Decentralized Genetic Algorithm for Dynamic Plant Layout Problem

Weidong Liang 1, Dianjun Fang 1, 2, *
1 Sino-German School for Postgraduate Studies, Tongji University, Shanghai, China
2 Sino-German Institute of Intelligent Technologies, Qingdao, China

*Corresponding author e-mail: fang@tongji.edu.cn

Abstract. The research of dynamic plant layout problem (DPLP) is to find the optimal layout scheme within multiple periods of production, with the increasing scale of calculation, the efficiency of the traditional centralized algorithm may be decreased. In this paper, we proposed a decentralized genetic algorithm (DGA) which combines genetic algorithm (GA) with decentralized optimization technology. Simulation experiments are carried out to compare the decentralized genetic algorithm with the centralized one, and the result shows that the decentralized method is better than the latter at convergence rate and optimization result. On the other way, the decentralized algorithm is a parallel computing technology, so it can improve computational efficiency especially in high-complexity systems.

1. Introduction
In the manufacturing field, the plant layout problem (PLP) is to study the schemes of arranging devices or facilities to meet the optimization goals such as minimizing the logistics handling costs among devices. With the increasingly complex market environment and the development of intelligent manufacturing technology, the production plan of the workshop fluctuates greatly, so the material handling flow between different devices in a plant may change with the production periods. Therefore, studying the dynamic plant layout problem with multi-period production can effectively improve the production efficiency of the plant.

In this paper, we propose a Decentralized Genetic Algorithm (DGA) that transfer DPLP into a decentralized optimization problem and combines it with the genetic algorithm.

2. Model of DPLP and Decentralized optimization
2.1. Model of DPLP
A multi-line quadratic assignment problem model is used for DPLP which can be described that dividing the layout into equal-area rectangles according to a certain number of lines and one rectangle for one device. For instance, a 6-device layout can be arranged in two lines, each line has three devices (2x3 layout), and use a serial number represent devices, so the layout can be transformed into a string of numbers, such as 245613.

The optimization goal of static PLP is to minimize the material handling costs between different devices, but in DPLP, the material flows between devices are diverse from different production periods,
and during the process of layout changing there exists the shifting or rearrangement costs, therefore, the target function of dynamic layout optimization is shown below:

\[
\min Z = \sum_{t=1}^{P} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{iik}d_{ij}M_{tij}M_{tkl} + \sum_{t=2}^{P} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} A_{tij}Y_{tij}
\]

Subject to:

\[
\sum_{t=1}^{n} M_{tij} = 1, j = 1, \ldots, n, t = 1, \ldots, P
\]

\[
\sum_{j=1}^{n} M_{tij} = 1, j = 1, \ldots, n, t = 1, \ldots, P
\]

\[
Y_{tij} = M_{(t-1)ij}M_{tij}, i, j, l = 1, \ldots, n, t = 2, \ldots, P
\]

Where \( P \) is the number of periods, \( n \) is the number of devices in the layout, \( i, k \) represent the device in the layout, \( j, l \) represent the locations in the layout, \( f_{iik} \) is the material flow cost from device \( i \) to \( k \) in the \( t \) period, \( d_{ij} \) is the distance from location \( j \) to \( l \), \( M_{tij} \) is a 0,1 variable for locating device \( i \) at location \( j \) in \( t \) period, \( A_{tij} \) is the shifting cost of device \( i \) from position \( j \) to \( l \) in the \( t \) period, \( Y_{tij} \) is the 0,1 variable for shifting \( i \) from position \( j \) to \( l \) in \( t \) period.

2.2. Model of Decentralized optimization

First, the centralized optimization problem in an \( n \)-subsystem connected system is defined as:

\[
X_{a} \in \mathbb{R}^{m_{a}} \quad \min \sum_{a=1}^{A} f_{a}(X_{a}, X_{(N_{a})})
\]

Subject to:

\[
g(X) \leq 0
\]

\[
\hat{g}(X) = 0
\]

Where \( A \) is the number of subsystems, \( X_{a} \in \mathbb{R}^{m_{a}} \) is the local optimization variable from ath subsystem, \( X = [X_{1}^{T}, X_{2}^{T}, \ldots, X_{A}^{T}]^{T} \in \mathbb{R}^{m} \) gathers all optimization variables of \( A \) subsystems, \( N_{a} \) is the neighborhoods of ath subsystem. \( g(X) \leq 0 \) and \( \hat{g}(X) = 0 \) are the inequality and equality constraints, respectively.

