A GENETIC ALGORITHM FOR PLANT LAYOUT DESIGN

Salas-Morera, L.; García-Hernández, L.; Arauzo-Azofra, A.(P); Cubero-Atienza, A.J.; Redel-Macías, M.D.

Abstract

A genetic algorithm for designing plant layouts in industries with facilities of unequal area is presented. This algorithm takes into consideration material handling costs, logical relations between spaces and the shape of each area. The algorithm is based on an own developed evaluation function and a new coding scheme of facility distribution on a plant. By tuning some parameters, this function may suit better the user needs on this kind of problems; and the coding scheme, by providing a condensed representation of most possible distributions using rectangular shapes, allows GA to reach very good solutions in a reasonable time.

Keywords: Facilities Layout Design; Genetic Algorithms; Heuristics; Production.

Resumen

Se presenta un algoritmo genético para el diseño de distribución en planta en industrias con instalaciones de área desigual, considerando: el coste de flujo de materiales, las relaciones lógicas entre los espacios y la forma asignada a cada área. El algoritmo está basado en nuestra propia función de evaluación y una nueva codificación del esquema de distribución en planta. Ajustando algunos parámetros, esta función puede adaptarse mejor a las necesidades del usuario en este tipo de problemas; y el esquema de codificación, proporcionando una representación reducida de la mayoría de las distribuciones posibles usando formas rectangulares, permite alcanzar muy buenas soluciones en un tiempo razonable.

Palabras clave: Distribución en Planta; Algoritmos genéticos; Heurísticas; Producción.

1 Introduction

Plant layout design is extremely important for production efficiency (Kouvelis *et al*., 1992) in terms of both minimising material movements and ensuring logical relationships between production departments. It is estimated that between 20% and 50% of production costs can be attributed to material handling, although it is generally accepted that such costs can be reduced by at least 10% to 30% through efficient design (Tompkins *et al*., 2003). In practice, layout designs are normally based on the available area to which the layout must be adapted; the prior data are thus plant surface area and shape, which is normally rectangular. This problem is sometimes modelled as a Quadratic Assignment Problem (QAP) in which rectangular spaces are assigned to facilities in order to minimise a cost or target function. The problem may also be modelled using linear integer programming, mixed integer programming or the graph theoretic approach (Kusiak and Heragu, 1987). However, since optimal methods are limited by the number of facilities, other suboptimal methods have been developed to address more complex problems. Different techniques have been applied, such as Branch and Bound and other optimal methods (Meller *et al*., 1999; Castillo *et al*. 2005) or other based on Graph Theory (Carrie *et al*., 1978; Green and Al-Hakim, 1985; Goetschalckx, 1992; Kim and Kim, 1995; Welgama, *et al*. 1995). Researchers have recently focused more on heuristic methods, such as Tabu Search (Hasan and Osman, 1995; Chiang and Chiang, 1998; McKendall and Jaramillo, 2006), Simulated Annealing (Heragu and Alfa, 1992; Chwif et al., 1998; Baykasoĝlu and Gindy, 2001; McKendall *et al*., 2006) and Genetic Algorithms (GAs).

GAs (Goldberg, 1989) inspire in natural evolution to solve problems. Thanks to their flexibility GAs have been widely used in many different areas, from electronic circuit design to pop music production among many other examples. They can be used successfully for problems where the search space is large or not well understood, where the evaluation function is complex, or where the problem does not require a global optimum but only a *sufficiently good* solution (Mitchell, 1998), as is the case of the facility layout design problem.

