Journal of Theoretical and Applied Information Technology
20th February 2016. Vol.84. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SULE'S METHOD INITIATED GENETIC ALGORITHM FOR SOLVING QAP FORMULATION IN FACILITY LAYOUT DESIGN: A REAL WORLD APPLICATION

[1]FABRICIO NIEBLES ATENCIO, [2]DIONICIO NEIRA RODADO

[1]Asstt Prof., Department of Industrial Engineering, Universidad de la Costa (CUC), Barranquilla, COLOMBIA

[2]Asstt. Prof., Department of Industrial Engineering, Universidad de la Costa (CUC), Barranquilla, COLOMBIA

E-mail: [1]fniebles2@cuc.edu.co, [2]dneira1@cuc.edu.co

ABSTRACT

This paper considers the Quadratic Assignment Problem (QAP) as one of the most important issues in optimization. This NP-hard problem has been largely studied in the scientific literature, and exact and approximate (heuristic and meta-heuristic) approaches have been used mainly to optimize one or more objectives. However, most of these studies do not consider or are not tested in real applications. Hence, in this work, we propose the use of Sule's Method and genetic algorithms, for a QAP (stated as a facility Layout Problem) in a real industry application in Colombia so that the total cost to move the required material between the facilities is minimized. As far as we know, this is the first work in which Sule's Method and genetic algorithms are used simultaneously for this combinatorial optimization problem. Additionally the proposed approach was tested using well-known datasets from the literature in order to assure its efficiency.

Keywords: *Facilty Layout Design; QAP; Genetic Algorithm; Sule's Method*

1. INTRODUCTION

In order to improve competitiveness, manufacturing and service companies require to constantly implement formal procedures to optimize their processes. In this regard, Quadratic Assignment Problem (QAP) formulation is very important because it helps the decision maker to represent real life situations. The QAP tackles the problem of assigning N facilities to N locations; each of these assignments implies a certain cost. QAP formulation is useful to address problems such as airport terminal location, keyboard design, facility layout problems, among others. Particularly the Facility Layout Problem (FLP) is considered one of the most critical problems in the conformation of a new industry. Basically, it tackles the allocation of several resources in different settings [2], and is a problem in which numerous components have to be considered; for instance, distance and costs between the resources and the flow of materials. As these two are determining factors, it is essential to generate the mathematical formulation of the problem based on them. [5]

According to this, one of the conventional methods to determine a facility layout conformation was proposed by Dileep Sule. According to a basic goal: to obtain a neat and practical set –up of work stations, in order to reduce the movement of materials and people to a minimum level, which at the same time gives the possibility to have enough work in process.

It has been proved that the QAP belongs to the class NP-hard [6] and it is believed that this problem cannot be solved to optimality within polynomial bounded computation times even for smaller size problems, i.e. number of facilities less than or equal to 20 [7], the exact and conventional methods of resolution such as linear programming, integer and mixed programming, among others, are not efficient in terms of computing time to reach the optimal solution. Therefore, it becomes necessary to use alternative approaches to solve these problems in a reasonably

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

short time, even more if these decision have to be made daily. Meta-heuristics are within these approaches, and consist of formal procedures that are developed in order to overcome this difficulty encountered with traditional methods. Meta-heuristics solve instances of problems that are believed to be hard in general, by exploring the usually large solution search space of these instances. These algorithms achieve this by reducing the effective size of the space and by exploring that space efficiently. The most common meta-heuristic procedures to solve combinatorial problems are: genetic algorithms, tabu search, ant colony and simulated annealing, among others.

Genetic Algorithms (GA) are included in the group of metaheuristics, which are characterized by using one, although random, structured search approximation. Its exploration process is based on the mechanism of natural evolution in which individuals who are more adapted to the environment, survive and prosper. The algorithm opens a space to the evolution of a new solution through mutation (creation of a new solution through the combination of two solutions in the population) by maintaining and using a series of solutions available.

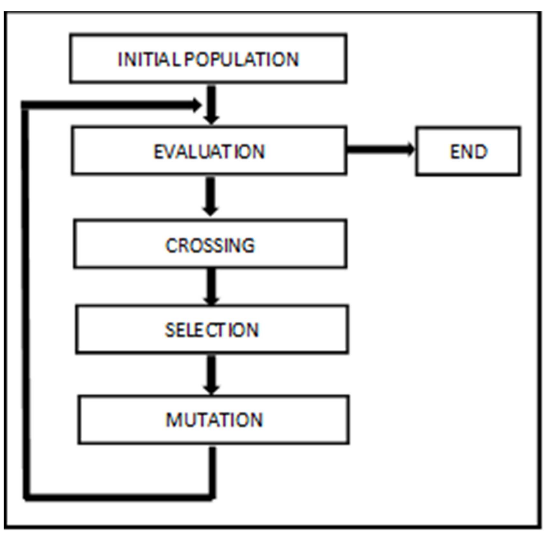

Figure 1: General Ga Scheme

As shown in figure 1, Genetic Algorithms operate in the following way: an initial population of solutions is generated; then, in order to obtain the value of the objective function, an evaluation of each individual is done. The objective of this evaluation is to make a selection for the better duplication criteria of individuals (the most suitable solutions). Subsequently, a crossing is done; through this procedure with the individuals selected or parents, two new individuals or sons are created; the new individuals suffer random mutation to generate a new individual (a single parent generates a new son). Finally, the new population suffers process iteration, from its evaluation to a new mutation. It becomes necessary then, to specify the GA variables: crossing probability and mutation probability, number of generations.

