i .
4. Dismember M_i into its constituent BSUs and apply SHA restricted to them.
5. Terminate successfully.

Note that SHA is applied restricted to the BSUs of that form the merged market, what means that none of the rest of the markets in the solution will absorb any of those BSUs and therefore none of them changes after this operation.

This operator would be the fusion of two more elemental operations: the union of two related groups, and the partition of a group. In fact, we implemented those union and partition mutations (U and P), but our preliminary experimentation showed that using mutation R made it pointless their inclusion, since it offered better performance without them. The reason is quite obvious after an insightful analysis of the tests. Unions always produce valid individuals but few of them survive for the next generation because of the fitness loss, as unions only exert a positive influence on the evolution when the individuals they produced manage to survive until being selected for another operation (as division) that takes advantage of the big size of the market that results from the merger to find a better partition. In the case of the partition mutation, the problem is that they never produce a valid individual when the population is close to the optimal values and most markets are close to their minimum constraints’ thresholds so they can not be divided. This mutation R takes the advantage of the unions to greatly increase the percentage of valid and useful individuals produced by the division operator, being a good provider of fitness improvement during all the stages of the evolution.

4.5.2. Mutation I: inclusion into a market of adjacent, related BSUs.

This operator tries to increase the size of a market with low self-containment by widening it through the inclusion of some of the *unnecessary* adjacent BSUs in the surrounding markets with which it shares higher interaction (it is the opposite to the mutation E). A BSU is considered unnecessary to its current market if it can be extracted without causing the market to break any constraint, i.e. if they are unnecessary for the their markets’ validity.

1. Select by 3-way tournament a market M_i with low self-containment.
2. Choose a random number r between 1 and 10% of the average number of BSUs per market.

3. Repeat r times:
 - Select, with probability proportional to the attraction, a BSU s_i adjacent to M_i and belonging to a market M_j different to M_i .
 - Change s_i from M_j to M_i if both markets continue being valid.
4. Terminate successfully if there has been any change, otherwise terminate unsuccessfully,

4.5.3. Mutation E: exclusion from a market of border BSUs with high external attraction.

This operator tries to reduce the size of a big market by choosing some border BSUs with lower interaction to the rest of the market and reassigning them to other related, adjacent markets. Its process is inverse to that of mutation I.

1. Select by 3-way tournament a market M_i of big size.
2. Choose a random number r between 1 and 20% of the amount of BSUs in M_i (*).
3. Repeat r times:
 - Select by 3-way tournament a BSU s_i from M_i with low self-containment in M_i and adjacent to other markets.
 - Select, with probability proportional to the attraction to s_i , a market M_j adjacent to s_i .
 - Change s_i from M_j to M_i if both markets continue being valid.
4. Terminate successfully if there has been any change, otherwise terminate unsuccessfully.

(*) Self-containment of a BSU in its market is measured as the proportion of incoming and outgoing flows of that BSU that respectively came from or go to BSUs in the same market (including itself).

4.5.4. Mutation D: dismembering of a market and assignation of its constituent BSUs to the adjacent markets.

This operators uses the same mechanism than mutation E, but finishing when the market disappears. Although the effect of this is similar to the one produced by the union operator—the number of markets decreases by one—, it is more flexible since it allows to reassign the BSUs to different markets, more related to them, resulting in lower fitness decreases or even small fitness improvements.

1. Select by 3-way tournament a market M_i with low self-containment.
2. Select by 3-way tournament a BSU s_i of M_i with low self-containment in M_i and adjacent to other markets.
3. Select, with probability proportional to the attraction to s_i , a market M_j adjacent to s_i .
4. Change s_i from M_i to M_j if M_j continues being valid after the change, otherwise terminate unsuccessfully.
5. Go to 2 if there are BSUs left in M_i , otherwise terminate successfully.

4.5.5. Mutation N: creation of a new market using a border BSU as seed.

This operator chooses an *unnecessary*, border BSU in a market of low self-containment, creates a new market from that BSU, and then tries to absorb other adjacent BSUs from surrounding markets (if unnecessary in their original markets) until the new market becomes valid, or there are no more available BSUs to absorb, or a certain size has been reached without qualifying as a valid market. To avoid parametrization of this threshold size (that can be done anyway), we choose at random three markets in the current individual and take the median size (we also call it 3-way tournament).

1. Select by 3-way tournament a market M_i with low self-containment.
2. Select by 3-way tournament an unnecessary, border BSUs s_i of M_i with low self-containment in its market. If it is not possible to found a BSU meeting this requisites, terminate unsuccessfully.
3. Create a new market M_j conformed by s_i ; if M_j is a valid market, terminate successfully.
4. Select by 3-way tournament a market M_c with average size (median of three).
5. If size of M_j is bigger than size of M_c , terminate unsuccessfully.
6. Select s_k among the BSUs adjacent to M_j , with probability proportional to the percentage of attraction to M_j .
7. Change s_k from its market M_k to M_j if M_k continues being valid after the change, otherwise terminate unsuccessfully.
8. Go to 5.

