Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada
DOI: 10.23977/acss.2023.070202
ISSN 2371-8838 Vol. 7 Num. 2

Research on the Application of Two-Population Genetic Algorithm in Production Line Balancing

Yuqin Pan

Business School, Shandong University of Technology, Zibo, Shandong, 255000 China

Keywords: Production lines; Two-population genetic algorithm; Balance optimisation

Abstract: Exploring ways to improve the overall balancing rate of the production line by adjusting the number of work stations in the line and allocating the work units in a rational way. Two-population genetic algorithm is introduced to take further improvements to this production line balancing problem by coding the job units and cross-mutating the job units in the optimisation scheme to eventually obtain a globally optimal solution. The effectiveness of the designed algorithm is verified with the classical Jackson problem, which shows that the Two-population genetic algorithm has excellent results in improving the balancing rate of the production line.

1. Introduction

The problem of line balancing was born as soon as companies started using flow lines. Line production also means that workpieces or products are processed along exactly the same route and their production speed depends entirely on the longest station in the line. In practice, line balancing takes into account priority relationships between jobs, external constraints, workstation capacity, etc. Therefore, the production line balancing process can be understood as the process of combining and optimising the relationship between production resources and the NP difficulty problem[0](2017). Production line balancing can be further divided into Assembly Line Balancing and Sequential Transfer Line balancing problems according to the type of production line.

To date, scholars both domestically and abroad have made significant progress in researching this problem, providing us with rich theoretical and practical solutions. In his study of the practical problem of balancing a mixed flow line, Thomopoulos improved the algorithm based on heuristic rules for a single flow and verified the effectiveness of the improved algorithm in solving the balancing of a mixed flow line[0] (1970). Sabuncuoglu et al. [0] (2000) used genetic algorithms to solve deterministic and single-model assembly line equilibrium problems, where the authors changed the dynamics of the chromosome structure to improve the algorithm's search efficiency. Mura and Dini [0] (2017) conducted a study of the manual assembly line problem for assembly line-based companies and added an investigation of the impact of tool quantity. They used a genetic algorithm to perform multi-objective optimisation of the assembly line balance. Bojun[0] (2018) improved and optimized the computational process of the genetic algorithm used in balancing mixed-flow automobile production lines to overcome its shortcomings. They proposed a "greedy search method" to enhance the algorithm's search ability, which ultimately resulted in a satisfactory balancing effect.Jieyun et al.[0] (2018) also used an improved genetic algorithm to solve and optimize the production line balancing problem, overcoming the problems of local optimal solutions and limited search space of traditional genetic algorithms. Foroughi and Gökçen [0] (2019) proposed a multi-rule genetic algorithm for multi-process assembly lines in solving the least-cost stochastic Assembly Line Balancing Problem (ALBP). Yadav and Agrawal [0] (2020) developed a mathematical model based on MATLAB in solving a mixed-model double-sided assembly line balancing problem and used LINGO software to solve the problem, the results show that the method can effectively reduce the number of assembly workers, reduce handling costs and increase production line balancing efficiency. In this paper, the two-population genetic algorithm is used to design an optimisation algorithm for production line balancing and the effectiveness of the designed algorithm is verified against the classical Jackson problem.

2. Overview of production line balancing

The aim of line balancing is to eliminate the loss of efficiency and overcapacity of companies with unbalanced production operations, by adjusting the load on all processes of the line so that all its processes are averaged to achieve a more efficient production line balance.

In the process of optimising the line balance, two quantitative analysis criteria were set up to compare and analyse the strengths and weaknesses of the line: the line balance ratio P and the smoothing index SI.

(1) Expression for line equilibrium rate P:

$$P = \frac{\sum_{k=1}^{m} T(S_k)}{m \times maxT(S_k)} \times 100\% \tag{1}$$

In the formula:

P: Indicates the line balance rate of the assembly line;

M: Indicates number of workstations;

S_k: Indicates the kth workstation k=1, 2, 3, ..., m;

$T(S_k)$: Indicates the time of the kth workstation k=1, 2, 3..., m;

$maxT(S_k)$: Indicates the maximum workstation time, i.e. the beat of the production line.

