Genetic - Local Hybrid Optimizer for Solving Advance Layout Problem

Dr.Imad.F.T.Yasseen*

Hind Saleem Ibrahim Habra**

Abstract

Advance layout problem (Alp) is used to search for an optimal layout of machines. This Research Describes a novel method, based on genetic algorithms (GA) to solve the machine layout problem, where developing machine layout is an important step in designing manufacturing , renovation of factories, distribution centers, hospitals, banks, department stores, military supply, depots, university, ect .The research studies the problems of adding the heterogeneous objects, continuous placement in the general spatial layout problem. Results are achieved through the use of local optimizer, separation algorithm with genetic algorithm also called hybrid simple genetic. Results show the potentiality of the proposed algorithm in solving the problem and outperforming previous algorithms.

خوارزمية جينية هجينة لحل مشكلة التوطين المكاني المتقدمة

الخلاصة

مسألة التوطين المكاني المتقدمة تكون مستخدمة كطريقة بحث لتوطين أمثل للالات. يتم وصف طريقة حديثة معتمدة على الخوارزميات الجينية في حل مسألة توطين الالات ،حيث تطوير هذه المسألة تعتبر خطوة مهمة في تطوير الصناعات ، المصانع،توزيع المراكز، المستشفيات،... الخ. هذا البحث يدرس المشاكل بالاضافه الى الكيانات الغير متجانسة ، الاماكن المستمرة في مسألة التوطين المكاني.النتائج تحقق من خلال استخدام أمثلي محلي و خوارزمية الفصل مع الخوارزميات الجينية هذه الطريقة يطلق عليها الخوارزميات الجنية الهجينة البسيطة.

النتائج تبين احتمالية الخوارزميات المقترحة في حل مسالة هذه المسالة تفوق انجازية الخوارزميات السابقة.

* Head Dept. of Computer information systems Al-Rifdain University Collage

** College of Science Al-Mustansiriyah University

Imadyasseen39@hotmail.com,Hindsaleem@yahoo.com

1. Introduction

Spatial layout problems affect architects, physicists and many other professionals in their attempts to find optimal molecular configurations or arrange components for micro-circuitry design. Spatial layout problems occur in many felids including component placement in micro circuitry design, modeling of theoretical physics problems such as the C 60 Bucky ball [Dea95],and operations research problems like the factory floor layout problem.

A genetic algorithms (GA) is an adaptive search technique, which imitates the process of biological evolution (Goldberg (1989)). A method using genetic algorithms has been developed to solve the machine layout problem[Col89][Mel98][Zbi96].

2.Definition of the Advance Layout Problem:-

The objective of the Advance 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, repitition between the circles. The distance between all pairs of entities are optimized according to weight matrix A [Goo94].

The weight matrix (A) [Goo94,VO97]: 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 that keep entities together. In the factory floor layout problem, these negative relationships might be due to hazardous chemical interactions 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. Table 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	I	U	U	U	U	
M12	U	U	U	U	U	U	U	U	U	I	U

Tabel1:Weight Matrix for twelve entity where, I=20,u=0,x=-20

3.The Proposed Methods

Hybrid GAs have been Known as the effective optimization technique for solving the complicated optimization problems**.**

Hybrid Simple Genetic (HSG)

Algorithm Description

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

```
Initialization (population);
Evaluation (population);
Generation ← 0;
While not stops criteria on do
        Selection (population, parent);
        Crossover (parent, offspring, pc);
                   - Non optimize(2x)
                   - local optimizer(greedy algorithm)
          Mutation (offspring, pm);
                   -Separation Algorithm; // only used with local  optimizer
 Evaluation (offspring);
 Population ← offspring;
Generation ← generation+1;
End while
```

♣ **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 simply (1-error). This correctness is then multiplied by relationship weighs and added to the total 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 relationships [Goo94].

♣ **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 staring 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 staring 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 then we should 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

Finally, separation algorithm is used after the mutation to prevent overlap of the machines.

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 the last iteration. Both work well, but the first method is used because it is independent of the local optimizer.

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 and is moved as little as possible to remove the overlap. This process is iterated until the object is clear of any other machines. After all objects are placed safely, the algorithm terminates.

5.Experimental Results

The hybrid simple genetic was run on two different problems (twelve and forty- five entity layout).

The first problem the twelve-entity layout problem is a theoretical problem in which there are two independent near global maxima without rotation. If rotation is included then this number is doubled due to symmetry. The term near optimal is used because small adjustments can be made to reach a true optima. The other global optimum is similar but has the 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 [Goo94].