The termination condition in step 5 is aimed at avoiding inefficient behaviour of the operator, caused when it is possible to find many small BSUs that can be absorbed from surrounding markets without significantly increasing (or even decreasing) its self-containment, despite of reaching a big size, giving place sometimes to markets of branched (undesirable) shapes and wasting a big amount of processing time. Therefore, it is advisable to find some trigger that stops the process when there are few chances of reaching a good solution, for example an upper threshold in size. We decided to use a stochastic procedure also here, instead of some statistics or fine tuning. Median and mode of the sizes of the markets in the individual could be also good references. The average size of markets is not suitable since there can be large BSUs (megalopolis) with sizes several orders of magnitude bigger than the smaller ones, biasing too much that statistic.

4.5.6. Mutation S: segregation of a subset of BSUs from a market and reassignment to another one.

This operators moves a set of BSUs from a big market to an adjacent market with high interaction.

1. Select by 3-way tournament a market M_i of big size.
2. Select by 3-way tournament a BSU s_i of M_i with low self-containment in M_i and adjacent to other markets. If not found then terminate unsuccessfully.

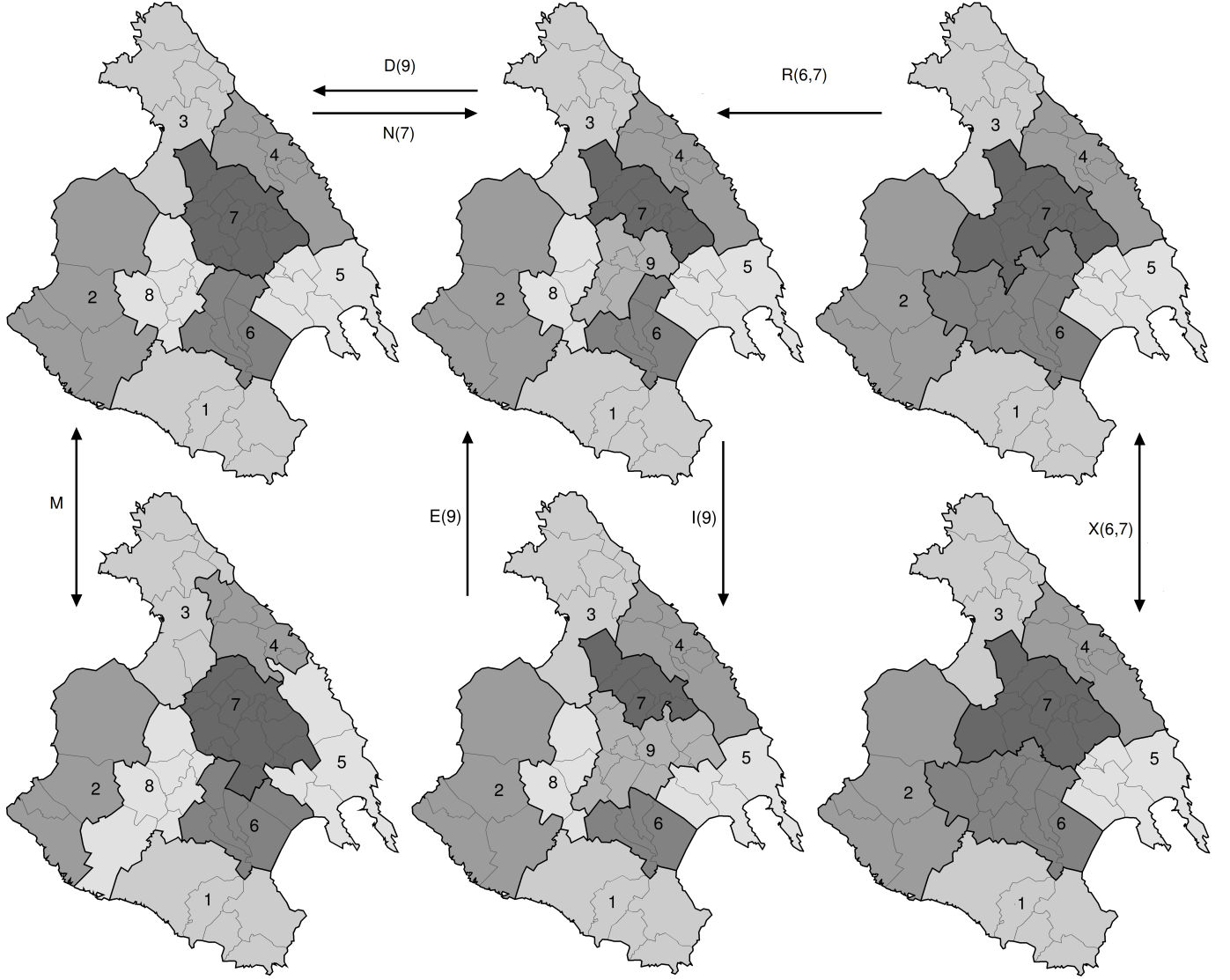


Figure 2: Examples of application of some operators. The maps depict different solutions of a (fictional) FADP, where the thin lines delimitate the BSUs, the thick lines delimitate the frontiers between markets, i.e. the partition, and the numbers are the group identifiers.