A number of authors have addressed the facility layout problem using GAs. Tam (1992), using a tree structure to represent spatial distribution, took into account in the evaluation function the product of the rectilinear distance between centroids of facilities and the volume of traffic between pairs of areas, as well as the aspect ratios of the rectangles. Along similar lines, Wu and Appelton (2002) proposed a GA for optimising space layouts considering material flows, isolated spaces between different work areas, and – in all cases - the requirements of a raw-material warehouse versus a product warehouse. Lee and Lee (2002) also considered material handling costs and free space between activity areas. The design proposed by Lee *et al*. (2003) included the possibility of considering pre-existing walls or aisles, thus adding flexibility to the design. They took, as a major element in the evaluation function, the product of material flow by the distance between area centroids. Gómez *et al*. (2003), using Muther's (1973) SLP method as a starting point, considered not only material handling costs but also the proximity and distance ratios required by production logic. Balakrishnan *et al*. (2003) proposed a simple programme with an interface for effectively resolving these types of problems. For example, they allowed two departments or areas to exchange locations in order to achieve a more adequate layout. These authors used simulated annealing and GAs and, like the other authors mentioned above, material handling costs. Wang *et al*. (2005) focused on the irregular-shaped areas prompted by the use of square unit modules, but took into account area shape in the evaluation function and checked their operation using three problems with 8, 10 and 12 departments taken from the literature. Aiello *et al*. (2006) developed a combined method in which solutions were obtained using a multiobjective GA and subsequently selected using the multicriteria decision-making procedure Electre. These authors considered the costs of material flows between work area centres, the shape of assigned work areas, and an adjacency function based on plant needs. They applied the algorithm to a specific problem and proposed two solutions. Balamurugan et al. (2008) proposed a GA that minimises the cost associated with material handling and the cost of unusable space.

In this paper, our goal is to develop a new GA specialized on rectangular facility areas that improves the set of techniques available for facility layout design. The contents are organized as follows. First, we state the concrete problem to be addressed. Second, the proposed GA is described with detail. After that, its application to a problem and experiments with different parameters are described.

2 Formulation of the facility layout problem

The problem addressed in this paper is how to place *n* facility areas (or departments) in a rectangular shaped plant:

- The plant is defined in size and shape by its length and width ($L \times W$).

- Each facility area ($i \in \{1..n\}$) is defined in size by its surface (s_i).
- All facilities must fit in the plant ($\sum_{i=1}^{n} s_i \leq L \times W$).
- Allocation preferences are given by:
  - Density of material flows (for example in kg/hour, unit/hour, etc.) between each pair of departments. This is represented in a matrix (*F*) of size $n \times n$ where each value f_{ij} represents the flow of material in the sense from department *i* to department *j* (it does not have to be symmetrical).
  - Proximity preferences due to the process. For example, this may include logical organisation of the production system, the absence or presence of noise, and safety reasons, among others. This can be represented in a triangular relational matrix (*REL*) (a symmetrical matrix where we are only interested in the $n(n-1)/2$ values over the diagonal). The value rel_{ij} will measure in some way the degree at which proximity between departments *i* and *j* is desired or refused.

3 Description of the genetic algorithm

In order to apply GAs to a given problem, the following elements must be designed: a) a **coding scheme** to represent possible solutions to the problem, b) a **fitness function** that allows comparing solutions and takes into account the limitations of the problem, c) a **crossover operator** between individuals, and d) a **mutation operator** of an individual. The success of this type of algorithm depends partly on the consideration given to restrictions in the evaluation function as well as in the chosen coding scheme and in the operators used (Michalewicz *et al*. 1996).

The proposed GA is designed to solve the facility layout problem described above. In the following sections, the design of aforementioned elements is described with detail, as well as the type of GA used and the selection mechanisms.

3.1 Fitness function

The evaluation of the solutions take into account both qualitative aspects -such as closeness or distance requests between activity centres due to the logic of the production process, information flows, existence of noise or thermal environments- and quantitative aspects such as material flows. The solution evaluation function considers three aspects: minimisation of the distance covered by material, compliance with the proximity preferences, and the rectangular aspect ratio of each department.

The global evaluation function is:

$$C = (\alpha_1 \times F_1 + \alpha_2 \times F_2 + 1) \times \delta \quad (1)$$

where F_1 , F_2 , and *δ* are respectively the fitness of: material flow, proximity preferences, and aspect ratio (all described below); while α_1 and α_2 are weighting factors. Users of this algorithm can variate α_1 and α_2 to change the relative importance given to material flow and proximity preferences. They can also choose between considering or ignoring the facility aspect factor; in the latter case, the value of *δ* should be forced to 1.

The following expression is used to evaluate distance covered by material:

$$F_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} \cdot f_{ij} \tag{2}$$

where:

D_{ij} : shortest distance between the centroids of the departments, as measured by the Manhattan distance, excluding the distance covered inside the departments (as shown in figure). In other words, if two departments are adjacent to one another, their distance is zero.

f_{ij} : density of the flow of materials from department i to department j.