(2) Expression for Smoothing index SI:

$$SI = \sqrt{\sum_{k=1}^{m} (CT - T(S_k))^2} \tag{2}$$

In the formula:

SI: Indicates smoothing index;

M: Indicates number of workstations;

S_k: Indicates the kth workstation k=1, 2, 3, ..., m;

$T(S_k)$: Indicates the time of the kth workstation k=1, 2, 3..., m;

CT: Indicates the beat of the production line.

3. Design of the two-population genetic algorithm for production line balancing problem

3.1 Coding

In this paper, in order to be close to the actual production situation, real number coding based on job priority sequences is used, where each process number represents a gene locus and the process numbers are linked into a real number string in job priority order to form a chromosome of length of the number of processes, i.e. each chromosome represents a possible way of doing the job. This encoding method is well adapted to the objective function and genetic operator operations. This encoding method is able to adapt to the objective function and the operation operator relatively well, thus making the algorithm results accurate[0].

3.2 Decoding

For assembly line balancing problems, there are mainly Type I and Type II assembly line balancing problems. The first type of assembly line balancing problem is to minimise the number of workstations with a defined production rate, with a view to reducing the investment in equipment, personnel etc. In the second type of assembly line balancing problem, the number of workstations is known and the production schedule of the assembly line is optimised. The purpose of assigning jobs under unknown beat conditions cannot be achieved. Therefore, when designing decoding rules for Type II assembly line balancing problems, the theoretical production beat CT needs to be set in advance and a test calculation is done in accordance with a specific increment during the evolution of the algorithm until the condition is satisfied.

The decoding process is as follows:

(1) The initial value of the beat is set in advance as the minimum theoretical beat, The formula is expressed as $CT^* = \frac{T}{m}$.

In the formula: T is the sum of the job unit times; M is the number of workstations required to be given.

(2) Under the condition that the initial value CT* is the beat, n jobs are assigned to m workstations according to the job priority relation, and the time of the workstations is $T(S_k)$, where k=1,2,3, ..., m. If the time of all workstations does not exceed CT*, the computation stops and CT* is the minimum beat sought and the job sort is the optimal sort, otherwise, go to the next step.

(3) The possible increments of the production beat are expressed as$\triangle$k = operating time of the first job unit at the k+1st workstation (k= 1, 2, 3,..., m - 1).

(4) Determine the beat increment and then add the beat increment to the initial beat to obtain the incremented production line beat. Let $CT = \max\{T(S_k)\}$ and $CT^* = \min\{T(S_k) + \Delta k\}$, if $CT \le CT^*$, then CT is the minimum beat in this case and the search stops, if $CT > CT^*$, then go back to step 2 and repeat the calculation.

3.3 Adaptation function

In this paper, the objective of the line balancing problem is to minimise the rate of line balancing losses and to minimise the smoothing index for a determined number of workstations. Therefore, the fitness function is

$$Fit = Fit1 + Fit2 \tag{3}$$

In the formula:

$$Fit1 = \frac{\sum_{k=1}^{m} T(S_k)}{m \times \max\{T(S_k)\}} \tag{4}$$

$$Fit2 = \frac{1}{\sqrt{\sum_{k=1}^{m} (CT - T(S_k))^2 + 0.01}} \tag{5}$$

3.4 Selection

Selection is the process of eliminating inferior individuals from a population and selecting superior individuals based on a defined fitness function. The higher the fitness, the greater the probability that an individual will be selected for the next generation, and the probability P that each individual will be selected is

$$S_p = \frac{Fit(p)}{\sum_{q=1}^{pop-size} Fit(q)} \tag{6}$$

In the formula: $Fit(p)$ is the fitness value of chromosome p, $Fit(q)$ is the fitness of any individual, $pop-size$ is the set of individuals of the population in the selected region, $\sum_{q=1}^{pop-size} Fit(q)$ is the fitness value of all chromosomes within the population.

