A REVIEW ON ENCODING SCHEMES USED BY GENETIC ALGORITHMS IN PLANT LAYOUT DESIGN

García-Hernández, L.; Araúzo-Azofra, A.[(p)]; Salas-Morera, L.

Abstract

The design of industrial plant layouts is a complex problem. In order to solve this problem, many authors have used Genetics Algorithms (GAs) with the goal of reaching an efficient plant layout design. The GAs require to define a coding scheme to represent the plant layout design as a data structure. This encoding defines the types of design that can be obtained and it influences heavily in GA ability to find good solutions. This work shows a review of encoding schemes reported in the specialized literature, identify their main characteristics classifying the general types of encoding, and propose approaches that can be applied into the encoding schemes.

Keywords: Plant Layout Design; Genetic Algorithms; Encoding Schemes; Heuristics.

Resumen

El diseño de la distribución de áreas en plantas industriales es un problema complejo. Para resolverlo, muchos autores han usado Algoritmos Genéticos (AGs) con el objetivo principal de alcanzar una diseño eficiente de la planta. Los AGs requieren definir un esquema de codificación que represente la distribución de la planta como una estructura de datos. Esta codificación define los tipos de distribución que se pueden obtener y en su fuerte influencia en la capacidad del AG para encontrar buenas soluciones. Este trabajo tiene los siguientes objetivos: presentar una revisión de los esquemas de codificación usados en la literatura, identificar sus principales características.

Palabras clave: Diseño de distribución en planta; Algoritmos Genéticos; Esquemas de codificación; Heurísticas.

1 Introduction

Facility Layout Design determines the placement of facilities or departments in a work plant with the main aim of obtaining the most effective arrangement according to some criteria or preferences and ensuring some constraints. Among others, the objectives can be: to minimize the material handling cost, to maximize the closeness relationship between each pair of facilities, to satisfy a desired aspect ratio. Plant layout design is extremely important for production efficiency [18] because it affects directly in manufacturing costs, lead times, work in process and productivity. A good placement of facilities contributes to the overall efficiency of operations and can reduce between 20% and 50% the total operating costs [35].

In order to deal with Plant Layout Design many techniques have been used. One of the most promising of them are the Genetic Algorithms (GAs). The GAs, conceived by Holland [15], are robust methods that are capable of working well in a broad spectrum of problems [13]. GA´s are a meta-heuristic search that simulates the natural evolution. In short, the procedure begins creating randomly a population of solutions. Then, simulating the biological selection, the most appropriated individuals are chosen for being recombined, in order to generate the new population.

The first stage of building a GA is to decide on a genetic representation of an individual (genotype), which must be concordant with a candidate solution in the problem (phenotype) after to apply the decoding procedure. To choose the encoding scheme is the most important and difficult part of designing a good GA[9]. The selected representation will determinate the operators that could be applied, such as crossover or mutation.

In this paper, we present a survey about the encoding schemes used by GAs in plant layout design. An analysis and classification of the coding schemes and its operators used in plant layout design is presented.

The remainder of this paper is organized as follows: First, in section 2, we present the preferences and constraints that can have into account for defining the problem. Section 3 shows the types of distribution used by the authors to design the plant layout. Section 4 explains the different purposes that it is possible to obtain with each encoding scheme. In section 5 and 6, we classify the types of encoding and its operators used in the studied works. In section 7, the conclusions are summarized.

2 Problem definition

There are numerous plant layout problems. A good representation of them are in [7]. Many techniques exists to solve this problem [19][29][26]. We will focus in the layout problem that pretending to divide a plant into different facilities or departments optimizing an objective function. Moreover, to design a plant layout, several constraints, characteristics and preferences can be considered. So that, in function of them, we can have different problems to solve.

Now, we will comment some of these characteristics which have been applied to this problem.

- Facility dimensions. The dimensions of the plant facilities can be proporcionated by the Design Manager (DM) and its can be fixed. Normally, the DM gives the total layout area and the area of each deparment that make up the plant. It is possible to have the following alternatives:

  - Equal. When all the facilities in the plant have the same height and length and width. So that, all the departments have the same size.

