



A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding

Giuseppe Aiello, Giada La Scalia*, Mario Enea

Dipartimento di Tecnologia Meccanica, Produzione ed Ingegneria Gestionale, University of Palermo, Italy

ARTICLE INFO

Keywords:

Facility layout problems
Multi objective genetic algorithm
Slicing structure

ABSTRACT

This paper proposes a new multi objective genetic algorithm (MOGA) for solving unequal area facility layout problems (UA-FLPs). The genetic algorithm suggested is based upon the slicing structure where the relative locations of the facilities on the floor are represented by a location matrix encoded in two chromosomes. A block layout is constructed by partitioning the floor into a set of rectangular blocks using guillotine cuts satisfying the areas requirements of the departments. The procedure takes into account four objective functions (material handling costs, aspect ratio, closeness and distance requests) by means of a Pareto based evolutionary approach. The main advantage of the proposed formulation, with respect to existing referenced approaches (e.g. bay structure), is that the search space is considerably wide and the practicability of the layout designs is preserved, thus improving the quality of the solutions obtained.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

The facility layout problem (FLP) concerns the optimal placement of a set of departments with known dimensions within the facility area, in order to minimize the operating cost and maximize the system efficiency. Generally speaking, the problem of designing a physical layout involves the minimization of the material handling cost as the main objective, and previous researches demonstrate that such minimization can lead to a cost reduction of 10–30% (Tompkins et al., 1996). Since the FLP is known to be NP-Hard, in recent years, a lot of sub-optimal approaches have been developed to solve facilities layout problems (Singh & Sharma, 2006). Recent developments are mostly based on iterative meta-heuristic approaches such as Simulation Annealing (Chwif, Marcos, & Lucas, 1998; Sahin & Turkbey, 2009), Tabu Searching (Mckendall & Hakobyan, 2010), Ant System (Komarudin & Wong, 2009, 2010; Konak-Kulturel & Konak, 2010) and Genetic Algorithms (GAs) (Islir, 1998; Al Hakim, 2000; Aiello, Enea, & Galante, 2006; Diego, Santamarina, Alcaide, & Cloquell, 2009). In recent years GAs have been widely implemented to solve combinatorial optimization problems (Gero & Kazakov, 1997; Gilkinson, Rabelo, & Bush, 1995; Goldberg, 1989; Nang, 1994; Whitley, Starkweather, & Shaner, 1991). The structure of a GA involves the production of an initial population and the achievement of a sub-optimal solution via recursive operations of reproduction, crossover, and mutation (Holland, 1992; Michalewicz, 1992). Several formulations have been proposed for unequal area facility layout problems (UA-FLPs),

among others, the flexible bay structure (FBS) is currently receiving many attentions from researchers (Aiello, Enea, & Galante, 2002; Konak, Konak-Kulturel, Norman, & Smith, 2006). In FBS formulation, the placement of departments generates columns or bays with different widths and number of departments. The width of a bay is automatically adjusted according to the number of departments contained. By means of such representation, the problem complexity is reduced into determining the departments placement order and the total number of departments each bay will contain. FBS has an advantage in that the bays will become candidates for aisle structures and this facilitates users to transform the model into an actual facility design. However, in the bay structure, the floor is always divided in one direction (vertically or horizontally) into rectangular blocks (bays). An alternative decomposition scheme for facility layout design is the slicing structure. A slicing structure results from dividing an initial rectangle either in horizontal or vertical direction completely from one side to the other (so-called guillotine cut) and recursively going on with the newly generated rectangles (Scholz, Jaehn, & Junker, 2010). This procedure is recursively applied to the sub matrices produced by the decomposition until the location matrix is fully decomposed into either row or column vectors. In this paper, in order to overcome the simplifying assumptions of the bay structure, we propose to solve the facility layout problem using a genetic algorithm encoded by a slicing structure. The objective of the problem considered here is to minimize the total Material Handling Cost, i.e., the sum of rectilinear distance costs weighted by flow amounts between the centroids of the facilities. Three additional objective functions, commonly employed in the literature (Harmonosky & Tothoro, 1992; Meller & Gau, 1996) have also been considered, namely: the distance

* Corresponding author.

