

Assembly line balancing based on a genetic algorithm

Xiumu Lin^{1, a *} and Youjun Zhang^{1, b}

¹ School of mechanical engineering, Shenyang University of Technology, Economical & Technological Development Area, Shenyang, Liaoning, China, 110870

^alinxiumu@qq.com, ^b979723626@qq.com

Keywords: assembly line balancing, genetic algorithm, optimization

Abstract. Assembly line balancing research is a key for the modern production. The implementation steps of genetic algorithm are proposed in this paper and it is used to optimize the assembly line balancing problems in engineering practice. The genetic algorithm operators based on feasible operating sequences of these job orders, which is guarantee the feasibility of the solution, and the optimal retention strategy is used to ensure the effective of the algorithm. Finally the assembly line reduce 2 work stations after calculation, and it makes each station load become more uniform.

Introduction

Assembly line balancing (ALB) is a thing that the specific basic work elements are distributed to the station reasonably, which makes the time of each word station be balance as far as possible, and meet the expected objectives and constraints. In essence, assembly line balancing problem (ALBP) is a problem of combinatorial optimization, but it becomes complicated as a result of facilities, operating procedures, manufacturing sites, and so on. In actual production, each station working hours is not balanced because of combination of manual station and automatic station. The longest time workstation is the bottleneck station. If the bottleneck station is automatic station, it cannot be changed because of the high cost of changing automatic equipment, so the cycle time (CT) of the assembly line is not changed. In this case, redistributing work elements and reducing manual station can achieve the balance [1]. This situation is studied in this paper.

In recent years, the main trend is adopting heuristic algorithm to solve the large-scale ALBP. Among them, genetic algorithm is suited for solving such problems because of fast optimizing speed, easily escaping from local optimal solution, high searching efficiency, simple structure and convenient [2].

Design of Genetic Algorithm

GA is proposed by J. Holland in 1975, which is inspired by biological evolution. GA is a general optimization algorithm and it can solve any form of objective function and constraints. Standard GA may produce the situation such as premature convergence and slow convergence, even no result because of infeasible operators. In view of this, this paper adopt the operators based on feasible sequence to ensure the feasibility of the operators, and adopt the optimal retention strategy to avoid the optimal solution lost or algorithm degradation [3].

Through the analysis of before in this paper, the problem studied can be defined as that $CT=C$, to divide the operating elements set $E=\{1,2,\dots,n\}$ to $\{S_k | k = 1, 2, \dots, m\}$, making the equation $\min \text{Goal} = \alpha \cdot m + \beta \cdot SI$. In this equation, m is number of workstations, α and β are weight coefficient. Equilibrium index for ALB is as given below:

$$SI = \sqrt{\sum_{k=1}^m (\max(T_k) - T_k)^2} \quad (1)$$

Where, S_k is a subset for E , which is a set of work elements for No. k workstation, and it also is No. k process. T_k is the time for No. k workstation. Balance goal is making location least or each

workstation loads more balanced, and focus is different, α and β can be adjusted to control the optimization results of the algorithm. The algorithm flow is shown in Fig. 1.

