Journal University of Kerbala , Vol. 16 No.2 Scientific . 2018

# Comparison between two methods for solving advance layout problem

# المقارنة بين طريقتين لحل مسألة التوطين المتقدمة

**Ikhlas Watan Gindawi Lamyaa Mohammed**
**Dept. of computer science AL- Mustansiriyah University**

## Abstract

This paper description on various methods for solving advance layout problem and describe a novel method based on genetic algorithms to solve machine layout problem. This research present an exhaustive study of hybrid simple genetic algorithm (HGS) and hybrid steady state genetic ( HGSS).

Genetic algorithms were used in conjunction with local optimizer and separation algorithm to find optimal spatial layouts. A single population was used in both twelve- machine (entity ) layout and more complex forty-five machine layout problem. In this work we used Delphi language to solve this problem.

**الخلاصة**

في هذا البحث يتم وصف طرق متنوعة لحل مسألة التوطين المتقدمة ووصف طريقة حديثة معنمدة على الخوارزميات الجينية لحل مسألة توطين الالات. في هذا البحث تقديم دراسة واسعة لطريقة الخوارزمية الهجينة البسيطة والخوارزمية الجينية الهجينة لحالة الاستقرار.

يتم استخدام الخوارزميات الجينية مع الامثل المحلي وخوارزمية الفصل لايجاد التوطين الامثل حيث يتم تطبيقها على مسألتين هما الاثنى عشر كيان والمسألة الاكثر تعقيد ال خمسة واربعون كيان لمسالة توطين الالات. حيث تتم استخدام لغة برمجة في هذا العمل تستطيع ان تتعامل بقوة مع هذة المشكلة وهي لغة Delphi .

## 1- Introduction

Advance layout problem (ALP) is concerned with the conceptual design of the physical enclosure manufacturing design or renovation of factories distributaries centers hospitals, bank, department stores, military supply, depots, university, ect:[1].

Currently layouts are only used as a guide for architects and engineers that are designing factory floor plans. These guides are used to give that designing factory plans. These guides are used to give additional information about interdependcies of machine that is not very easily seen in factory design. This information can then be used by human designer to improve the flow of materials, people and resources through the new plant, Although the ALP is still used mainly as a guide for designer, the ability to deal with extension such as specific continue placement, heterogeneous foot prints information make ALP more capable of directly generating practical floor plans that could be used as an advance starting point for design specialists[2].

## 2- Definition of the Advance Layout Problem

The ALP layout problem is to position n circular objects with heterogeneous radius I into an area of length L and width W such that there are no overlaps, repetition between the circles, The distance between all pairs of entities is optimized according to weight matrix [2].

**3- Definition of the Weight Matrix (A) [2,3]**

Is the n*n with each corresponding to specific priority of relationship between two objects. Each priority is indicated by a particular class type and each class type can be either positive or negative. Negative relationships are those that keep entities apart and positive relationships are those keep entities together, in the factory floor layout problem. These negative relationships might be due hazardous chemical interaction noise pollution to another machine or any such difficulty in keeping two machines in close proximity. Typical positive relationships are due to optimization of the material flow between machines in a sequence. Tale (1) shows an example matrix for the twelve entity problem where I,u,x represent the relationships between machines.

| | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | M11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| M2 | X | | | | | | | | | | |
| M3 | X | U | | | | | | | | | |
| M4 | I | X | X | | | | | | | | |
| M5 | I | X | X | U | | | | | | | |
| M6 | X | X | U | U | U | | | | | | |
| M7 | X | I | I | X | X | U | | | | | |
| M8 | X | U | U | X | X | U | I | | | | |
| M9 | I | X | X | U | U | U | X | X | | | |
| M10 | U | U | X | X | U | U | U | U | U | | |
| M11 | U | U | U | U | U | U | U | U | U | U | |
| M12 | U | U | U | U | U | U | U | U | U | I | U |

**Table (1): weight matrix for twelve entity where i= 20 , u = 0 , x = -20**

## 4- Approaches to Finding the Machine Layout

- Quadratic Assignment Problem (QAP)

Quadratic assignment problem is the most used method Classically, the QAP attempts to allocate N object in N discretely partitioned areas, one object per location, the input relationships are similar or that of the advance layout problem and the goal is also to optimize these relationships by placing the N object onto the designated areal[2]. Evolutionary methods have been applied to this problem with great success [FBT93, YP94]. The advance layout problem, however, places the objects in a continuous space instead of discrete partitions of the QAP[2].

- Traditional Methods

The factory layout problem is similar to the Quadratic assignment problem but must also find clustering information that is place in groups such as departments, into a typically unrestricted floor plan. This has been examined in detail and genetic algorithms have been shown to be superior to traditional methods [2].

