UNIVERSITAT POLITÈCNICA
DE CATALUNYA

The United States industrial organization pattern in mixed-model-assembly-line

Hao Niu and Anna Maria Coves Moreno

IOC-DT-P-2005-13
Juliol 2005

Institut d'Organització i Control
de Sistemes Industrials

THE UNITED STATES INDUSTRIAL ORGANIZATION PATTERN IN MIXED-MODEL-ASSEMBLY-LINE*

Hao Niu and Anna M. Coves Moreno

Institut d' Organització i Control de Sistemes Industrials (IOC)

Universitat Politécnica de Catalunya (UPC)

Av. Diagonal 647 planta 11, 08028 Barcelona, Spain

Abstract

In modern industries, mixed-model-assembly-lines (MMAL) are widely used. There are two principal kinds of industrial organization patterns, which called the *United States Pattern* and the Toyota (*Japan) Patter.* The *United States Pattern* will be studied in this article and the objective of this pattern is minimizing the utility and the idle time. Genetic algorithm is been used for solving the problem and more than 5000 examples have been designed. From the results, the better parameters for the assembly line and the genetic algorithm will be known.

Keywords**:** Mixed-model-assembly-lines (MMAL), *United States Pattern,* Utility time, Idle time

1. Introduction

In many assembly systems, products are mounted on a conveyor belt and operators move alongside the belt while working on the product [Scholl, A. 1999]. In the closed station assembly systems, the operators should finish their operation within their stations; while in the assembly system with open stations, although the operators can do their operation out of their stations, it is preferable that the operators finish their work within an optimal sequence and do it as quickly as possible. In the case of closed stations, if the operators can't finish their work within their stations, there are two quite different alternative approaches for completing the unfinished work; these are the *United States Pattern* and the *Japan Pattern*, which show the different industrial organization patterns in each country.

In *the United States Pattern*, if the operators can't finish their work within their stations, utility workers are employed on an ad hoc basis to finish work left undone by primary line operators [Tsai, Li, 1995], [Bhaba, R. et al 1998], while in Japan the operators push a stop button whenever they are unable to finish their work and let the conveyor move again when they have finished their work [Zhao, X. B. et al.1997] and [Zhao, X. B. et al. 2000]

---

* This research is supposed by the project of investigation of the Ministry of Science and Technology DPI2004 -03472 ¨Design and Equivalent of Assembly Line with Real Condition ¨, Spain.

In the assembly line, the work which can't be finished within the station is called overload. Usually, the work overload is dealt with through the use of utility workers who either are dispatched to assist the regular workers during peak load situations, or are stationed at various points along the assembly line to complete the unfinished operations. The other alternative is to increase staff on the assembly line to handle estimated peak loads. This is an expensive solution, so a typical "real" solution would involve a combination of slight overstaffing and utility work. No matter what the staffing policy is, it is clear that minimizing the work overload contributes to reducing the total load cost [Yano, C. A. et al 1991].

The mathematical formula for this objective is $Min\sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{ut}_{\mathrm{n}}^{\mathrm{k}}$, where ut_n^k the total utility time (Utility time is the time needed for completing the unfinished work-pieces which the normal operator has not completed when he arrives at the border of the station) of the conveyor, *k* and *n* is are the number of stations and the work-piece models respectively; in this article, while minimizing the utility time, the idle time is also minimized. The mathematical formula is $Min\sum_{n=1}^{N}\sum_{k=1}^{K}it_n^k$, where it_n^k is the total idle time (a positive difference between the cycle time and the station time is called idle time) of the assembly line.

This article will talk about the *United States Pattern;* the objective of this pattern is minimizing the utility and the idle time. Genetic algorithm is been used for solving the problem and more than 5000 examples have been designed. The results will show the better parameters for the assembly line and the genetic algorithm.

According to the classification of assembly line [Niu, H. et al 2005], *the United States Pattern* can be classified as follows:

- The products:

In this article, it is supposed that the type of product corresponds to the mixed-model-assembly-line; the launching interval discipline is fixed rate launching; and the position of the products is also fixed.

- The assembly line:

In this article, it is supposed that the layout of time is serial; the line is paced; the type of the station is left-side open (except the first station) and right side closed; the length of the line is deterministic; this pattern is the *United States Pattern* without consideration of the set-up time.

- The operator:

In this article, it is supposed that the velocity of the operator is considered and can vary from operator to operator; both utility and idle time exist; the operation time is deterministic; the operator schedules is Later Start and the operator only works at one station.

- The objective:

The objective of the pattern is to minimize the utility time and the idle time.

The above considerations can be seen in Fig. 1.

Thompoulos, N. T. [1967] and Macaskill, J. L. C. [1972] developed approximate algorithm to minimize the sum of utility work, idle time, deficiency and congestion, their paper shows that single-model line balancing techniques are adaptable to mixed-model schedules, and also shows that sequencing can be used to increase efficiency on the mixed-model assembly line. Okamura, A. et al [1979] presented a heuristic algorithm to minimize the risk of conveyor stoppage. Their heuristic approach sought to reduce maximum displacement by inserting and interchanging products. Their paper proposes one method only for small-scale, mixed-model sequencing problems and uses an improved branch and bound method as its application. Yano, C. A. et al [1991] gave a mathematical programming formula for the problem of minimizing total utility work at a single station and for products with arbitrary processing times.

Although there has been research into this objective of minimizing the utility time and idle time, until now there have not been many articles concerning the use of genetic algorithm with the added consideration of the velocity of the operator.

The thesis starts with the introduction. The second part aims at explaining the notations which used in this article; the third part presents the process of the *United States Pattern* and the calculation part; the fourth part describes program and the genetic algorithm of this article; the fifth is the design experiment which has been done in this research and the last part is the conclusion of this article.

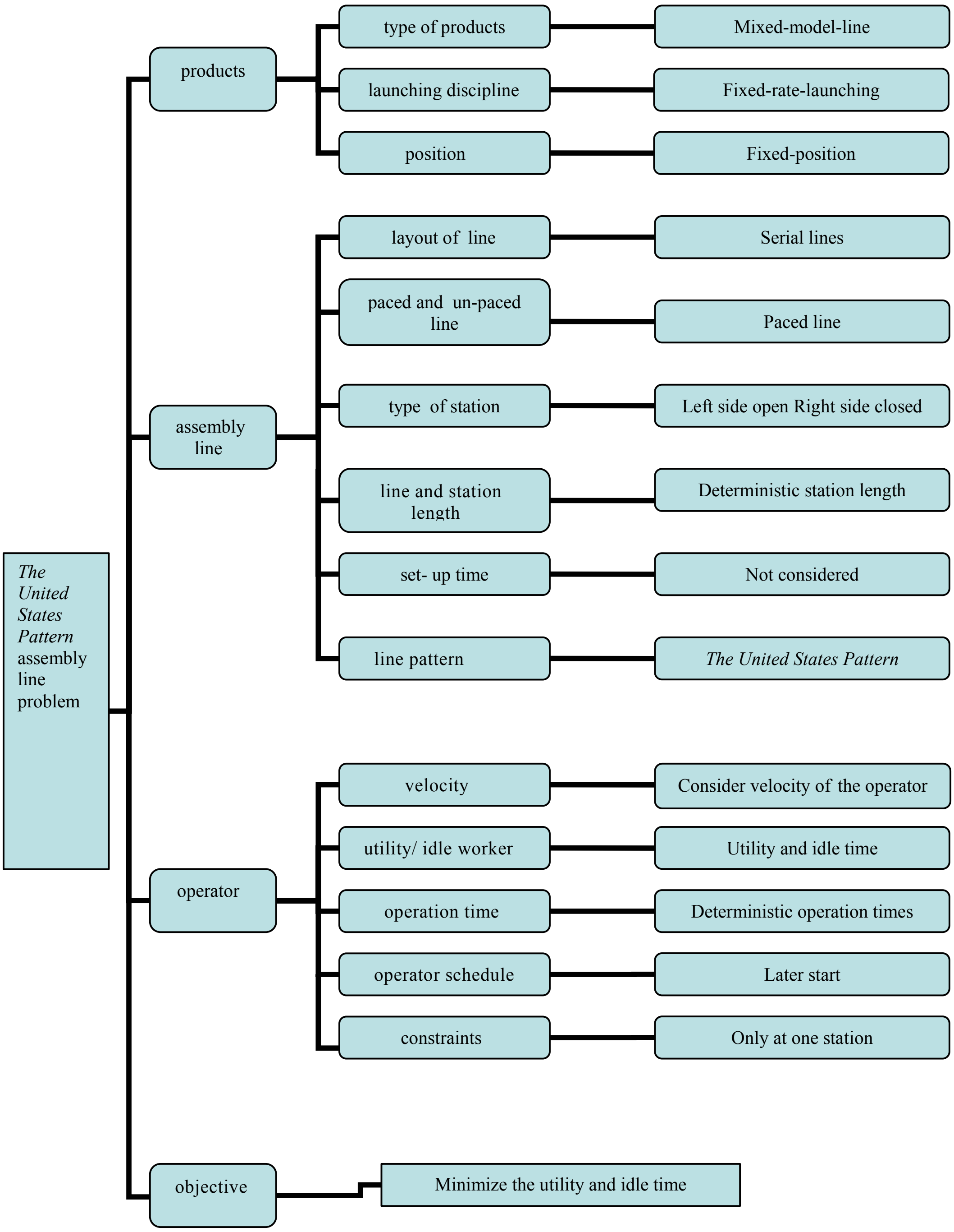


Fig.1 *the United States Pattern* Assembly Line Classification

2. Notations and Equations of the *United-States Pattern.*

2.1. Notations

In this article, the following notation is used:

t_m^k	Operation time by worker *k* for model *m (1,…, M)*
$\pi(n)$	The *nth* unit in sequence $\pi = \{\pi(1),\ldots, \pi(n)\}$
$T_{\pi(n)}^k$	Operation time by worker *k* for *nth* unit in the sequence
L^k	The length of *k* work stations (*k=1,…K*)
M_m	Minimal part set demand for model *m* (*m=1,…, M*)
p_n^k	Starting position from the upstream boundary of work station *k* for the *nth* unit in the sequence.
d_n^k	Upstream distance of work station *k* for the *nth* unit in the sequence
f_n^k	Completing position from the upstream boundary of work station *k* for the *nth* unit in the sequence
ut_n^k	Assembly line utility time caused by worker *k* for the *nth* units in the sequence
it_n^k	Idle time of worker *k* reaching the upstream boundary of work station after completing the operation for the *nth* unit in the sequence
$UT(\pi)$	Total utility time of the sequence π
$IT(\pi)$	Total idle time of sequence of π
V_c	Velocity of the assembly line
V_o	Velocity of the operator when walking upstream to the left boundary
t_c	Window cycle time or launch interval time

For all the notations above, K, t_c, V_c, V_o, M, L^k, $\pi(n)$, t_n^k are the input variables

2.2. Calculation Process of *the United States Pattern.*

Fig. 2 shows an example of the movement of operators in a left-side open station (except the first one) assembly line. In this assembly line, the conveyor has 6 work-pieces, 3 models and 3 stations.

