ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp.53 - 59
doi: 10.18178/wcse.2017.06.009

Dynamic Layout Optimization Based on Improved Genetic Algorithm

Yunfang Peng and Lingzhi Fan+
School of management, Shanghai University

Abstract. In order to reduce the total cost of the workshop layout under dynamic environment, a dynamic layout optimization model is established to minimize the cost of material handling and equipment adjustment. An improved genetic algorithm is proposed to solve the model, and the initial population with high fitness is generated to reduce the search time. The effectiveness of the improved genetic algorithm and the accuracy of the model are verified by some numerical examples.

Keywords: dynamic facility layout optimization, Basic genetic algorithm, improved genetic algorithm

1. Introduction

With the increasingly fierce market competition and the growing demand of customers, the demand for production is developing towards the direction of diversity. In order to solve the problem of how to adapt to the dynamic change of production requirements, the layout optimization problem in dynamic environment is proposed. According to the relevant data， it is shown that 40% of corporate sales from new products [1]. As a result of these changes, the cost of material handling will be increased, which makes the efficiency of the current production layout low. Therefore, it is of great theoretical and practical significance to study the layout of workshop in dynamic environment.

At present, the research method on the layout problem of the workshop in dynamic environment can be divided into dynamic approach and robust approach [2]. Dynamic layout optimization is to adjust the layout of workshop facilities according to the demand of production in different stages, which minimizes the total cost of the workshop layout and the cost of material handling in the whole stage. Robust layout is to keep one layout constantly corresponding to different stages of production needs. Among them, it is not necessarily the best solution to a certain stage, but it is the best overall layout in the whole period [3]. The robust approach is mainly suitable for small batch, multi-stage manufacturing. But in today's market environment, many industries are required to meet the customer's large-volume demand, so dynamic layout optimization is more suitable for them.

In recent years, the research ondynamic layout optimization has been developed in many directions. Assuming that the system is independent to each other in the facility layout problem, some scholars have proposed a new model which gives a detailed calculation of transportation distance, cost, and volume. [4]. In addition, some scholars have used the confidence level P, theory of reliability and the subjective factors of managers to further improve the classic model [5]. Some scholars set the flow rate as a random number and study the stability of the production line and the ability to resist risks from the perspective of the entire production line [3]. Xi Yao introduced the signal to noise ratio into the dynamic layout optimization problem. Based on the signal to noise ratio, decision makers can judge the stability and anti risk ability [6]. Other scholars put the problem of material delivery system into the dynamic layout optimization problem and consider the renewal of conveying equipment. These considerations make the dynamic layout optimization model closer to the actual production [7]. On the basis of the original model, Saeed uses the multi-objective

+ Corresponding author. Tel.: +15201935829.
E-mail address: fanlingzhi@i.shu.edu.cn.

model, and puts forward the distance coefficient [8]. Fariborz and other scholars consider the dynamic layout problem under the condition of unequal size, a calculation method is put forward with the actual logistics distance step function. Based on the mathematical model to describe the different attitude of the production facilities, can more accurately determine the specific direction of each production facility in different time periods [9].Most of the workshop layout problem is a NP-hard problem, the general heuristic algorithm is difficult to obtain a satisfactory solution. At present, genetic algorithm and improved genetic algorithm are widely used, and it has been widely used in dynamic layout optimization problems [10]. However, the speed of the algorithm is usually slow because the search space is too large. In this paper, a population initialization strategy is proposed to reduce the search space and improve the efficiency of solving the model.

2. Dynamic Layout Optimization Problem

2.1. Problem description

Select the production facilities and transportation equipment as the research object. The entire production cycle includes T stages, I production facilities and G kinds of conveying equipment, production facilities need to be placed in the L position. Assumptions are as follows:

(1) At any stage, a facility can only be placed on a single location. A location can only be placed on one facility.

(2) If there is material transportation between two facilities, only one conveying equipment must be selected.