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 area(27),width of the area(18),radius of the area(2), were averaged 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 the near optimal goal are indicated in the first column. The second column, indicate the average of the best fitness over sixteen runs, the third column indicates the average number of the generations to reach the fitness given.

	optimal	Best fitness	Gen
Local optimize	(16/16)	0.930795232000	56.25
Non local optimize	(0/16)	0.56866408122	1578.75

Table 2 :Results of Twenty Layout Problem

The hybrid simple genetic with local optimizer clearly performed much better than the 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 fitnesses 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 simple genetic was tested on the forty –five entity layout problem which is again a theoretical problem but one that more closely resembles the typically attributes larger scale machine layout . Table 3 show the solution found by the genetic**.**

Operator name	Optimal	Best fitness	Gen
Local optimize	(16/16)	0.92498701021	312.5
Non Local optimize	(0/16)	0.855891202431	881.25

Table 3: Results of The Forty-Five-Entity Layout Problem

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 3 shows the results of hybrid simple genetic with and without local optimizer. Again, the method with local optimization perform significantly better than the hybrid simple genetic without local optimization. The hybrid simple genetic was run once with parameters similar to what has used before. A mutation rate (0.2), crossover rat (0.8), pop size (80), length of the area(27),width of the area(18),radius of the area(2), and the results of sixteen trials.

6. Comparative Study

The results of sixteen trials 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 HSG	(0/16)	0.584380596777261	208.125
	E.Goodman	(0/16)	0.74822656259	1000
local-optimize	Proposed method in HSG	(16/16)	0.9349474684091	29.0625
	E.Goodman	(16/16)	0.845813625	173.125

Table4: The Comparison Between HSG and Pervious Works For The Twelve-Entity Layout Problem

Operator name	Method Name	Optimal	Best fitness	Gen
Non-Optimize	Proposed method in HSG	(2/16)	0.85534196303	287.75
	E.Goodman	(0/16)	0.8252123125	2000
Local Optimizer	Proposed method in HSG	(16/16)	0.916663071485	87.5
	E.Goodman	(16/16)	0.9123779375	506.125
	Hand optimize	N/A	0.907205	N/A
	Spiral algorithm	N/A	0.852815	N/A

Table 5: Comparison between H SG and pervious work for the forty-five-entity layout problem

Table 4 and table 5, show that :
The results of the hybrid simple genetic (HSG) 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 without local optimization. The results of the hybrid simple genetic with local optimization were marginally better than the expert's in the field and significantly better than the of SPIRAL method's better than E.Goodman. The SPIRAL method left method left too much space between and would need an algorithm to optimize the distance between machines to compete With the locally minimized algorithm. The hybrid simple genetic without the local optimizer was not even able to get to the level of the SPIRAL method.

Even though the hybrid simple genetic 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 the fitness score. The 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 .

7.Conclusions

The following points can be concluded from the 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.
- ♣ . 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 data on the cost incurred per distance of movement, also to find the relative location of machines in designated areas.
- ♣ The best fitness function was selected among many of fitness functions which were studied in this research.
- ♣ The advance layout problem adds features to spatial layout problem ;such as continuous placement and heterogeneous objects.

Reference

[Dea95] D.M.Deaven and K.M., Ho,"*Molecular Geometry Optimization with a Genetic Algorithm*", in physical review letters, 10, july 1995.

[Goo94] E.Goodman,"*an introduction to GALOPPS the' Genetic Algorithm optimized for portability and parallelism' system*", Tech. Report, Michigan state university, 1994. Email **http://isl.cps.mus.edu/GA.**

[Nor95] Norhashimah Morad,"*Genetic Algorithms Optimization for the Machine Layout Problem*", School of Industrial Technology Universities Sains Malaysia 11800, Minden, and Penang Malaysia 1995. E-mail: nhashima@usm.my.

[VO97] Volker Schnecke and Oliver Vornberger, "*Hybrid Genetic Algorithms for Constrained Placement Problems*", IEEE1998 Appeared in IEEE Transactions on Evolutionary Computation, Vol.1, No.4, November 1997, pp.266-277.

[XLW01] Y.G.XU,G.R.Liu and Z.P.Wu,"*Anovel Hybrid Genetic Algorithm Using Local Optimizer Based On Heuristic Pattern Move*",Appl.Artif.Intell.-an Int.J.vol.15,pp.601-631,2001.

[Gol89] D.E.Goldberg,"*Genetic Algorithm in search, optimization, and Machine learning*", Addison-Wesley, 1989.

[Mel98] Melanie Mitchell, "*An Introduction to Genetic Algorithms*", Printice- Hall, India, 1998.

[Zbi96] Zbigniew Michalewicz,"*Genetic Algorithms +Data Structures= Evolutionary programs*", 1996.