3. Select a market M_j adjacent to s_i , with probability proportional to the percentage of attraction to s_i .
4. Change s_i from M_i to M_j if both markets continue being valid after the change; otherwise terminate unsuccessfully.
5. Select by 3-way tournament another BSU s_i of M_i with high attraction to M_j and adjacent to it.
6. Change s_i from M_i to M_j if both markets continue being valid after the change and then go to 5; otherwise terminate successfully.

4.5.7. Mutation A: *annexation to a market of a subset of adjacent BSUs belonging to another market.*

This operators behaves as the inverse process of mutation S. As it is a one-to-one operator it could result in some redundancy (the annexation by M_i of part of M_j would be the same as the segregation of part of M_j and reassignment to M_i). However, their behaviour is not totally equivalent, since the way the markets are selected differs in each mutation, and preliminary tests

showed small loss in performance when disabling one of those operators.

1. Select by 3-way tournament a market M_i with low self-containment.
2. Select a market M_j adjacent to M_i , with probability proportional to the attraction to M_i . If not found terminate unsuccessfully.
3. Select by 3-way tournament a BSU s_i from M_j with high interaction with M_i and adjacent to it. If not found, terminate unsuccessfully.
4. Change s_i from M_j to M_i if both markets continue being valid after the change; otherwise terminate unsuccessfully.
5. Select by 3-way tournament another BSU s_i of M_j con high attraction a M_i and adjacent to it.
6. Change s_i from M_j to M_i if both markets continue being valid after the change and go to 5; otherwise terminate successfully.

4.5.8. Mutation X: exchange of two BSUs between two markets.

This operator chooses two related markets and tries to simultaneously move one BSU from each market to the other. This operator can be useful at points in which the simultaneity of the changes is mandatory to avoid generating an invalid market.

1. Select by 3-way tournament a market M_i with low self-containment.
2. Select, with probability proportional to the attraction to M_i , a market M_j adjacent to M_i . If not found, terminate unsuccessfully.
3. Select by 3-way tournament a BSU s_i of M_i adjacent to M_j with high interaction with it. If not found, terminate unsuccessfully.
4. Change s_i from M_i to M_j . If M_i is not continuous any more terminate unsuccessfully.
5. Select by 3-way tournament a BSU s_j of M_j with high interaction with M_i and adjacent to it, different from the previously exchanged s_i .
6. Change s_j from M_j to M_i . If M_j is not continuous any more terminate unsuccessfully.
7. Terminate successfully if M_i and M_j are valid; otherwise terminate unsuccessfully.

The first prototypes of this operator were more complex, and tried to interchange simultaneously more than one BSU from each market, but that way it was difficult to find valid or un-repeated solutions (very often the BSUs reassigned from one market to the other are the ones more likely to be selected and moved back). We therefore opted for this simpler version.

4.5.9. Mutation M: random changes.

This operator chooses at random border BSUs with low interaction in their markets and tries to move them to other related, adjacent markets. This is the only many-to-many operator that we have included, and the closer to the concept of standard mutation in genetic algorithms.

1. Choose a random number r between 1 and 2% of the number of BSUs.
2. Repeat r times:
 - Choose by 3-way tournament a border BSU s_i with low interaction with its market M_i .
 - Select a market M_j adjacent to s_i , with probability proportional to the attraction to s_i .
 - Change s_i from M_i to M_j if both markets continue being valid.
3. If there has been any effective change terminate successfully; otherwise terminate unsuccessfully.

4.5.10. Crossover.

We have designed a grouping crossover operator to work over the group-number encoding, based on the standard grouping crossover described by Falkenauer. A random selection of the markets of one parent is copied over the other, changing the codification so that none of the copied markets' shares their

code number with any of those already present in the other parent. The integrity of the copied markets is preserved while the markets of the other parent sharing BSUs with the copied markets can become invalid in terms of size, self-containment or contiguity (in this last case getting fragmented). We then apply SHA to all the individual, so that invalid fragments of markets are assigned to related, adjacent markets until all of them are valid. This procedure can also change the copied markets, initially preserved, if any of them absorbs invalid fragments.