- Spiral Algorithm

  One of the better factory layout problem algorithms is called SPIRAL, it attempts to place object in graph with arcs extending only in hexagonal direction [4]. The algorithm is much like the ALP in that it is used to search for an optimal layout of machines. SPIRAL, however, has a fixed are length so that placement is not continuous. SPIRAL also uses an interactive approach and requires additional user post processing. Although the purposes of SPIRAL and the advance layout problem are not the same , an attempt is made in the results section to compare the two methods even though comparisons might not be fair to either methods, and so should be considered only as a general guideline[5].

- Conventional Approaches

  The conventional technique to find the best layout possible involves the use of diagrams and graphs. A detailed analysis needs to be done on the routing of parts, volume moved, distance, frequency of move, rate of which the part travels and cost of the move. Some common techniques used are assembly charts, which give the graphical representation of the flow of parts and from/to charts, etc[6]

- Quantitative Approach

  The quantitative approaches on the other hand, are primarily concerned with optimizing the location of equipment with the movement of material, and include techniques from the operations research in addition to mathematical techniques. The measure of effectiveness used in the quantitative approach is the distance traveled by the parts [6]

**Other Techniques**

More recent approaches to solve the machine layout problem include simulated annealing, and genetic algorithms [6].

## 5- The Proposed Methods

Implementation of hybrid genetic algorithm (HGA), to attain the purpose of this paper the following method was employed. Each with both sets of parameters and operation type is executed 16 trails.

After studying the genetic operation group that includes (selecting, crossover and mutations), I found out the following:- HSG works in a better way with the selection of the roulette wheel (RWS), and HSSG works in a better way with the selection of binary tournament sized 2(B.T).
In general, two a methods have been applied to solve advance layout problem.

## 5.1 The first method is Hybrid Steady State Genetic (HSSG)

## Algorithm Description

Basically the proposed algorithm in this study in the further development for hybrid GA called HSSG. As the HSSG has been discuses which may be used as reference to explain in mechanism of HSSG, it is decided here in to only give a brief of description for the implementation process of HSSG as follows:-

1- Initiations (population )
2- Evaluation (population)
3- Generations = 0
4- While not stop criteria on do
5- Selection (population, parent); Binary tournament selection
6- Crossover (parent, offspring, pe)
- Without local optimizer (2x)
- With local optimizer
7- Mutation (offspring, pm);
8- Separation algorithm;
9- Evaluation (offspring);
10- Replacement (population , offspring)
11- Generation = generation +1;
12- End while
13- Extract the best individual
14- End

- **Encoding**
  Chromosome representation is based on machine numbers, where the location of the gene represents the location of the machine.
- **Fitness function**
  The fitness takes the distance between each pairs of machines and finds the percentage of correctness of that distance relative to optimal distance indicated by their relationship matrix. The correctness is simple (1-error). This correctness is then multiplied by relationship weight and added to the fitness for the given layout. The total fitness after all relationships have been measured is then divided by the total weight represented in the relationship matrix. The final fitness is guaranteed to be zero ( theoretical worse fitness) to one ( theoretical best fitness ). Usually the theoretical best fitness is not attainable because every relationship cannot be optimized, due to competing relationship[2].
- **Local optimizer**
  A local optimizer is used to prevent repetition of the machines. The particular optimizer used for this problem uses greedy algorithm.
  First step: staring point (first gene ) is to be selected randomly from one of the parents OR starting point "first gene" would be chosen according to the biggest error sums for one of the parents. In both cases it would be considered the starting point for the child tour (first gene of the child ).

  Second step: comparing the two edges that leave the starting points in the parents, choosing the edge with less error (current gene) and putting it in the child. Before putting the current gene(c ) in a child it should be compared with other genes present in him, if repetition occurred than we shuld move for the following gene (c+1) and compare it with the gene of the other parent, choosing the less error edge, comparing it with the child's genes and thus we continue this process until this extension may lead to from a cycle. If no repetition occurred, then we should continue partial tour by choosing the less error edge of two edges in the parents which mean extending the tour.

- **Separation Algorithm [2]**

The separation algorithm first sets a priority schedule according to the distance of the objects from the center of the floor. Those objects close to the center of the floor are considered volatile and thus are given lowest priority. While those in corners of the floor are considered fairly stable. This ranking is also similar to measuring the distance that the machine has changed since that last iteration.

Next, objects are placed according to the priority schedule, with highest priority being placed first. If the object overlaps with another machine, it is moved in the direction in which the two machines overlap iterated until the object is clear of any other machines. After all objects are placed safely. The algorithm terminates.