The following expression was used to evaluate compliance with the logical relationships between activities:

$$F_2 = \sum_{i=1}^{n} \sum_{j=i+1}^{n} E_{ij} \tag{3}$$

where:

$$E_{ij} = \begin{cases} rel_{ij} \cdot D_{ij}^2 & \text{if } rel_{ij} \geq 0 \\ \dfrac{|rel_{ij}|}{D_{ij}^2 + 10^{-6}} & \text{if } rel_{ij} < 0 \end{cases} \tag{4}$$

Figure 1: Measurement of the distance between two departments

The rel_{ij} values depend on the logical relationship existing between the pair of facilities *i* and *j*. To allow better interaction with the designer that defines these relations, we have defined the following labels according to Muther (1973): A, proximity is essential; E, proximity is highly important; I, proximity is important; O, proximity is desired; U, indifferent; and X, undesirable proximity. Each pair of facilities will be assigned one of these labels, using U as the default for the non specified. To numerically evaluate proximity preferences, each label has been translated to the following values: A=40, E=12, I=4, O=1, U=0 and X=-1. Nevertheless, they may be modified according to the characteristics of the problem.

Users of this GA may deliberately or inadvertently assign proximity needs in the REL table based on material flow, which would be redundant. Although this is not necessarily a problem when generating layouts, it should be taken into account, specially when setting weighting parameters.

Finally, a value is established for the desired aspect ratio for the areas assigned to the departments (b). The general compliance of facility layout with desired aspect ratio is valued as follows:

$$\delta = \sum_{i=1}^{n} \frac{\sqrt{(b - x_i)^2}}{2} \tag{5}$$

where:

b : desired aspect ratio.

x_i : aspect ratio of department i (largest side divided by smallest side).

3.2 Coding scheme of layouts

This is a fundamental part of the GA. The coding scheme will be used to store the individuals that evolve in a data structure and it will predispose how operators modify solutions. The goal of the coding scheme proposed is to represent most possible solutions with rectangular facilities in a compressed representation. As a smaller representation leads to a reduced search space, it is expected to reach better solutions faster.

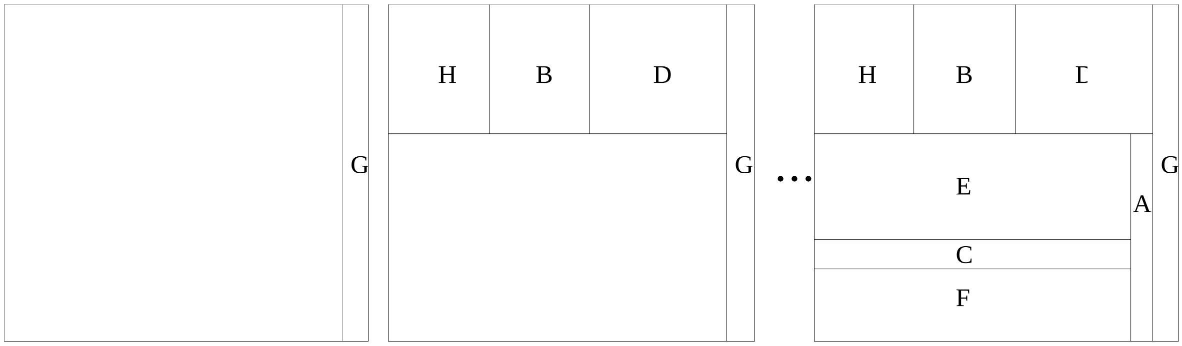
The scheme proposed codes the possible solutions of the problem using three *n*-tuples:

- First tuple sets the order in which the departments are arranged over the available surface area. The elements are labels representing each of the *n* facility areas. Each area can only appear once, so the domain is all the permutations of labels.
- Second tuple makes groups of facilities. The splits of groups are marked by the changes of value in second tuple. Formally, those labels that are together on first tuple and have the same value on second tuple belong to the same group and, transitively, if two areas belong to the same group and one of them belong to the same group that another area, all of these areas belong to the same group. The elements of this tuple are binary values.
- Third tuple points which border will each group adhere to in the remaining space. Only the first value of each group will be used. The elements of this tuple can have four values: 0, left border areas will be placed vertically; 1, right border placed vertically; 2, bottom border placed horizontally; 3, top border placed horizontally. Where areas are placed vertically the order is bottom-up and, where horizontally, left to right.