3.5 Crossover

The crossover strategy in this paper uses a two-point crossover approach. A two-point crossover is the random selection of two crossover points on a chromosome and the exchange of genes for the segments between the crossover points. Two chromosomes are selected arbitrarily from the obtained chromosome pool, and two points in the interval [1, n] are chosen arbitrarily to make gene crossover points, Cross 1 and Cross 2, respectively, to cut the chromosomes into three segments: head, body and tail. The body segments between the two parental chromosome gene crossovers, Cross 1 and Cross 2, were genetically swapped to obtain two offspring chromosomes, which were revised so that the chromosomes were conflict-free.

3.6 Variation

According to the set mutation probability, the corresponding chromosome is selected among the chromosomes, at this time there are n genes on this chromosome, a random integer k is generated, so that the 1st to kth genes are the genes that do not need to be mutated, the gene fragment is kept unchanged directly to the offspring chromosome, and then the k+1th to nth genes are recombined. The recombination is carried out by eliminating the genes before the corresponding mutation point in the priority matrix, then rearranging the remaining gene fragments in the same way as the initial population was generated, and finally splicing them back together with the gene fragments before the mutation point to form a new chromosome.

3.7 Population gene exchange

The swap operator is an operational operator unique to the two-population genetic algorithms, which can effectively avoid the local optimum problem and improve the convergence rate. To enhance the exchange of individuals between populations, two gene swaps will be performed, one in which the population gene with the best fitness is selected and swapped between the two populations, and the other in which the remaining gene is randomly selected and swapped between the two populations.

4. Verification of the effectiveness of the two-population genetic algorithm

According to the algorithm rules designed in the above paper, the programming and operation of the two-population genetic algorithm are implemented in Matlab software. Since the production line balancing problem was proposed, there exist many classical production line balancing problems. In this paper, the Jackson problem is used for verification to ensure the validity of the program design, and the priority relationship matrix of the Jackson problem is obtained based on the Jackson problem flowchart as follows.

$$[Time, Matrix] = [6\ 2\ 5\ 7\ 1\ 2\ 3\ 6\ 5\ 5\ 4]$$

$$\begin{aligned} Matrix = [&0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \\ &0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\ &0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \end{aligned}$$

0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0]

According to the designed arithmetic program, and after running the Jackson problem data into the designed program, the workstation mountain product diagram is obtained as shown in Figure 1, in Figure 1, the horizontal coordinate is the workstation, the vertical coordinate is the time, and on the bar graph in the figure: the number on the left indicates the number of the assembly task, the number on the right indicates the completion time of the assembly task, and the sequence between the assembly tasks is bottom-up, for example For example, in Figure 1, the sequence of assembly tasks at workstation 1 is 1 → 2 → 5, with task completion times of 6s, 2s and 1s respectively.