  - Unequal. This happens when at least a facility has different dimensions than the another facilities in the plant.

- Department shape. This characteristic determines the permitted facility shape. Which can be:

  - Regular. It is the most used in the studied works and normally is rectangular.

  - Irregular. This shape is less used due to the fact that is very difficult to handle with irregular shapes.

- Floor considerations. This constraint will determine if the problem is:

  - Single-floor. When only an unique floor is considered in the problem definition. That is to say, the facility distribution is realized in the same floor.

  - Multi-floor. When the facility planning is performed for several floors. This consideration adds difficult to plant design because more constraints (i.e. vertical distance) and/or devices must be had into account (stairs, elevators).

- Planning horizon. This constraint includes the different production requeriments that maybe exist in certain periods of time.

  - Static layout. When the problem is defined thinking that the environment is stable and it not changes.

  - Dynamic layout. In this case, the design of the manufacturing plant is carried out in order to adapte the plant to a changing environment. Usually, for solving this problem, the planning horizon is splited into periods and each of them, is represented by a static layout, so that, the dynamic plant layout will makes up for several static plants.

- Aspect ratio. Normally, this factor is given by means of an upper and lower bounds. This value is used in order to not appear strange shape in the departments. For instance, facilities with a big value of lenght and very small value for width, which proporcionates infeasible facilities.
- Objective function. This function is the essential part in the problem definition. The objetive function contains the objective or objectives that we desire to optimize. The objective function is the tool that allows us to evaluate each problem solution determinating if the solution is good or not.
- Another devices. It possible to add certain devices into the problem planning, those can be: passages or aisles, inner walls, elevators and stairs, among others.

3 Types of distribution

For solving the problem proposed, mainly, the following structures has been used:

- Bays. This approach divides the plant into a number of bay blocks, which can be fixed or variable and then, each facility is placed until to complete this bay, when it happens, the facility is put in the next bay. Also, it is possible that the bay width be equal for all of the bays that make up the layout, or be unequal. In the figure 1, an example taken from [14] of using bay blocks can see.

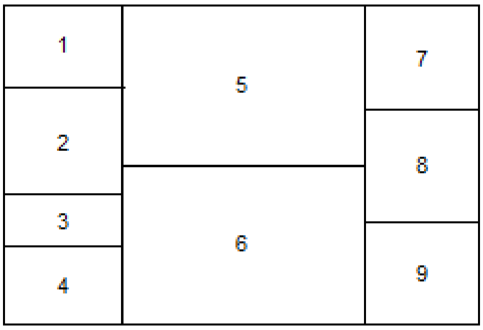

Figure 1: Bay Structure

- Tree Structure. In this case, the manufacturing plant is represented by a slicing tree structure, in which, each leaf node represents a facility in a plant and, each intern node represents the slicing operator that cuts the layout into portions or allocations. These operators can be vertical or horizontal cuts, or they can be more detailled, such as, bottom, up, rigth or left cuts. We can see the slicing tree and its plant representation in figure 2, this example is taken from [30].

