

# **Assembly Line Balancing Problem of Sewing Lines in Garment Industry**

**James C. Chen**

**Department of Industrial Engineering and Engineering Management  
National Tsing Hua University  
Hsinchu, Taiwan, R.O.C.**

**Chun-Chieh Chen**

**Department of Industrial and Systems Engineering  
Chung Yuan Christian University  
Chung Li, Taiwan, R.O.C**

**Yi-Jhen Lin, Chun-Ju Lin, and Tiffany Y. Chen**

**Department of Industrial Engineering and Engineering Management  
National Tsing Hua University  
Hsinchu, Taiwan, R.O.C.**

## **Abstract**

Garment manufacturing is a traditional industry with global competition. The most critical manufacturing process is sewing, as it generally involves a great number of operations. A balanced sewing line can reduce labor requirement, increase production efficiency, and reduce production cycle time. Assembly line balancing problem (ALBP) is known as an NP-hard problem. Thus, heuristic methodology could be a better way to plan the sewing lines within a reasonable time.

This paper uses grouping genetic algorithm (GGA) to solve types I ALBP with different labor skill levels in sewing lines of garment industry. Type I ALBP minimizes the number of workstations for a given cycle time. Genetic algorithm (GA) is a search heuristic that generates solutions to optimization problems using techniques inspired by natural evolution, such as mutation, selection, and crossover. GGA is a GA modified to suit the structure of grouping problems. Real data from garment factories and experimental design are used to evaluate GGA's performance. Production managers can use the research results to quickly design sewing lines for important targets such as short cycle time and high labor utilization.

## **Keywords**

Assembly line balancing problem, Grouping genetic algorithm, Garment industry, Labor skill level

## **1. Introduction**

Textile industry is one of the world's major industries and the garment industry is a substantial one within the supply chain of textile industry. As shown in Figure 1, the production process of garments is separated into four main phases: designing/clothing pattern generation, fabric cutting, sewing, and ironing/packing. The most critical phase is the sewing phase, as it generally involves a great number of operations.

The sewing line consists of a set of workstations in which a specific task in a predefined sequence is processed. In general, one to several tasks are grouped into one workstation. Tasks are assigned to operators depending on the constraints of different labor skill levels. Finally, several workstations in sequence are formed as a sewing line. Shop floor managers are concerned about the number of workstations. Inappropriate workstations assignment will lead to the increase of labor cost, WIP, cycle time and poor throughput. These shop floor supervisors arrange tasks to workstations based on their experience in practice. As a result, the line balance performance cannot be guaranteed from one supervisor to another with different assignment preference and/or work experience.

This study presents grouping genetic algorithm (GGA) to solve the assembly line balancing problem (ALBP) with different labor skill levels and minimum number of workstations in garment industry. This paper is organized as follows. Section 2 reviews related literature. Section 3 defines the problem. Next, Section 4 and 5 present the algorithm and results. Finally, Section 6 draws conclusions and provides future work directions.



Figure 1: Garment manufacturing processes.

## 2. Literature review

Research related to assembly line balancing problem is first reviewed. Research in genetic algorithm (GA) and GGA is then discussed.

### 2.1. Assembly line balancing problem

Assembly line balancing is the problem of assigning various tasks to workstations, while optimizing one or more objectives without violating any restrictions imposed on the line. ALBP has been an active field of research over the past decades due to its relevancy to diversified industries such as garment, footwear and electronics. Ghosh and Gagnon (1989) as well as Erel and Sarin (1998) provided detailed reviews on these topics. Configurations of assembly lines for single and multiple products could be divided by three line types, single-model, mixed-model and multi-model. Single-model assembles only one product, and mixed-model assembles multiple products, whereas a multi-model produces a sequence of batches with intermediate setup operations (Becker & Scholl, 2006). This paper solves single-model line balancing problem with real application. Boysen, Fliedner, and Scholl (2007, 2008) classified ALBPs and pointed out that there were less than 5% articles explicitly solving line balancing of real world assembly systems. As a result, for practical consideration, this paper focuses on the real case of an assembly line in garment manufacturing.