2.2. Mathematical model

The classical dynamic layout optimization problem is based on the objective of minimizing the total material cost and the minimum cost[11]. According to the classical dynamic layout model, material delivery costs as equation (1), production facility adjustment cost as equation (2), objective function of the model as equation (3), equation (4) ensures that each production facility can only be placed in one position at each stage, equation (5) ensures that a location can be placed at a maximum of one production facility at each stage, equation (6) ensures that at most one conveyor can be used between two facilities at each stage, equation (7) ensures that when there is a material requirement between two production facilities at any stage, a conveying device must be used for material handling.

$$MHC = \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{j=1}^{I}\sum_{k=1}^{L}\sum_{h=1}^{L}\sum_{z=1}^{g} x_{tik}x_{tjh}y_{tzij}D_{tkh}Q_{ijt}P_{tkhz} \tag{1}$$

$$AC = \sum_{t=1}^{T-1}\sum_{i=1}^{I}\sum_{j=1}^{I}\sum_{k=1}^{L}\sum_{h=1}^{L} x_{tik}x_{t+1ih}C_{kh} \tag{2}$$

$$TC = \min\{MHC + AC\} \tag{3}$$

$$\sum_{k=1}^{L} x_{tik} = 1, \forall t, i \tag{4}$$

$$\sum_{i=1}^{I} x_{tik} \le 1, \forall t, k \tag{5}$$

$$\sum_{z=1}^{g} y_{tzij} \le 1, \forall t, i, j (i \ne j) \tag{6}$$

$$x_{tik}x_{tjh}Q_{tij}\sum_{z=1}^{g} y_{tzij} = Q_{tij}, \forall t, i, j, k, h, z \tag{7}$$

In model: i and j represent workshop production facilities, $i,j \in [1,2,\ldots,I]$. k and h represent workshop facilities location, k, $h \in [1,2,\ldots,L]$. z indicates the kinds of equipment used, $z \in [1,2,\ldots,g]$. D_{tkh} indicates the amount of material transferred from position k to position h in the T phase, P_{tkhz} indicates the cost of conveying unit material from position k to position h in the T stage by using the z conveyor, Q_{tij} indicates the total material requirement from the facility equipment i to the facility equipment j during the T stage. x_{tik} is 0-1 decision variable, $x_{tik} = \begin{cases} 1 & \text{facility equipment } i \text{ placed in position } k \text{ during the } T \text{ stage} \\ 0 & \text{Other} \end{cases}$. y_{tzij} is 0-1 decision variable, $y_{tzij} = \begin{cases} 1 & \text{production facilities } i \text{ and facility } j \text{ using conveyor equipment } z \text{ during the } T \text{ stage} \\ 0 & \text{Other} \end{cases}$.

3. Design improved genetic algorithm

Genetic algorithm is a heuristic algorithm based on biological evolution, it consists of three basic operations: selection, crossover and mutation [12-13]. The basic methods of operation are different, in order to improve the efficiency of the algorithm, proportional fitness assignment are used in the selection phase. Since the decision variables are all 0-1 variables, a single point crossover is used in the crossover phase and binary mutation is used in the mutation phase. The optimization criterion of genetic algorithm is based on the hypothesis that the world algebra exceeds the preset value. The maximum number of generations is set to 50. The specific flow chart of the algorithm is shown in figure 1.