In decentralized optimization, each subsystem has its own local target function and constraints. Each subsystem is coupled to each other through these constraints. The target function of the system is composed of the local target functions in each subsystem, that is, the global optimization target is to minimize the sum of the target functions of each subsystem. The decentralized optimization problem of ath subsystem is defined as:

\[
X_{a} \in \mathbb{R}^{m_{a}} \quad \min f_{a}(X_{a} | X_{(N_{a})})
\]

Subject to:

\[
g_{a}(X_{a} | X_{(N_{a})}) \leq 0
\]

\[
\hat{g}_{a}(X_{a} | X_{(N_{a})}) = 0
\]

Where functions \( f_{a}(X_{a} | X_{(N_{a})}) \), \( g_{a}(X_{a} | X_{(N_{a})}) \leq 0 \) and \( \hat{g}_{a}(X_{a} | X_{(N_{a})}) = 0 \) represent the functions of \( X_{a} \) when neighboring variables are given.

2.3. Model of Decentralized DPLP

According to the principle of decentralized optimization, the process of optimizing each period for each layout scheme can be regarded as a subsystem which means that there are \( P \) subsystems in a \( P \)-period DPLP. The mathematical formulation for decentralized DPLP is shown as:

\[
x_{t} \in \mathbb{R}^{m_{t}} \quad \min f_{t}(X_{t} | X_{(N_{t})}), t = 1, \ldots, P
\]

Subject to:
\[ g_t(X_t | X_{(N+3)}) \leq 0 \]
\[ h_t(X_t | X_{(N+3)}) = 0 \]

Where \( X_t \) is the layout scheme in the period \( t \) and \( f_t(X_t | X_{(N+3)}) \) is the local target function. Since the first period is the initial layout, there is no device shifting cost, the local target function of each subsystem can be expressed as below:

\[
f_t(X_t | X_{(N+3)}) = \sum_{i=1}^{\sum} \sum_{j=1}^{\sum} \sum_{k=1}^{\sum} f_{tik} d_{tij} M_{tij} M_{tik} t = 1 \]

\[
= \sum_{i=1}^{\sum} \sum_{j=1}^{\sum} \sum_{k=1}^{\sum} f_{tik} d_{tij} M_{tij} M_{tik} + \sum_{i=1}^{\sum} \sum_{j=1}^{\sum} \sum_{k=1}^{\sum} A_{tij} M_{(i-1)ij} M_{tij}, 1 < t \leq P \quad (8)\]

The topology relationship between each subsystem is a ring, that is, the neighbours of subsystem 1 are subsystem \( P \) and subsystem 2, and the neighbours of subsystem 2 are subsystem 1 and subsystem 3, the neighbours of subsystem \( P \) are subsystem \( P-1 \) and subsystem 1.

3. Decentralized Genetic Algorithm

Genetic Algorithm (GA) is an adaptive global optimization algorithm that simulates the genetic and evolutionary processes in the natural environment. GA is commonly used to generate high-quality solutions to optimization and search problems by encoding, selecting, crossing and mutating. The DGA based on GA and decentralized optimization to deal with DPLP, each subsystem executes GA synchronously. In each iteration, a subsystem needs to exchange information with its neighbours, and obtain the optimal solution of the whole system while achieving the local optimum of each subsystem. DGA is a synchronous algorithm and all subsystem runs the same codes. The flowchart of the DGA in one subsystem is shown in Figure 1.

![Figure 1. The flowchart of DGA in one subsystem](image)

3.1. Encoding

Based on the DPLP model described in Section 2.1, we select the permutation encoding as the encoding method of DGA, that is, each device is arranged into an n-length string according to the position in the layout, for example, the code of a 2x3 layout containing 6 devices with P-period is shown in Figure 2.

| Period/Subsystem1 | Period/Subsystem2 | Period/SubsystemP |
|-------------------|-------------------|-------------------|
| 2 4 5 6 1 3       | 5 4 3 2 6 1       | 1 3 6 4 2 5       |

![Figure 2. Encoding example](image)

3.2. Initialization

The initialization of DGA can be divided into two parts. One is to set basic parameters, such as the number of population \( NP \) and iteration \( G \), mutation probability \( P_m \) and the material flow cost \( f_{tik} \) in each subsystem. The other one is to generate the initial population as the initial value of iteration.
3.3. Populations exchange

In this step, all subsystems will exchange populations with their neighbours for the calculation of target functions. As shown in Figure 1, the communication is implemented by sending $X_t$ and receiving $X_{(N_t)}$.