E-mail address: lascalgia@dtm.unipa.it (G. La Scalia).

4	3	0
1	0	6
2	7	9

Fig. 1. The localization matrix.

and the closeness requirements among the departments, and the desired aspect ratio. As a consequence a multi objective genetic algorithm (MOGA) has been developed in order to optimize such objectives simultaneously. The approach proposed refers to the class of Pareto-based approaches suggested by Goldberg (1989) and it is developed according to the framework of non-dominated sorting GA (NSGA) proposed by Srinivas and Deb (1995). Pareto based approaches involve the evolution of the Pareto front constituted by the fitness of a generic individual corresponding to each optimality criterion considered. The GAs belonging to this class generally outperform the non-Pareto based approaches (Tamaki, Kita, & Kobayashi, 1996; Zitzler & Thiele, 1999). In addition the following constraints are considered for feasibility: (1) blocks, which specify locations and shapes of departments, must not overlap each other; and (2) the size of each block must be equal to the area of its corresponding department (area constraints), (3) the aspect ratio must be within a specified range. The remainder of this paper is organized as follows. Section 2 gives an encoding scheme for representing solutions. Section 3 describes the genetic algorithm implemented in this study for the facility layout problem and how various parameters are selected for the algorithm. To show performance of the suggested algorithm, computational experiments are done in Section 4. Finally, Section 5 concludes the paper with a short summary and the results are reported.

2. Space partitioning method

The GA here proposed is based on the referenced slicing structure, where a solution is represented by an $r \times c$ matrix M , called location matrix, which contains information about the relative locations of the departments on the floor. In our representation, in order to obtain a uniform genetic encoding scheme, only quadratic matrices ($r=c$) are considered. Consequently, given N departments, the rank (r) of the corresponding location matrix is determined as:

$$r = \lceil \sqrt{N} \rceil \quad (1)$$

where $\lceil x \rceil$ is the ceiling function that denotes the smallest integer greater than or equal to x .

The number of elements (r^2) in the matrix is thus greater than or equal to the number of departments. When r^2 is strictly greater than N , ($r^2 - N$) dummy departments with null area are introduced. These dummy departments have null material fluxes from/to other departments and are indexed as zero.

Let m_{ij} be the value of the element of the location matrix at row i and column j in M . Then, using the aforementioned encoding scheme, the department m_{ij} is to be placed to the right of the department $m_{i-1,j}$, to the left of the department $m_{i+1,j}$, below the department $m_{i,j-1}$, and above the department $m_{i,j+1}$ on the floor. In this manner, the relative locations of all departments on the floor are determined by M . For example, Fig. 1 shows a 3×3 location matrix where there are seven (real) departments and two dummy departments.

The floor is partitioned by a guillotine cut in such a way that the area of each block is equal to the sum of areas of the departments included in the submatrix corresponding to the block. The blocks always have rectangular shapes since only either a vertical or horizontal guillotine cut is used for partitioning. Considering every

Table 1

Areas of facilities in the example problem.

Department	1	2	3	4	5	6	7	8	9	10
Area	10	6	6	4	12	6	10	10	8	18

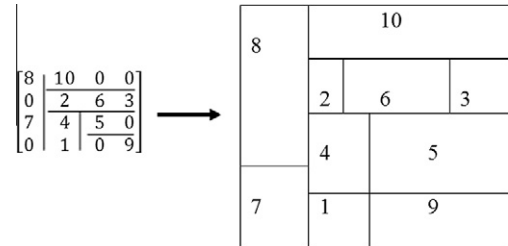


Fig. 2. Partitioning method.

possible alternative for the guillotine cuts for the decomposition, the maximum number of cuts in which the matrix can be divided is calculated by the following equation:

$$T = 2r - 3 \quad (2)$$

In our representation, in order to always obtain a quadratic matrix, at each horizontal cut follows a vertical cut and vice versa. This method is illustrated with an example problem with $N = 10$, $L = 9$ and $W = 10$. Let L and W be the length and the width of the floor, respectively. The areas of the facilities are presented in Table 1, the rank of the location matrix is $\lceil \sqrt{10} \rceil = 4$ and the maximum number of cuts is 5.