1. Copy all the information of the first parent into the child.
2. Choose a random number r with uniform distribution between 1 and 66% of the amount of markets in the second parent.
3. Select at random r markets from the second parent and copy them into the child, changing the codification so that none of that markets share their code with any market in the offspring.
4. Check that all the markets are continuous and divide into its continuous parts those markets that are not continuous.
5. Apply SHA over all the markets of the child (repair of broken markets of the first parent).
6. Terminate successfully.

5. Results

We have tested our proposal against the method and results presented in [16], the latest evolutionary proposal for this problem. We use two cases of real territories: first the Region of Valencia, in Spain, with 541 BSUs and daily travel-to-work data derived from the Spanish Census of Population [17], which was the real-world case study analysed in the experimentation section of the cited paper; and second the whole case of the Spanish municipalities from the same census, with 8104 BSUs. The second case allows testing how the new method deals with bigger problem sizes, typical in real-world, something what was not feasible with previous EAs in the literature —our implementation of the EA proposed in [16] needed several minutes for each operation, with an estimation of millions of operations to converge—. For both study cases we have used the same parameters of the LLMA delimitation problem: $\beta_1 = 70\%$, $\beta_2 = 75\%$, $\beta_3 = 20000$ and $\beta_3 = 3500$ (these were the threshold values used in [3, 6, 14, 16]).

To allow comparison with relevant official procedures we have included also the results of applying the UK travel-to-work areas delineation method [6] and its update [30], using the same parameter values.

In each case we have built a square matrix of origin-destination commuting flows, where each cell represents W_{s_i, s_j} (the number of residents of s_i that works in s_j). For this kind of problem those matrices are always sparse (most cells have value zero), therefore memory usage and complexity of the operators can be greatly reduced if only non-null flows are coded.

5.1. Region of Valencia (Spain)

Parameters of the GGA were set in these values: size population $N_p = 10$ and new offspring per generation $N_o = 1$. The

Table 1: Comparison of results for Region of Valencia

Method		Fitness value	Number of LLMA's
1986's TTWA [6]		143.10	50
2008's TTWA [30]		136.05	49
EA [16]	Max	192.78	63
	Mean	183.27	59.84
	σ	3.70	1.81
Adaptive EA [16]	Max	190.48	62
	Mean	184.73	60.44
	σ	3.70	1.72
GEA $t_{max} = 3min$ 100 runs	Max	200.78	65
	Mean	193.61	62.79
	σ	2.09	0.81
GEA $t_{max} = 10min$ 100 runs	Max	200.42	65
	Mean	195.57	63.43
	σ	1.96	0.75
GEA $t_{max} = 30min$ 70 runs	Max	203.45	67
	Mean	198.18	64.37
	σ	2.62	0.91

probabilities of application of the operators were uniform (as noted previously, crossover and mutations are evenly treated in a single phase). The condition of termination is maximum elapsed time $t_{max} = 1800s$. These are the same EA parameters employed in [16]. We present results also for $t_{max} = 180s$ and $t_{max} = 600s$, from independent runs of the algorithm. All the tests over the Region of Valencia have been run in a PC with CPU AMD Athlon II x2 2GHz and 2GB of RAM. The characteristics of the computers used by the authors of [16] were not published.

The new set of operators dramatically improves the results obtained in [16] (Table 1), even when the process is allowed to run for only 10% of the time given to previous EAs. It must be noted that the solutions provided by the TTWA's methods and the previous EAs are usually non contiguous, another positive aspect of our new grouping operators.

Fig. 3 plots the fitness evolution over generations for the first 650k generations (164.5 s. on average). Plotting over seconds causes little changes to these curves. The figure shows that the new operators produce an evolution quite faster both in terms of absolute increase of fitness per generation (needing less generations to reach a better solution) and relative generations to converge (on average, the new method needed 10k generations to get 95% of its maximum final fitness, and 165k generations to reach 99% of that value; the referenced method needed 51k and 250k generations respectively), and the final solution are better in any statistic (the average fitness value with runs of 3 minutes is higher than the best solution presented in [16] with runs of 30 minutes, and our best solution has a fitness 5.5% higher than that).

The previous EAs against which we compare our results [16]

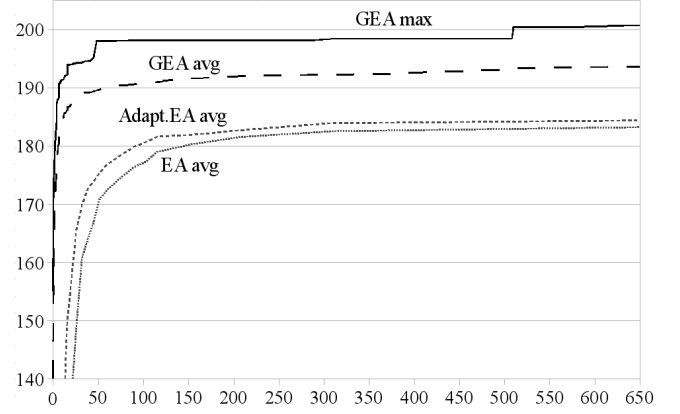


Figure 3: Fitness per generations (thousands)