3.4. Calculation of fitness function

The fitness function is a function used to evaluate individuals. By establishing a mapping relationship between the target function and the fitness function, the optimal result can be achieved in the process of iteration. The process of calculating fitness function is as follow:

3.4.1. Local target function. As shown in Eq. (1), the local target function consists of the costs of material handling flow and device shifting. The variable of device location $M_{ij}$ and the distance between two positions $d$ are critical in this function. An $n \times n$ matrix $M$ is set to describe $M_{ij}$ that the row of $M$ represents the position and the column of $M$ represents the device. For example, if the 2nd device is on the 4th position, then $M(2,4) = 1$, the rest elements of the 2nd row and 4th column are 0.

In order to calculate $d$, we set a coordinate system in the layout which takes the top left corner as the origin, the horizontal right direction is the x-axis, and the vertical down direction is the y-axis. To simplify the model, each position is a square with length 1 and take the lower right vertex as the coordinate point. The distance $d$ is the Manhattan distance between each coordinate point and an $n \times n$ matrix $D$ is set to save the distances. An example of a 2x3 layout shown as Figure 3.

3.4.2. Local target function exchange. In this step, all subsystems will exchange target functions with their neighbours to achieve global optimization of the system. A subsystem sends a local target function $f_t$ to its neighbours and receives the target functions sent by neighbours, which called the feedback $\varphi_{N_t}$. The total target function of the subsystem as below:

$$ F_t^{pop} = f_t^{pop} + \varphi_{N_t}^{pop}, pop = 1, \ldots, NP $$

Where $pop$ is the number of populations, $F_t^{pop}$ is the total target function of the $pop$th population in the subsystem $t$.

3.4.3. Fitness function. In DGA, the lower value of the target function, the higher fitness of the population and the higher probability of the population being selected. Therefore, the fitness function in DGA is constructed as follows.

$$ F_t^{pop} = \frac{(F_t)_{max} - F_t^{pop}}{(F_t)_{max} - (F_t)_{min}} $$

Where $(F_t)_{max}$ and $(F_t)_{min}$ are the maximum and minimum values of the total target function in the current iteration of the subsystem.
3.5. Selection
As shown in Eq. (10), \( 0 \leq \text{Fit}_t^{\text{pop}} \leq 1 \), when \( \text{Fit}_t^{\text{pop}} = 0 \) the population is the worst and when \( \text{Fit}_t^{\text{pop}} = 1 \) the population is the best, so the \( \text{Fit}_t^{\text{pop}} \) can be directly taken as the selective probability. We set a random variable \( r_1^{\text{pop}} \) which is subject to uniform distribution, the \( \text{pop} \)th population will be selected when \( \text{Fit}_t^{\text{pop}} > r_1^{\text{pop}} \).

3.6. Crossover and mutation
From the selected population, two populations A and B are randomly selected for crossover operation, and a certain number of digits are selected for exchange according to the length of the string. For instance, let \( A=245613 \) and \( B=543261 \) exchange in the 3rd site, the first step is \( A=243613, B=545261 \), then exchange the duplicate number with the original number which will be \( A=243615 \) and \( B=345261 \).

After the crossover operation, the mutation operation is performed with a certain probability \( P_m \). We set a random variable \( r_2^{\text{pop}} \) which is subject to uniform distribution, when \( r_2^{\text{pop}} < P_m \), randomly select two sites in the crossed string for swapping.

Adding the crossed and mutated populations to the new generation pool and repeat the process until the number of populations in new generation pool reaches \( NP \).