Fig. 2 shows a possible decomposition and its corresponding layout, corresponding to the following sequence of cuts: vertical, horizontal, horizontal, vertical and horizontal.

3. Genetic algorithm

Goldberg (1989) indicated that traditional optimization search methods consist of calculus-based methods, enumerative schemes, and random search algorithms. The calculus-based methods, such as the hill climbing approach, utilize the single point search to gradually approximate the maximum value of the objective function. They are therefore unable to reach the global optimum since their search space is limited to local areas. Enumerative schemes such as the dynamic programming and branch and bound approaches calculate the objective function of each point in finite (discretised infinite) search spaces and reach the optimal solution. However, they are time consuming and inefficient and practically inapplicable to NP-Hard problems. Random search algorithms improve the drawbacks of the aforementioned methods and are able to save search time. However, the solution may just be a near-optimal solution. GAs are peculiar random search algorithms which simulate the evolutionary competition and survival fitness in natural evolution. They use a parallel processing, robust, and multiple-points algorithm in searching the solution space. Such approach enhances the opportunity to achieve the global optimal solution without falling into a local optimum. In recent years multi objective genetic algorithms have been investigated. Studies on multi-objective GAs started with the pioneering work by Schaffer in 1984 (Fonseca & Fleming, 1995) with the fundamental motivation to capture multiple Pareto optimal solutions in a single run. In most of the literature multiobjective GAs are divided into non-Pareto and Pareto-based approaches (Tamaki et al., 1996; Zitzler & Thiele, 1999). The category of non-Pareto GAs, uses one population and associated selection metric for each individual objective function. Different

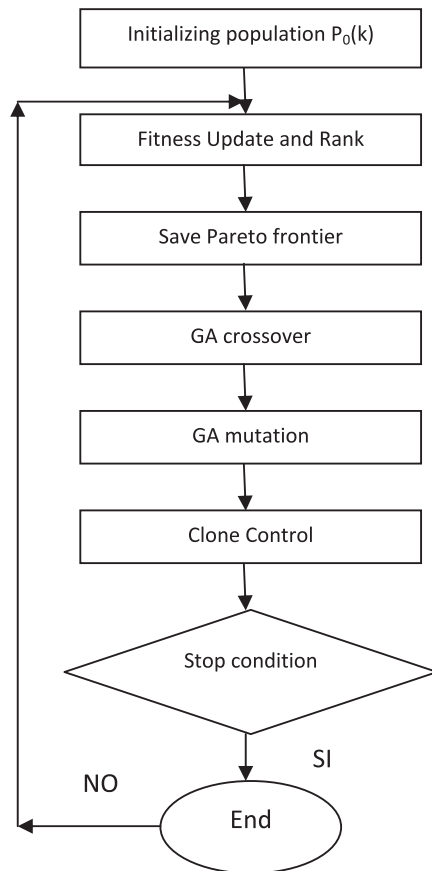


Fig. 3. Flow chart of GA optimization procedure.

objective selection methods and diversity maintenance mechanisms are tried within these approaches, however, they tend to converge to a subset of the Pareto-frontier while they leave a large part of it unexplored (Harik, 1995). The Pareto-based approaches are mainly motivated by a suggestion of a non-dominated GA by Goldberg (1989). Pareto-based approaches have been proven to be quite successful in maintaining diversity and exploring the Pareto-front in several benchmark numerical multiobjective optimization problems.

The algorithm here proposed belongs to the class of Pareto-based approaches and consists of six steps (Fig. 3): initializing the population, computing fitness values and rank, save Pareto frontier, performing crossover, mutation and clone detection, and reiterating steps 2–6 until the termination conditions are satisfied.