The model sequence is shown on the horizontal while the station and the line length are shown on the vertical. The horizontal is labeled as *j(m),* meaning sequence *j* of model *m* where *j* is the index for a model that varies from 1-6, and *m* is the index for a model that varies between A, B and C. The horizontal solid arrows in Fig. 2 show the operators' downstream movements while walking along with the work-piece. The tail and head of an arrow indicate the beginning and the end of a task on a work-piece at each station, respectively. The starting point p_n^k of a work-piece n and the processing time t_m^k of a model m at station k are indicated by the general notation $p_n^k \cdot (t_m^k)$ on each arrow in the figure. The diagonally drawn broken lines denote the operators' upstream movements and the operator's walk starts, in this example, from sequencing model B to C to B and so on at each station. In the left-side open station, an operator does not necessarily have to start working within his station; he can begin work at a station upstream by crossing his own station boundary. The starting point and the stopping point of the operator, the idle time and the utility time are also shown in this figure.

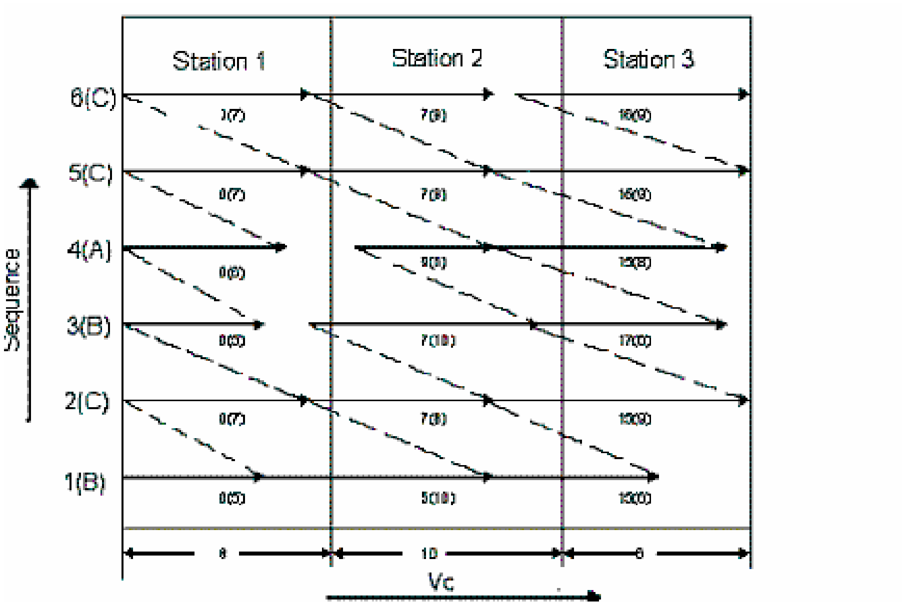


Fig. 2 Left –side open station (Except the 1st one) and the movement of the assembly line

Given the basic information and definitions of the assembly line, a mathematical equation can be formulated to relate utility time, idle time, upstream distance, the starting points and the completing points. These are as follows.

2.2.1. Starting Point and Completing Point

Starting point

When an operator reaches the downstream boundary of his station, no matter whether he has finished the work-piece or not, he walks back at a constant speed of V_o to the starting point upstream at distance d_n^k. The position of the starting point is: (some of the equations in this part are adapted from Bhaba, R. S et al [1998] and some changes have been made in the notations)

$$p_{n+1}^k = p_n^k + \mathrm{V_c} \cdot T_{\pi(n)}^k - \mathrm{V_c} \cdot ut_n^k - d_n^k \tag{1}$$

If the operator starts assembling the work-piece at a position upstream, he is not allowed to interfere with the operator who works in the preceding station. Suppose p_{n+1}^k, is the starting point at *(n+1)th* station for the *kth* work-piece, then p_{n+1}^k must be beyond the end of the *nth* station upstream for the same work-piece. Therefore the constraint of the starting point is

$$p_n^{k+1} \geq p_n^k + V_c \cdot T_{\pi(n)}^k \tag{2}$$

The equation above shows that the next starting point of an operator at station n for the next work-piece depends on the current starting point of the work and the assembly time of scheduled work-piece. The configuration of the starting point is shown in Fig. 3.

Completing Point

Here two cases are considered:

A) If the operator finishes the work-piece before he arrives at the right-side boundary of his station there is no utility time and he goes back upstream to work on the next work-piece; the completing point is where the operator finishes his work-piece

B) If the operator arrives at the right-side boundary and doesn't finish his work, then there is utility time; in this situation, the completing point is the left-side boundary. The mathematical equation for the completing point is as follows:

$$f_n^k = p_n^k + \mathrm{V_c} \cdot T_{\pi(n)}^k - \mathrm{V_c} \cdot ut_n^k \tag{3}$$

Comparing the equations (1) and (3), the completing point can be expressed as:

$$p_{n+1}^k = f_n^k - d_n^k \tag{4}$$

The configuration of the completing point is shown in Fig.3.

2.2.2. Utility time and Idle Time

Utility Time

In the right-side open station, the operator moves downstream at a speed of V_c from the starting point p_n^k while assembling the *kth* work-piece. The total lengths of all preceding stations including the current station is $L_1 + L_2 + \ldots L_n$. Therefore the distance that the operator works on this work-piece is the difference between $L_1 + L_2 + \ldots L_n$ and the starting point, that is: $L_1 + L_2 + \ldots L_n - p_n^k$. The time that is used by the conveyor to move through this distance is $(L_1 + L_2 + \ldots L_n - p_n^k)/ V_c$. When the operator can't finish his work-piece by the time he reaches the right-side boundary, a utility operator is employed to work on the incomplete work-piece offline for a time which is:

$$T_{\pi(n)}^k - (L_1 + L_2 + \ldots L_n - p_n^k)/V_c$$

Because $ut_n^k \geq 0$, the mathematical equation of utility time can be expressed as:

$$ut_n^k = \text{Max } \{T_{\pi(n)}^k - (L_1 + L_2 + \ldots L_n - p_n^k)/V_c , 0\} \tag{5}$$

From the equation above, it is clear that the utility time depends on the assembly time of the sequenced model, the starting point of the work-piece and the lengths of stations.

Idle Time

Idle time results when an operator is kept waiting for the next work-piece to arrive when he has finished the previous work-piece and has gone back upstream. Idle time occurs when the time needed for an operator to move upstream to meet the next work-piece is longer than the maximum time allowed for this.

Suppose that the time for an operator moving upstream until the starting point p_{n+1}^k is d_n^k/V_0, and the time for the work-piece moving through the distance is $t_c - d_n^k/V_c$, then the idle time it_n^k for an operator waiting for *(k+1)th* work-piece to enter his work area is $t_c - d_n^k/V_c - d_n^k/V_0 = t_c - d_n^k (1/V_c + 1/V_0)$. Since $it_n^k \geq 0$, the mathematical equation of idle time can be expressed as:

$$it_n^k = \text{Max } \{t_c - d_n^k (1/V_c + 1/V_0), 0\} \tag{6}$$

From the equation above, one can observe that the idle time depends on the cycle time or the launch interval time, the operator's upstream distance and the speed of the conveyor and the worker.

The configuration of the utility time and the idle time are shown in Fig. 3.

2.2.3. Upstream Distance

The operator's upstream distance differs from model to model due to variable assembly time, starting point and length of station (Fig.3). There are two cases:

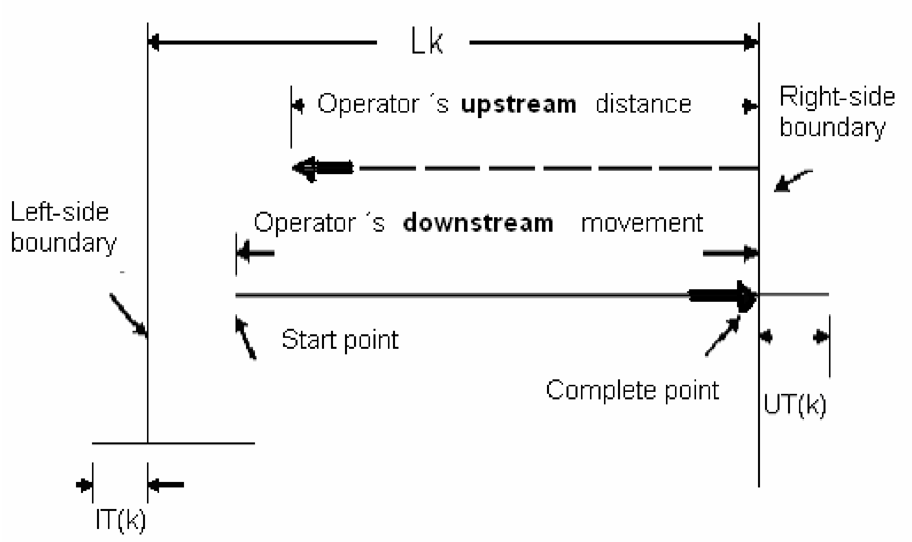


Fig. 3 Operator Moving within a Station (Adopted from Bhaba, R. S. et al [1998])

The operator waits for the work-piece to enter his allowed working area. All stations are considered left-side open except the first one. When the operator waits for the next work-piece to enter the station area, the maximum upstream distance d_n^k is as follows:

$$d_n^k = p_n^{k+1} - p_n^k + V_c \cdot T_{\pi(n)}^k - V_c \cdot ut_n^k \tag{7}$$

The operator starts work within his station

If the next work-piece has entered his station as the operator goes back to the left-side boundary, then the new starting point will be within this boundary. For the cycle time or the launch interval and the conveyor speed are t_c and V_c respectively, the distance between any adjacent work-pieces is $t_c \cdot V_c$; the speed of the operator returning upstream is $V_{0,}$ the time for the operator to access the work-piece is $d_n^k / V_{0.}$ The time for the work-piece moving downstream to meet the operator is $(t_c \cdot V_c - d_n^k)/V_c$, these two times are the same, that is:

$$\frac{d_n^k}{V_0} = \frac{t_c \cdot V_c - d_n^k}{V_c} \quad \text{Or} \quad d_n^k = t_c \cdot V_0 \cdot \left(\frac{V_c}{V_0 + V_c}\right) \tag{8}$$

Considering the equations (7) and (8), upstream distance d_n^k can be expressed as follows:

$$d_n^k = \text{Min}\ \{\ p_n^{k+1} - p_n^k + V_c \cdot (T_{\pi(n)}^k - ut_n^k), t_c \cdot V_0 \cdot (\frac{V_c}{V_0 + V_c})\ \} \tag{9}$$

3. Program

Considering the complexity of the mixed-model-assembly line, in this article a combinatorial optimization program called MMAL is used.