One of the greatest advantages of GA´s by which it is considered superior when compared to other meta-heuristics, especially in facility layout problem [7], is that its search pattern focuses in a random and parallel way. It has been demonstrated that this search pattern gives a better performance when it has been compared to other serial approaches [7]. On the other hand, Genetic Algorithms can handle several parameters in a parallel way. This make them work properly when solving problems in which the space for solutions is very large.

In this paper is considered a real life application, analyzing the problem of locating facilities in the configuration of a new manufacture facility that is going to make diverse products for electrical, telecommunications and building infrastructures. The company of this case study is planning to build a new facility and has estimated the allocation of the different assets and areas using basic tools as the spaghetti diagram. Nevertheless they considered that it is necessary to optimize this initial solution. The decision making process of allocating these assets and areas is known as the Facility Layout Problem (FLP), which can be formulated as a Quadratic Assignment Problem (QAP). This NP-hard problem has been largely studied in the scientific literature, and exact and approximate (heuristic and meta-heuristic) approaches have been used mainly to optimize one or more objectives. However, the most of these studies do not consider real applications. Hence, in this work, we propose the use of Sule's Method and genetic algorithms, for facility layout in this real industry so that the total cost to move the required material between the facilities is minimized. This paper shows preliminary and final results from the execution of this real case application.

To the best of our knowledge, this is the first work in which Sule's Method and genetic algorithms are used simultaneously for this combinatorial optimization problem. According to [8], the utilization of initialization method to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

generate an initial solution (or subpopulation) for the genetic algorithm, improve its performance obtaining better solutions in a shorter computational time. The proposed approach was useful for the analyzed company, but in order to prove the efficiency of the proposed model, computational experiments are carried out using well- known datasets from the literature. Results show the efficiency of our approach, and allow us to estimate the deviation against the optimum in this problem.

The rest of this paper is organized as follows: Section 2 is devoted to present the review of literature related to the solution of some particular facilities layout problems. Section 3 shows the formulation and mathematical model of the problem under study. Section 4 presents in detail the proposed hybrid approach, while Section 5 is devoted to computational experiments and the analysis of results. This paper ends in Section 7 by presenting some concluding remarks and suggestions for further research.

2. LITERATURE REVIEW

Many researchers in multiple disciplines have analyzed the optimization of the Facility layout Problem over the last decades; among them: [9], [2], [10] and [11], in which they present different surveys that expose various examples of methods to solve the Facility Layout Problem. In addition, [12] reviewed the state-of-the-art papers on facility layout problem with quadratic assignment model and mixed integer-programming models. More exhaustive surveys of the heuristic algorithms for the QAP can be found in [13], [14] and [15]. However, due to the intrinsic complexity in Facility Layout Problem, which are of the NP-hard type -like we previously said- the attention of the researchers is focused on the development of heuristics and metaheuristics for solving this problem with the less computational effort. These procedures can produce good answers within reasonable time constraints. There are following categories of heuristics for the QAP: construction methods, limited enumeration methods, improvement methods, and metaheuristics. Construction methods create suboptimal permutations by starting with a partial permutation which is initially empty. The permutation is expanded by repetitive assignments based on set selection criterion until the permutation is complete. The CRAFT (Computerized Relative Allocation of Facilities Technique), used for the layout of facilities was first introduced by [16]. Limited enumeration methods are motivated when one expects that an acceptable suboptimal solution can be found early during a brute force enumeration examination. Thus, due to the hardness of the QAP for heuristic methods [17], in recent times, this problem is a suitable testing platform for innovative intelligent optimization techniques or improvement methods like metaheuristics [18]. These methods work by starting with an initial basic feasible solution and then attempting to improve it. Therefore, approaches like: ant colony optimization [19], [20], [21] and [22]; evolution strategies [23], genetic algorithms [24], [25], [26], [27], [28] and [29]; greedy randomized adaptive search procedures [30]; hybrid heuristics [31], [32] and [33]; iterated local search [34]; simulated annealing [35], [36]; tabu search and very large-scale neighbourhood search [37], [38]. Thus, the design of the enhanced heuristic approaches for the QAP - which is also stimulated by numerous practical applications-, remains an active area of research.

3. PROBLEM FORMULATION

Considering that QAP formulation is useful to model the problem of allocating n facilities to n locations with the goal of optimizing one or more objectives, the objective of the FLP application considered in this study was to minimize the cost associated with the distance and flow between the facilities. Each of these assignments implied a certain cost. The determining factors for cost assignment are: distance and the flow of materials between facilities. As these two are determining factors, it is essential to generate the mathematical formulation of the problem based on them, in the following way:

$$min\, F = min\left[\sum_{i}\sum_{j} K_{ij} + \sum_{i}\sum_{j} C_{ij}\right] \cdot W_{ij}$$

With:

i= 1, 2, 3,…, n
j= 1, 2, 3,…, n
And:
$K_{ij=}f_{ij}d_{ij}$
If $r_{ij} < 0 \rightarrow C_{ij} = |r_{ij}| \div d_{ij}$

If $r_{ij} \geq 0 \rightarrow C_{ij} = r_{ij} \cdot d_{ij}$

Where:

pt: wo

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

rk station
n: amount of *pt* and facilities
f_{ij}: flow of materials from *pt i* to *pt j*
d_{ij}: distance between facility *i* and facility *j*
r_{ij}: proximity relation between *pt i* and *pt j*

Variables:

$$W_{ij} \begin{cases} If \;\; i \; is \; assigned \; to \; j \rightarrow 1 \\ Otherwise \rightarrow 0 \end{cases}$$

$$\sum_i W_{ij} = 1$$

$$\sum_j W_{ij} = 1$$

$$W_{ij} \in \{0,1\}$$

To define r_{ij} it is necessary to take into account that it can take negative values (because the non-desirable relation is quantified as a negative value. For this reason, two alternatives for the calculation of costs are given based on the proximity relation or C_{ij}, the formula $|r_{ij}| \div d_{ij}$, is done, since a negative r_{ij} value which is calculated $r_{ij} \cdot d_{ij}$, would generate an inconsistency in the result of the target function, even if this is represented as an absolute value : $|r_{ij}| \div d_{ij}$

Hence, a non-desirable relation should be weighed as a raise in the total cost which generates lower impact in the desirable relations. If the designer considers it appropriate, a non-desirable relation could also be quantified as a higher value in any desirable relation and in the formulation of C_{ij} only $r_{ij} \cdot d_{ij}$ would be considered.