3.1. Genetic encoding

The encoding method in the natural system is regarded as chromosomes. In the artificial system, it is a string of genes, coded by fixed length binary values (0,1) or alphabetical characters (A,B,C) or arabic numbers (1,2,3). In this study, the strings of genes are subdivided in two segments constituted by numerical values the first, and binary variables the second. The first segment gives the department placement sequence, i.e. the elements of the location matrix read from up-left to bottom-right. The second segment shows the type of cutting, where 0 indicates a sequence of horizontal-vertical cuts, while 1 a sequence of vertical-horizontal cuts. According to this decomposition scheme, after each couple of cuts a quadratic matrix is always obtained, the number of cuts required is always maximum (see Eq. (2)), the length of the second portion of the chromosome is the same for all elements in the population and the last

I segment	II segment
4 3 0 1 0 6 2 7 9	0 1

Fig. 4. Genetic encoding.

4	3	0
1	0	6
2	7	9

Fig. 5. Location matrix and cutting scheme corresponding to the sample encoding.

cut will not be performed since the number of cuts required is always odd. This means that the number of genes in the second chromosome is strictly equal to the ceiling function of the square root of T . The strings of genes (i.e. a solution to a layout problem), which comprise these two segments, represent the whole chromosome encoding of a generic floor layout. In Fig. 4, a sample chromosome is reported.

Having regard to the considerations made in Section 2, the resulting location matrix and cutting scheme is given in Fig. 5.

3.2. Crossover

Crossover operator for the first chromosome is based upon the generic uniform-based scheme. A binary string is randomly generated. The children F1 and F2 are constructed copying the genes of a parent G1 in a child F1, in correspondence of the “1” values in the binary string. To complete F1, avoiding gene duplication (i.e. wrong coding), the missing genes are introduced according to their sequence in G2. The second son is generated oppositely. The breakpoint chromosome of F1 is chosen randomly between the breakpoint chromosomes of G1 and G2; analogously for the other child F2. For each couple of parents, hence, a couple of sons is generated. Being k the number of elements in the population, the crossover routine involves the initial partitioning in $k/2$ couples of parents randomly selected, and subsequently in the generation of an offspring population of k sons. The two populations are subsequently merged in a population of $2k$ elements, and the first k elements are extracted to constitute the new population according to the ranking procedure described below.

3.3. Mutation

Mutation operator is carried with a fixed probability at each step. Mutation operates alternatively on the first or on the second chromosome with an assigned probability. On the first chromosome, the mutation operator consists in swapping a fixed number of genes, while on the second chromosome, it consists in randomly switching “0” values to “1” and vice versa. This ensures the feasibility of the resulting chromosomes.

3.4. Objective functions

In the proposed approach, four different aspects of the block layout problem are taken into account: handling cost, adjacency requests, distance requests and aspect ratio of departments. The following objectives are therefore considered:

1. minimization of material handling cost,
2. maximization of the satisfaction of weighted adjacency,
3. maximization of the satisfaction of distance requests,
4. maximization of the satisfaction of aspect ratio requests.

Table 2

Departments areas.

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Areas	10	12	23	15	13	24	30	34	21	18	18	11	18	11	11	23	22	16	30	20

Table 3

Departments flows.