did not employ informed solutions in the initial population, causing that all the evolutions started in the minimum possible fitness value (all the territory of the problem as a market), that for the Region of Valencia was 0.27. In this exercise we have successfully used the stochastic agglomerative algorithm to produce valid solutions in the initial population, starting from fitness around 145. This can be noticed in Fig. 3. Thus, the new method skips a part of the initial evolution that the previous one was bound to calculate, something that could be considered as an advantage for our method biasing the comparison. This can however be discarded since the result of applying the SHA heuristic over a whole territory is precisely one of those informed solutions with which we seed the initial population, and it is the only applicable operator when there is only one market. Time consumed by the generation of the initial population is taken into account in the results of both methods.

Our implementation of the algorithms described in [16] gave quite similar results to those published in that paper, although slightly worse (what can be consequence of a slower machine) but not statistically significant. The average number of generations performed in 30 minutes with that implementation was 7.2 millions, while the method presented in this paper performed on average 7.1 millions of generations in the same time. Therefore, it seems that our operators are not on average faster, in terms of operations per second, than those of the previous EAs. So we can credit merit for the great increase in performance not to a reduction of complexity due to the introduction of the spatial contiguity constraint into the mutations mechanics, but to the increase of the percentage of valid (and useful) individuals by the operators, and also to the way these mutations operate, individually and in cooperation with the rest of them, consistently being able to perform a better exploration and exploitation of the search space.

From Table 1 it is also clear that our proposal manages to identify a larger number of LLMA's, something considered as desirable in the policy-making context as stated in section 2.2.

Table 2 summarises some statistics about the operators: the first two columns shows valid non-repeated individuals and individuals that improve their (best) parent as a percentage of total applications of the respective operator; third column displays

Table 2: Operators’ statistics, Region of Valencia

Oper.	Valid indiv.	Impr. indiv.	Accum. fitness	CPU t. (s)
cross.	59,66%	28,71%	24,17%	22,29%
mut.R	64,74%	4,42%	40,54%	16,54%
mut.N	1,23%	0,91%	10,80%	13,66%
mut.D	84,44%	0,18%	1,13%	10,83%
mut.E	48,17%	2,05%	3,18%	4,25%
mut.I	69,58%	3,85%	7,31%	14,47%
mut.S	45,23%	1,80%	3,17%	4,26%
mut.A	26,88%	1,36%	2,82%	3,70%
mut.M	90,82%	3,49%	5,67%	7,64%
mut.X	4,65%	0,48%	1,20%	2,35%

Table 3: Comparison of results for Spain

		Fitness value	Number of LLMA's
TTWA 1986's method		14304.93	462
TTWA 2008's method		13547.21	458
GEA	Best	23286.45	659
$t_{max} = 16h$	Mean	23036.15	657
5 runs	σ	137.03	2.0

Table 4: Operators’ statistics, Spain

Oper.	Valid indiv.	Impr. indiv.	Accum. fitness	CPU t. (s)
cross.	67,46%	31,78%	46,70%	40,61%
mut.R	69,51%	1,25%	15,39%	16,42%
mut.N	0,22%	0,10%	3,20%	4,62%
mut.D	83,50%	0,03%	0,58%	10,01%
mut.E	48,47%	0,95%	5,20%	3,85%
mut.I	75,83%	2,07%	11,23%	5,46%
mut.S	47,66%	0,74%	4,44%	3,92%
mut.A	33,91%	0,52%	3,61%	2,68%
mut.M	99,41%	0,51%	7,48%	10,95%
mut.X	10,74%	0,27%	2,16%	1,47%

the accumulated increase in fitness over (best) parent as a percentage of total accumulated fitness achieved by all the operators; and the last column shows CPU time as a percentage of total time employed by all the operators. Compared to the same statistics about the operators in [16] it seems clear that our new operators produce a much higher percentage of valid individuals, and individuals that improve their parent’s fitness. This is consequent with the results obtained and the discussion above.

5.2. Whole territory of Spain

In this case the parameters were set as in the previous case study except for the finalisation time, set to $t_{max} = 16h$ (a higher value for a much bigger problem).

The set of operators we propose in this paper is able to produce very good results in reasonable time when applied to large problems, as is the case of Spain’s territory (Table 3), something that was impractical making use of the proposals described in the previous EA in literature —in our tests with the implementation of the algorithms in [16] with the case of Spain, it only managed to complete 7 mutations and obtain 2 feasible solutions after one hour of computation, while our method generated more than half a million individuals in the same time.