The variable Wij ensures that, firstly, all the working stations are used and, secondly, all the facilities have a assigned working station

On the other hand, distance is calculated through the rectilinear norm: the distance between two points is not determined by the straight line that joins them, but, by the number of streets (making an analogy of the distance crossed by a car in a city) or positions (making reference to the generation of any initial solution to the problem) that should be crossed. It is formulated in the following way:

$$d_{ij} = |X_i - X_j| + |Y_i - Y_j|$$

Where:

X_i: Coordinate in *X* from point *i*
X_j: Coordinate in *X* from point *j*
Y_i: Coordinate in *Y* from point *i*
Y_j: Coordinate in *Y* from point *j*

4. PROPOSED APPROACH

4.1 Sule´s Conventional Method

In order to generate a preliminary facility distribution, [3] suggests the completion of a series of steps: (the tables and figures shown at this point are the results obtained from the practical exercise).

4.1.1. Establish the necessary content for each workstation, determining its area. The sum of these areas will determine the total required area. For this study, it is needed to make sure that the total required area does not exceed the total available area.

4.1.2. Determine the amount of material to be moved between workstations or f_{ij} by using a single unit of measurement in order to handle raw materials in a generic way, as well as product in process and finished goods. For the case of study of this paper, the unit Kg/Hour is used.

4.1.3. Closeness can be determined by the flow of materials, personal needs in multiple workstations, communication requirements, security restrictions and any other aspect to be considered.

4.1.4. Then a graphic display of the relationships table is generated. To do this, it is necessary to illustrate an initial arrangement of workstations by using a nodal diagram. Later, a grid or net representation in which the initial arrangement can be seen in the form of blocks is generated.

The representation of the solutions is shown in a matrix, as follows:

$$\begin{bmatrix} 8 & 5 & 3 \\ 6 & 1 & 2 \\ 4 & 9 & 7 \end{bmatrix}$$

Journal of Theoretical and Applied Information Technology
20th February 2016. Vol.84. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Material Flow Rates.

1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	11	11	11	
1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	11	11	11	22
1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	18	18	18	12	13
1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	18	18	18	12	13
8	8	10	10	10	10	10	10	10	10	10	10	21	21	21	21	21	21	20	20	20	0	0	93	0	0	0	0	0	18	18	18	12	13
8	8	10	10	10	10	10	10	10	10	10	10	21	21	21	21	21	21	20	20	20	4	4	4	4	4	4	4	4	18	18	18	12	13
8	8	10	10	10	10	10	10	10	10	10	10	22	22	22	22	22	22	8	0		4	4	4	4	4	4	4	4	18	18	18	17	17
19	19	10	10	10	10	10	10	10	10	10	10	22	22	22	22	22	22	16	16	16	16	16	16	16	16	16	16	16	16	16	16	17	17
19	19	7	7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	6	6	9	9	14	14	8	56	17	17
19	19	7	7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	6	6	9	9	14	14	0	0	0	0
19	19	7	7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	6	6	9	9	14	14	0	0	0	0
14			0	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	6	6	6	0	0	15	15	0	0	0	0
15			0																							0	0			0	0	0	0

16	1400	0	88	0	0	0	0	0	0	0	0	0	0	0	0	---	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	---	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	---	0	0	0	0
19	6	0	0	0	0	7	0	0	0	0,8	0	0	0	0	0	0	0	0	---	0	0	0
20	0	0	0	0	0	0	0	0	0	56	0	0	0	0	0	0	0	0	0	---	0	0
21	0	0	0	0	0	22	0	0	0	22	0	0	0	0	0	0	0	0	0	0	---	0
22	0	0	0	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	---

The location of each working station can be represented through the matrix position. For instance, working station 8 is located in position *1,1* at the matrix and working station 7 is located in position *3,3*.

Table 2: Available Area Vs Required Area.

AREA REQUIREMENT			
Facility	Length (m)	Width (m)	Area (m2)
1	20,0	8,0	160
2	20,0	8,0	160
3	38,0	4,0	152
4	16,0	4,0	64
5	16,0	16,0	256
6	15,0	3,0	45
7	7,0	1,4	9,8
8	8,4	1,4	11,76
9	13,0	1,0	13
10	18,0	9,0	162
11	5,0	5,0	25
12	3,0	3,0	9
13	3,0	3,0	9
14	5,0	5,0	25
15	2,0	2,0	4
16	20,0	2,0	40
17	4,0	6,0	24
18	11,0	5,0	55
19	7,0	1,4	9,8
20	5,0	2,0	10
21	12,0	4,0	48
22	12,0	4,0	48
Total required área			1340,36
Total available area (Parte			1482
Complete available area			2613

Table 3A: Relationship Nomenclature (from Sule D. 2001)

Closeness Relationship Associated Cost	
Absolutely Necessary	50
Primarily Important	30
Important	20
Ordinarily important	10
Without importance	0
Not Desirable	-10

Table 3B: Associated Cost

Closeness Relationship Associated Cost	
Absolutely Necessary	50
Primarily Important	30
Important	20
Ordinarily important	10
Without importance	0
Not Desirable	-10

TABLE 4: EXAMPLE OF SULE'S PERFORMANCE

Journal of Theoretical and Applied Information Technology
20th February 2016. Vol.84. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2 Genetic algorithm (GA)

A Genetic Algorithm is a problem solving technique that uses the concepts of evolution and hereditary to produce good solutions to complex problems that typically have enormous search spaces and are therefore difficult to solve. Figure 2 illustrates the general phases of a GA. The biggest difference with other meta-heuristics is that GA maintains a population of solutions rather than a unique current solution. Solutions are coded as finite-length strings called chromosomes and a measure of their adaptation (the fitness) is computed by an engine. Starting from an existing population representing the initial solution of the problem, a set of iterations generate new chromosomes (solutions) by applying crossover and mutation operators, according to a probability, to two chosen parents. The main advantage of GA is its intrinsic parallelism, which allows the exploration of a larger solution space.