### 5.2 The Second Methods Hybrid Simple Genetic (HSG)

### Algorithm Description

Basically the proposed algorithm in this study in the further development for hybrid GA called HSG. As the HSG has been discuses which may be used as reference to explain in mechanism of HSG. It is decided here in to only give a brief of description for the implementation process of HSSG as follows:

1- Initiations ( population )
2- Evaluation ( population )
3- Generations = 0;
4- While not stop criteria on do
5- Selection (population, parent);
Binary tournament selection
6- Crossover (parent, offspring, pc),
- Without local optimizer (2x)
- With local optimizer
7- Mutation (offspring, pm);
8- Separation algorithm
9- Evaluation (offspring)
10- Population = offspring;
11- Generation = generation +1
12- End while
13- Extract the best individual
14- End

The description of the HSG model is similar to that of HSSG in all its components but with some differences in HSG will include the RWS, instead of tournament (B.T.), the replacement strategies (generational ).

## 6- Experimental results

The hybrid genetic algorithms were run on two different problems (twelve and forty- five entity layout). All problems were run by using (HSSG, HSG).

The first problem the twelve- entity layout problem is theoretical problem in which there are two dependent near global maxima without rotation. If rotation is included then this number id doubled due to symmetry. The term near optimal is used because small adjustment can be made to reach at rue optimal. The other global optimum is similar but has two independent pairs of machines grouped in one corner. The genetic algorithm with local optimizer converges one of these two near global optima. The second optima can be not globally optimal but is within a small epsilon of the global optimum [2][7].

The twelve- machine layout problem was run both with and without local optimizer. The results of sixteen trials each with a population size of (20). Probability of crossover (0.8), probability of mutation (0.1). length of the areal (27) width of area (18). Radius of the area (2). Were average for each method and are given in table (2). Predetermined near global optima was based on test received results and the run was terminated after this mark was met, after specific generations. The numbers that reached then near the average of the best fitness over sixteen runs, the five column indicated the average number of generations to reach the fitness given.

| M.N. | Operator name | Optimal | Best fitness | Gen |
|---|---|---|---|---|
| HSG | Local optimize | (16/16) | 0.931368044074 | 163.75 |
| | Non Local optimize | (0/16) | 0.608628317341 | 1106.25 |
| HSSG | Local optimize | (16/16) | 0.930795232000 | 56.25 |
| | Non Local optimize | (0/16) | 0.56866408122 | 1578.75 |

**Table 2: Results of the Twelve- Entity Layout Problem in HSSG and HSG**

The hybrid steady state genetic and hybrid simple genetic algorithms with local optimizer clearly performed much better than the hybrid steady state genetic and hybrid simple genetic algorithms alone. The local optimizer is the reason for much of the success in this project, which can be seen simply from the fact that most of the fitness's in the random population with local optimizer are close to the best value achieved without local optimizer. These locally optimized values, however, are not fit enough to be global optima even in large scale search space, as is evident by the number of generations taken to reach the optimal mark.

The second problem that the hybrid steady state genetic and hybrid simple genetic algorithms were tested on the forty- five entity layout problem which is again a theoretical problem but one that more closely resembles the typically attributed larger scale machine layout [2][7].

**Table (3) show the solution found by the genetic.**

| Operator name | Optimal | Best fitness | Gen | M.N. |
|---|---|---|---|---|
| Local optimize | (16/16) | 0.92498701021 | 312.5 | HSG |
| Non Local optimize | (0/16) | 0.855891202431 | 881.25 | |
| Local optimize | (16/16) | 0.930587010519 | 250 | HSSG |
| Non Local optimize | (0/16) | 0.848537431724 | 2187.5 | |

**Table (3): Results of the Forty –Five-entity layout problem in HSG and HSSG**

This problem was run similar to twelve –entity layout problem with predefined optimal fitness value. Due to the increased complexity of the problem , however , the maximum number of generations increased and the number of individuals in the population increased as well as . table (4-4) and table (4-5), show the results of hybrid simple genetic and hybrid steady state genetic algorithms with and without local optimizer. Again , the methods with local optimization perform significantly better than the hybrid simple genetic and hybrid steady state genetic algorithms without local optimization . The hybrid steady state genetic and hybrid simple genetic algorithms were run once with parameters similar to what has used before . A mutation rate (0.2), crossover rate (0.8),pop size (80), length of the area (27), width of the area (18) , radius of the area (2) and the results of the sixteen trials.