G	H	B	D	F	C	E	A
1	0	0	0	1	1	1	0
1	3	0	1	2	0	3	1

Table 1: Coding scheme of a solution.

a) 1st group placed b) 2nd group placed c) All facilities placed



An example of a coded solution is shown in table 2, with its corresponding graphical representation shown on figure 2. The facilities are distributed in the plant in the order established by first tuple (G, H, B, D, F, C, E, A). The groups marked in second tuple are: {G}, {H, B, D}, {F, C, E}, and {A}. To assign space to facilities in the plant, first group (G facility) is placed on the right side, as indicated by the 1 value on third tuple. The left border of G is placed where required to reserve the required area (s_G) for this facility (see figure 2.a). Using the remaining space, next group ({H,B,D}) is placed, according to third tuple value (3), on top with the three facilities distributed horizontally in order. The lower border is determined by the sum of required areas of the three facilities, while the inner borders are placed according to the required space of each one (figure 2.b). The last two groups are placed on the same way leading to the final plant distribution of figure 2.c.

3.3 Crossover method

Crossover is performed by interchanging one of the three tuples completely between two individuals. The tuple to be interchanged is chosen randomly with equal probability for the three tuples.

3.4 Mutation mechanism

Once an individual is selected for mutation, one of its tuples is chosen randomly with equal probability. The mutation mechanism depends on selected tuple. If first tuple is selected, two different positions are chosen randomly and its values are interchanged. When second tuple gets selected, the mutation changes the bit value of one random position. If third tuple is selected, the mutation changes the orientation value of one random position. The mutation always changes some value, never leaves previous value of group neither orientation.

3.5 Selection and replacement methods

The proposed GA follows the generational approach. In this approach, all individuals are evolved on each iteration. Various selection methods have been implemented in order to allow testing to find the most appropriate method. These methods are roulette wheel, sigma scaling, and rank selection (Mitchell, 1998). The option of applying elitism to keep the best individual between one generation and the next has been considered.

4 Computational results

In this section, the application of the GA to the well known case taken from Aiello et al. (2006) is described.

Parameters	**Considered values**
Mutation probability	0, 50%, 100%
Crossover probability	0, 50%, 100%
Selection method	Sigma scaling
Elitism	Yes, No

Table 2: GA parameter values considered in experiment

In order to apply the GA, some parameter values should be chosen. A population of 500 individuals is used and 10000 iterations are performed on each run. Complete experiment design with all values of parameters shown in table 2 is performed. This leads to 54 tests. Each test is repeated three times to avoid discarding good parameters by random effect. Once parameter values are chosen (mutation probability 50% and 100%, crossover probability 50% and 100%, sigma scaling and elitism included), GA is run three times with best values changing random seed and best solution ever found is reported. The weights α_1 and α_2 of the evaluation function are fixed according to each case requirements.