4.2.1 Solution Representation and Initial Population:

In a broad way, the genetic algorithm presented here is an optimization procedure that seeks to minimize the total cost of facility layout design proposed. Once the values of decision variables are found, the total cost is computed by the procedure shown in figure 2.

The structure of each individual in the solution is a chain of chromosomes, each one giving the values of decisions variables (*see problem formulation section*) for a specific layout conditions. That is we have a total of v chromosomes, each with m × z genes representing the location of each working station obtained by sule's method.

4.2.2 Selection, Crossover and Mutation:

The selection procedure selects the best individuals to be considered for the next generation. In our procedure, the number of selected individuals is limited by the size of population and by the constraints of the problem (i.e., capacity constraints). The individuals with best values of the fitness function are selected.

Table 5: Closeness Relationship Quantification

Sule's Relationship Table																						
Facility	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	-	4	0	0	0	0	4	4	0	0	0	0	-1	0	0	3	0	0	4	0	0	0
2		-	4	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
3			-	4	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
4				-	4	0	0	0	3	0	0	0	-1	1	1	2	0	0	3	0	0	0
5					-	4	0	0	3	0	0	0	-1	0	0	0	0	0	0	0	0	0
6						-	2	2	0	0	0	0	-1	0	0	0	0	0	2	0	3	3
7							-	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
8								-	0	4	0	0	0	0	0	0	0	0	0	0	0	0
9									-	0	0	0	-1	0	0	0	0	0	0	0	0	0
10										-	0	0	-1	1	1	0	0	0	4	4	4	4
11											-	0	0	0	0	0	0	3	0	0	0	0
12												-	0	0	0	0	0	0	0	0	0	0
13													-	-1	-1	0	3	0	0	0	0	0
14														-	4	0	-1	0	0	0	0	0
15															-	0	-1	0	0	0	0	0
16																-	0	0	0	0	0	0
17																	-	0	0	0	0	0
18																		-	0	0	0	0
19																			-	0	0	0
20																				-	0	0
21																					-	0
22																						-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

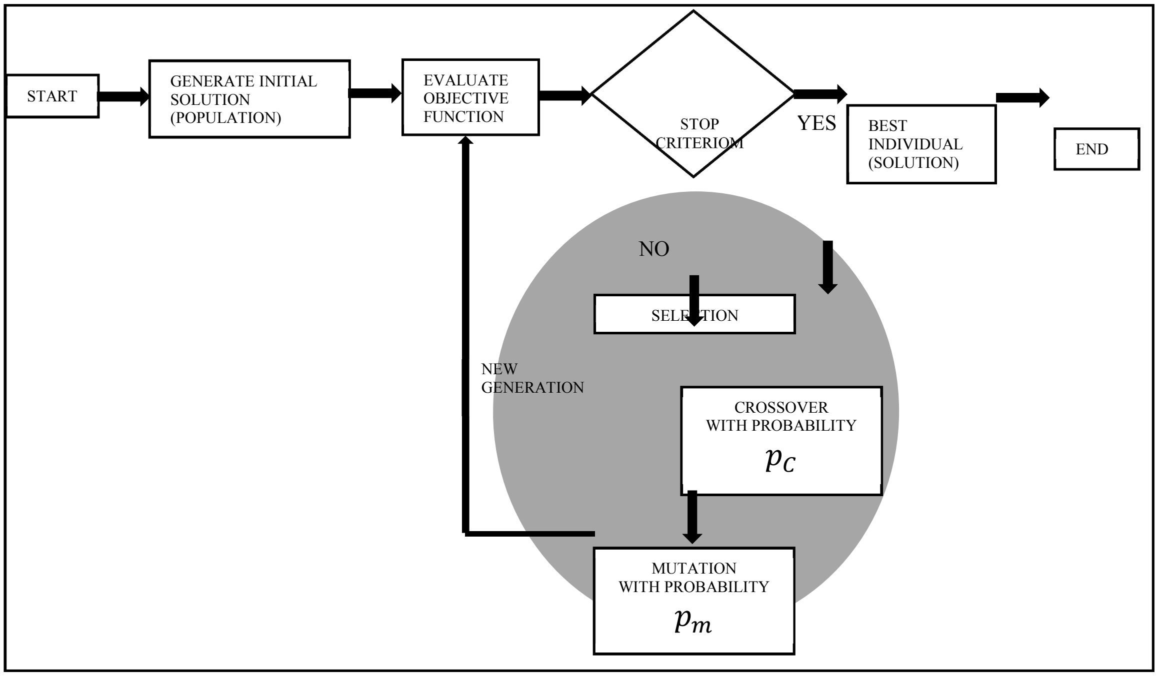

Figure 2: Generic Structure of a Genetic Algorithm

Partial Mapped Crossover (PMX) is done over all individuals of the population and elitism procedure is carried out, registering all information in order to compare with the next generation. Four individuals are generated as follows: Son 1 is composed of genetic material from the first half of chromosome Father 1 with second half of chromosome Father 2, Son 2 is composed of genetic material from the first half of chromosome Father 2 with second half of chromosome Father 1, Son 3 is composed of genetic material from the first half of chromosome Father 1 with first half of chromosome Father 2, and finally Son 4 is composed of genetic material from the second half of chromosome Father 1 with second half of chromosome Father 2. This procedure is carried out for each matrix giving all the decision variables of the problem. After crossover, it is necessary to verify that the resulting individuals correspond with a feasible solution.