## 7. Comparative Study

The results of sixteen trails each with a population size of (20,80), pc (0,8) , pm (0.1, 0.2) from using parametric study (parameter found by using parametric study). Generally the parameters are the same as the other methods.

| Operator name | Method name | Optimal | Best fitness | Ava.Gen |
|---|---|---|---|---|
| Non optimize | Proposed method in HSSG | (0/16) | 0.631684071958178 | 1018.75 |
| | Proposed method in HSG | (0/16) | 0.584380596777261 | 208.125 |
| | E.Goodman | (0/16) | 0.74822656259 | 1000 |
| Local - optimize | Proposed method in HSSG | (16/16) | 0.92201772538331 | 52.625 |
| | Proposed method in HSG | (16/16) | 0.9349474684091 | 29.0625 |
| | E.Goodman | (16/16) | 0.845813625 | 173.125 |

**Table (4): The Comparison between HSSG, HSG and previous works for the Twelve- Entity Layout problem.**

The results of the hybrid simple genetic and hybrid steady state genetic algorithms with and without local optimizer , as well as the results of the SPIRAL methods and the hand placed results of an expert in the field for comparison of the basic layout capabilities , Again , the method with local optimization performs significantly better than the hybrid simple genetic and steady state genetic algorithm without local optimization in both methods (HSSG,HSG). The results of the hybrid steady state genetic and hybrid simple genetic algorithms with local optimization were marginally better than E.Goodman . The SPIRAL method left too much space between machines to compete with the locally minimized algorithm.

| Operator name | Method name | Optimal | Best fitness | Ava.Gen |
|---|---|---|---|---|
| Non optimize | Proposed method in HSSG | (0/16) | 0.86873373792768 | 618.75 |
| | Proposed method in HSG | (0/16) | 0.85534196303 | 287.75 |
| | E.Goodman | (0/16) | 0.8252123125 | 2000 |
| Local - optimize | Proposed method in HSSG | (16/16) | 0.9244187114164 | 106.25 |
| | Proposed method in HSG | (16/16) | 0.916663071485 | 87.5 |
| | E.Goodman | (16/16) | 0.9123779375 | 506.125 |
| | Hand optimizer | N/A | 0.907205 | N/A |
| | Spiral algorithm | N/A | 0.852815 | N/A |

**Table (5): The Comparison between HSSG, HSG and previous works for the forty-five-- Entity Layout problem.**

The hybrid steady state genetic and hybrid simple genetic algorithms without the local optimizer were not even able to get to the level of the SPIRAL.

Even though the hybrid steady state genetic and hybrid simple genetic algorithms with local optimizer improved over the other methods in comparison , it is evident that a small amount of optimization is still possible. This advancement however would only yield an improvement of fitness score. This positioning of a single object with many relationships tends to affect total score more than one single relationship as this . such minor improvement are usually improved by larger populations.

## 8. Conclusions

The following points can be concluded from the research :

1- In this research , we have solved the machine layout problem by using GA.In comparison with the proposed hybrid available techniques that can be used for solving this problem , the genetic algorithms perform significantly better.

2- In this research we applied two methods of GA(HSSG,and HSG) and we found that HSG , and HSSG carry specific attributes (adjective or character). Also we noted that the HSGis more efficient than the HSSG because it taken a less average number of generations. On other hand , we can say that the HSSG is more efficient from than the HSG in time execution programs.

3- The advantage of the proposed algorithm is that the machine layout can be determined using minimal amount of data, offering an advantage in areas where the cost of changing from one object to another is noticed to lack availability of the data on the cost incurred per distance of movement, also to find the relative location of machines in designated areas.

4- The best fitness function was selected among many of fitness functions which were studied in this research.

5- The advance layout problem adds features to spatial layout problem such as continuous placement and heterogeneous objects.

## 9. References


1 Christos Papahristodoulou " A Pare Binary LP Model to the facility Layout Problem ", Department of Economic , Uppsala University, Email : Christos. Papahristodo.ulou@nek.uu.se,1997.

2 E.Goodman, " an introduction to GALOPPS the Genetic Algorithm optimized for portability and parallelism system", Tech, Report, Michigan state University, 1994.

3 Baidaa Abd Al-Kaleck, "Comparison between Genetic algorithms(GAs) and Adaptive Simulated Adaptive simulated annealing(ASA) In Training Neural Network", Master Thesis, 2005.

4 M. Geotshalckx, "An interactive layout huristic based on hexagonal adjacency graphs", in European Journal of Operational Research , vol 63,1992,pp: 304-321.

5 Imad F.T.Yassen,"Solution of Some Hard Problems Using Genetic Algorithms (GA)", Ph.D. Thesis Dep of computer science ,University of Pune .India,1998.

6 Norhashimah Morad , " Genetic Algorithm Optimization for the Machine Layout Problem", School of Industrial Technology University sains Malaysia 11800, Minden , and Penang Malaysia 1995.

7 Dr.Mohamed El-sherbin Ali ,"Neural network and Genetic Algorithms", lectures , 2003.