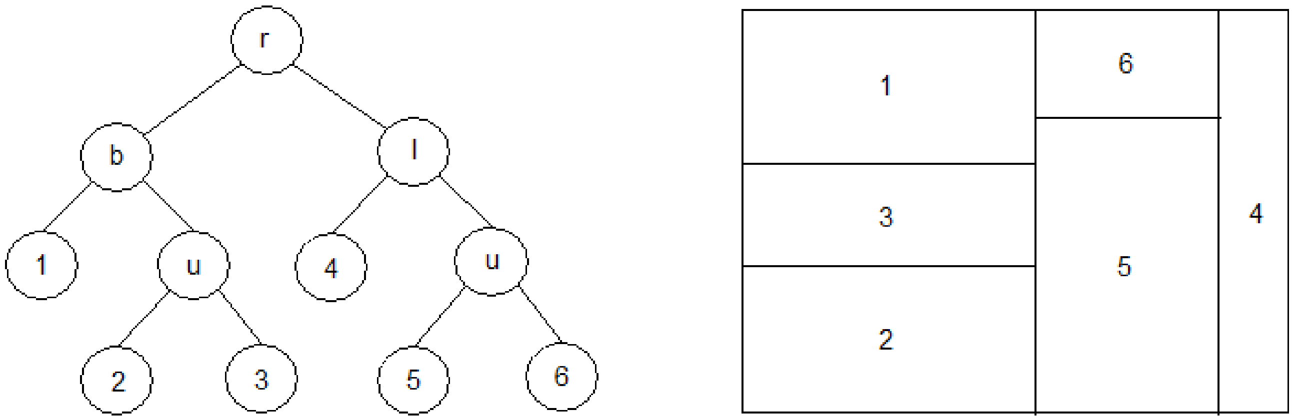

Figure 2: A Slicing Tree and the Slicing Structure, respectively.

- Grid. This approach divides the plant into squares which are the same area and dimensions. If the problem that we expect to solve has the facilities with equal dimensions and regular shape, we are in a simply problem of allocating n departments into m positions. However, if the dimensions are unequal and/or the shape are irregular, it will be necessary another structure that allows to identify each square within a determined department as the Space Filling Curve(SFC). Depends on the selected SFC we can obtain different department shapes. In the Figure 3, taken from [3], a grid with SFC is shown.

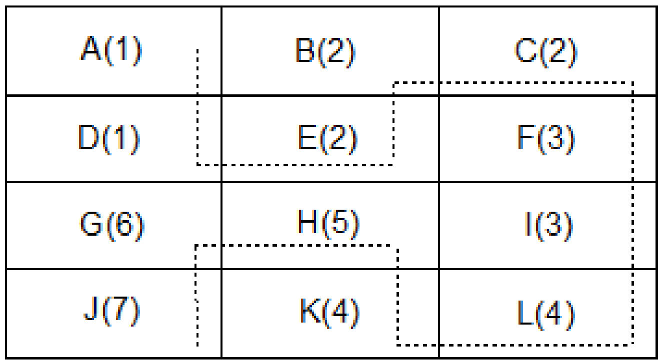

Figure 3: An example of a 12 square layout with SFC.

In the table 1, we show the structure been used by the revised works, we can see that the grid structure is the most used, followed by the bay structure and, finally, the tree structure. After revising the three structures, we have seen that is possible to obtain the bay solutions by the tree structure, but this not happens inversely, that is to say, we can´t find the tree solutions by the bay structure. On the other hand, in the grid structure, we have perceived that as much small is the square, the precision will be better.

Used Distribution	Blocks of Bays	Tree Structure	Grid Structure
Aiello 2006 [1]	慶		
Balakrishnan 2003a [2]			慶
Balakrishnan 2003b [3]			慶
Chae 2006 [4]	慶		
Chan 1994 [5]			慶
Conway 1994 [6]			慶
El-Baz 2005 [10]			慶
Enea 2005 [11]	慶		
Gómez 2003 [14]	慶		
Honiden 2004 [16]		慶	
Hu 2004 [17]			慶
Lee 2002 [20]	慶		
Lee 2003 [21]	慶		
Lee 2005 [22]	慶		
Mak 1998 [23]			慶
Matsuzaki 1999 [24]		慶	
Norman 2006 [25]	慶		
Ramkumar 2008 [27]			慶
Singh 1998 [28]			慶
Tam 1992 [30]		慶	
Tate 1995a [32]			慶
Tate 1995b [33]	慶		
Tavakkoli 1998 [34]			慶
Wang2005 [36]			慶
Wu2002 [37]		慶	

Table 1: Structures used in the revised works

4 Atomic tools for plant design (Phenotype)

To design the plant structure various strategies are combined. Here, the atomic tools used in these strategies are identified. These tools, combined in different ways, lead to a complete plant structured design. By recognizing them, we can analyse which of these elements are shared or similar, and to study the options that are available and its purpose. Each of these tools can be encoded in different ways that are described in the next section. The identified atomic tools are:

- Set position. The first function is to set position, which places a facility in a location determined by x and y coordinates. If we consult the figure 4, we can see that the encoding schemes that realize this purpose are the float permutations [8] and the float string without restrictions [21][22]. In the first of them, the float string are permutations of the departments center coordinates. The second encoding proportionates the distance between the origin and the center of aisles.
- Sort. In the studied problem, the most important and necessary function is to sort, which is the action to put in order the facilities in the layout. This function determines the department

sequence in a plant. For example, we have a different sequence of facilities, if we read the facility order from top to bottom and from left to right that if we read inversely. If we see the figure 4, we can find that for obtaining this purpose, normally a string of permutations is used which its elements are the facility sequence. Logically, in the combination of integer string and integers [24], also appears the sequence of the departments in the layout. In [30] is possible to find a new way to short. For this, a comprised value between '0' and '1' is assigned to each department randomly, then, all this values are sorted from minor to mayor for determining the facility sequence. The last encoding that allows to short the elements in the plant is the float string with the restriction that the addition of all the string elements is equal or less of the total area in the plant [21][22].

- Make partitions. With grouping purpose exits the function of making partitions, wich divides the plant into portions. This happens, for instance, when the layout is divided into bay blocks. If we observe the figure 4, we can see that for grouping, we have several possibilities of encoding. The first of them is the combination of integer permutation and integers. This case is used by [24] in the slicing tree, where the integer permutations indicate the facility sequence and the integers are operators which allow to divide (or to group) the plant into sections. The second possibility is the string of increasing positive integer [33], this structure indicates where are the breakpoints. The third way of grouping is through the integer string, in this case, the elements are inserted without restrictions, this elements are operators that indicate the plant divisions [16][30][37], or its show the way of grouping the layout facilities by the sweeping direction and the sweeping band [36][17]. Another possibility of grouping is the integer string, which the elements addition is the total number of facilities that exists in the layout [1][11][4][22]. This string indicates how many facilities are in each bay block, and so, the string has the same elements that bays are in the plant. The last possibility for grouping is the binary encoding [14][36][17][8]. This encoding used a binary string with a size equal to the facility sequence string. When in the binary string appears the value '0' indicates that the equivalent facility is in the same group or block, if in the string appears '1', then, it is the moment to begin with another block.
- Set orientation. If we continue seeing the figure 4, we can find that appears another function, which is to set orientation. that function allows to rotate a department over it central axis with regard to origin point. For setting orientation to department [37] used a integer string, which each element is a operator that proporcionates the facility orientation.