Mutation operator is defined as an interchange operation. This means that two positions of the chromosome are interchanged as follows: the first position with the third one, the third position with the last one, etc. (see Figure 3). Chromosomes to be mutated are those with the lowest value of the total cost (objective function) after the selection process.

At the same way of crossover, after mutation, it is necessary to verify that the resulting individuals corresponds with a feasible solution.

4.2.3 Fitness Function

The fitness function allows the algorithm to compare the quality of the individuals in the population (i.e., to evaluate the quality of the different solutions). Let $f(x_1), \ldots, f(x_n)$ be the values of the objective function for each individual. Since the objective is to minimize the total cost, chromosomes with the lowest probability are selected. That is, the lower the value of $f(x_h)$. The higher the probability p_h of being selected. Hence, individual x_h will be selected for reproduction.

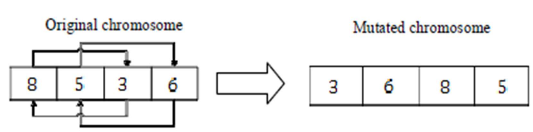

Figure 3: Example of Mutation Operation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. COMPUTATIONAL EXPERIMENTS AND RESULTS

The main objective of design of experiments (DOE) is to study the effect of various factors on the response variables and select, for these factors, the best option among multiple levels. The kind of design is a 3^3 factorial and the response variable is the cost result in the objective function.

A group of initial solutions is generated by using the Sule's method and those solutions that will make part of the initial population of genetic algorithm (those that conceived a better result of efficacy) are chosen. The initial population is used as a basis for the experiment design pilot-test. The genetic algorithm was coded in Visual Basic®. Computational experiments were carried out on a PC under Windows 7 Home Premium (32 bits) operating system and with processor Intel® Core TM 2 Duo CPU T7100 1.8GHz and RAM 2.00 GB RAM.

5.1 Evaluation Of The Amount of Generations

Initially, the amount of ideal generations to be studied is evaluated. Five runs in which the results for 5, 10, 15 and 20 generations are done, then; the cost percentage decrease is quantified from the results obtained in the initial population.

For the group of solutions conceived by the crossing, the solutions with the best result in the target function emerged from the first 5 generations.

In the group of solutions conceived by mutation, better solutions are obtained in 5, 10 and 20 generations. One of the best solutions was found in the group of the 20 generations; which leads to considering the group in the evaluation; especially because of its high decrease percentage with respect to the solution generated in the initial population (21.0%)

For most of the runs (except the third one), the best solutions obtained through mutation are better than those obtained through crossover; which shows the effectiveness of the mutation operation in the algorithm. It is established that the amount of generations to be evaluated are 5, 10 and 20 generations.

5.2 Crossover Probability And Mutation Probability

As Genetic Algorithms behaves in a probabilistic manner, the analysis of computational experiments were carried out following a proper statistical methodology. A formal planned experimentation was used following the principles of statistical experimental design (or Design of Experiments, DOE), in which a set of factors that may affect a response variable defined in advance are evaluated [38]. Our first experiment consisted on a 3^3 full factorial design which means that three factors and three levels were defined (see Table 6). The response variable was the total cost. A pilot sample was analyzed first by running four replications for all nine levels of the three factors, giving a total of 108 runs. The analysis of results was done using SPSS® statistical software. The aim of this pilot test was to verify the assumptions of experimental design. Based on the results obtained, the basic assumptions of the design of experiments (independence of observations, normal distribution of the residuals and homogeneity of variances) were all verified [38].

This first test allowed us also to have some insights about the possible interactions between factors. Results of the ANOVA (analysis of variance) test are presented in Table 7.

We see that at least one factor (population size) causes an effect on the response variable, while crossover and mutation probabilities do not affect the total cost. However, the interaction between the number of generations and mutation probability, with their respective levels, does influence the value of the total cost.

From the insights given by these results, we next analyzed the behavior of the proposed genetic algorithm in terms of the objective function (total cost) in order to evaluate its convergence over the number of generations (see Figure 4 to 6). As shown in Figure 4, there is not a clear convergence of the algorithm when the probability of crossover is 0.7; the value of the objective function tends to improve when mutation probability is 0.2, and the initial solution is not improved at all in the case of mutation probability of 0.1. The value of the total cost moves within the range between USD 132467 and USD 177456, Pc=0.7 with Pm=0.05 and Pc=0.7 with Pm=0.1, respectively.

Figure 5 shows the evolution of the objective function value with crossover probability of 0.8 and different values of mutation probability. We observe a converge starting from the iteration number 80. For the case of Pc=0.8 with Pm= 0.05, the solution value improves over the number of generations, while the

Journal of Theoretical and Applied Information Technology
20th February 2016. Vol.84. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE 6: FACTORS AND LEVELS FOR THE EXPERIMENTAL DESIGN

Factors	***Notation***	***Values***		
Number of generations	*Ngen*	50	100	125
Crossover probability	*Pc*	0.7	0.8	0.9
Mutation probability	*Pm*	0.05	0.1	0.2

opposite phenomenon occurs with *Pc*=0.8 and *Pm*=0.2. The value of the minimum cost moves between USD 131389 and USD 166099 for *Pc*=0.8 with *Pm*=0.05 and *Pc*=0.8 with *Pm*=0.1, respectively.