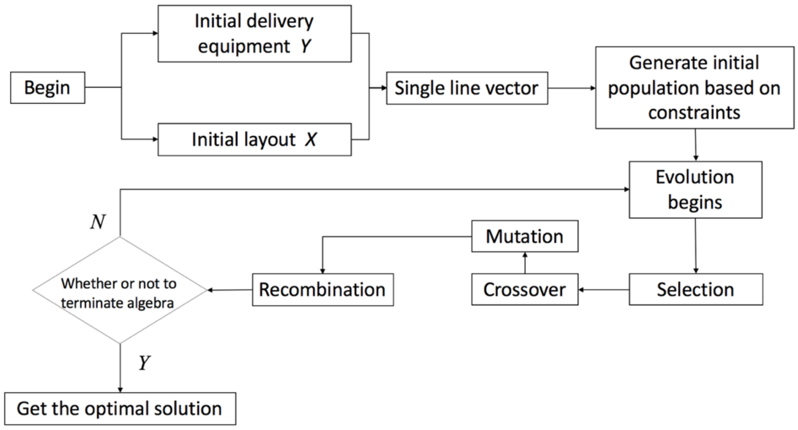

Fig. 1: Flow chart of algorithm.

3.1. General algorithm

Step one: randomly generated initial population, a certain number of individuals, each individual is expressed as a chromosome gene coding.

Step two: calculate the fitness of the individual, and determine whether the optimization criteria is fitted. If it meets the criteria, then the optimal solution of the optimal individual and the result is produced, and the computation is finished, otherwise it turns to the third step.

Step three: regenerate individual based on the fitness, individuals with high fitness are selected with high probability and individuals with low fitness may be eliminated. Roulette selection is used in this phrase.

Step four: generate new individuals based on the certain crossover probability and crossover method based on single point mutation.

Step five: generate new individuals based on the certain mutation probability and mutation method.

Step six: a new generation consists of individuals by crossover and mutation , and return to the second step.

3.2 Improved algorithm

In this paper, the decision variables of the model are all 0-1 variables, so the coding difficulty of genetic algorithm is reduced, and the precision of the algorithm is reduced, so the coding length is shorter which can greatly improves the efficiency of the algorithm. Among them, the expression of chromosome is $\{x_{111}, \ldots, x_{TIL}, y_{1111}, \ldots, y_{TgII}\}$, the length is *TIL*+*TgII*. In order to speed up the solution, according to the characteristics of the model, improved population initialization. According to the characteristics of decision variables, the individual chromosomes were divided into two segments, the first is decision variable x_{tik}, the second is decision variable y_{tijz}, generate the initial population with constraint conditions. The following strategies are adopted for the initial population:

(1) In a single chromosome, the gene at the first *TIL* position is decision variable x_{tik}: First generate *T* *L***L* dimensional matrix, each matrix is composed of only 0 and 1, and the sum of each row is 1, then form a three-dimensional array P_1. After each matrix randomly selected *I* line, which I<L, form *T* *I***L* dimensional matrix. Then form a three-dimensional array P_2. Then the array P_2 is arranged according to the sequence of variables in the chromosome to form the initial chromosome.

(2) Genes located in the posterior *TgII* position are decision variables y_{TgII}: According to (1), first generate *I**I* Max*(*T*, *g*)* *Max*(*T*, *g*) dimensional matrix, each matrix is composed of only 0 and 1, and the sum of each row is 1, then form a four-dimensional array P_3. After each matrix randomly selected *Min*(*T*, *g*) line, form four-dimensional matrix P_4. According to the constraint condition (7), the assignment operation is carried out again, and the four dimensional array is formed according to the order of the variables in the chromosome.

(3) According to the steps of (1) and (2) generate the vector of the two segments, and then combine the two vectors to form a chromosome, and then repeat the operation *N* times to generate the initial population.

3.3. Fitness function

The model involves the cost of material handling in the workshop *MHC* and equipment adjustment cost *AC*, take right set, the objective function is reduced to equation (3). Genetic algorithm is an unconstrained optimization algorithm, but in the model, there exists a constraint of (4) ~ (7). So in the solution process, the penalty function is introduced to deal with the constraint conditions. The fitness value of population *N*(*i*) *Fit*(*N*(*i*)) is calculated as follows:

$$Fit(N(i))=1/(PK+Ob) \tag{8}$$

$$PK(i)=\begin{cases} 0 & no \\ inf & When\ the\ population\ is\ not\ satisfied\ (4)\sim(7)\ constraints \end{cases} \tag{9}$$

Equation: *PK* is a penalty function, *inf* is a large natural number.