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	150	200	280	240	55	0	50	95	55	890	0	0	320	655	395	0	845	80	95
2	25	0	0	235	935	0	0	515	0	0	0	75	0	0	95	840	50	915	135	0
3	935	0	0	910	675	0	0	265	60	800	180	0	70	0	0	0	90	0	655	995
4	620	75	160	0	80	60	180	165	0	385	175	890	955	0	60	0	430	345	555	90
5	0	75	0	50	0	80	0	70	520	95	70	435	0	0	65	95	50	0	175	0
6	0	0	925	50	55	0	510	0	760	0	0	95	0	0	0	0	195	900	110	0
7	55	60	305	235	385	0	0	305	0	0	0	0	95	760	60	60	75	80	0	555
8	380	65	50	80	525	90	975	0	85	120	400	980	0	95	0	830	0	55	90	0
9	50	70	65	0	90	95	60	530	0	0	0	90	80	265	0	60	0	85	95	85
10	170	50	90	90	0	665	50	975	85	0	0	75	90	55	640	0	255	385	0	80
11	50	0	90	765	0	0	0	980	180	425	0	75	65	725	515	930	180	85	515	65
12	90	195	0	110	0	585	185	385	95	60	55	0	0	545	0	0	0	0	220	670
13	0	0	135	550	55	690	310	410	60	0	0	50	0	165	85	0	195	545	80	55
14	65	90	220	0	80	0	65	605	90	425	0	0	70	0	65	750	95	0	0	785
15	80	0	0	55	70	0	75	0	70	0	545	90	95	0	0	0	80	775	70	755
16	65	140	0	0	730	80	55	0	80	50	90	80	740	65	0	0	95	0	455	70
17	0	70	880	95	0	0	50	685	70	0	870	725	0	450	0	70	0	0	450	50
18	70	75	0	60	80	65	0	50	65	0	430	80	0	130	190	80	90	0	765	0
19	80	725	310	0	0	0	85	0	885	50	0	85	85	90	75	0	485	0	0	65
20	80	60	365	0	80	0	0	0	50	0	50	0	0	110	65	0	90	0	0	0

Table 4

Closeness request.

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6							4		4		4									
9														8						8
11							2							2						2

Table 5

Distance request.

Department	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1				8	8			8					8		8			8		
2				2	2			2					2		2			2		
3				4	4			4					4		4			4		

3.4.1. Material handling cost

The first objective function expresses the total material handling cost to be minimized:

$$\text{cost} = \sum_i \sum_j (f_{ij} c_{ij}) d_{ij} \quad (3)$$

where f_{ij} is the material flow between the departments i and j , c_{ij} is the unit cost (the cost to move one unit load one distance from department i to department j) and d_{ij} is the distance between the centres of departments using a pre-specified metric.

3.4.2. Closeness request

The second objective has been modeled by the maximization of the adjacency function:

$$\text{adjacency} = \sum_i \sum_j r_{ij} l_{ij} \quad (4)$$

where r_{ij} is the closeness rating and l_{ij} is the contact perimeter length between departments i and j .

3.4.3. Distance request

The third aspect taken into account is the distance request among some departments. A designer's objective in fact can be the separation of some departments from some others. Reasons for such requirement may be, for example, environmental issues like noise, vibration, pollution or aspects related to the security of workers or to risks of fire or explosion. The objective function, to be maximized, is expressed in this case by the following equation:

$$\text{separation} = \sum_i \sum_j s_{ij} d_{ij} \quad (5)$$

where s_{ij} is the distance rating of departments i and j and d_{ij} is the distance between the centres of departments using a pre-specified metric.

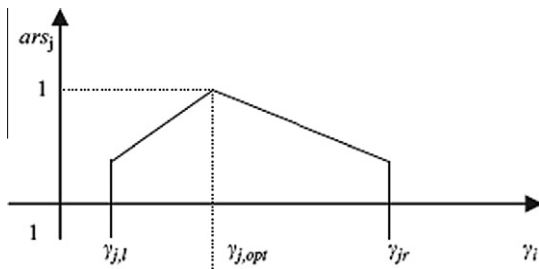
3.4.4. Aspect ratio

For each department, a proper aspect ratio is required, typically to optimize the placement of the machines inside. Let h and w be

Table 6

Best solutions obtained with the two different approaches.

Solution	Chromosome	Aspect ratio	Handling cost	Distance	Adjacency
1	0-0-0-0-0-1-20-9-6-7-3-12-14-11-15-19-17-10-8-13-2-16-4-5-18	0.896	2.77×10^5	5399	978
2	0-0-0-0-0-1-20-9-6-7-3-12-14-11-15-19-17-10-8-13-2-16-4-5-18	0.890	2.74×10^5	5403	952
3	0-0-0-0-0-1-20-9-6-7-3-12-14-11-15-17-19-10-8-4-16-2-13-5-18	0.901	2.80×10^5	5396	978
4	0-0-0-0-0-1-20-9-6-7-3-12-14-11-15-17-10-8-4-5-19-16-2-13-18	0.886	2.68×10^5	5154	989
1*	1-3-17-10-7-6-11-19-14-9-12-16-4-20-18-2-8-5-13-15	0.844	2.88×10^6	5120	937
2*	1-3-17-10-6-7-19-11-14-9-12-4-16-20-18-8-2-5-13-15	0.881	2.90×10^6	5140	911