Finally, Figure 6 presents the evolution of the minimum cost over the number of iterations with crossover probability of 0.9. We observe that the value of the objective function trends to converge from the 80th iteration. For the case of *Pc*=0.9 with *Pm*= 0.1, the solution value improves over the number of generations, while the opposite phenomenon occurs with *Pc*=0.9 and *Pm*=0.05. The value of the minimum cost moves between 134500 and 152435 for *Pc*=0.9 with *Pm*=0.1 and *Pc*=0.9 with *Pm*=0.05, respectively.

5.3. Comparison of the proposed model

A comparison with some instances proposed in QAPLIB (Burkard, Çela, Karisch, & Rendl, 2011) is done. These have been the subject of multiple comparisons through the years.

Specifically, the instances proposed by [36] and in Comparison of iterative searches for the quadratic assignment problem are selected, to then carry out 10 runs with the proposed method for each instance.

Additionally, in order to maintain certain coherence in the experimental analysis, a relative deviation index, in percentage, was employed, as shown in the following equation, where F_x^{GA} corresponds to the averages values of the objective function (i.e., total cost) obtained using the proposed genetic algorithm (GA). Also, F_x^{TAI} corresponds to the best values of the objective function based on mentioned Taillard instances. These values are shown in Table 5. It is necessary to clarify that the instances that were chosen are the most comparable with the problem under study. Not all the instances accomplished with this.

$$\%dev = \frac{F_x^{GA} - F_x^{TAI}}{F_x^{TAI}} \times 100\%$$

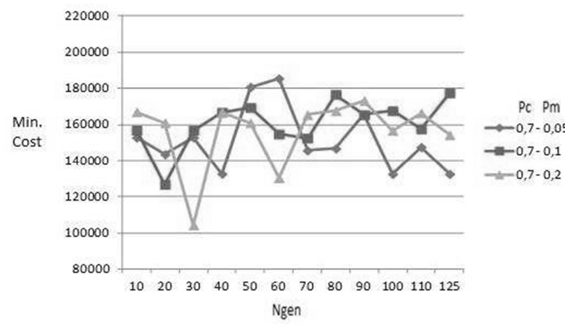

Figure 4: Evolution of the Total Cost with Crossover Probability of 0.7

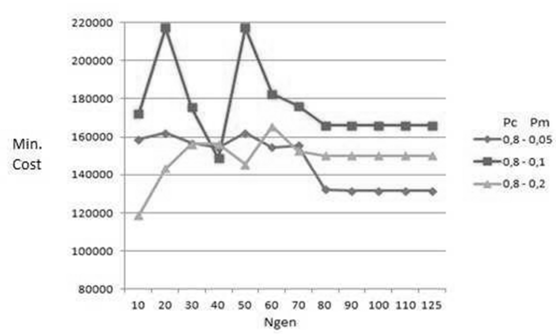

Figure 5: Evolution of the Total Cost with Crossover Probability of 0.8

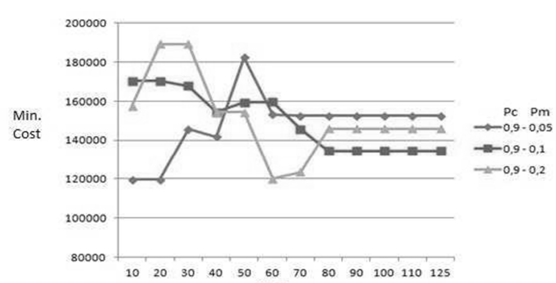

Figure 6: Evolution of the Total Cost with Crossover Probability of 0.9

Journal of Theoretical and Applied Information Technology
20th February 2016. Vol.84. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE 7: ANOVA FOR THE PILOT TEST (DEPENDENT VARIABLE: TOTAL COST)

Origin	Sum of squares type III	Dof	Square mean	F	Sig.
Corrected model	26367194188.667	26	1014122853.410	2.344	0.002
Intersection	2966047611200.127	1	2966047611200.127	6855.396	0.000
Ngen	4766367057.722	2	2383183528.861	5.508	0.006
Pc	232984132.167	2	116492066.083	0.260	0.765
Pm	559181762.667	2	279590881.333	0.646	0.527
Ngen*Pc	1588022998.111	4	397005749.528	0.918	0.458
Ngen*Pm	9788990126.611	4	2447247531.653	5.656	0.000
Pc*Pm	4053057158.667	4	1013264289.667	2,342	0.062
Ngen*Pc*Pm		8	672323869.090	1.554	0.152
Error		81	432658844.102		
Total		108			
Total corrected	61412560560.917	107			

aR-square 0.429 (corrected R-square 0.246)