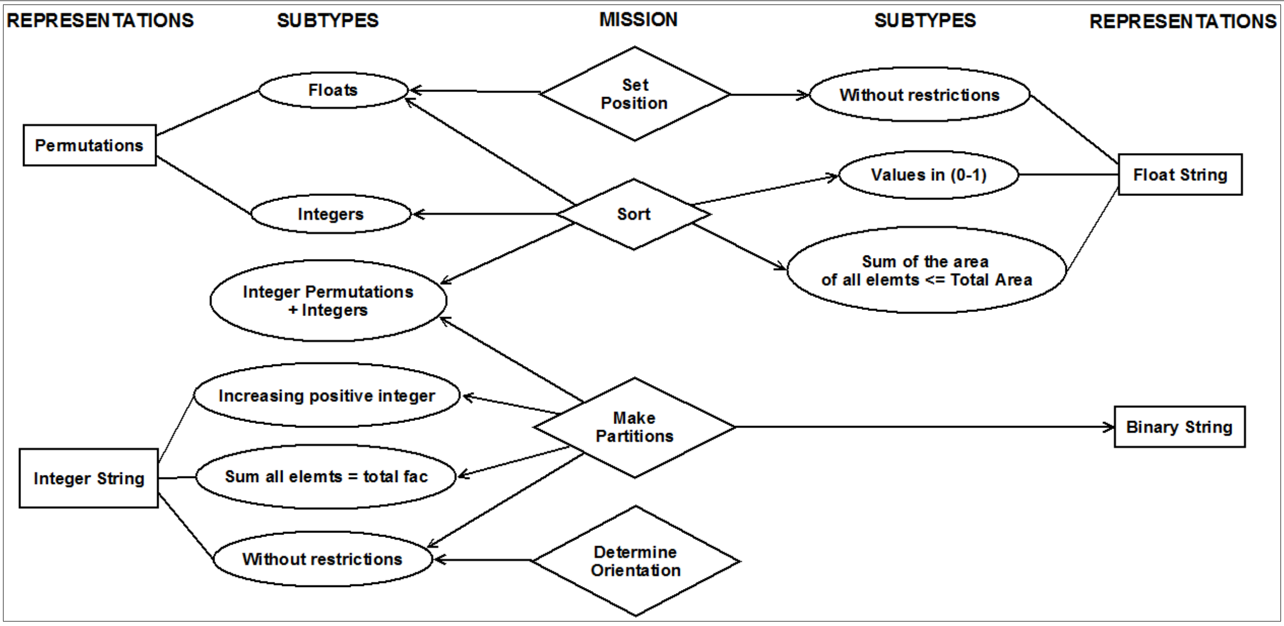

Figure 4: Relations between the encoding representations and their associated missions.

5 Encoding (Genotype)

After of analysing the studied works, we can to classify various types of encoding schemes:

- Integer/Real permutations: This encoding has the mission to determine the department sequence that is make up of plant. Generally, this encoding is a string of *n* size, where *n* is the number of facilities in the layout. Logically, it can´t have repeated elements in the string, because, the same department can´t be placed in two locations of the layout. Many authors have employed this encoding to establish their facility sequence. The great majority of them [1][2][3][4][5][10][11][12][15][16][17][21][22][23][27][28][32][33][34][36][37] have used integer strings to represent the department sequence, however, only some of them have been used real strings [21][22]. In the case of dynamic layout, various authors have been represented the corresponding layout for each period with permutations of integers [2] [6] or floats [8]. In the last case, the author uses the real string at the same time to stablish the facility sequence and to put the position of the department centers.
- Integer permutations + integer: This encoding was proposed by [24], and it is used at the same time to determine the facility sequence and for grouping the departments in the layout. For the first intention, it uses an integer elements which can´t be repeated. In order to represent the cut operators, he use 4 characters (that can be traduced to integers) which can be repeated. Both types of elements are mixed in the string. The size of the string is the sum of the number of facilities and the numbers of cut operators which are equal to facilities number minus one.
- Integer string: this type of encoding scheme can be divided into tree cases. In all cases, the chromosome is an integer string.

  - Increasing positive integer. The integer string is created considering that each element is bigger than the next one. [33] uses this encoding to group into a bays the elements of the layout, and it consists in an integer string where each element represents the last department of the bay. The total elements of the string plus one, is total number of the bays in the plant.

  - Sum all elements = total facilities. The integer string is created having into account that each element is bigger than '1', and the sum of all elements is equal than the number of facilities which make up the layout. Some authors [1][4][11][22] have used this encoding with the objective of grouping the facilities into bays, they propose an integer string that contains as many elements as the total bays in the layout.

  - Without restrictions. The integer string is created without restrictions: this encoding is used to group departments and to determine the facility orientation. [16] employed an integer string to shows the grouping order of facilities. [30] uses a string composed by chars (that can be traduced to integers), each of them will be a value of four possible operators, these operators determines the cut of tree, and its are bottom, upper, right and left. [36][17] utilizes this coding scheme to show the sweeping band, that determines the way of grouping the layout. [37] uses this encoding for 2 functions. On the one hand, he employs as string of integers which represents the cutting levels and allows to group the facilities in the plant. On the other hand, he uses another integer string to indicate the department orientation. Each element of this string can have associated one value of four possibilities which are 0º, 90º, 180º and 270º.
- Float: this encoding scheme can be divided into tree cases too. In all cases, the chromosome is a float string.