4. Example Analysis

Because there is no standard example, so sets up different examples as article [7], and compares the basic genetic algorithm with the improved genetic algorithm proposed in this paper. A total of 15 examples of different sizes, and through sensitivity experiments, the algorithm parameters are set as follows: the number of population is 20, the evolutionary algebra is 50, the crossover probability is about 0.7, and the mutation rate is 0.1. The penalty factor is different according to the size of the numerical example. In order to eliminate the influence of the random factors, the average value of the 20 simulation results is compared with the reference as [13]. The effectiveness of the proposed algorithm is verified by a numerical example. The operating environment of the algorithm is as follows: 2.7GHz Intel Core i5, 8G RAM, MacOS Sierra 10.12.1, by matlab2014b.

4.1 Comparison of various scale models

The improved genetic algorithm and genetic algorithm are used to solve 15 different scales, and the results are compared with the results of table 1:

Table1: Comparison of algorithms.

Scale	Basic genetic algorithm		Improved genetic algorithm	
T_I_L_g	Time(s)	Total Cost	Time(s)	Total Cost
2_2_3_2	14.5634	27	3.4521	27
2_4_6_3	29.344	41	8.4437	41
2_6_10_4	56.4692	64	19.3768	64
4_6_10_2	64.441	89	26.72	87
4_8_12_4	90.3476	147	40.457	140
4_10_16_6	149.887	198	58.7843	178
6_10_16_4	190.443	278	77.892	249
6_14_20_6	287.4328	341	94.83	312
6_18_24_8	379.12	399	112.452	349
8_18_24_4	482.45	468	140.177	405
8_20_26_8	590.483	508	180.243	478
8_24_30_10	700.48	680	210.479	648
10_24_30_8	849.21	778	260.434	750
10_28_32_12	1013.4377	948	300.422	938
10_30_36_14	1286.2144	1099	380.4339	1078

Based on the table 1, get a comparison chart, as shown in figure 2:

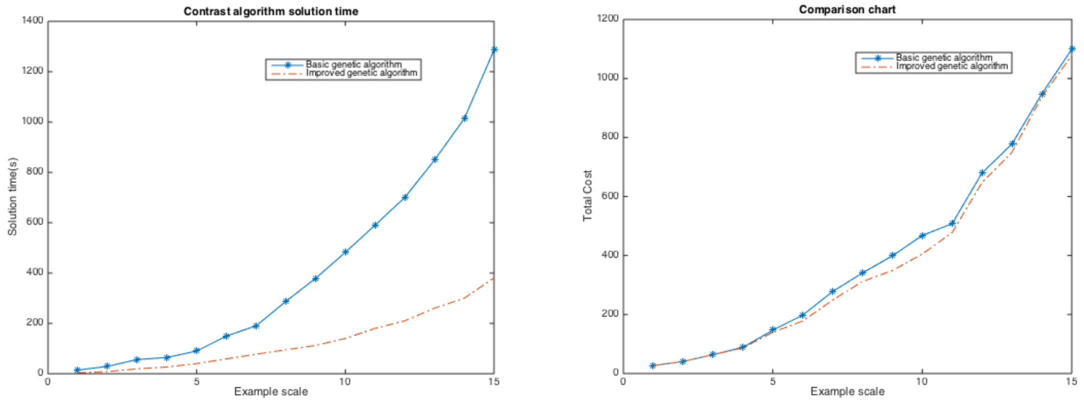

Fig. 2: Algorithm comparison chart.

Above all, it is found that the basic genetic algorithm and the improved genetic algorithm used in this paper have little difference in the validity of the solution, but the improved genetic algorithm is superior to the basic genetic algorithm in solving the speed. Especially in the solution of huge scale numerical example, the improved genetic algorithm in this paper has more advantages in solving speed. Therefore, the algorithm proposed in this paper is more optimized.