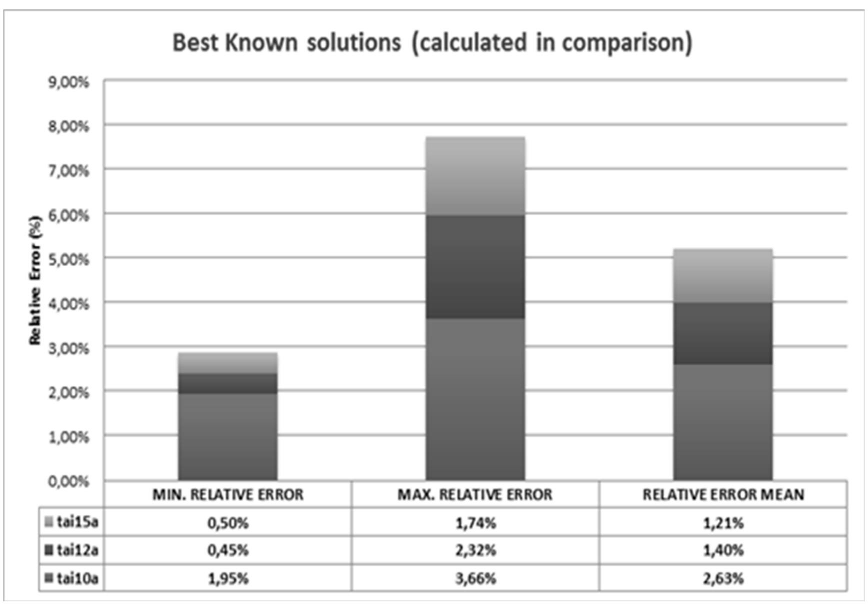

Figure 7: Comparission between results and QAPLIB Instances

As it was evidenced in the table above, solutions with low relative error are found when compared to the best solutions found for each of the instances.

The solution with the lowest relative error was founded for tai12a (0.45%) followed by those generated by tai15a (0.5%) and tai10a. The instance which the method evidenced its best behavior was tai15a, whose average and relative error range among the 10 solutions found, where the lowest (1.21% and 1.2% respectly). Hence, the results shown in Table 5 shows that the proposed algorithm has very good performance with respect to the best value known for this problem.

On the other hand, the solutions with the lowest relative error belong to the comparisons made with Tai-a instances, these last instances mentioned were generated randomly, unlike tai-b instances, which belong to real problems.

Finally, as we previously stated in section I, the company initially made an estimation of the new

Journal of Theoretical and Applied Information Technology
20th February 2016. Vol.84. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7: Results And Qaplib Instances Comparission

INSTANCES vs PROPOSED METHOD (COMPARISON)			
INSTANCE	BEST KNOWN SOLUTION	BEST OBTAINED SOLUTION	RELATIVE ERROR
tai10a	135028	137662	1,95%
tai10b	118376	132277	11,74%
tai12a	224416	225430	0,45%
tai12b	394695	403852	2,33%
tai15a	388214	390160	0,50%
tai15b	576528	600987	4,02%
tai17a	491812	508939	3,48%
tai20a	703482	725155	3,08%
tai20b	122459	130148	6,27%

Facility layout using basic tools such as the spaghetti diagram. However, this method is not efficient in terms of total time of production process, total amount and time spent in movements. Thus, with the proposed approach, the company obtained a reduction the total time of flow materials in a 15% [41]. In addition, the total cost for the proposed layout was reduced in a 20% comparing it with the initial estimation of the company [41] (See Table 8)

TABLE 8: Savings

Initial estimated total layout cost (USD)	Total layout cost after optimization (USD)	Saving (USD)
156178	130148	26030

CONCLUSIONS

The integration of the proximity measure in the proposed model turns out to be highly important: it forces the inclusion of important factors in the total cost. Besides flow and distance, it is also important to consider the need for distance between working stations due to risks in the security of the facility´s operative personnel; or, closeness issues, due to the fact that a group of people are required to distribute their daily activities between several working stations.

The proposed algorithm generated solution that compared with instances from other methods, provides a sign that the proposed method can behave satisfactorily both with problems conceived randomly and with real-life problems. Additionally, it is important to point that such behavior can be even more evidenced in problems with less than 15 working stations.

As for the practical exercise, the combination of QAP together with the additional variable proposed by Sule permitted the consideration of the manufacture system´s stochastic nature, while optimal close solutions were derived.

For further research, several lines are still open. For example, many other issues of the problem under study could be included in the analysis in order to keep the problem much more realistic: probabilistic constraints, i.e. stochastic capacities in production plant. The procedure can be improved by implementing: different procedures to generate the initial population, other types of crossover or mutation strategies, or even other fitness function, etc. Other heuristic procedures could be employed to hybridize the genetic algorithm. Finally, because of the NP-completeness of the problem under study, researchers could be interested in analyzing the behavior of various meta-heuristic algorithms such as GRASP (Greedy Randomized Adaptive Search Procedure), Tabu Search, etc.

Additionally, it is important to use this kind of formulation for other applications such as police station allocation, hospital areas allocation, among others in order to test the applicability of the QAP formulation and the proposed solution method.

REFERENCES:

[1] Niebles-Atencio, F., Solano-Charris, E.L., Montoya-Torres J.R.. "Ant Colony Optimization Algorithm to Minimize Makespan and Number of Tardy Jobs in Flexible Flowshop Systems. Proceedings 2012 XXXVIII *Conferencia Latinoamericana en Informática (CLEI 2012), Medellin, Colombia,* 1-5 Oct. 2012, pp. 1-10. DOI: 10.1109/CLEI.2012.6427154.

[2] Eldrandaly K. A., G.M. Nawara, M.A. Shouman, and A. H. Reyad. 2001. Facility Layout problem and intelligent techniques: A Survey, *7th International Conference on Production Engineering, Design and Control, PEDAC*, 2001, Alex Egypt. Pp.:409-422

[3] Logistics of facility location and allocation. Dileep Sule. 2001.

[4] Koopmans T, Beckmann M (1957) Assignment problems and the location of economic activities. *Econometrica* 25:53–76

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[5] Sahni S, Gonzalez T (1976) P-complete approximation problems. J ACM 23:555–565.

[6] Meller R. D, Gau K. Y, The facility layout problem: recent and emerging trends and perspectives, *Journal of Manufacturing Systems*, 15 (1996) 351-366.

[7] Azadivar, F., & Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), 4369–4383.).

[8] Coello, C.A.C., Lamont, G.B., van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn.. 2007. Pp 145–156.