**Fig. 6.** Satisfaction of aspect ratio request.

within a certain interval. When the value of aspect ratio is not acceptable, the score function drastically drops to 0 because manufacturing resources cannot be placed inside the department anymore. The simplest shape of such score function is given in Fig. 6, where ars_j represents the aspect ratio satisfaction function and $\gamma_{j,opt}$ represents the optimal aspect ratio. The presence of even a single department with unfeasible shape makes the whole layout unfeasible and therefore the feasibility can be evaluated by:

$$\text{aspect ratio} = \min(ars_i) \quad (7)$$

or

$$\text{aspect ratio} = \prod ars_i \quad (8)$$

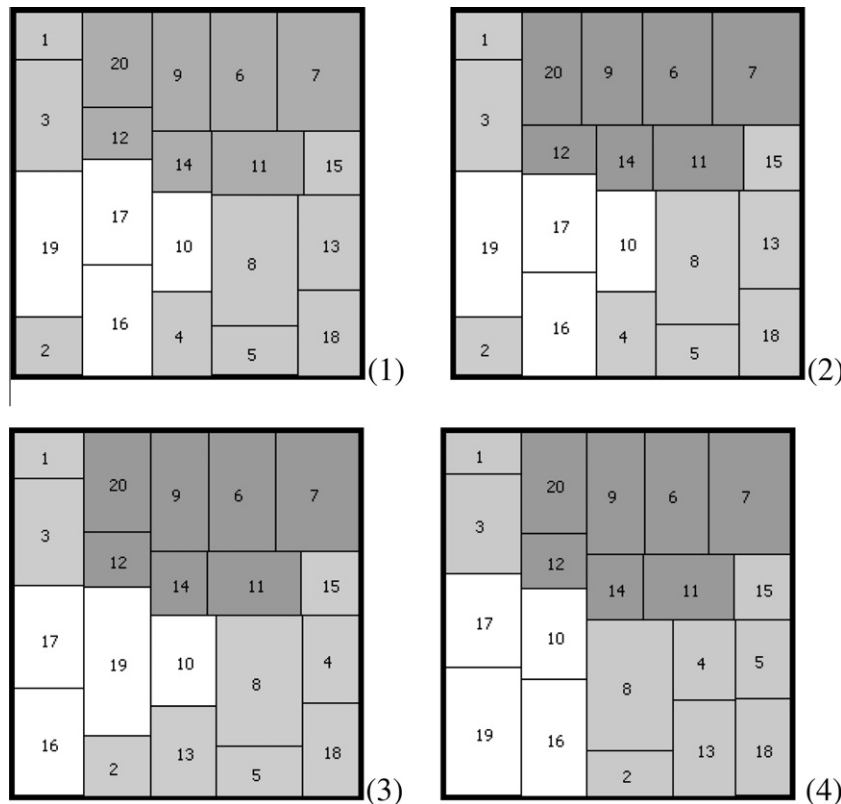
A null value implies an unfeasible solution. The multi-objective genetic algorithm, described in Section 3, requires a measure of the constraint violation in order to rank the unfeasible solutions; hence, the following parameter has been defined:

$$V = V_r + V_l \quad (9)$$

the two dimensions of the rectangle; the aspect ratio of department j is defined as:

$$\gamma_j = \frac{\max\{h_j, w_j\}}{\min\{h_j, w_j\}} \quad (6)$$

Shape ratio is still considered acceptable, with a decreasing level of satisfaction as long as it deviates from the optimal shape,