  - Without restrictions. The real or float string is created without restrictions: this type of encoding is used to set position elements in the layout. This proposal was suggested by [21][22] in order to allocate the vertical and horizontal passages in the plant. For this purpose, he uses a string with the position values of them.

- Values in (0,1). The real or float string is created having into account that each element is a value included in the range (0-1). [25] uses this encoding with the aim to put in order the facilities of the plant, he creates the department sequence assigning a random value between '0' and '1' for each facility and then, he arranges the string of smaller to higher value.

- Sum of the area of all elements <= Area total. The float string is created considering that each value of element is included between lower and higher bounds, moreover, it is necessary that the sum of all elements be lower or equal than the total area of the distribution. In this case, the string is formed of float elements (that are organized as the string of department sequence) which offer the area information of each facility in the layout.

- Binary string: this encoding has the function of grouping the facilities in the layout and to determine the facility orientation. [14] employed a binary string of *n* elements in order to divide the plant into bays. When '1' appears in the string, the facility is the last department of the bay, in other case, it appears a '0'. Moreover, [8] used a binary string to stablish the orientation of the department in the dynamic layout. When the value is '0' then the orientation is vertical, in other case, the orientation is horizontal.

Representation — Structure	Permut. of		Int Permut + int	String of integers			String of floats			String of bin
Subtype	Int	Float		Incr. Pos. Int.	Sum all elem = tot fac.	With out rest.	With out rest.	Val. (0-1)	Sum all elems A <= tot A	String of bin
Aiello 2006 [1]	■				■					
Balakrishnan 2003 [2]	■									
Balakrishnan 2003 [3]	■									
Chae 2006 [4]	■				■					
Chan 1994 [5]	■									
Conway 1994 [6]	■									
Dunker 2005 [8]		■								■
El-Baz 2005 [10]	■									
Enea 2005 [11]	■				■					
Ficko 2004 [12]	■									
Gómez 2003 [14]	■									■
Honiden 2004 [16]	■					■				
Hu 2004 [17]	■					■			■	■
Lee 2003 [21]	■	■					■			
Lee 2005 [22]	■	■			■		■			
Mak 1998 [23]	■									
Matsuzaki 1999 [24]			■							
Norman 2006 [25]								■		
Ramkumar 2008 [27]	■									

Singh 1998 [28]	■									
Tam 1992 [30]						■				
Tate 1995 [32]	■									
Tate 1995 [33]	■			■						
Tavakkoli 1998 [34]	■									
Wang2005 [36]	■					■			■	■
Wu2002 [37]	■					■				

Table 2: The different encoding schemes used by the revised authors

6 Crossover and Mutation operators

The crossover is the process whereby a new individual solution is created from the information contained within two (or more) parent solutions, is considered by many to be one of the most important features in GAs[9].

Representation		**Crossover**					
Structure	**Subtype**	**Uniform**	**PMX**	**OX**	**CX**	**One point**	**Select the best of set**
Permutation of	integers	慶	慶	慶	慶	慶	慶
	floats	O	慶	O	O	慶	O
Integer permutations + Integers		慶	O	O	O	O	O
String of Integers	Increasing positive	X	X	X	X	O	O
	Sum all = total fac.	X	慶	O	O	X	O
	Without restrictions	O	O	O	O	慶	O
String of floats	Without restrictions	O	O	O	O	慶	O
	Values (0-1)	X	X	X	X	O	O
	Sum of the elements area <= total area.	X	慶	O	X	X	O
String of bin	Binary values	O	慶	O	X	O	O

Table 3: Relation of the crossover methods used by each encoding scheme.

The mutation is the process whereby a parent solution is modified to obtain a new offspring. The purpose of mutation in GAs is to allow the algorithm to escape from local minima. Normally, this process is applied with a low probability called mutation rate.