4. Simulation experiment and analysis

4.1. Simulation experiment setup
In this section, we compare and analyse the DGA and centralized GA for solving DPLP by 12 sets of simulation experiments. The experiments are different in the number of devices, the number of periods and the material flow costs. The data settings are shown in Table 1, and the setting details are as follow:

(1) The layouts are: 6-device (2x3 layout), 12-device (3x4 layout).
(2) The total material handling flow in one period was subject to uniform distribution in a certain range.
(3) The material handling flows (from one device to another) in one period generate proportionally so that their sum equaled a pre-set total material flow which is the same for every period in an experiment.
(4) The shifting cost of a device is 15% of its average flow cost in one period.
(5) Flow dominance is introduced by randomly selecting between 1 to 3 devices (1 for 5-period, 2 for 10-period, 3 for 15-period) from each period and increasing the flow to these devices by a factor of 3-5.
(6) For each set, 30 independent numerical experiments are performed.

| Num. | Layout | Period | Material flow range |
|------|--------|--------|---------------------|
| 1-1  | 6 (2x3)| 5      | 1000-2000           |
| 1-2  | 6 (2x3)| 10     | 8000-10000          |
| 1-3  | 6 (2x3)| 15     | 1000-2000           |
| 1-4  | 6 (2x3)| 15     | 8000-10000          |
| 1-5  | 6 (2x3)| 15     | 1000-2000           |
| 1-6  | 6 (2x3)| 15     | 8000-10000          |
| 2-1  | 12 (3x4)| 5     | 1000-2000           |
| 2-2  | 12 (3x4)| 5     | 8000-10000          |
| 2-3  | 12 (3x4)| 10    | 1000-2000           |
| 2-4  | 12 (3x4)| 10    | 8000-10000          |
| 2-5  | 12 (3x4)| 15    | 1000-2000           |
| 2-6  | 12 (3x4)| 15    | 8000-10000          |
4.2. Result analysis
In order to verify the effectiveness and superiority of the proposed algorithm, the result of DGA is compared with the centralized GA. The processes of iteration are shown in Figure 4 and Figure 5. It can be found that the convergence rate and final value of the target function in DGA are better than those in GA, as the complexity of system increases, the advantages of DGA are more obvious. Furthermore, in the two figures, it can be found that the volatility of DGA is larger, that may be due to the fact that decentralized algorithms cannot obtain global information, and are vulnerable to small fluctuations of neighbours when calculating target functions, further lead to the fluctuations in global results.

5. Conclusion
In this paper, we proposed a DGA by combing genetic algorithm and decentralized optimization to solve the multi-period DPLP. Each production period is regarded as a subsystem and each subsystem takes the minimum of the sum costs of material handling and device shifting as the target function. The subsystems achieve the global optimality by exchanging the populations and the target functions with their neighbours and get the optimal layout scheme for each period. By simulating numerical experiments of the different layouts, periods and material handling flow, the convergence rate and optimal result of DGA are better than centralized DA especially in high-complex systems.

References
[1] Misevicius. A, A modified simulated annealing algorithm for the quadratic assignment problem, J. Informatica, 2003, 14(4): 497-514.
[2] K. L. Mak, Y. S. Wong, F. T. S. Chan, A genetic algorithm for facility layout problems, J. Computer Integrated Manufacturing Systems, 1998, 11(1-2): 113-127.

[3] Jaydeep. Balakrishnan, Chun. Hung. Cheng, Daniel. G. Conway, A hybrid genetic algorithm for the dynamic plant layout problem, J. International Journal of Production Economics, 2003, 86(2): 107-120.

[4] Nobuhiro. Yokoyama, Decentralized Algorithm Applicable to Parallel Trajectory Planning of Multiple Aircraft, A. International Conference on Control, Automation and Systems, C, 2007: 1393-1398.

[5] Zhuang. L, Chen. X, Guan. X, A decentralized ordinal optimization for energy saving of an HVAC system, A. American Control Conference, C, 2016: 611–616.

[6] Le Jian, Zhou Qian, Wang Cao, Li Xingrui. Research on Voltage and Power Optimal Control Strategy of Distribution Network Based on Distributed Collaborative Principle, J. Proceedings of the CSEE, 2019, 7.

[7] Guanghong Han, Xi Chen and Qianchuan Zhao. Decentralized Differential Evolutionary Algorithm for Large-Scale Networked Systems, A. International Conference on Smart City and Intelligent Building, C, 2018: 407-417.

[8] Rosenblatt, M. J, The dynamics of plant layout, J. Management Science, 1986, 32 (1): 76 – 86.

[9] S. P. Singh, R. R. K. Sharma, A review of different approaches to the facility layout problems, J. The International Journal of Advanced Manufacturing Technology, 2006, 30(5-6):425-433

[10] Sadan. Kulturel. Konak, The zone-based dynamic facility layout problem, J. Information Systems and Operational Research, 2019, 57(1): 1-31