The program include two parts, the first is the calculation part, which calculates the starting point, the completing point, utility time, idle time and the upstream distance; the second part uses the genetic algorithm to obtain the optimal selection. These two parts were carried out in Visual Basic 6.0 on a Pentium 200 MHz computer. The general diagram of the program is shown as Fig. 4.

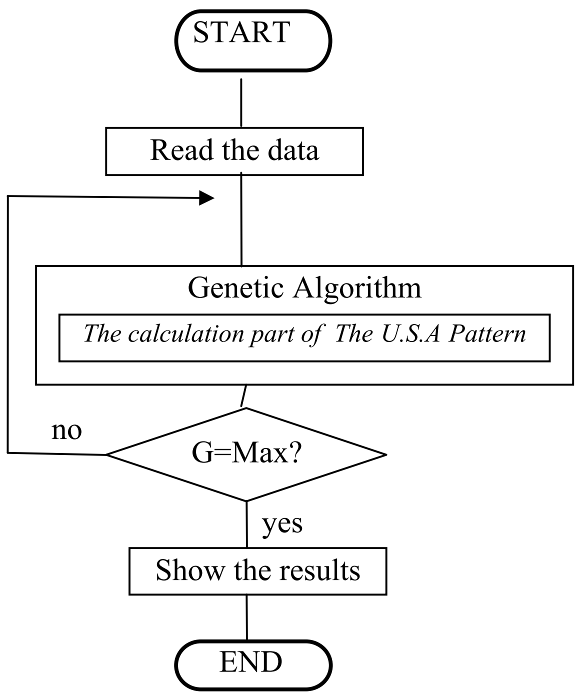


Fig. 4 General Diagram of the Program

3.1 Calculation Part

The following notation is used:

The processes of the calculation are follows:

- ✓ Input the data of K, t_c, v_c, v_k, L^k, M and t_n^k.

- ✓ Input the data of $\pi(n)$.
- ✓ The beginning of the program.
- ✓ Begin to calculate ut_n^k of every station.
- ✓ Calculate every station when there is utility time.
- ✓ Calculate f_n^k
- ✓ Calculate p_n^k
- ✓ Calculate d_n^k
- ✓ Calculate it_n^k

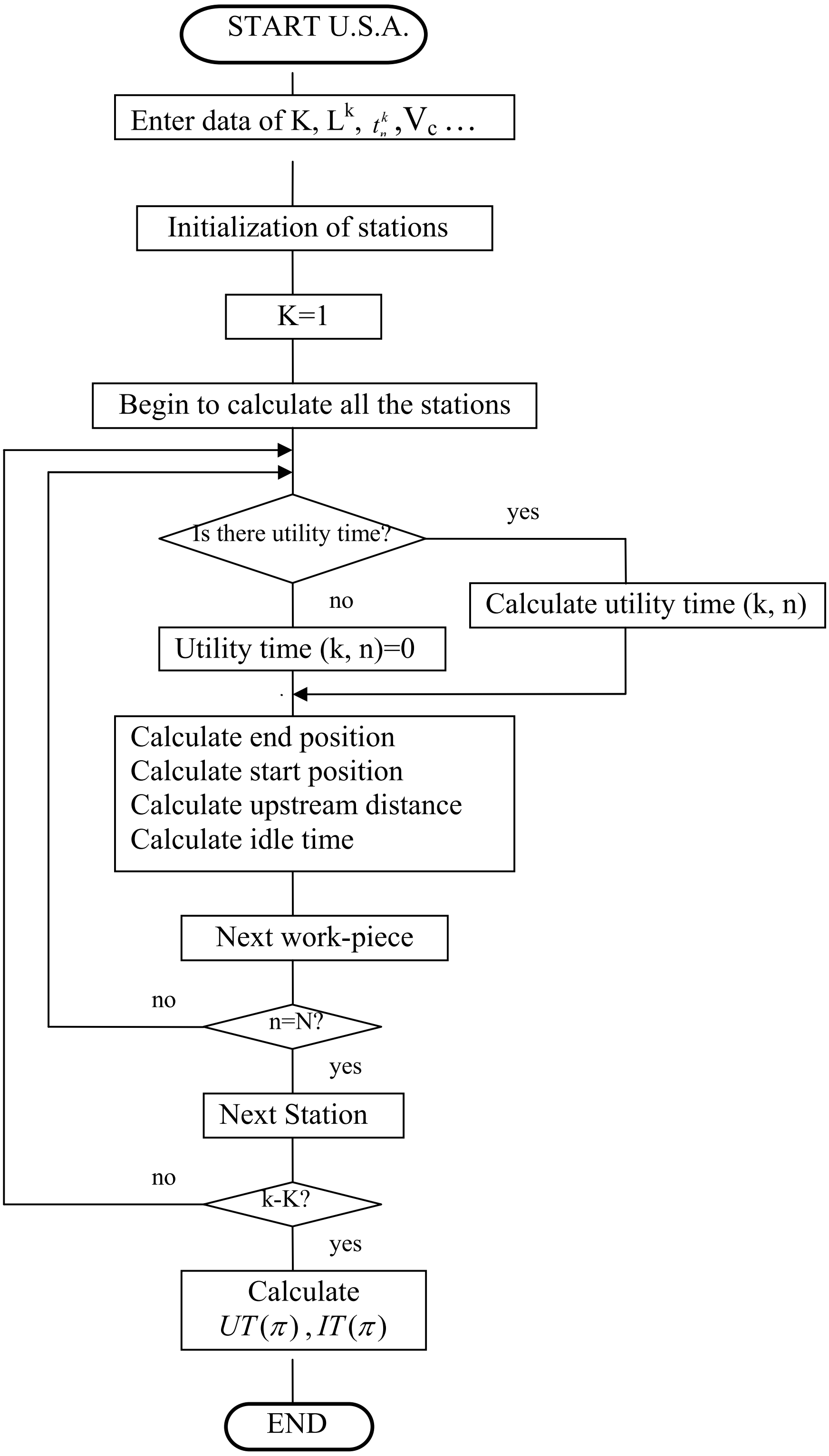


Fig. 5 Calculation Part of the Program of *the United States Pattern*

3.2 Genetic Algorithm Part

The diagram of the second part of the program that is the Genetic Algorithm part is shown as Fig. 6.

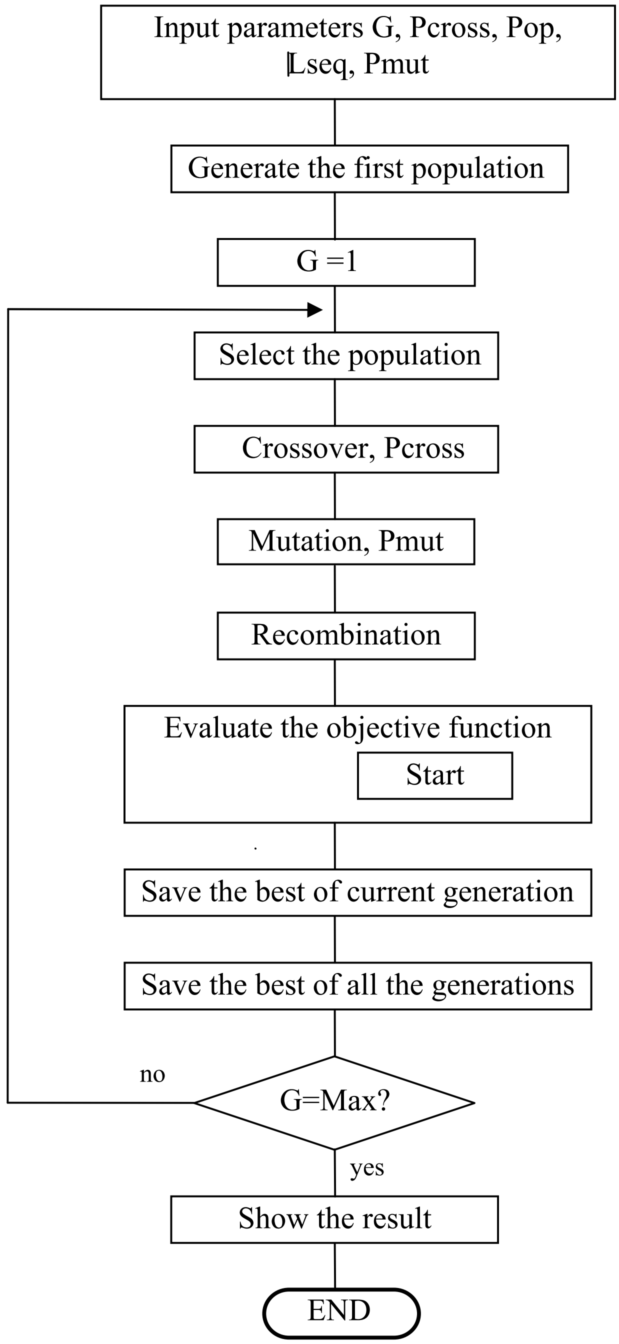


Fig. 6 Genetic Algorithm Part of the Program of *the United States Pattern*

3.2.1 Initialization

Initialization is the first step of the GA. Normally, a population of chromosomes is created, and each chromosome is initialized randomly.

3.2.2 Selection

For the selection process, a roulette wheel with slots sized according to fitness is used. The roulette wheel is constructed as follows [Michalewicz, Z. 1999]:

Calculate the fitness value *eval (vi)* for each chromosome *vi (i=1, …, pop_size).*

Find the total fitness of the population F $=\sum_{i=1}^{pop\_size} eval(v_i)$.