Representation		Mutation								
Structure	**Subtype**	**PM**	**AM**	**SM**	**Inverse**	**PM if improve**	**Ins/Del a gen**	**Inc/Dec a gen**	**Div**	**Join**
Permutations of	integers	慶	慶	慶	慶	慶	X	X	X	X
	floats	慶	O	O	O	O	X	X	X	X
Integer permutations + Integers		O	O	O	慶	O	O	O	慶	慶
String of Integers	Increasing positive	X	X	X	X	X	O	O	X	X
	Sum all = total fac.	慶	O	O	O	O	慶	慶	O	O
	Without restrictions	慶	O	O	O	O	X	X	X	X
String of floats	Without restrictions	O	O	O	O	O	X	X	X	X
	Values (0-1)	X	X	X	X	X	X	O	X	X
	Sum of the elements area <= total area.	X	O	O	O	X	X	慶	X	X
String of bin	Binary values	慶	O	O	O	O	X	X	X	X

Table 4: Relation of the mutation methods used by each encoding scheme

In the table 3 and table 4, we can see the crossover and mutation methods, respectively, which have been studied marked with '慶'. The methods that is not possible to apply are market with 'X' by the encoding nature, and finally, we have marked with 'O' the methods that we can be applied, but have not been reported as used in literature.

7 Conclusions

After of performing this work we have extracted the following conclusions:

- The approaches applied for designing plant layouts using GA´s have been studied. The main characteristics and preferences that define the problem have been identify. We have classified the different types of distribution that are used in the studied works.
- We have identified the different purposes that are possible to obtain with the use of each encoding scheme . After, we have extracted the existing relationships between the found encoding schemes and the different suggested utilities.
- We have shown the crossover and mutation methods that have been used in the revised works. Moreover, we have evaluated the possibility of applying another methods. As future work, we plan to evaluate of all these methods with the aim of improving results and recommend the best methods for each encoding scheme.

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.; Cheng, C.H.; Conway, D.J. and Lau, C.M., “A hybrid Genetic Algorithm for the Dynamic Plant Layout Problem”, *International Journal of Production Economics*, Vol 86, 2003, pp.107-120.

[3] Balakrishnan, J.; Cheng, C.H. and Wong, K.F., “FACOPT: A User Friendly FACility Layout OPTimization System”, *Computers & Operations Research*, Vol 30, 2003, pp.1625-1641.

[4] Chae, J. and Peters, B.A., “Layout Design of Multi-Bay Facilities with Limited Bay Flexibility”, *Journal of Manufacturing*, Vol 25, 2006.

[5] Chan, C. and Tansri, H., “A Study of Genetic Crossover Operations on the Facilities Layout Problem”, *Computers & Industrial Engineering*, Vol 26, 1994, pp.537-550.

[6] Conway, D. and Ventakaramanan, M., “Genetic Search and the Dynamic Facility Layout Problem”, *Computers & Operations Research*, Vol 21, 1994, pp.995-960.

[7] Drira, A.; Pierreval, H. and Hajri-Gabouj, S., “Facility Layout Problems: A Survey”. *Annual Reviews in Control*, Vol 31, 2007, pp.255-267.

[8] Dunker, T.; Radons, G. and Westkämper, E., “Combining Evolutionary Computation and Dynamic Programming for Solving a Dynamic Facility Layout Problem”. *European Journal of Operational Research*, Vol 165, 2005, pp.55-69.

[9] Eiben, A.E. And Smith, J.E., “Introduction to Evolutionary Computing”. *Springer*. 2007

[10] El-Baz, M., “A Genetic Algorithm for Facility Layout Problems of Different Manufacturing Environments”, *Computers & Industrial Engineering*, Vol 47, 2004, pp.233-246.

[11] Enea, M.; Galante, G. and Panascia, E., “The Facility Layout Problem Approached using a Fuzzy Model and a Genetic Search”, *Journal of Intelligence and Manufacturing*, Vol 16, 2005, pp.303-315.

[12] Ficko, M.; Brezocnik, M. and Balic, J., “Designing the Layout of Single- and Multiple-Rows Flexible Manufacturing System by Genetic Algorithms”, *Journal of Materials Processing Technology,* Vol 157-158, 2004, pp.150-158.