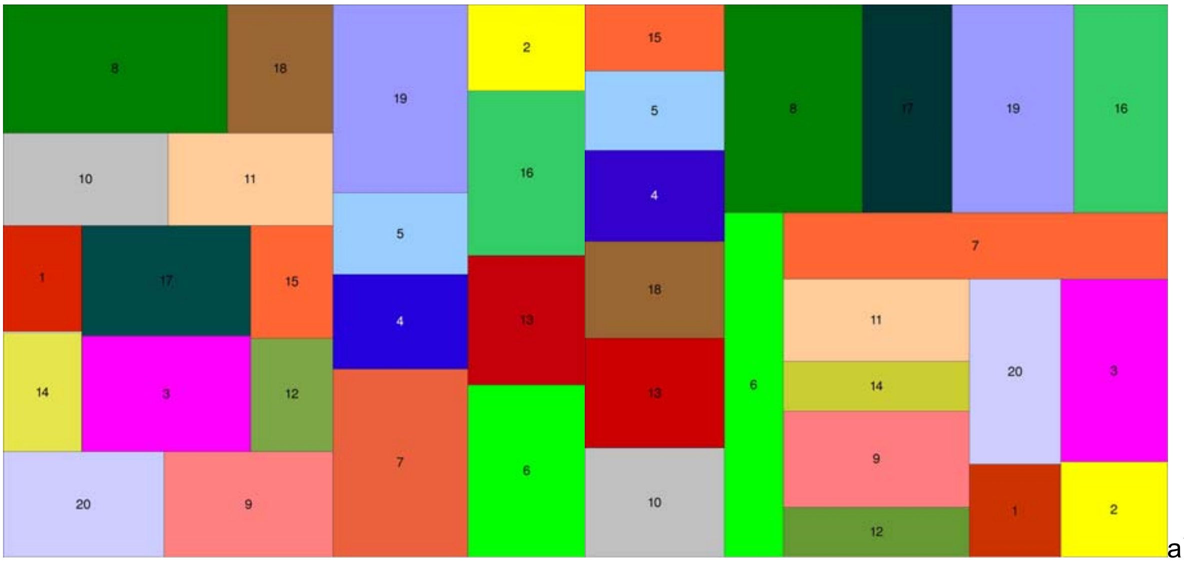
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1 (100)	---	U	U	X	X	U	U	X	U	U	U	U	X	U	X	U	U	X	U	U
2 (120)		---	U	X	X	U	U	X	U	U	U	U	X	U	X	U	U	X	U	U
3 (230)			---	X	X	U	U	X	U	U	U	U	X	U	X	U	U	X	U	U
4 (150)				---	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
5 (130)					---	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
6 (240)						---	E	U	E	U	E	U	U	U	U	U	U	U	U	U
7 (300)							---	U	U	U	I	U	U	U	U	U	U	U	U	U
8 (340)								---	U	U	U	U	U	U	U	U	U	U	U	U
9 (210)									---	U	U	A	U	A	U	U	U	U	U	A
10 (180)										---	U	U	U	U	U	U	U	U	U	U
11 (180)											---	U	U	I	U	U	U	U	U	I
12 (110)												---	U	U	U	U	U	U	U	U
13 (180)													---	U	U	U	U	U	U	U
14 (110)														---	U	U	U	U	U	U
15 (110)															---	U	U	U	U	U
16 (230)																---	U	U	U	U
17 (220)																	---	U	U	U
18 (160)																		---	U	U
19 (300)																			---	U
20 (200)																				---

Table 3: Ratios between activities (surface areas in brackets).

The case taken from Aiello *et al*. (2006), addressed the problem of finding layouts to minimise material handling costs and to maximise an adjacency function that qualitatively expressed closeness requests and distance requests. It also took into account the shape of the activities, considering those with an aspect ratio of 1.5. The problem consisted of 20 activity areas or centres with strongly inter-related movements of materials between most of these centres. In order to adapt this to our algorithm, the values shown in Table 3 were used to evaluate compliance with the logical relationships. To solve these problems using the GA, the mentioned test strategy was used with the results shown in Table 4. Moreover, figure 3 shows the two best solutions found with both search strategies: a) shows how compliance with closeness requests and distance requests was partly traded off in order to obtain the lowest possible movement of materials; while b), since the evaluation function did not consider material flows, shows that compliance with the closeness and distance requests was more weighted and that all these requests were satisfied. The intense movements of materials between all the areas made it very difficult to obtain low values in this part of the evaluation function, in spite of this aspect being marked as a priority; therefore, designers must once again consider which is the best solution for satisfying the actual conditions of the problem, taking into consideration constructive aspects.

Repetition	Mutation	Crossover	α1	α2	Iteration	F_1	F_2	δ	C
1	100	100	0.5	0.5	8668	47746.1	1581590.0	1.45	1180100
2	100	100	0.5	0.5	9795	53546.0	1598530.0	1.72	1421790
3	100	100	0.5	0.5	8378	97385.3	1570880.0	1.35	1129500
1	100	50	0.5	0.5	9774	**25227.6**	1460650.0	1.63	1216110
2	100	50	0.5	0.5	7032	127681.0	1411620.0	1.17	897587
3	100	50	0.5	0.5	8255	185497.0	1800500.0	1.33	1315220
1	50	100	0.5	0.5	2831	73568.2	1488140.0	1.15	900356
2	50	100	0.5	0.5	9824	86302.3	1538800.0	1.15	935820
3*	50	100	0.5	0.5	8056	52254.8	**1362820.0**	**1.06**	**750242**
1	50	50	0.5	0.5	4863	29490.8	1551820.0	1.59	1253590
2	50	50	0.5	0.5	7810	85858.5	1538150.0	1.60	1299010
3	50	50	0.5	0.5	9692	72504.3	1526880.0	1.70	1357210
1	100	100	1	0	7461	0.5	1618650.0	23.95	35.81
2	100	100	1	0	6836	0.3	1723270.0	18.95	24.95
3	100	100	1	0	9440	0.3	1702430.0	15.44	19.65
1	100	50	1	0	9897	113.7	1757110.0	12.94	1483.43
2*	100	50	1	0	6972	**0.3**	1750120.0	**9.55**	**12**
3	100	50	1	0	8261	0.4	1411100.0	32.74	44.82
1	50	100	1	0	9646	0.3	1354190.0	26.34	35.08
2	50	100	1	0	5025	0.3	1750560.0	16.38	21.7
3	50	100	1	0	8183	0.3	1495820.0	18.24	23.25
1	50	50	1	0	5996	0.5	1612020.0	17.10	25.71
2	50	50	1	0	9963	0.4	1463710.0	23.05	33.07
3	50	50	1	0	9103	0,4	1485450,0	30,20	41,43