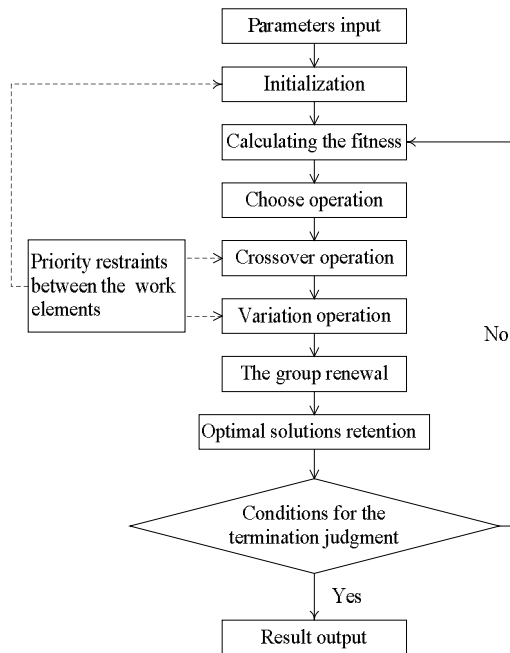


Fig. 1 Flow chart of GA

Parameters input. Including coding and decoding. Coding process is equivalent to the biological phenotype corresponding to genotype, which is the problem of state space corresponding to the GA code space. The coding of this paper is establishing a corresponding relationship between the sequence of work elements and chromosomes. Every gene is named with work elements' number, which makes work elements' numbers in a line according to the priority, each work elements corresponding to a gene. Decoding process is equivalent to the biological genotype corresponding to phenotype. Chromosome chart after coding is shown in Fig. 2. The chromosome illustrates the order of each element distributed, element 1 is the first, element 3 is the second, ..., element 5 is the sixth, element 7 is the last one.

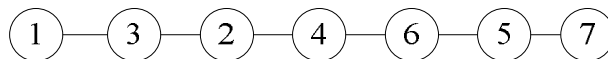


Fig.2 Chromosome chart after coding

Calculating the fitness. Choose fitness function according to the objective function is as given below:

$$F_{(i)} = \frac{1}{\text{Goal}} = \frac{1}{\alpha \cdot m + \beta \cdot SI} \quad (2)$$

Calculating the fitness of each individual to control the evolution direction and make the population development towards the direction of the continuous optimization, gradually close to the global optimal solution.

Choose operation. Calculating the fitness value to eliminate the smaller $F_{(i)}$ individual by using the method of expected value and to speed up the convergence speed.

Crossover operation. With the method of improved order exchange. Let P is for crossover probability, and choose $N(q) \cdot P$ chromosomes randomly to mate. If the quantity is odd number, get rid of one. Take two of these chromosomes, and produce two intersections in the interval $[1, n - 1]$ (N is the quantity of work elements.) to divide each chromosome into three parts. Retain the head and the tail genes of the first chromosome, at the same time delete the first keep parts in the second. And combine the rest of both to new progeny chromosomes. Delete the mated parents chromosomes, and put the progeny chromosomes to the next generation populations [4].

Variation operation. With the method of displacement variation. Randomly choose an individual and randomly choose one of the genes, find the feasible range of work elements order, and insert the variable genes in any position.

Using the optimal retention strategy to accelerate the algorithm convergence and prevent random roaming [5].

Case Analysis

A cylinder head assembly line of an engine corporation's engine is the annular arrangement layout, including manual and automatic. There are 15 processes in it, including 7 automatic stations, 3 semiautomatic stations, 5 manual stations and 7 workers. Bottleneck process is an automatic process, CT is 70s. Due to the high cost of equipment upgrades, only adjust other processes without changing the CT. The priority of work elements is shown in Fig. 3. And the cylinder head assembly line work elements and workstation time is shown in Table 1.

Table1 Cylinder head line work elements and workstation time

No.	Work elements	Precedence work No.	Time/s
1	Load cylinder head to pallet		20
2	Scan 2D matrix	1	5
3	Get spindle loosening tool	2	2
4	Loosen cam caps with tool	3	20
5	Clean cylinder head	4	5
6	Remove cam caps and place to pallet	4	6
7	Place 8 intake valve seals to rack for detect spring presence	5、 6	10
8	Place valve seals to semi auto press head (a)	7	6
9	Place 8 dxhaust valve seals to rack for detect spring presence	5、 6	10
10	Place valve seals to semi auto press head (b)	9	6
11	Lubricate & install 8 intake valves	8	32
12	Lubricate & install 8 exhaust valves	10	32
13	Press 2 oil check valves	11、 12	42
14	Install 16 valve springs	11、 12	12
15	Install retainers	14	10
16	Install & secure 4 spark plugs	11、 12	20
17	Install Int. & Exh. key (a)	15	62
18	Install Int. & Exh. key (b)	15	62
19	Valve key laser check	17、 18	35
No.	Work elements	Precedence work No.	Time/s
20	Valve run-in	19	35
21	Valve blow-by leak test	20	38
22	Shake Out and final visual check	13、 16、 21	70
23	Place exhuaast cam caps to head	22	6
24	Snug exhuaast cam caps bolts	23	12
25	Place intake cam caps to head	22	6
26	Snug intake cam caps bolts	25	12
27	Unload cylinder head assembly	24、 26	12