4.2 Example results

Taking a small scale example as an example, T=2,I=4,L=6,g=3. Evolution process shown in figure 2. Evolution process shown in figure 2. According to figure 2, the improved genetic algorithm is used to solve the small scale numerical example, the convergence speed is very fast, and the solution can be found in the third generation. The ordinary genetic algorithm is used to solve the problem, and the solution can be found in the eleventh generation. The search speed of the ordinary genetic algorithm is slower than the the search speed of improved genetic algorithm. At the same time, the average fitness of the improved genetic algorithm is obviously higher than that of the common genetic algorithm.

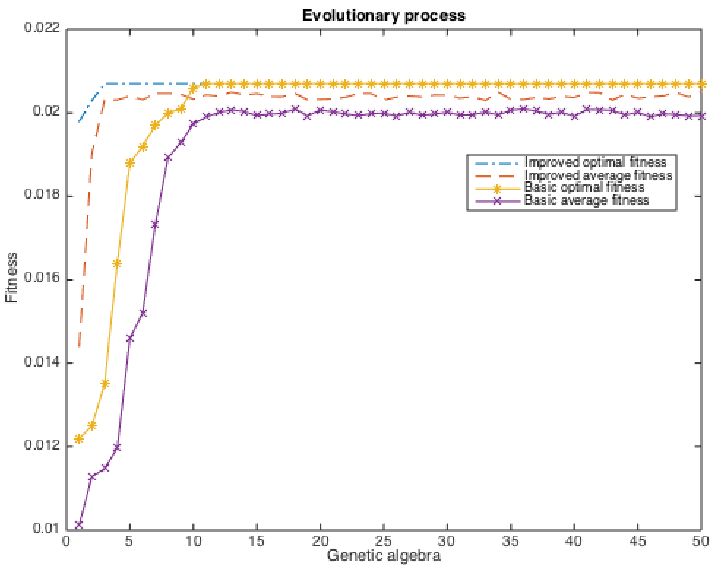

Fig. 3: Small scale evolutionary process.

Taking a huge scale example as an example, T=8,I=20,L=26,g=8. Evolution process shown in figure 3. According to figure 3, it can be found that the improved model can be used to solve large scale numerical examples and its convergence speed is faster and the solution can be found in the Eleventh generation. Ordinary genetic algorithm can search the solution in the sixteen generation. The fitness of the improved genetic algorithm is higher than that of ordinary genetic algorithm. The average fitness of the improved genetic algorithm is obviously better than that of the common genetic algorithm.

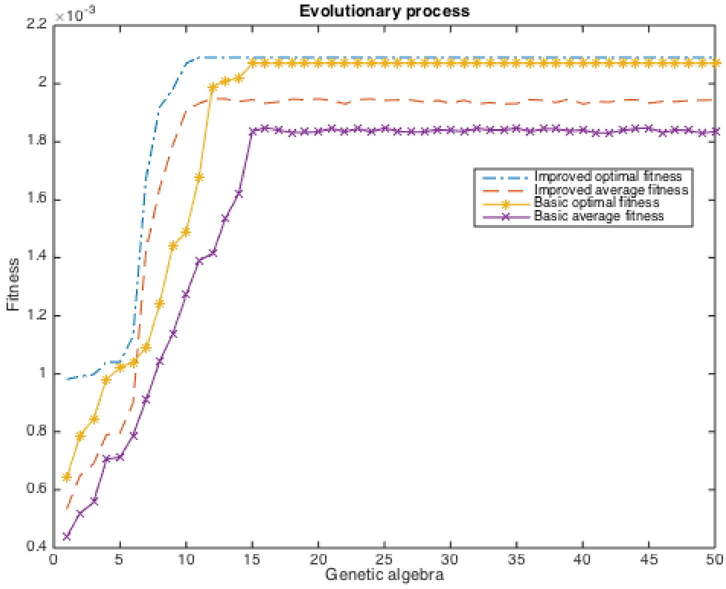

Fig. 4: Huge scale evolutionary process.