[9] Kratica J., Tošić D., Filipović V., Dugošija Đ.A. New genetic representation for quadratic assignment problem. 2011. *Yugoslav Journal of Operations Research*. 21 (2011), Number 2, 225-238. DOI: 10.2298/YJOR1102225K

[10] S. P. Singh, R. R. K. Sharma. A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology*. September 2006, Volume 30, Issue 5-6, pp 425-433

[11] Drira A, Pierreval H, Hajri-Gabouj S (2007) Facility layout problems: a survey. *Annual Reviews in Control*, 31(2), 255—267.

[12] Burkard RE, Çela E, Pardalos PM, Pitsoulis L (1998) The quadratic assignment problem. In: Du DZ, Pardalos PM (eds) *Handbook of Combinatorial Optimization,* vol 3. Kluwer, Dordrecht, pp 241–337

[13] Loiola EM, de Abreu NMM, Boaventura-Netto PO, Hahn P, Querido T (2007) A survey for the quadratic assignment problem. *Eur J Oper Res* 176:657–690

[14] Voß S (2000) Heuristics for nonlinear assignment problems. In: Pardalos PM, Pitsoulis L (eds) *Nonlinear Assignment Problems. Kluwer, Boston*, pp 175–215 Armour and Buffa

[15] Angel E, Zissimopoulos V (2002). On the hard ness of the quadratic assignment problem with metaheuristics. *J Heurist* 8:399–414

[16] Gambardella LM, Taillard ED, Dorigo M (1999) Ant colonies for the quadratic assignment problems. *J Oper Res Soc* 50:167–176

[17] Maniezzo V, Dorigo M, Colorini A. The ant system applied to the quadratic assignment problem. *Technical report IRIDIA/94-28. Belgium: Universite Libre de Bruxelles*; 1994.

[18] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern*—Part B 1996;26:29–41.

[19] Stutzle T, Dorigo M. ACO algorithms for the quadratic assignment problem. In: Corne D, Dorigo M, Glover F, editors. *New ideas in optimization. Maidenhead, UK: McGraw-Hill Ltd*; 1999.

[20] Nissen V (1994) Solving the quadratic assignment problem with clues from nature. *IEEE Trans Neural Netw* 5:66–72 (Drezner 2003

[21] Merz P, Freisleben B (2000) Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Trans Evolut Comput* 4:337–352

[22] Misevicius A (2004) An improved hybrid genetic algorithm: new results for the quadratic assignment problem. *Knowl Based Syst* 17:65–73

[23] Fleurent C, Ferland J. Genetic hybrids for the quadratic assignment problem. DIMACS *Ser Math Theor Comput Sci* 1994;16:190–206.

[23] Tate DM, Smith AE. A genetic approach to the quadratic assignment problem. *Comput Oper Res* 1995;22(1):78–83.

[24] Ahuja KR, Orlin JB, Tiwari A. A greedy algorithm for the quadratic assignment problem. *Comput Oper Res* 2000;27:917–34.

[25] Li Y, Pardalos PM, Resende MGC. A greedy randomized adaptive search procedure for the quadratic assignment problem. Quadratic assignment and related problems. In: Pardalos PM, Wolkowicz H, editors. *DIMACS series in discrete mathematics and theoretical computer science. American Mathematical Society*; 1994. p. 237–61

[26] McLoughlin JF III, Cedeño W (2005) The enhanced evolutionary tabu search and its application to the quadratic assignment problem. In: Beyer H-G, O'Reilly U-M (eds) *Proceedings of the genetic and evolutionary computation conference—GECCO* 2005. ACM Press, New York, pp 975–982

[27] Tseng L, Liang S (2005) A hybrid metaheuristic for the quadratic assignment problem. *Comput Optim Appl* 34:85–113

[28] Xu Y-L, Lim MH, Ong YS, Tang J (2006) A GA-ACO-local search hybrid algorithm for solving quadratic assignment problem. In: Keijzer Metal (eds) *Proceedings of the genetic and evolutionary computation conference-*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GECCO 2006, vol 1. ACM Press, New York, pp 599–605.

[29] StützleT (2006) Iterated local search for the quadratic assignment problem. *EurJOperRes*174:1519–1539.

[30] Bölte A, Thonemann UW (1996) Optimizing simulated annealing schedules with genetic programming. *Eur J Oper Res* 92:402–416

[31] Wilhelm MR, Ward TL. Solving quadratic assignment problems by simulated annealing. *IIE Trans* 1987; 19:107–19.

[32] Battiti R, Tecchiolli G (1994) The reactive tabu search. *ORSA J Comput* 6:126–140

[33] Drezner Z (2005). The extended concentric tabu for the quadratic assignment problem. *Eur J Oper Res* 160:416–422

[34] JamesT, Rego C, Glover F (2009) A cooperative parallel tabu search algorithm for the quadratic assignment problem. *Eur J Oper Res* 195:810–826

[35] Misevicius A (2005) A tabu search algorithm for the quadratic assignment problem. *Comput Optim Appl* 30:95–111

[36] Taillard ED (1991) Robust taboo search for the QAP. Parallel Comput 17:443–455

[37] Voß S (1995) Solving quadratic assignment problems using the reverse elimination method. In: Nash SG, Sofer A (eds) The Impact of Emerging Technologies on Computer Science and Operations Research. Kluwer, Dordrecht, pp 281–296

[38] Skorin-Kapov J. Tabu search applied to the quadratic assignment problem. ORSA J Computing 1990; 2:33–45.

[39] Ahuja RK, Jha KC, Orlin JB, Sharma D (2007) Very large-scale neighborhood search for the quadratic assignment problem. *INFORMS J Comput 19:646–657*

[40] Escobar, I. Thesis: Ejercicio Práctico de Distribución de Planta mediante Algoritmos Genéticos. Universidad de la Sabana. 2013.