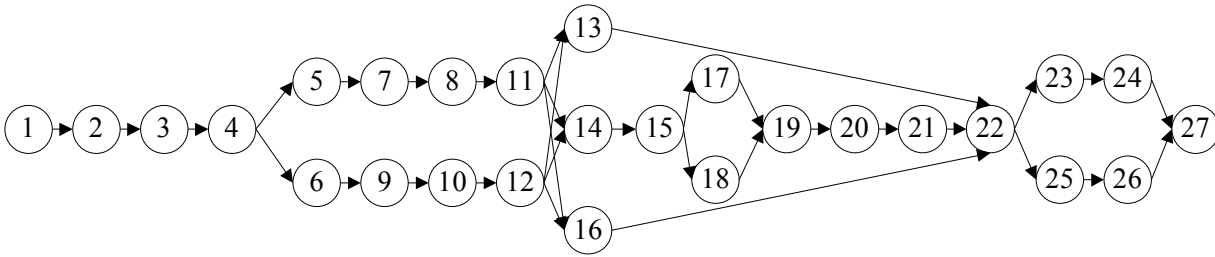


Fig. 3 The priority of work elements

The CT of the cylinder head assembly line is 70s, the population size is 10, the crossover probability is 0.9, variation probability is 0.1. The algorithm is implemented through the C++. The result is 13 workstations after calculating, including 7 automatic workstations, 3 semiautomatic workstations, 3 manual workstations and 7 workers. In actual production, the station CH1030={11}, CH1035={12}, CH1050={14, 15, 16}, CH1105={23, 24, 25, 26}, CH1110={27}. The result is station CH1030={11, 14, 15}, CH1035={12, 16}, CH1105={23, 24, 25, 26, 27}. This algorithm reduces two stations and two workers.

Calculating the delay index of ALB is as given below:

$$D = \frac{m \cdot T - T}{m \cdot T} \times 100\% \quad (3)$$

After calculation, the actual balance delay index of cylinder head line $D_1 = \frac{15 \times 70 - 588}{15 \times 70} \times 100\% = 44.00\%$,

the optimized balance delay index $D_2 = \frac{13 \times 70 - 588}{13 \times 70} \times 100\% = 35.38\%$, the improvement is obvious. Actual equilibrium index after calculated by the formula (1) $SI_1 = 132.97$, and the optimized equilibrium index $SI_2 = 105.22$. The optimized workstation loads are more balanced than before.

Conclusion

The GA for solving cylinder head ALBP is presented in this paper and is applied to engineering practice. The method reduces the quantity of workstations from 15 to 13 and reduces 2 workers. After optimization, the balance effect is better and workstations loads are more balanced than before. At the same time, the method reduces the cost and improves the efficiency. It provides the technical base for the enterprise to solve this kind of problem later.

References

- [1] S.O. Tasan and S. Tunali: J. Intell. Manuf. Vol. 19 (2008), p. 49
- [2] P. Sivasankaran and P. Shahabudeen: Int. J. Adv. Manuf. Technol. Vol. 73 (2014), p. 1665
- [3] J. Yu and Y. Yin: Int. J. Adv. Manuf. Technol. Vol. 48(2014), p. 347
- [4] H. Wang and H. Zhang: J. Micro Computer Information. In Chinese. Vol. 25(2009), p. 306
- [5] L. Hou, Y. Wu, R. Lai, and C. Tsai: Int. J. Adv. Manuf. Technol. Vol. 70(2014), p. 1775