Table 4: Results of tests with different parameter values.



a) First set b) Second set

Figure 3.- Best layout found on each set of tests.

5 Conclusions

- A useful Genetic Algorithm (GA) has been proposed to design plant layouts with unequal areas. The GA can be used laying out industrial facilities.
- A new, simple and easy-to-implement way of coding possible solutions has been designed. This coding scheme, by providing a condensed representation of many possible distributions using rectangular shapes, reduces the search space and allows GA to reach very good solutions in a reasonable time. In addition, the solutions are flexible comprising useful rectangular areas that can be easily refined by the designer.
- The GA is based on an own developed evaluation function. This evaluation function considers material flow, aspect ratio of areas and adjacency preferences -such as closeness requests between activity centres, or the existence of noise or thermal environments. The function can be tuned depending on the characteristics of the problem case, making it possible to achieve good solutions over a wide range of practical scenarios.
- Good solutions have been reached in the case tested.
- As future work it may be interesting to extend the coding scheme to consider fixed locations and to fit the areas on a given non rectangular plant shape.

References

[1] Aiello, G.; Enea, M. and Galante, G., “A multi-objective approach to facility layout problem by genetic search algorithm and Electre method”, *Robotics and Computer-Integrated Manufacturing,* Vol. 22, 2006, pp.447–455.

[2] Balakrishnan, J.; Chengb, C.H. and Wongb, K.F., "FACOPT: A User Friendly FACility Layout OPTimization System", *Computers & Operations Research*, Vol 30, 2003, pp.1625-1641.

[3] Balamurugan, K.; Selladurai, V. and Ilamathi, B., "Manufacturing facilities layout design using genetic algorithm", *International Journal of Manufacturing Technology and Management*, Vol 14 (3-4), 2008, pp.461-474.

[4] Baykasoĝlu, A. and Gindy, N.; "A Simulated Annealing Algorithm for Dynamic Layout Problem", *Computers & Operations Research*, Vol 28, 2001, pp.1403-1426.

[5] Carrie, A.S.; Moore, J.M.; Roczniak, M. and Seppanen, J.J., "Graph Theory and Computer Aided Facilities Design". *Omega*. *International Journal of Management Science*, Vol. 4, Nº 6, 1978, pp.353–361.

[6] Castillo, I.; Westerlund, J; Emet, S. and Westerlund, T., "Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods", *Computers and Chemical Engineering*, Vol 30, 2005, pp.54–69.

[7] Chiang, W-C. and Chiang, C., "Intelligent Local Search Strategies for Solving Facility Layout Problems with the Quadratic Assignment Problem Formulation", *European Journal of Operational Research*, Vol 106, 1998, pp.457-488.

[8] Chwif, L.; Pereira Barretto, M.R. and Moscato, L.A., "A Solution to the Facility Layout Problem Using Simulated Annealing", *Computers in Industry*, Vol 36, 1998, pp.125–132.

[9] Goetschalckx, M., "An interactive Layout Heuristic Based on Hexagonal Adjacency Graphs", *European Journal of Operational Research*, Vol 63-2, 1992, pp.304-321.

[10] Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning". *Addison-Wesley*, New York, 1989.

[11] Gómez, A.; Fernández, Q.I.; De la Fuente García, D. and García, P.J., "Using Genetic Algorithms to Resolve Layout Problems in Facilities where there are Aisles", *International Journal of Production Economics*, Vol 84, 2003, pp.271-282.