Regarding the operators performance in this case, table 4 shows the same statistics about the operators in the tests conducted over the case study of Spain. There are noticeable differences in performance compared to the statistics obtained working with the Region of Valencia, for example the fitness per second (accumulated fitness over CPU time) contributed by the operator R is much lesser here, while mutation I greatly improves in that aspect. Differences in the performance of the operators can be noticed when comparing the statistics at different stages of the evolution in the same problem. All this is expectable given the huge difference in size of the two problems and that the complexities of the different operators depends differently on the number or markets in the solution and/or the number of BSUs within markets, and also because a bigger problem will probably need more search steps to achieve the optimal value or the stagnation.

6. Conclusions and current works

The delineation of functional areas is a very relevant problem in the domains of Economics and Geography, among others. Setting appropriate boundaries between highly self-contained and cohesive sets of basic spatial units largely influences the quality of policy making at local levels in fields such as housing, transportation or labour markets.

Being inspired by previous works that tackled the Local Labour Market Areas (LLMAs, a typical kind of functional areas) delimitation through a multi-operator evolutionary algorithm [14, 16], we have presented a new set of group-based operators carefully designed to deal with that and other related problems. Among our design principles, opposed to the inspiring work, we have included: to produce as few infeasible solutions as possible to narrow the search, to incorporate stochastic (but not optimum) knowledge into all the operators to further accelerate convergence, and to consider any constraint of the problem into the operators as well as in the whole optimization process. All of these three principles were discarded by the authors of the inspiring work as it is arguable that their application could lead to an insufficient exploration of the search space, what could systematically avoid reaching optimal solutions. However, the results contradict these initial concerns since they outperform the best solutions from previous evolutionary algorithms applied to the problem of LLMA delimitation, in terms of both fitness of the better solution and average fitness. Moreover, the experimentation showed that the speed of the new method is at least one order of magnitude higher in the case study of Region of Valencia (541 BSUs), and much faster in the case of Spain (8108 BSUs) where the previous EAs

were unable to solve in reasonable time. These results prove that introducing the contiguity requirement has not negatively constrained the search ability of the new operators.

We have also considered the possibility of allowing invalid solutions to remain in the population and mate with valid parents, but that would imply major changes in the general method and in the operators themselves, since all of them are designed under the premise that any market in a solution *a priori* meets the constraints—and doing so allows the procedure to save many costly checks of constraints and fitness. Moreover, invalid solutions (always in terms of constraints fulfilment) could have higher fitness than valid ones, leading to a stagnation in a population full of good-but-invalid solutions, if they are not penalised somehow. So the adequacy of that idea deserves a more extensive consideration.

The contiguity constraint introduced in the new set of operators seems to play an important role in the speed up, since it naturally introduces the concept of contiguity in the behaviour of the operators, allowing to consider only adjacent markets or BSUs in the operations when selecting them instead of the whole set of them or the functional neighbours (a BSU tends to have many more flows than adjacent neighbours). We thought that this would reduce the complexity of many subroutines frequently used in the method, noticeably accelerating the number of operations per second in comparison to the same operators without the constraint, but the experimentation showed that it has had slight impact on that aspect when dealing with small size problems (Region of Valencia case). Thus the main boost in performance in those cases is due to the higher percentage of valid solutions produced by the operators. In the case of Spain the operators of our proposal were much faster than those of earlier EAs, so we can not discard the impact of the contiguity constraint in the method's complexity.

We were reluctant to include that requirement in the operators, because it is not used in the greedy TTWA's method [6, 30], and because the previous EAs and the TTWA's method were able to produce almost continuous solutions without including the information on adjacency. However, the results prove that our reluctance here was wrong. Moreover, the resulting regionalisations made by this new method are continuous without the need of applying a final procedure to fix discontinuities [4] that could make part of the effort of producing a good but discontinuous solution useless. The inclusion of this constraint seems to have resulted in only good effects, considerable narrowing the search space without excluding good solutions or hindering the exploration.

In the cited EAs with which we compare our proposal, some operators produced totally random changes while other made optimal ones (optimal in the local search of the operator, but not necessarily leading to the global optimal solution) and some other were informed (probability proportional to some attribute or relationship), and all of them were on the whole more *ad hoc*, specialised in the particular problem itself. In this work all the operators have been made informed (but still stochastic), and it has been tried to exhaustively cover all the possible operations that could facilitate the search in the exploration or the exploitation in a GPP context, even those that intuitively seemed less

adequate for our particular problem. So maybe performance of the method is also higher due to a more versatile search strategy, but still could happen that smaller subsets of the presented operators perform better in the reported problem. Nevertheless, a proper study of the specific performance of each operator is out of the scope of this paper.