Calculate the probability of a selection pi for each chromosome *vi (i=1,…, pop_size): pi =eval(vi)/F.*

Calculate a cumulative probability qi for each chromosome *vi (i=1,…, pop size):*

qi $=\sum_{j=1}^{i} p_j$

The selection process is based on spinning the roulette wheel pop_size times; each time select a single chromosome for a new population in the following way:

Generate a random number *r* from the range [0…1].

If $r < q_i$ then select the first chromosome *(v_i)*; otherwise select the *i-th* chromosome v_i ($2 \le i \le pop\_size$) such that q_i<r<q_i .

According to the Schema Theorem [Michalewicz, Z. 1999], the best chromosome acquire more copies, the average stay even, and the worst die off. So obviously, some chromosomes would be selected more than once.

3.2.3 Crossover

For realizing crossover between the solutions, it should be defined that one result would be one possible sequence for the assembly line. This means that the model and the number of every model of the work-piece should be equal before and after crossover. In this chapter, order crossover (OX) is utilized. After the crossover, the number of the offspring is equal to their parents.

3.2.4 Mutation

Sometimes, mutation is performed so that a solution or a sequence has an occasional trait that is unique from its parents. This is done so that diversity remains in the gene pool, where the diversity component contributes to robustness in the solution population. In this chapter, the traditional way is used.

It is the same as the process of crossover. The result should be defined so that the model and the number of every model of the work-piece are equal before and after mutation, the number of the offspring is equal to their parents.

4. Design Experiment

For finding the best sequence for the work-pieces with the least idle time and the utility time of the conveyor, this section will discuss the problem of how to select the parameters of the genetic algorithm according to the different parameters of the assembly line (numbers of units, models, stations, and the minimal part set) so that this objective can be achieved.

Name	Description	Range	Value
Unit	Total number of the work-piece needed to fabricate	1)Medium 2)Large	1) 20 2) 50
Model	Total kinds of the work-piece needed to fabricate	1)Medium 2)Large	1) 6 2) 12
Station	Total number of the station in the conveyor	1)Medium 2)Large	1) 5 2) 15
Minimal part set	A smallest possible set of product type quantities, to be called the multiplicities, in which the numbers of assembled products of the various types are in the desired ratios.	1)A/B/C…/N A=B=C…N 2)A/B/C…/N A=50% B=C…N	For example: 20 work-pieces 5 stations: 1)3/3/3/3/5 2)10/2/2/2/2/2
Population size	Number of individuals in each generation of the GA	1)Small 2)Medium 3)Large	1) 25 2) 50 3) 70
Maximum generation	Maximum number of generation	1)Small 2)Medium 3)Large	1) 30 2) 75 3) 100
Crossover ratio	Fraction of selected pairs undergo crossover	1)Small 2)Medium 3)Large	1) 20 2) 45 3) 80
Mutation ratio	Percentage of genes in the population which are replaced with random values each generation.	1)Small 2)Medium 3)Large	1) 10 2) 40 3) 60

Table 1 Parameters Utilized in the Design Experiment

Here, a design experiment will be performed with the parameters of the assembly line and the parameters of Genetic Algorithm. The parameters of the assembly line are: number of units, number of models, number stations and the minimal part set - the first three parameters will be changed randomly between medium and large quantity (the small quantity is quite easy and will not be explained here); the minimal part set will be changed in two cases: one is where the number of all the models of the work-pieces are the same, which is called Case A; the second case is where the number of one model represent 50% of the quantity of all the models, the others models are the same within the other 50% of all the quantity of the work-pieces, which is called Case B. The parameters of the Genetic Algorithm are the population size, maximum generation, crossover ratio and mutation ratio; they will be changed randomly from small, medium to large quantity. Table 1 shows the parameters utilized in the design experiment

4.1 How to Make the Design Experiment

There are 20 parameters in this experiment, 6 of the assembly line, 12 of the genetic algorithm and 2 cases of the minimal part set. While one of them is fixed, other parameters change, so the numbers of experiments that will be done are $2^4 \cdot 3^4=16 \cdot 81=1296$. The combinations of the parameters are: Table 2.

Unit-model-station	Number of every model	Number of every model
20-6-5	A) 3/3/3/3/3/5	B) 10/2/2/2/2/2
20-6-15	A) 3/3/3/3/3/5	B) 10/2/2/2/2/2
20-12-5	A) 2/2/2/2/2/2/2/2/1/1/1/1	B) 9/1/1/1/1/1/1/1/1/1/1/1
20-12-15	A) 2/2/2/2/2/2/2/2/1/1/1/1	B) 9/1/1/1/1/1/1/1/1/1/1/1
50-6-5	A) 8/8/8/8/9/9	B) 25/5/5/5/5/5
50-6-15	A) 8/8/8/8/9/9	B) 25/5/5/5/5/5
50-12-5	A) 4/4/4/4/4/4/4/4/4/4/5/5	B) 25/3/3/3/2/2/2/2/2/2/2/2
50-12-15	A) 4/4/4/4/4/4/4/4/4/4/5/5	B) 25/3/3/3/2/2/2/2/2/2/2/2

Table 2 Combination of the Parameters in the Design Experiment

Because the method of genetic algorithm is quite stochastic, in this experiment, every combination will be made four times, so the number of experiments is $1296 \cdot 4=5184$. The processes and the results of the experiments are shown in the figures and tables.

4.2 The Processes and Results of the Experiments

Unit-model station	Number of every model	Number of every model
20-6-5	A) 3/3/3/3/3/5	B) 10/2/2/2/2/2
20-6-15	A) 3/3/3/3/3/5	**B) 10/2/2/2/2/2 (Example 1)**
20-12-5	A) 2/2/2/2/2/2/2/2/1/1/1/1	**B) 9/1/1/1/1/1/1/1/1/1/1/1 (Example2)**
20-12-15	A) 2/2/2/2/2/2/2/2/1/1/1/1	B) 9/1/1/1/1/1/1/1/1/1/1/1
50-6-5	**A) 8/8/8/8/9/9 (Example 3)**	B) 25/5/5/5/5/5
50-6-15	A) 8/8/8/8/9/9	B) 25/5/5/5/5/5
50-12-5	A) 4/4/4/4/4/4/4/4/4/4/5/5	B) 25/3/3/3/2/2/2/2/2/2/2/2
50-12-15	**A)4/4/4/4/4/4/4/4/4/4/5/5(Example 4)**	B) 25/3/3/3/2/2/2/2/2/2/2/2

Table 3 Examples (in Boldface Type) in the Design Experiment

From 5184 times of experiment with the 16 examples (8 examples and every one of them includes A and B cases), four examples of these 16 will be shown (in Table 3, in boldface type), they are 20-6-15, when the minimal part set is case B (Example 1); 20-12-5, when the minimal part set is case B (Example 2); 50-6-5, when the minimal part set is case A (Example3); 50-12-15, when the minimal part set is case A (Example 4).

Example One, 20-6-15

The first Figure that will be shown is example 1, where the parameters of the assembly line are medium size in units and models, large size in stations and the minimal part set is case B; that is one model represents 50% of all the numbers of units, while the rest of the models are equal in number. In example 1, this means there are 10 work-pieces of one model, while for the other five models there are two.

M=10(mutation ratio is 10)

P=25(population size 25, first column) P=50(population size 50,second column) P=70(population size 70,the third column)

G=30(maximum generation is 30)

C=20(crossover ratio is 20)

Idle time(in black color) 322(population size 25, idle time 322s) 6779(population size 50, idle time 6779s) 10303(population size 70, idle time 10303s)

Utility time(in red color) 3(population size 25, utility time 3s) 38(population size 50, utility time 38s) 19(population size 70, utility time 19s)

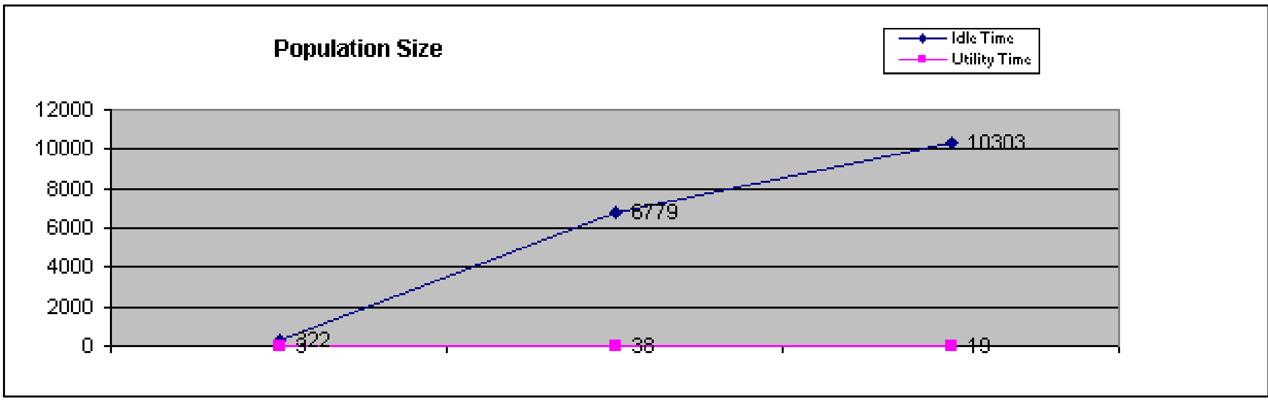


Fig 7 the First Part of Figure A-1 (Time is measured in seconds)

Figure 7 shows one part of Figure A-1. Here the numbers and the letters are all explained to help understanding of Fig A-1.

The results of this combination are shown followed:

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
G		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
C	*	Small		V.G	V.G	V.G
		Medium				
		Large	V.G			
M		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G

Table 4 Experiment Result of Example 1

From the table above, we can conclude that when the parameters of the assembly line are within the range of medium size in unit and model, large size in station, and the minimal part set is case B, the optimal range of the parameters of the genetic algorithm, population size, maximum generation, crossover ration and mutation ratio are large, medium, small and large size respectively.

The detail of other 3 examples 20-12-5, 50-6-5 and 50-12-15 are shown in Appendix.

5. Conclusions

In this article, we have looked at the *United States Pattern.* That is, that if the task can not be finished within the work stations of the assembly line, the operator continues to work on another work-piece, while a utility worker is employed to complete the unfinished work Here the objective is to minimize the utility time and the idle time.

The sequencing problem with the goal of minimizing the total conveyor utility time and idle time has been formulated taking into account the walking times of the operators. Using genetic algorithm, a program has also been developed which is applicable to different scales of assembly lines. From the results it should be said that the program based on genetic algorithm works well and the calculation times for getting the results are all less than two minutes.