**Fig. 7.** Block layout corresponding to the optimal solutions.

where

$$V_r = \sum_{i=1}^N \gamma_i - \gamma_{ir} \quad \forall i | \gamma_i > \gamma_{ir} \quad (10)$$

$$V_l = \sum_{i=1}^N \gamma_{il} - \gamma_i \quad \forall i | \gamma_i < \gamma_{il} \quad (11)$$

For feasible solutions, the aspect ratio satisfaction is considered a further objective. In order to rank these solutions, the satisfaction level is measured by the mean of the aspect ratio satisfaction score of each department:

$$\text{aspect ratio} = \frac{\sum_{i=1}^N \text{ars}_i}{N} \quad (12)$$

In order to simplify the input data, a unique satisfaction function has been hypothesized for each department and the aspect ratio satisfaction function of each department is shown in Fig. 4. In particular defined in the interval $1 < \gamma < 2.5$ and reaching the optimal aspect ratio in correspondence to $\gamma = 1.5$.

The above objectives have been chosen because they are frequently cited in literature and involve quantitative (1) and qualitative (2,3) aspects as well as an aspect (4) that configures as a problem constraint.

3.5. Raking procedure

The ranking procedure here employed is referred to the degree of dominance. According to this approach, first non-dominated individuals within the population are identified, they are given the rank 1, and removed from the population. Then, the non-dominated individuals within the reduced population are identified and given the rank 2, followed by their removal from the population. This procedure is repeated until the whole population is ranked.

4. Numerical example

The proposed approach has been applied to a layout of 20 departments, that must be placed within a square area, equal to the sum of the departments' area given in Table 2. In order to compare the results with other referenced approaches, the input data employed are the same employed in past applications. In particular, material flows between departments, randomly generated, are given in Table 3. The flow f_{ij} in Eq. (1) is equal to the sum of f_{ij} and f_{ji} reported in the Table 3. The unit cost (the cost to move one unit load per unit distance from department i to department j) is fixed equal to 1 for each couple of departments, while the distance between departments is the Manhattan distance between their centres. The closeness requests between departments, r_{ij} parameters, have been expressed by a binary scale and are shown in Table 4. The distance requests s_{ij} are expressed again by a binary scale and reported in Table 5 and Euclidean distance has been employed. The scores have been limited to few couples of departments, avoiding incongruence with the previous objective.

5. Results and conclusions

The genetic algorithm allowed to obtain a set of feasible Pareto-optimal solutions characterized by different non-dominated values of the objective functions. Among such solutions the most representative ones are given in Table 6. The convergence of the algorithm largely depends upon the genetic parameters, and it is significantly influenced by the number of individuals in the population and by the probability of mutation. These solutions have been compared with those obtained in a previous work using the

bay structure approach (Aiello et al., 2006). All the solutions reported are dominant towards those obtained in the previous approach, thus showing the effectiveness of the genetic algorithm developed and the efficiency of the slicing structure representation. In Fig. 7, the departments having an adjacency and distance requests (Tables 4 and 5) are highlighted in light grey and dark grey areas, respectively. Solutions 1* and 2* (see table 6) are referenced solutions reported in Aiello et al. (2006).

Results obtained finally confirm the effectiveness of GA in solving the FLP even when multiple objectives are considered. Concluding, the main advantage of the proposed approach, is its capability to explore a wide space of solutions, preserving the practicability of the design, thus it configures as a good basis for the development of advanced and modern support tools to aid engineers in determining the most effective layout configurations through a real multi-objective approach.