The new operators presented, although being designed for the problem of LLMA's delimitation are directly applicable to the delimitation of other functional areas as HMAs [1] and FTAs [25] subject to the availability of interaction matrices (such as commuting, migrations, freight and passengers flows or connections, movements of mobile phones between cells, etc.). The proposed operators are simple and general, and the whole EA is able to deal with large instances of FADPs. Therefore, a good approach to many other graph and network partitioning problems—such as sparse matrix product, load balancing, timetabling, vehicle routing problem, etc.—could be the adaptation of the presented method by choosing proper fitness function and constraints, and adjusting the heuristics in the operators and the repair/construct routine.

We are currently adapting the algorithm to work in parallel with island and cellular models, in order to enhance the global search procedure of a single run as well as to improve the parallelisation. This could lead to a significant improvement in quality and variance of the results, specially in large instances.

We are working as well on an adaptive version that will take advantage of the fact that each operator is more helpful in specific stages of the search, or under certain conditions of the individual(s) over which the operator is applied. As happened with the EA presented in [16], the data about the variable efficiency of the operators point to a possible increase of performance of the whole method by fine-tuning or more likely (self-)adapting the probabilities of application of each operator depending on its outcome or the stage of the evolution, so that each operator is used when its contribution to the optimisation process can be higher.

Finally, a multiobjective version of the procedure will be worthy of consideration, since its ability to present a set of variate good solutions (the Pareto front) could facilitate or even allow to skip the parameter settings of the FADP by the practitioners. Alternative fitness functions could assess cohesiveness, geographic compactness and/or size homogeneity.

Acknowledgements.

This work was supported by the Spanish Ministry of Science and Innovation and the European Regional Development Fund (ERDF), project SEJ2007-67767-C04-02.

References

- [1] Brown, P. J. B., & Hincks, S. (2008). A Framework for Housing Market Area Delineation: Principles and Application. *Urban Studies* 45(11), 2225–2247.
- [2] Casado-Díaz, J. M. (2000). Local Labour Market Areas in Spain: A Case Study. *Regional Studies* 34(9), 843–856.
- [3] Casado-Díaz, J. M., & Coombes, M. G. (2005). The Delineation of 21st Century Local Labour Market Areas (LLMAs). In: *Proceedings of the 8th Nectar Conference*, Las Palmas de Gran Canaria, Spain, 2005, unpublished.