With different combinations of the parameters of the assembly and genetic algorithm, more than 5,000 experiments have been done and we have seen the results above. That is with *the United States Pattern*, with a varied range of assembly lines, with the objective of minimizing the utility time and idle time, the optimal parameters of the genetic algorithm are shown as follows:

For the parameters of the assembly line:

- If the unit is in the medium range (in this article there are 20 units), the optimal parameters of the genetic algorithm are: population size, medium; maximum generation, small; crossover ratio, small and mutation ratio, medium, respectively.

- If the unit is in the large range (in this article there are 50 units), the optimal parameters of the genetic algorithm are: population size, small; maximum generation, medium; crossover ratio, large and mutation ratio, small, respectively.

- If the model is in the medium range (in this article there are 6 models), the optimal parameters of the genetic algorithm are: population size, large; maximum generation, medium; crossover ratio, small and mutation ratio, small, respectively.

- If the model is in the large range (in this article there are 12 models), the optimal parameters of the genetic algorithm are all in medium sizes.

- If the station is in the medium range (in this article there are 5 stations), the optimal parameters of the genetic algorithm are: population size, medium; maximum generation, medium; crossover ratio, small and mutation ratio, small, respectively.

- If the station is in the large range (in this article there are 15 stations), the optimal parameters of the genetic algorithm are: population size, small; maximum generation, small; crossover ratio, medium and mutation ratio, small, respectively.

- If the minimal part set is case A (in this article this means that all the models of the work-pieces are equal in number), the optimal parameters of the genetic algorithm are: population size, medium; maximum generation, small; crossover ratio, small and mutation ratio, small, respectively.

- If the minimal part set is case B (in this article this means that one model represents 50% of all the work-pieces, and the other models are equal in number), the optimal parameters of the genetic algorithm are: population size, medium; maximum generation, medium; crossover ratio, small and mutation ratio, medium, respectively.

From the resume above, it can be shown that for the *United States Pattern* most of the optimal parameters of the genetic algorithm are in small and medium sizes while cases of large sizes are very few.

Since work overloads happen frequently in many mixed-model-assembly-lines, minimizing the total conveyor utility time and idle time is vitally important. The methods and results of this article will be useful for future research and investigation.

References

1. Anderson, E. J. and Ferris, M. C. (1994) Genetic Algorithms for Combination Optimization: The Assembly Line Balancing Problem. *OESA Journal on Computing 6, 161-173.*

2. Bautista, J.; Companys, R. and Corominas, A. S. (1996) Heuristic and Exact Algorithm for Solving the Monden Problem. *Europe Journal of Operational Research 88, 101-113.*

3. Bhaba R. S. and Pan, H. X. (1998) Designing a Mixes-Model Assembly Line to Minimize the Costs of Idle and Utility Times. *Computers Industry Engineering, Vol.34, No. 3, 609—628.*

4. Bhaba R. S. and Pan, H. X. (2001) Designing a Mixed-Model, Open Station Assembly Line Using Mixed-Integer Programming. *Journal of Operation Research Society. 52, 545-558.*

5. Dar-EL, E. M. and Cother, R. F. (1975) Assembly Line Sequencing for Model Mix. *International Journal of Production Research 13, 463-477.*

6. Macaskill, J. L. C. (1972) Production Line Balances for Mixed-Model Lines. *Management Science, 19, 423-434.*

7. Michalewicz, Z. (1999) Generate Algorithm + Date Structures = Evolution Programs. ISBN 3-540-58090 2nd ed. Spring-Verlag Berlin Heidelberg New York

8. Miltenburg, J and Goldstein (1989) Scheduling Mixed-Model Multi-Level Just-In-Time Production Systems. *International Journal of Production Research 27, 1487-1509.*

9. Miltenburg, J. (1989) Level Schedule for Mixed-Model Assembly Lines in Just-In-Time Production Systems. *Management Science 35, 192-207.*

10. Miltenburg, J. and Goldstein, D. E. (1991) Developing Production Schedules which Balance Part Usage and Smooth Production Loads for Just-In-Time Production Systems. *Naval Research Logistics 38, 893-910*.

11. Miltenburg, J. and Goldstein, D. E. (1992) Algorithms for Scheduling Multi-Level Just-In-Time Production Systems. *IIE Transactions 24, 121-130*.

12. Monden, Y. (1993) Toyota Production System: An Integrated Approach to Just-In-Time. Second Edition.

13. Niu, H. and Coves, A. M. M. (2004) Scheduling and Sequencing Problem of Mix-Model-Assembly-Line". *¨IE&EM'2003¨, Shanghai, R. P. China.*

14. Niu, H. and Coves, A. M. M. (2005) Sequencing and Balancing Problem of Mixed.Model-Assembly-Line with Window Cycle Time. *Thesis of Doctor's Degree*. Universidad Politécnica de Cataluña.

15. Okamura, K and Yamashina, H. (1979) A Heuristic Algorithm for the Assembly Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor. *International Journal of Production Research 17, 233-247.*

16. Plan, J. and Corominas, A. (2000) A Classification Scheme for Assembly Line Problems. *IOC-DF-P-2000-20, Universitat Politècnica de Catalunya, Spain*

17. Thomopoulos, N. T. (1967) Line Balancing Sequencing for Mixed- Model Assembly. *Management Science 14, B59-B75.*

18. Tsai, L. (1995) Mixed Model Sequencing to Minimize Utility Work and the Risk of Conveyor Stoppage. *Management Science 41, 485—495.*

19. Scholl, A. and S. Voß, S. (1996): Simple Assembly Line Balancing - Heuristic Approaches. *Journal of Heuristics 2,. 217-244.*

20. Scholl. A. (1999) Balancing and Sequencing of Assembly Lines. *Second Edition. Darmstadt, Germany*

21. Yano, C. A. and Rachamadugu, R. (1991) Sequencing to Minimize Work Overload in Assembly Line with Product Options. *Management Science 37, 572-586.*

22. Zhao, X.B. and Ohno, K. (1994) A Sequencing Problem for a Mixed-Model Assembly Line in a JIT Production System . *Computers Industrial Engineering. Vol. 27, 71—74.*

23. Zhao, X. B. and Ohno, K. (1996) Algorithms for Sequencing Mixed Models on An Assembly Line in a JIT Production System. *Computers Industrial Engineering. Vol. 31, No.1 47—56.*

24. Zhao, X. B. and Zhao, Z. (1999) Algorithm for Toyota ´s Goal of sequencing mixed Models on An Assembly Line with Multiple Workstations. *Journal of the Operation research society 50, 704—710.*

25. Zhao, X. B. and Ohno, K. (2000) Properties of A Sequencing Problem for A Mixed Model Assembly Line with Conveyor Stoppages. *Europe Journal of Operation Research 124, 5*

Appendix

In this appendix, the detail of the design experiment of the four examples will be explained.

1. Example 20-6-15

About the Figure A-1, the explanation is:

1) P means Population size, G means Maximum generation, C means Crossover ratio and M means Mutation ration.
2) The idle time and the utility time are shown in figures respectively.
3) All the figures in this example are named Population size because the first, second and the third point of every figure show the corresponding idle time and the utility time when the population size is equal to 25, 50 and 70 respectively.
4) The first, second and third column of figures are the results of M=10, M=40 and M=60 respectively.

5) The first three figures of the first line are the results of C=20, the second three figures of the second line are the results of C=45 and the third three figures of third line are the results of C=80. Again, the fourth three figures are the results C=20, and so on.

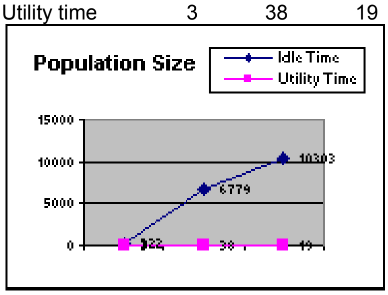
6) The first nine figures are the results of when G=30, the second nine figures are the results of when G=75 and the third ones are when G=100. (Fig. A-1).

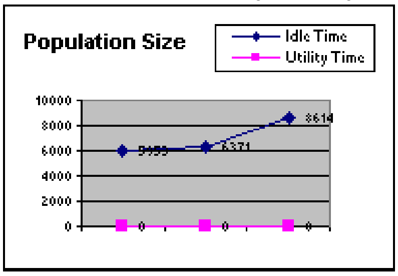
M=10 M=40 M=60

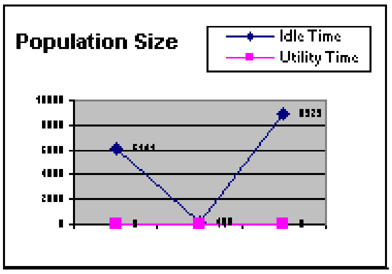
G=30

C=20

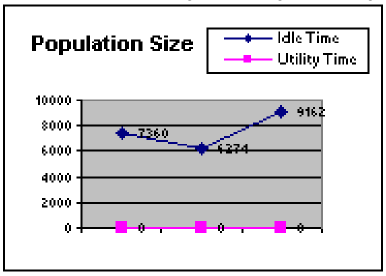
	P=25	P=50	P=70	P=25	P=50	P=70	P=25	P=50	P=70
Idle time	322	6779	10303	5953	6371	8614	6141	158	8923
Utility time	3	38	19	0	0	0	0	11	0

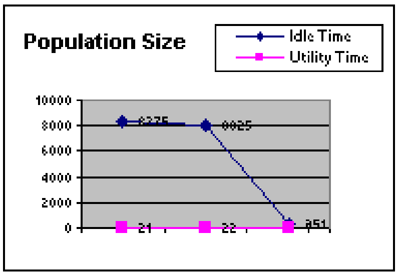


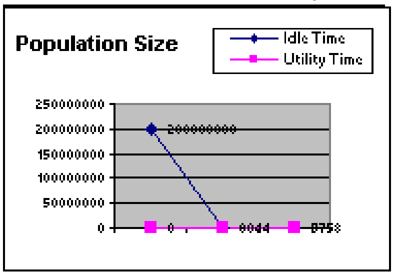




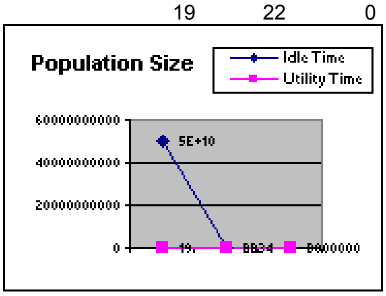
7360	6274	9162	8275	8025	251	2E+08	8044	5758
0	0	0	21	22	7	0	0	0