[13] Goldberg, D. E., “Genetic Algorithms in Search, Optimization and Machine Learning”, *Addison-Wesley*, 1989.

[14] 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.

[15] Holland, J.H. “Adaptation in Natural and Artificial Systems”, *MIT Press*, 1975.

[16] Honiden, T., “Tree Structure Modeling and Genetic Algorithm-based Approach to Unequal-area Facility Layout Problem”, Industrial Engineering & Management Systems, Vol 3, pp. 123-128, 2004.

[17] Hu, M.H. and Wang, M.J., “Using Genetic Algorithms on Facility Layout Problems”. *International Journal of Advanced Manufacturing Technology*, Vol 23, 2004, pp.301-310.

[18] 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.

[19] Kuturel-Konak, S., “Approaches to Incertainties in Facilities Layout Problems: Perspectives at the Beginning of the 21st Century”, *Journal of Intelligence and Manufacturing*, Vol 18, 2007, pp.273-284.

[20] 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.

[21] 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.

[22] Lee, K.Y.; Roh, M. and Jeong, H., “An Improved Genetic Algorithm for Multi-Floor Facility Layout Problems Having Inner Structure Walls and Passages”, *Computers & Operations Research*, Vol 32, 2005, pp.879-899.

[23] Mak, K.L.; Wong, Y.S. and Chan, F.T.S., “A Genetic Algorithm for Facility Layout Problem”,

Computers Integrated Manufacturing System, Vol 11, 1998 , pp.113–127.

[24] Matsuzaki, K.; Irohara, T. and Yoshimoto, K., “Heuristic Algorithm to Solve the Multi-Floor Layout Problem with Consideration of Elevator Utilization”. *Computers & Industrial Engineering*, Vol 36, 1999, pp.487-502.

[25] Norman, B.A. and Smith, A.E., “A Continuos Approach to Considering Uncertainty in Facility Design”, *Computers & Operation Research*, Vol 33, 2006, pp.1760-1775.

[26] Pierreval, H.; Caux, C.; Paris, J.L. and Viguier, F., “Evolutionary approaches to the design and organization of manufacturing systems”, *Computers & Industrial Engineering*, Vol 44, 2003, pp.39-364.

[27] Ramkumar, A.S.; Ponnambalam, S.G.; Jawahar, N. and Suresh, R.K., “Iterated Fast Local Search Algorithm for Solving Quadratic Assignment Problems”, *Robotics and Computer-Integrated Manufacturing,* Vol. 24, 2008, pp.392–401.

[28] Singh, J.; Foster, B.T. and Heragu, S.S., “A Genetic Algorithm for the Unequal Area Facility Layout Problem”, *Computers & Operations Research,* Vol 25, 1998 , pp.583–594.

[29] Singh, S.P.; Sharma, R.R.K., “A Review of Different Aproaches to the Facility Layout Problems”, *International Journal of Advanced Manufacturing Technology ,* Vol 30, 2006 , pp.425–433.

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

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

[32] Tate, D.M. and Smith A.E., “A Genetic Approach to the Quadratic assignment zroblem”, *Computers & Operations Research,* Vol 22, 1995, pp.73–83.

[33] Tate, D.M. and Smith A.E., “Unequal Area Facility Layout Using Genetic Search”, *IEE Transactions,* Vol 27, 1995, pp.465–472.

[34] Tavakkoli-Moghaddain, R. and Shayan, E. “Facilities Layout Design by Genetic Algorithms”. *Computers & Industrial Engineering*, Vol 35, 1998, pp. 527-530.

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

[36] 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.

[37] 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.

8 Corresponding

Laura García Hernández

Área de Proyectos de Ingeniería. Universidad de Córdoba.

Edificio Leonardo Da Vinci – Campus Universitario de Rabanales – Carretera de Madrid km. 396 – 14.071 - Córdoba

Teléfono +34 957 21 20 50

E-mail: ir1gahel@uco.es