References

- Aiello, G., Enea, M., & Galante, G. (2002). An integrated approach to the facilities and material handling system design. *International Journal of Production Research*, 40(15), 4007–4017.
- Aiello, G., Enea, M., & Galante, G. (2006). A multiobjective approach to facility layout problem by genetic search algorithm and Electre method, robotics and computer. *Integrated Manufacturing*, 22, 447–455.
- Al Hakim, L. (2000). On solving facility layout problems using genetic algorithms. *International Journal of Production Research*, 38(11), 2573–2582.
- Chwif, L., Marcos, R. P. B., & Lucas, A. M. (1998). A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36, 125–132.
- Diego, M. J. A., Santamarina, S. M. C., Alcaide, M. J., & Cloquell, B. V. A. (2009). Solving facility layout problems with strict geometric constraints using a two phase genetic algorithm. *International Journal of Production Research*, 6(15), 1679–1693.
- Fonseca, C., & Fleming, P. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3, 1–18.
- Gero, J. S., & Kazakov, V. (1997). Learning and reusing information in space layout planning problems using genetic engineering. *Artificial Intelligence in Engineering*, 11(3), 329–334.
- Gilkinson, J. C., Rabelo, L. C., & Bush, B. O. (1995). A real world scheduling problem using genetic algorithm. *Computers & Industrial Engineering*, 29, 177–181.
- Goldberg, D. E. (1989). Genetic algorithms. In *Search optimization and machine learning*. Wesley: Addison.
- Harik, G. (1995). Finding multimodal solutions using restricted tournament selection. *Sixth International Conference on Genetic Algorithms*.
- Harmonosky, C. M., & Tothoro, G. K. (1992). A multifactor plant layout methodology. *International Journal of Production Research*, 30(8), 1773–1789.
- Holland, J. H. (1992). *Adaption in natural and artificial system*. Ann Arbor, MI: University of Michigan Press.
- Islier, A. A. (1998). A genetic algorithm approach for multiple criteria facility layout design. *International Journal of Production Research*, 36(6), 1549–1569.
- Komarudin & Wong, K. Y. (2009). Applying ant system for solving unequal area facility layout problems. *European Journal of Operations Research*, 202, 730–746.
- Komarudin & Wong, K. Y. (2010). Solving facility layout problems using flexible bay structure representation and ant system algorithm. *Expert Systems with Applications*, 37, 5523–5527.
- Konak, A., Konak-Kulturel, S., Norman, B. A., & Smith, A. E. (2006). A new mixed integer formulation for optimal facility layout design. *Operations Research Letters*, 34, 660–672.
- Konak-Kulturel, S., & Konak, A. (2010). Unequal area flexible bay facility layout using ant colony optimization. *International Journal of Production Research*, 1–26. iFirst.
- Mckendall, A. R., & Hakobyan, A. (2010). Heuristic for the dynamic facility layout problem with unequal, area departments. *European Journal of Operations Research*, 201, 171–182.
- Meller, R. D., & Gau, K. Y. (1996). Facility layout objective function and robust layouts. *International Journal of Production Research*, 34(10), 2727–2742.
- Michalewicz, M. (1992). *Genetic algorithm + data structure = evolution program*. Berlin, Heidelberg, New York: Springer.
- Nang, J. H. (1994). A simple parallelising scheme of genetic algorithm on distributed, memory multiprocessors. *International Journal of High Speed Computing*, 6, 451–474.
- Sahin, R. A., & Turkbey, O. (2009). Simulated annealing algorithm to find approximate Pareto optimal solutions for the multi, objective facility layout problem. *International Journal for Manufacturing Science and Technology*, 41, 1003. 10018.
- Scholz, D., Jaehn, F., & Junker, A. (2010). Extensions to STaTS for practical applications of the facility layout problem. *European Journal of Operational Research*, 204, 463–472.

- Singh, S. P., & Sharma, R. R. K. (2006). A review of different approaches to the facility layout problem. *International Journal of Advanced Manufacturing Technology*, 30, 425–433.
- Srinivas, N., & Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2, 221–248.
- Tamaki, H., Kita, H., & Kobayashi, S. (1996). Multiobjective optimization by genetic algorithms: A review. 1996 *IEEE International Conference on Evolutionary Computation*, ICEC'96, Nagoya, Japan.
- Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. A., & Trevino, J. (1996). *Facilities planning*. New York: Wiley.
- Whitley, D., Starkweather, T., & Shaner, D. (1991). The traveling salesman and sequence scheduling: quality solutions using genetic edge recombination. In L. David (Ed.), *Handbook of genetic algorithm* (pp. 350–372). New York: Van Nostrand Reinhold.
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transaction on Evolutionary Computation*, 3, 257–271.