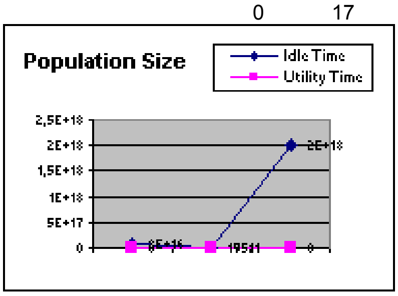


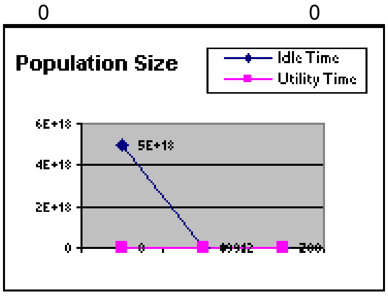




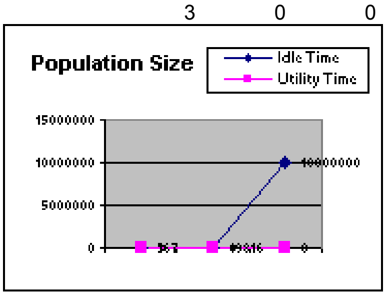
5E+10	9934	3E+06	8E+16	19511	2E+18	5E+18	19912	200
19	22	0	0	17	0	0	0	7

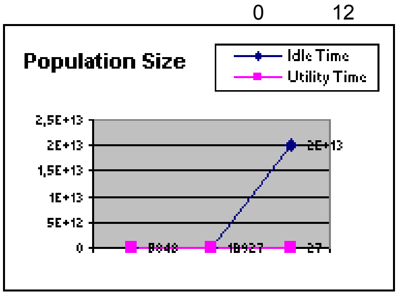


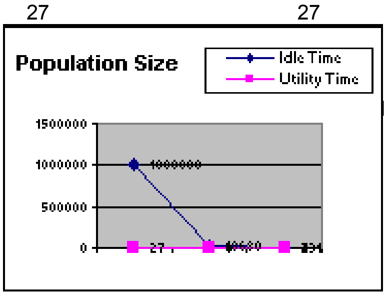




167	19816	1E+07	5848	19927	2E+13	1E+06	18680	231
3	0	0	0	12	27	27	0	4

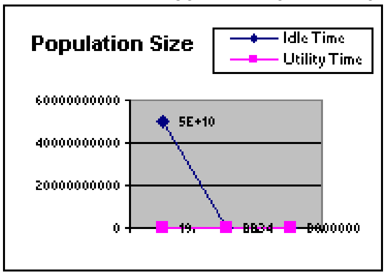


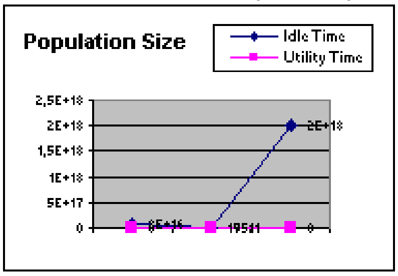


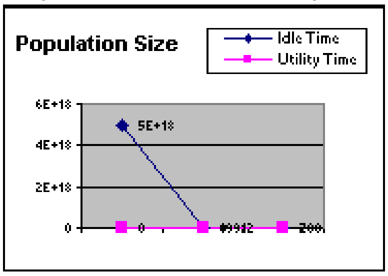


72 153

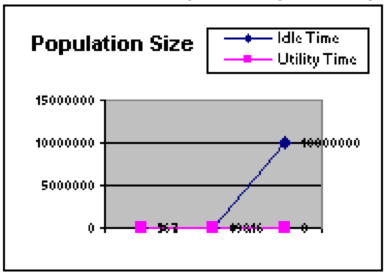
30 0 0 0 0 0 0 0 10

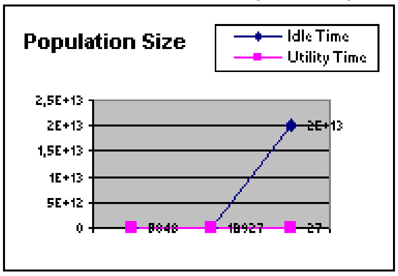


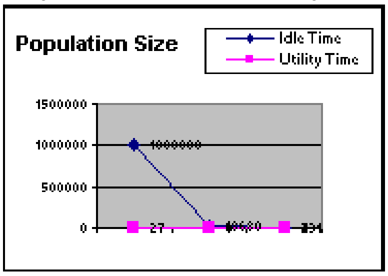




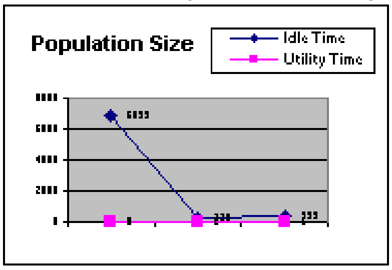
3E+18 164 6E+09 3E+18 8770 6E+09 119 9181 6E+09
0 10 0 0 0 0 19 0 0

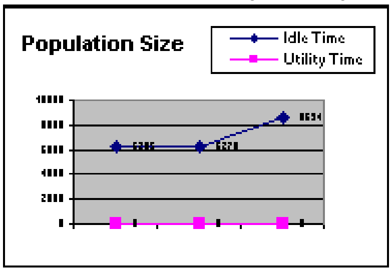


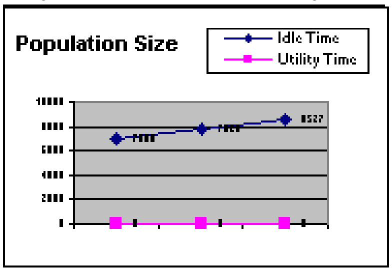




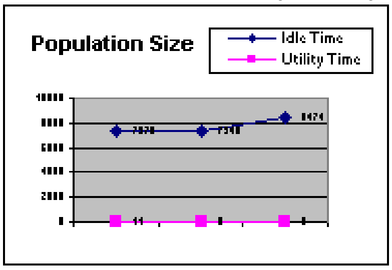
6833 228 339 6285 6270 8634 7000 7820 8527
0 2 5 0 0 0 0 0 0

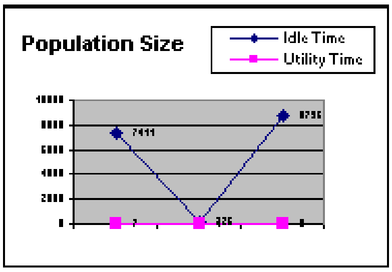


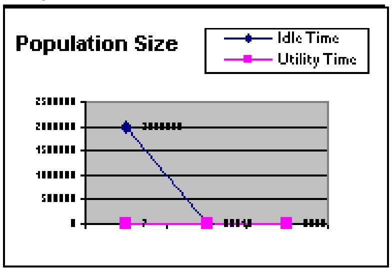




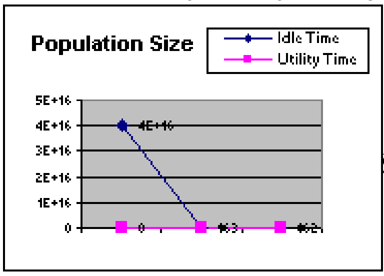
7370 7348 8474 7411 226 8796 2E+06 14010 8893
11 0 0 7 4 0 7 26 0

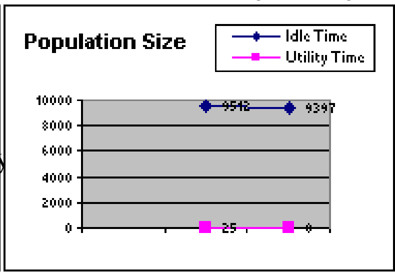


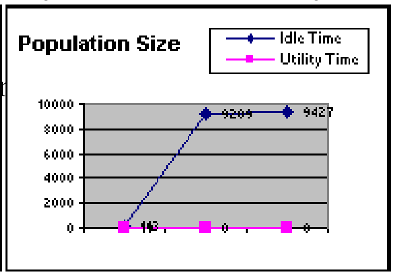




4E+16 163 162 1060 9512 9397 113 9209 9427
0 9 8 8 25 0 19 0 0







From the figures above, it is shown that:

A: About the idle time:

In all the figures, there are thirteen cases of idle time less than 300 units (some explanations about selecting the standard for judging the idle time will be shown in 5.6.2.2.2). When P=25 there are 3 cases, which represent 23.1% of the thirteen cases, when P=50 and P=70, there are 5 cases respectively, which represent 38.5% of all the thirteen cases respectively; when G=30, there are 3 cases, which represent 23.1% of the thirteen cases, and when G=75 and 100, there are 5 cases respectively, which represent 38.5% of all the thirteen cases respectively; when C=20 there are 4 cases which represent 30.8% of the thirteen cases, when C=45 and C=80, there are 3 and 6 cases respectively, which represent 23.1% and 46.1% respectively; when M=10, there are 5 cases, which represent 38.5% of the thirteen cases, when M=40 and M=60, there are 2 and 6 cases respectively which represent 15.4% and 46.1% respectively.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	3	23.1%		3	23.1%		4	30.8%		5	38.5%	
Medium size	5	38.5%	*	5	38.5%	*	3	23.1%		2	15.4%	
Large size	5	38.5%	*	5	38.5%	*	6	46.1%	*	6	46.1%	*
Total number and percentage	13	100%		13	100%		13	100%		13	100%	

Table A-1 Analysis of the Idle Time of Example 1

So, when the idle time is less than 300 units, the optimal parameters of the genetic algorithm are P=50 and P=70, which are the medium and large size, G=75 and G=100, which are the medium and large size; C=80, which is large size and M=60 which also is the large size. The process of analysis can be explained in Table A-1, where * means the optimal results.

B: About the utility time:

There are 49 cases in which the utility time equals zero, meaning that there isn't any utility time, which is also what we want. Within these, there are 15 cases when P=25 and P=50 respectively, representing 30.6% respectively of all the 49 cases; 19 cases when P=70 which represent 38.8% of 49 cases; when G=30, 75 and 100, there are 16, 18 and 15 cases respectively, which represent 32.7%, 36.7% and 30.6% respectively; when C=20, 45 and 80, there are 16, 17 and 16 cases respectively, which represent 32.7%, 34.7% and 32.7% respectively; when M=10, 40 and 60, there are 14, 17 and 18 cases respectively, which represent 28.6%, 34.7% and 36.7% respectively.