Therefore, by comparing the two large and small scale examples, it is found that the improved genetic algorithm is superior to the ordinary genetic algorithm in the search speed, the effectiveness of the solution and the average fitness. The evolutionary process chart of different size examples can be found, as regard to small scale examples, the feasible solution is less and the search space is small. In this paper, the improved genetic algorithm has a high degree of adaptability of initialization population, so the search speed is greatly improved, even in the 1 to 3 generations can be searched for efficient solutions. But, as regard to the huge scale examples, the more feasible solution, so it is difficult to search for high quality solutions in a very short period of time.

5. Concluding Remarks

In this paper, genetic algorithm population initialization strategy is improved, reduce the search space, greatly improve the fitness of the understanding, and improve the speed of solving dynamic layout optimization. By setting different scale examples, the improved genetic algorithm is compared with the basic genetic algorithm. By comparison, it is found that the improved genetic algorithm is superior to the basic genetic algorithm in solving speed. Therefore, the improved genetic algorithm proposed in this paper is more suitable for solving the dynamic layout problem. In addition, the initialization strategy proposed in this paper can be combined with various heuristic algorithms to further improve the search efficiency.

6. Acknowledgements

Project supported by the National Natural Science Foundation, China (NO. 51405283)

7. References

[1] E.Saeed, Nookabadi.Managing a new multi-objective model for the dynamic facility layout problem [J]. *International Journal Advanced Manufacture Technology* ,2013, 68:2215–2228

[2] P. Madhusudanan, I.Hunagund, K.Krishnan. Design of robust layout for dynamic plant layout problems[J].*Computer and Industrial Engineering* ,2011, 61(3):813–823.

[3] P.Kouvelis, A.Kurawarwala, J.Gutierrez. Algorithms for robust single and multiple layout planning for manufacturing systems[J].*European Journal of Operational Research* ,1992, 63(2):287–303.

[4] B.Marcello, Z.Simone, Z.Lucio. Robust versus stable layout design in stochastic environments [J]. *Production Planning and Control*,2005, 16,(1):71–80.

[5] M.Ghorbanali.Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing

systems [J]. *Journal of Intelligent Manufacturing*,2012,23:1849–1860.

[6] Yao Xi, Liu Long, Cen Kefeng. Stochastic facility layout method based on SNR[J].*Industrial engineering and management*, 2012,17 (4):54 – 64.

[7] S.Amir , N.Hamidreza, B.Masume. Dynamic Facility Layout Problem：A New Bilevel formulation and Some Metaheuristic Solution Methods [J]. *Ieee transaction on engineering management*, 2015,62(3)：396-410.

[8] I.Fariborz ,M.Reza, T.Mohammad. A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations [J]. *International Journal of Production Research*,2012,50(15): 4279–4293

[9] Zhou Na, Mi Weijian, Xu Ziqi, Shu fan. The improved adaptive genetic algorithm to solve the multi line equipment layout problem based on [J], *Journal of Shanghai Jiao Tong University*, 2013,47 (12):1924-1929.

[10] Liu Qiong, Xu Jinhui, Zhang Chao. The robustness of the workshop layout based on modified shuffled frog leaping algorithm[J]. *Computer integrated manufacturing system*, 2014,20 (8): 1879 - 1886

[11] H.Mohammad，E.Kourosh，N.Hossein. A bi-objective MIP model for facility layout problem in uncertain environment [J]. *International Journal Advanced Manufacture Technology* ,2015, 81:1563–1575.

[12] K.Mojtaba,M.Vahid ,S.Amiri. A new intelligent algorithm for dynamic facility layout problem in state of fuzzy constraints [J]. *Neural Compute and Applic*, 2014, 24:1179–1190.

[13] R.Alan, H.Artak. Heuristics for the dynamic facility layout problem with unequal-area departments [J]. *European Journal of Operational Research*, 2010, 201:171–182.