- [4] Casado-Díaz, J. M., Martínez-Bernabeu, L., & Flórez-Revuelta, F. (2010). "Los mercados locales de trabajo españoles. Una aplicación del nuevo procedimiento británico". In: J. M. Albertos and J. M. Fera (eds.): *La ciudad real en España: procesos urbanos y metropolitanos en la primera década del siglo XXI*, Madrid: Thomson-Civitas, 2010, 275–313.
- [5] CLG (2010). Functional Economic Market Areas: An economic note. London: Communities and Local Government.
- [6] Coombes, M. G., Green, A. E., & Openshaw, S. (1986). An efficient algorithm to generate official statistical reporting areas: the case of the 1984 Travel-to-Work Areas revision in Britain. *Journal of the Operational Research Society* 37, 943–953.
- [7] Coombes, M. G. (2009). Functional Economic Areas. In: *Refreshing the West Midlands Evidence Base: a State of the Region Report*. Birmingham: West Midlands Regional Observatory.
- [8] Datta, D., Figueira, J. R., Fonseca, C. M., & Tavares-Pereira, F. (2008). Graph Partitioning Through a Multi-Objective Evolutionary Algorithm: A Preliminary Study. In: GECCO'08, Atlanta, USA, 2008, ACM.
- [9] EUROSTAT & Coombes, M. G. (1992). Étude sur les zones d'emploi. Document E/LOC/20. Office for Official Publications of the European Communities, Luxembourg.
- [10] Falkenauer, E. (1998). *Genetic Algorithms and Grouping Problems*, John Wiley & Sons, 1998.
- [11] Feng, Z. (2008). Fuzziness of Travel to Work Areas. *Regional Studies* 43(5), 707–720.
- [12] Fjallstrom, P. (1998). Algorithms for Graph Partitioning: A Survey. *Linköping Electronic Articles in Computer and Information Science* 3(10).
- [13] Flórez-Revuelta, F., J.M. Casado-Díaz & Martínez-Bernabeu, L. (2007). Specific crossover and mutation operators for a grouping problem based on interaction data in a regional science context. In: *IEEE Congress on Evolutionary Computation*, Singapore, 378–385.
- [14] Flórez-Revuelta, F., J.M. Casado-Díaz & Martínez-Bernabeu, L. (2008). An Evolutionary Approach to the Delineation of Functional Areas Based on Travel-to-work Flows. *International Journal of Automation and Computing* 05(1), 10–21.
- [15] Flórez-Revuelta, F., J.M. Casado-Díaz, Martínez-Bernabeu, L. & Gómez-Hernández, R. (2008). A Memetic Algorithm for the Delineation of Local Labour Markets. In: *Parallel Problem Solving from Nature*, Dortmund, 1011–1020.
- [16] Flórez-Revuelta, F., J.M. Casado-Díaz & Martínez-Bernabeu, L. (2009). Adaptive Evolutionary Algorithms for the Delineation of Local Labour Markets. In: *IEEE Congress on Evolutionary Computation*, Trondheim, 2354–2360.
- [17] Instituto Nacional de Estadística (2005). Censo de población 2001, <http://www.ine.es>.
- [18] ISTAT (2005). I sistemi locali del lavoro 2001. Available: <http://www.istat.it>.
- [19] James, T. L., Brown, E. C., & Keeling, K. B. (2007). A hybrid grouping genetic algorithm for the cell formation problem. *Computers & Operations Research* 34, 2059–2079.
- [20] James, T. L., Vroblefski, M., & Nottingham, Q. (2007). A hybrid grouping genetic algorithm for the registration area planning problem. *Computer Communications* 30, 2180–2190.
- [21] Johnson, K.P. & Kort, J.R. (2004). Redefinition of the BEA Economic Areas. *Survey of Current Business* November 2004, 68–75.
- [22] Jones, D. A., & Beltramo, M. A. (1991). Solving partitioning problems with genetic algorithms. In: *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, 442–449.
- [23] De Jong, K. A. (2006). *Evolutionary Computation: a unified approach*. Cambridge, MA: MIT Press.
- [24] Krasnogor, N., & Gustafson, S. (2004). A Study on the use of the "self-generation" in memetic algorithms. *Natural Computing* 3, 53–76.
- [25] Krygsman, S., de Jong, T., & Nel, J. (2009). Functional transport regions in South Africa: An examination of national commuter data. In: *Proceedings of the 28th Southern African Transport Conference*.
- [26] Lewis, R., & Paechter, B. (2006). Finding feasible timetables using group-based operators. *IEEE Transactions on Evolutionary Computation* 11(3), 397–413.
- [27] Limbourg, S., & Jourquin, B. (2010). Location and market area of rail-road terminals. ORBEL 24.
- [28] Masser, I., & Brown, P. J. B. (1975). Hierarchical aggregation procedures for interaction data. *Environment and Planning A* 7(5), 509–523.
- [29] Noktehdan, A., Karimi, B., & Kashan, A. H. (2010). A differential evolution algorithm for the manufacturing cell formation problem using group based operators. *Expert Systems with Applications* 37, 4822–4829.
- [30] ONS, Coombes, M. G., & Bond, S. (2008). Travel-To-Work Areas: the 2007 review. Office for National Statistics, London.
- [31] Pankratz, G. (2005). A Grouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows. *Operations Research Spectrum* 27, 21–41.
- [32] Papps, K., & Newell, J. O. (2002). Identifying Functional Labour Market Areas in New Zealand: A Reconnaissance Study Using Travel-to-Work Data. Institute for the Study of Labor (IZA), Bonn, Discussion Paper No. 443.
- [33] Rasheed, K., Ni, X., & Vattam, S. (2005). Methods for Using Surrogate Models to Speed Up Genetic Algorithm Optimization: Informed operators and Genetic Engineering. In: Y. Jin (ed.) *Knowledge Incorporation in Evolutionary Computation. Studies in Fuzziness and Soft Computing* 167, 103–122.
- [34] Rota, G. C. (1964). The Number of Partitions of a Set. *The American Mathematical Monthly* 71(5), 498–504.
- [35] Sandou, G., Font, S., Tebbani, S., Huret, A., & Mondon, C. (2008). Enhanced Genetic Algorithm with Guarantee of Feasibility for the Unit Commitment Problem. In: N. Monmarché et al. (eds.): *EA 2007, LNCS 4926*, 291–302.
- [36] Singh, A., & Baghel, A. S. (2009). A new grouping genetic algorithm approach to the multiple traveling salesman problem. *Soft Computing* 13, 95–101.
- [37] Strancar, J., Koklic, T., Arcov, Z., Filipic, B., Stopar, D., & Hemminga, M. A. (2005). Spin Laber EPR-Based Characterization of Biosystem Complexity. *Journal of Chemical Information and Modeling* 45, 394–406.
- [38] Watts, M. (2004). Local Labour Markets in New South Wales: Fact or Fiction? In: Carlson, E. (ed.) *A Future that Works*. Callaghan, N.S.W: University of New Castle, Centre of Full Employment and Equity, 461–472.
- [39] Watts, M. (2009). Rules versus Hierarchy: An Application of Fuzzy Set Theory to the Assessment of Spatial Grouping Techniques. M. Kolehmainen et al. (eds.): *ICANNGA 2009, LNCS 5495*, 517–526.
- [40] Welsh Assembly Government (2006). Local Housing Market Assessment Guide.
- [41] Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1, 67–82.