So, when utility time equals zero, the optimal parameters of genetic algorithm are P=70, G=75, C=45 and M=60, which are large, medium, medium and large size respectively. The analysis process can be explained in Table A-2.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	15	30.6%		16	32.7%		16	32.7%		14	28.6%	
Medium size	15	30.6%		18	36.7%	*	17	34.7%	*	17	34.7%	
Large size	19	38.8%	*	15	30.6%		16	32.7%		18	36.7%	*
Total number and percentage	49	100%		49	100%		49	100%		49	100%	

Table A-2 Analysis of the Utility Time of Example 1

From A and B, that is, with the consideration of both the idle time and the utility time, we can make a preliminary experiment conclusion about this part:

Item	Best	Range	First running
P		Small	
		Medium	
	*	Large	V.G
G		Small	
	*	Medium	V.G
		Large	
C		Small	
		Medium	
	*	Large	V.G
M		Small	
		Medium	
	*	Large	V.G

Table A-3 Results for the first test of example 1

For the combination example 1, that is for the parameters of the assembly line, medium size of unit and model, large size for the station, the optimal genetic algorithm parameters are all in large sizes. The results can be shown in Table A-3 (Where: G means good result and V.G means very good result, * means the best value).

Because the genetic algorithm is quite stochastic, the test should be repeated in order to obtain more exact results.

With the same parameters of the assembly line and genetic algorithm, the experiment is repeated another three times as above. Comparing all the figures, these are the following results: Table 4 (This table has been explained before).

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
G		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
C	*	Small		V.G	V.G	V.G
		Medium				
		Large	V.G			
M		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G

2. Example Two, 20-12-5

The combination of example 2, which is 20-12-5, that is 20 units, 12 models and 5 stations, the minimal part set is case B.

2.1 Results Analysis

A: About the idle time:

There are nine cases equal to or less than 100 units (some explanations can be seen follows), when P=25, 50 and 70 there are 2, 3 and 4 cases respectively, which represent 22.2%, 33.3% and 44.5% of the nine cases respectively; when G=30, 75 and 100, there are 3, 2 and 4 cases respectively, which represent 33.3%, 22.2% and 44.5% of nine cases respectively; when C=20, 45 and 80, there are 5, 2 and 2 cases respectively, which represent 55.6%, 22.2% and 22.2% respectively; when M=10, 40 and 60, there are 6, 2 and 1 cases respectively, which represent 66.7%, 22.2% and 11.1% respectively.

So, when the idle time is less than 100 units, the optimal parameters of the genetic algorithm are P=70, which is the large size; G=100, which is the large size; C=20, which is small size and M=10 which is also the small size.

The analysis process is explained in Table A-4.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	2	22.2%		3	33.3%		5	55.6%	*	6	66.7%	*
Medium size	3	33.3%		2	22.2%		2	22.2%		2	22.2%	
Large size	4	44.5%	*	4	44.5%	*	2	22.2%		1	11.1%	
Total number and percentage	9	100%		9	100%		9	100%		9	100%	

Table A-4 Analysis of the Idle Time of Example 2

Here two things should be noted:

1) The standard for judging the idle time is different within the two examples above; for example 1 this is less than 300 units while for example 2 it is less than 100 units, because of the different combinations of the parameters of the assembly line, and also of the genetic algorithm. Therefore, with the change in the actual idle time, the benchmark should be changed accordingly.

2) In these experiments, the judging standard is decided by the results of idle time and utility time, and the number of the cases which are within the standard is variable with different combinations. Another method for deciding is that no matter the combination, the benchmark is always the best 10% or 20% of all the results of the idle time and utility time. This method can be used in later research.

B: About the utility time:

There are 19 cases where the utility time is equal to zero. Within them, when P= 25, 50 and 70 respectively, there are 2, 12 and 5 cases respectively, which represent 10.5%, 63.2% and 26.3% respectively; when G=30, 75 and 100, there are 7, 6 and 6 cases respectively, which represent 36.8%, 31.6% and 31.6% respectively; when C=20, 45 and 80, there are 5, 6 and 8 cases, which represent 26.3%, 31.6% and 42.1% respectively; when M=10, 40 and 60, there are 4, 8 and 7 cases respectively, which represent 21.1%, 42.1% and 36.8% respectively.

So, when utility time equals zero, the optimal parameters of genetic algorithm are P=50, G=30, C=80 and M=40, which are medium, small, large and medium size respectively. The analysis process is explained in Table A-5.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	2	10.5%		7	36.8%	*	5	26.3%		4	21.1%	
Medium size	12	63.2%	*	6	31.6%		6	31.6%		8	42.1%	*
Large size	5	26.3%		6	31.6%		8	42.1%	*	7	36.8%	
Total number and percentage	19	100%		19	100%		19	100%		19	100%	

Table A-5 Analysis of the Idle Time of Example 2

From A and B, a preliminary experiment conclusion about this part can be made as follows:
For example 2, that is for the parameters of the assembly line which are medium size in units and station and large size in models, the optimal parameters of genetic algorithm are when population size is in medium size, maximum generations in small size, crossover ratio in large size, and mutation ratio in medium sizes.

With the same parameters of the assembly line and genetic algorithm, the experiment is repeated another three times as above. Comparing all the figures, the results are as follows: Table A-6.

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
G		Small			G	G
	*	Medium	V.G		G	G
		Large		V.G	G	G
C	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
M		Small		G	G	G
	*	Medium	V.G	G	G	G
		Large				

Table A-6 Experiment Result of Example 2

From this table, we can conclude that when the parameters of assembly line, number of units and stations are medium size, and the number of models is large size, the optimal range of the parameters of the genetic algorithm, population size, maximum generation, crossover ration and mutation ratio are medium, medium, small and medium size respectively.

3. Example Three, 50-6-5

The combination of 50-6-5, which the units is 50 (large size), the number of model is 6 (medium size) and the number of the station is 5 (media size), the minimal part set is case A.

3.1 Results Analysis

A: About the idle time:

There are nineteen cases less than 50 units, when P=25, 50 and 70 there are 6, 7 and 6 cases respectively, which represent 31.6%, 36.8% and 31.6% of the nineteen cases respectively; when G=30, 75 and 100, there are also 6, 7 and 6 cases respectively; when C=20, 45 and 80, there are 7, 4 and 8 cases respectively, which represent 36.8%, 21.1% and 42.1% respectively; when M=10, 40 and 60, there are 9, 4 and 6 cases respectively, which represent 47.4%, 21.1% and 31.6% respectively.

So, when the idle time is less than 50 units, the optimal parameters of the genetic algorithm are P=50, which is the medium size; G=75, which also is the medium, C=80, which is the large size and M=10 which is the small size.

The analysis process is explained in Table A-7.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	6	31.6%		6	31.6%		7	36.8%		9	47.4%	*
Medium size	7	36.8%	*	7	36.8%	*	4	21.1%		4	21.1%	
Large size	6	31.6%		6	31.6%		8	42.1%	*	6	31.6%	
Total number and percentage	19	100%		19	100%		19	100%		19	100%	

Table A-7 Analysis of the Idle Time of Example 3

B: About the utility time:

There are 34 cases where the utility time is equal to zero. Within them, when P= 25, 50 and 70 respectively, there are 11, 10 and 13 cases respectively, which represent 32.4%, 29.4% and 38.2% respectively; when G=30, 75 and 100, there are 6, 14 and 14 cases respectively, which represent 17.6%, 41.2% and 41.2% respectively; when C=20, 45 and 80, there are also 6, 14 and 14 cases; when M=10, 40 and 60, there are 10, 13 and 11 cases respectively, which represent 29.4%, 38.2% and 32.4% respectively.

So, when utility time equals zero, the optimal parameters of genetic algorithm are P=70, G=75 or 100, C=45 or 80 and M=40, which are large, medium or large, medium or large and medium size respectively. The analysis process is explained in Table A-8.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	11	32.4%		6	17.6%		6	17.6%		10	29.4%	
Medium size	10	29.4%		14	41.2%	*	14	41.2%	*	13	38.2%	*
Large size	13	38.2%	*	14	41.2%	*	14	41.2%	*	11	32.4%	
Total number and percentage	34	100%		34	100%		34	100%		34	100%	

Table A-8 Analysis of the Idle Time of Example 3

From A and B, a preliminary experiment conclusion about this part can be made:

For example 3, that is for the parameters of the assembly line which are medium size in model and station, and large size in units, the optimal parameters of genetic algorithm are when population size is in large size, maximum generations in medium size, crossover ratio in large size, and mutation ratio in small sizes.

With the same parameters of the assembly line and genetic algorithm, the experiment is repeated another three times as above. Comparing all the figures, the results are as follows: Table A-9.

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
		Medium			V.G	
	*	Large	V.G	V.G		V.G
G		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
C		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
M	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

Table A-9 Experiment result of Example 3

From the table above, a conclusion can be made that when the parameters of assembly line, number of units and stations are the medium size, the number of model is the large size, and the minimal part set is case A, the optimal range of the parameters of the genetic algorithm, population size, maximum generation, crossover ration and mutation ratio are large, medium, large and small size respectively.

4. Example Four, 50-12-15

The example in this part is 50-12-15, where the number of units is 50 (large size), the number of models is 12 (large size) and the number of stations is 15 (large size); the minimal part set is case A.

4.1 Results Analysis

A: About the idle time:

There are 10 cases equal to 136,000 units, when P=25, 50 and 70, there are 5, 3 and 2 cases respectively, which represent 50%, 30% and 20% of 10 cases respectively; when G=30, 75 and 100, there are 3, 6 and 1 cases respectively, which represent 30%, 60% and10% respectively; when C=20, 45 and 80, there are 3, 4 and 3 cases respectively, which represent 30%, 40% and 30% of the 10 cases respectively; when M=10, 40 and 60, there are 6, 2, and 2 cases respectively, which represent 60%, 20% and 20% of the 10 cases respectively.