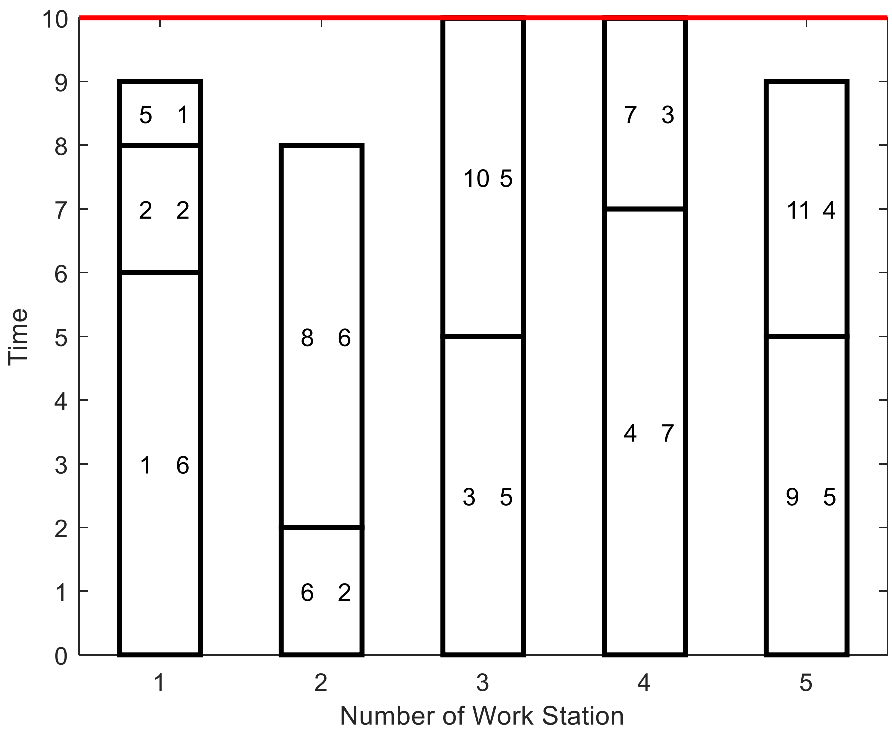

Figure 1 Mountain product diagram of a workstation solved by the two-population genetic algorithm for the Jackson problem

Table 1 Comparison of the results of Jackson's enumeration method with those of the two-population genetic algorithm

	Working hours					Balance rate	Smoothing Index
	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5		
Jackson Enumeration	10	7	10	10	9	92%	3.16
Two-population genetic algorithm	9	8	10	10	9	92%	2.45

The above results obtained with the improved genetic algorithm were compared with the results obtained by Jackson using the enumeration method and the results were compared as shown in Table 1 below.

5. Conclusions

This paper aims to improve the traditional genetic algorithm, which is limited in search space and is prone to local optima during global optimization. The two-population genetic algorithm is adopted to transform the production line balancing problem into a mathematical model, and mathematical modeling and algorithm design are carried out. The two-population genetic algorithm is implemented with the aid of Matlab software, which enables the production line to rearrange and reallocate its processes in accordance with the priority order of operations, resulting in a higher production balance rate. In addition, the classical Jackson problem was used to verify the effectiveness of the algorithm and the results were compared with the traditional enumeration method, which showed that both methods achieved the same production line balance rate of 92%, but the value of the smoothing index for the two-population genetic algorithm was less than that of the enumeration method. The two-population genetic algorithm is more effective in reducing the production rate of the production line and improving the balance rate of the production line.

References

[1] Li X. Study of automotive line balancing problem with improve genetic algorithm [D]. University of Science and Technology Beijing, 2017.

[2] Thomopoulos N. T. Mixed Model Line Balancing with Smoothed Station Assignments [J]. Management Science, 1970, 16(9).

[3] Sabuncuoglu I., Erel E., Tanyer M. Assembly line balancing using genetic algorithms [J]. Journal of Intelligent Manufacturing, 2000, 11(3).

[4] Mura M. D., Dini G. A multi-objective software tool for manual assembly line balancing using a genetic algorithm [J]. CIRP Journal of Manufacturing Science and Technology, 2017, (6):72-83.

[5] Chai B. Research on multi-objective balance of mixed assembly line based on improved genetic algorithm [D]. Northeast forestry university, 2018.

[6] Situ J., Yu Y., Yang H., et al. Optimization of type-II balancing problem of assembly line based on improved genetic algorithm [J]. Light Industry Machinery, 2018, 36(1):99-102.

[7] Foroughi A., Gökçen H. A multiple rule-based genetic algorithm for cost-oriented stochastic assembly line balancing problem [J]. Assembly Automation, 2019, 39(1).

[8] Yadav A., Agrawal S. Mathematical Modeling of a Cost-Oriented Bilateral Assembly Line Equilibrium Using an Exact Solution Method [J]. Journal of Research in Science and Engineering, 2020, 2(10).

[9] Li J., Gao J., Sun L. Balancing mixed model assembly line with capacity adjustment to satisfy uncertain demands [J]. Systems Engineering-Theory&Practice, 2014, 34(01):70-76.