[12] Green, R.H. and Al-Hakim, L.A.R., "A Heuristic for Facilities Layout Planning", *Omega. International Journal of Management Science*, Vol. 13, Nº 5, 1985, pp. 469-474.

[13] Hasan, M. and Osman, "Local Search Algorithms for the Maximal Planar Layout Problem", *International Transactions in Operational Research,* Vol 2, 1995, pp.89-106.

[14] Heragu, S.S. and Alfa, A.S., "Experimental Analysis of Simulated Annealing Based Algorithms for the Layout Problem", *European Journal of Operational Research,* Vol 57, 1992, pp.190-202.

[15] Kim, J-Y and Kim, Y-D. "Graph Theoretic Heuristics for Unequal-sized Facility Layout Problems", *Omega. International Journal of Management Science*, Vol 23, Nº 4, 1995, pp.391-401.

[16] Kouvelis, P.; Kurawarwala, A.A. and Gutiérrez, G.J., "Algorithms for Robust Single and Multiple Period Layout Planning for Manufacturing Systems", *European Journal of Production Research*, Vol 63, 1992, pp.287-303.

[17] Kusiak, A. and Heragu, S.S., “The facility layout problem”, *European Journal of Operational Research*, Vol 29, 1987, pp.229-251.

[18] Lee, Y.H. and Lee, M.H., “A Shape-based Block Layout Approach to Facility Layout Problems Using Hybrid Genetic Algorithm”. *Computers & Industrial Engineering*, Vol 42, 2002, pp.237-248.

[19] Lee, K.Y.; Han, S.N. and Myung, I.R., “An Improved Genetic Algorithm for Facility Layout Problems Having Inner Structure Walls and Passages”, *Computers & Operations Research*, Vol 30, 2003, pp.117-138.

[20] McKendall, A.R. and Jaramillo, J.R., “A Tabu Search Heuristic for the Dynamic Space Allocation Problem”, *Computers & Operations Research*, Vol 33, 2006, pp.768-789.

[21] McKendall, A.R.; Shang, J. and Kuppusamy, S., “Simulated Annealing Heuristics for the Dynamic Facility Layout Problem”, *Computers & Operations Research*, Vol 33, 2006 , pp.2431–2444.

[22] Meller, R.D.; Narayanan, V. and Vance, P.H., “Optimal Facility Layout Design”, *Operations Research Letters*, Vol 23, 1999, pp.117-127.

[23] Michalewicz, Z.; Dasgupta, D.; Le Richie, R. and Schoenauer. M., “Evolutionary Algorithms for Constrained Engineering Problems”, *Computers & Industrial Engineering*, Vol 30, 1996, pp.851-870.

[24] Mitchell, M., “An Introduction to Genetic Algorithms”. *The MIT Press*. 1998.

[25] Muther, R., “Systematic Layout Planning”, *Cahners Books,* 2^{d} ed, 1973.

[26] Tam, K.Y., “Genetic Algorithms, Function Optimization, and Facility Layout Design”. *European Journal of Operational Research*, Vol 63, 1992, pp.322-346.

[27] Tompkins, J.A.; White,J.A.; Bozer, Y.A. and Tanchoco, J.M.A., “Facilities Planning”, *Wiley*, 3rd ed, New York, 2003.

[28] Wang, M.J.; Hub, M.H. and Ku, M.Y., “A Solution to the Unequal Area Facilities Layout Problem by Genetic Algorithm”, *Computers in Industry*, Vol 56, 2005, pp.207–220.

[29] Welgama, P.S.; Gibson, P.R. and Al-Hakim, L.A.R., “Facilities Layout: A Knowledge-based Approach for Converting a Dual Graph into a Block Layout”, I*nternational Journal of Production Economics*, Vol 33, 1995, pp.17-30.

[30] Wu, Y and Appelton, E. “The Optimisation of Block Layout and Aisle Structure by a Genetic Algorithm”. *Computers & Industrial Engineering*, Vol 41, 2002, pp. 371-387.

Corresponding

Laura García-Hernández

Área de Proyectos de Ingeniería. Universidad de Córdoba.

Campus Universitario de Rabanales. Edificio Leonardo Da Vinci – Carretera de Madrid km. 396 – 14.071 – Córdoba - Teléfono +34 957 21 20 50 - E-mail: ir1gahel@uco.es