So, when the idle time is less than 136,000 units, the optimal parameters of the genetic algorithm are P=25, which is the small size; G=75, which is the medium size; C=45, which is medium size and M=10 which is the small size. The analysis process is explained in Table A-10

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	5	50%	*	3	30%		3	30%		6	60%	*
Medium size	3	30%		6	60%	*	4	40%	*	2	20%	
Large size	2	20%		1	10%		3	30%		2	20%	
Total number and percentage	10	100 %		10	100 %		10	100 %		10	100 %	

Table A-10 Analysis of the Idle Time of Example 4

B: About the utility time:

There are 52 cases when the utility time is equal to zero, within them, when P=25, 50 and 70, there are 12, 16 and 24 cases respectively, which represent 23.1%, 30.8% and 46.1% respectively; when G=30, 75 and 100, there are 18, 18 and 16 cases respectively, which represent 34.6%, 34.6% and 30.8% respectively; when C=20, 45 and 80, there are 19, 17 and 16 cases respectively, which represent 36.5%, 32.7% and 30.8% respectively of 52; when M=10, 40 and 60, there are 17, 16 and 19 cases respectively, which represent 32.7%, 30.8% and 36.5% respectively of all the 52 cases

So, when the utility time equals to zero, the optimal parameters of the genetic algorithm are P=70, which is the large size; G=30 or 75, which is the small or medium size; C=20, which is small size and M=60 which is the large size. The analysis process is explained in Table A-11.

Parameters of GA / Range	Population size			Maximum generation			Crossover ratio			Mutation ratio		
Small size	12	23.1%		18	34.6%	*	19	36.5%	*	17	32.7%	
Medium size	16	30.8%		18	34.6%	*	17	32.7%		16	30.8%	
Large size	24	46.1%	*	16	30.8%		16	30.8%		19	36.5%	*
Total number and percentage	52	100%		52	100%		52	100%		52	100%	

Table A-11 Analysis of the Idle Time of Example 4

With the consideration of A and B, a preliminary experiment conclusion about this part can be made:

For example 4, that is when the parameters of the assembly line are all in large size, the optimal parameters of genetic algorithm are when population size are small and large size, maximum generations is medium size, crossover ratio is medium size and mutation ratio is small size.

With the same parameters of the assembly line and genetic algorithm, the experiment is repeated three times as above. Comparing all the figures, the results are shown in Table A-12.

From the table, we can conclude that in example 4, that is, under the condition of large size of all the parameters of the assembly line, the optimal parameters for the genetic algorithm are small population size and mutation ratio respectively, medium size for maximum generation and crossover ratio respectively.

Item	Best	Range	First running	Second running	Third running	Fourth running
P	*	Small	V.G			G
		Medium			V.G	
		Large		V.G		G
G		Small				
	*	Medium	V.G	G	V.G	V.G
		Large		G		
C		Small		V.G		
	*	Medium	V.G		V.G	V.G
		Large				
M	*	Small		V.G	V.G	V.G
		Medium				
		Large	V.G			

Table A-12 Experiment result of Example 4

5. <u>Resume</u>

With different combinations of the parameters of the assembly line and genetic algorithm, more than 5,000 tests have been done. From these experiments, the following tables and conclusions can be made:

- Table A-13, 20-6-5 in case A

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
G		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
C	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
M	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

From the table above, the conclusion of example 20-6-5, that is, under the conditions of small sizes of all the parameters of the assembly line, the optimal parameters for the genetic algorithm are medium size for population size, large size for maximum generation, small size for crossover ratio and mutation ratio respectively.

- Table A-14, 20-6-5 in case B

Item	**Best**	**Range**	**First running**	**Second running**	**Third running**	**Fourth running**
P		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
G		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
C	*	Small	V.G	V.G		V.G
		Medium				
		Large			V.G	
M		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				

From the table above, when the parameters of the assembly line are all in medium size and the minimal part set is case B, the optimal parameters of the genetic algorithm are large population size, medium size maximum generation, small size crossover ratio, and medium size mutation ratio respectively.

- Table A-15, 20-6-15 in case A

Item	**Best**	**Range**	**First running**	**Second running**	**Third running**	**Fourth running**
P		Small	V.G			
	*	Medium		V.G		V.G
		Large			V.G	
G	*	Small		V.G	V.G	
		Medium	V.G			V.G
		Large				
C	*	Small	V.G	V.G		V.G
		Medium				
		Large			V.G	
M		Small	G	G		G
		Medium	G	G	V.G	G
	*	Large				G

From the table above, when the parameters of the assembly line are medium size in units, medium size in models and large size in stations respectively, and the minimal part set is case A, the optimal parameters of the genetic algorithm are medium size population size, small size maximum generation and crossover ratio respectively, and medium size mutation ratio.

- Table 20-6-15 in case B (example 1)

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
G		Small				
	*	Medium	G	V.G	V.G	V.G
		Large	G			
C	*	Small		V.G	V.G	V.G
		Medium				
		Large	V.G			
M		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G

This table has been explained in before

- Table A-16, 20-12-5 in case A

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
G	*	Small	G	G	V.G	V.G
		Medium	G	G		
		Large				
C	*	Small	V.G		V.G	V.G
		Medium		G		
		Large		G		
M	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

From the table above, when the parameters of the assembly line are medium size in units, large size in models and medium size in stations respectively, and the minimal part set is case A, the optimal parameters of the genetic algorithm are medium in population size, and small size in maximum generation, in crossover ratio and in mutation ratio respectively

- Table 20-12-5 in case B (example 2)

Item	Best	Range	First running	Second running	Third running	Fourth running
	*	Medium	V.G	G	G	G
		Large				
P		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
G		Small			G	G
	*	Medium	V.G		G	G
		Large		V.G	G	G
C	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
M		Small		G	G	G

This table has been explained in before

- Table A-17, 20-12-15 in case A

Item	Best	Range	First running	Second running	Third running	Fourth running
P	*	Small	V.G		V.G	V.G
		Medium		V.G		
		Large				
G	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
C		Small			G	
		Medium				
	*	Large	V.G	V.G	G	V.G
M		Small		V.G		
		Medium				
	*	Large	V.G		V.G	V.G

From the table above, when the parameters of the assembly line are medium size in units, large size in models and stations respectively, and the minimal part set is case A, the optimal parameters of the genetic algorithm are small size in population size and in maximum generation respectively, large size in crossover ratio, and in mutation ratio respectively

- Table A-18, 20-12-15 in case B

Item	Best	Range	First running	Second running	Third running	Fourth running
P	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
G	*	Small	V.G	V.G		V.G
		Medium			V.G	
		Large				
C	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
M		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G

From the table above, when the parameters of the assembly line are medium size unites, large size models and stations respectively, and the minimal part size in case B, the optimal parameters of the genetic algorithm are small size in population size, maximum generation and in crossover ratio respectively, large size in mutation ratio.

- Table 50-6-5 in case A (example 3)

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
		Medium			G	G
	*	Large	V.G	V.G	G	G
G		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
C		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
M	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

This table has been explained in before

- Table A-19, 50-6-5 in case B

Item	Bes t	Range	First running	Second running	Third running	Fourth running
P		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
G		Small	V.G			
		Medium		G	G	
	*	Large		G	G	V.G
C	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
M		Small	G	G	G	
		Medium				G
	*	Large	G	G	G	G

From the table above, when the parameters of the assembly line are large size in units, medium size in models and stations respectively, and the minimal part set is case B, the optimal parameters of the genetic algorithm are large size in population size, maximum generation and in mutation ratio respectively, small size in crossover ratio.

- Table A-20, 50-6-15 in case A

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small				
	*	Medium			V.G	V.G
		Large	V.G	V.G		
G	*	Small		V.G	V.G	V.G
		Medium	V.G			
		Large				
C		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
M	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

From the table above, when the parameters of the assembly line are large size in units, medium size in models, large size in stations respectively, and the minimal part set is case A, the optimal parameters of the genetic algorithm are medium size in population size and crossover ratio respectively, small size in maximum generation and in mutation ratio respectively

- Table A-21, 50-6-15 in case B

Item	Best	Range	First running	Second running	Third running	Fourth running
P	*	Small		V.G	V.G	V.G
		Medium				
		Large	V.G			
G		Small				
		Medium				
	*	Large	V.G	V.G	V.G	V.G
C		Small				
	*	Medium	V.G	V.G	V.G	V.G
		Large				
M	*	Small	G	V.G	G	V.G
		Medium				
		Large	G		G	

From the table above, when the parameters of the assembly line are medium size in units, large size in models and medium size in stations respectively, and the minimal part set is case B, the optimal parameters of the genetic algorithm are small size in population size and mutation ratio respectively, large size in maximum generation, medium size in crossover ratio.

- Table A-22, 50-12-5 in case A

Item	Best	Range	First running	Second running	Third running	Fourth running
P	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
G	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				
C	*	Small	G	V.G	V.G	V.G
		Medium				
		Large	G			
M	*	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

From the table above, when the parameters of the assembly line are large sizes in units and models, and medium size in stations respectively and the minimal part set is case A, the optimal parameters of the genetic algorithm are all in small sizes.

- Table A-23, 50-12-5 in case B

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small		V.G		
	∗	Medium	V.G		V.G	V.G
		Large				
G		Small				
		Medium				
	∗	Large	V.G	V.G	V.G	V.G
C		Small				
	∗	Medium	V.G	V.G	V.G	V.G
		Large				
M	∗	Small	V.G	V.G	V.G	V.G
		Medium				
		Large				

From the table above, when the parameters of the assembly line are large sizes in units and models and medium size in stations respectively, and the minimal part set is case B, the optimal parameters of the genetic algorithm are medium size in population size and crossover ratio, large size in maximum generation size and small size in mutation ratio.

- Table 50-12-15 in case A (example 4)

Item	Best	Range	First running	Second running	Third running	Fourth running
P	∗	Small	V.G			V.G
		Medium			V.G	
		Large		V.G		
G		Small				
	∗	Medium	V.G	G	V.G	V.G
		Large		G		
C		Small		V.G		
	∗	Medium	V.G		V.G	V.G
		Large				
M	∗	Small		V.G	V.G	V.G
		Medium				
		Large	V.G			

This table has been explained in before.

- Table A-24, 50-12-15 in case B

Item	Best	Range	First running	Second running	Third running	Fourth running
P		Small	G			V.G
	∗	Medium	G	V.G	V.G	
		Large	G			
G	∗	Small	G	V.G	V.G	V.G
		Medium	G			
		Large				
C		Small	V.G			G
	∗	Medium		V.G	V.G	G
		Large				G
M		Small				
	∗	Medium	V.G	V.G	V.G	V.G
		Large				

From the table above, when the parameters of the assembly line are all in large size, and the minimal part set is case B, the optimal parameters of the genetic algorithm are all in medium size.

In future research, investigation and practical application, the genetic algorithm above can be used if the industrial organization pattern is just like that of *the United States Pattern*, and the objective of the assembly line is to minimize idle time and utility time., This article has identified the optimal parameters corresponding to the genetic algorithm to be selected according to the scale of the assembly line.