ALBP with different objectives could be classified into several types called ALBP-I to ALBP-V (Tasan and Tunali 2008, Kim et al. 1996), and among these ALBP-I, ALBP-II, and ALBP-III are the most popular ones addressed in recent literature. ALBP-I and ALBP-II have a dual relationship. ALBP-I is formulated with the objective of minimizing the number of workstations used to meet a target cycle time. It can result in low labor costs and reduced space requirements. ALBP-II tries to minimize the cycle time for a given number of workstations and therefore can maximize the production rate. ALBP-III maximizes the workload smoothness, for a given number of workstations. Workload smoothness is accomplished by leveling the workload distribution as evenly as possible, and it can lead to the throughput increase of a line and the equity in task assignment among workers. ALBP-IV maximizes the work relatedness by arranging the tasks in a workstation that interrelated tasks are arranged into the same workstation as much as possible, so that work efficiency can be improved. ALBP-V is corresponding to the multiple objectives with ALBP-III and ALBP-IV considered simultaneously. ALBP-I problem will be the type of problem addressed in this research.

1. ALBP-I: minimizes the number of workstations, for a given cycle time.
2. ALBP-II: minimizes the cycle time, for a given number of workstations.
3. ALBP-III: maximizes the workload smoothness, for a given number of workstations.
4. ALBP-IV: maximizes the work relatedness by arranging the tasks in a workstation.
5. ALBP-V: consider the multiple objectives with ALBP-III and ALBP-IV simultaneously.

### 2.2. Genetic algorithm and grouping genetic algorithm

Among the methods, GA (presented by John Holland in 1975) is the most popular one having been employed in solving ALBP, as compared with simulated annealing and tabu search (Holland, 1975). Based on evolutionary computation techniques, GA method has received much attention and has been applied successfully in many research fields. Within recent years, many studies have been made on the applications of GA to ALBP of different industries. GA is also one of the most popular heuristic algorithms in the deterministic approach adopted to solve ALBP (e.g., Brown & Sumichrast, 2005; Yong, Yeo, & Yongkyun, 1998). Recently, GA was applied to the production planning and scheduling problems (e.g., Balin, 2011; Chan, Au, & Chan, 2006; Chan, Chung, & Chan,

2005; Chen & Ji, 2007; Chen, Pan, & Lin, 2008; Musharavati & Hamouda, 2011) and line balancing problems in textile manufacturing (e.g., Hsu, Hsiung, Chen, & Wua, 2009; Wong, Chan, & Ip, 2000). Wong, Mok, and Leung (2006) also used GA to optimize the assignment of operators for a garment assembly line.

A number of researchers have successfully applied GA to solve problems in garment manufacturing. Wong, Kwong, Mok, Ip, and Chan (2005) described a GA approach to solve the job sequencing optimization problem of the manual fabric-cutting process for garment manufacturing. Wong, Leung, and Au (2005) focused on pre-sewing stage and proposed a real-time GA-based rescheduling approach to handle the production planning and scheduling problem in dynamic garment manufacturing environment. Wong, Mok, and Leung (2006) applied GA to optimize the operator assignment and minimize makespan of garment assembly line. Song, Wong, Fan, and Chan (2006) proposed an optimal operator allocation solution for solving ALBP with the consideration of the operator efficiency variance. Guo, Wong, Leung, Fan, and Chan (2006) developed a GA universal mathematical model of the job shop scheduling problem of the garment assembly process. Hsu et al. (2009) presented a GA scheduling approach for yarn-dyed textile manufacturing. These publications illustrate the potential of GAs to address the garment industry assembly line balancing problems.

However, the disadvantage of GA is that the algorithm does not incorporate weak problem-specific information. If better solutions can be achieved by including this type of information, it may be advantageous to enhance GA in this manner. Brown and Sumichrast (2005) indicated that GGA indeed is a more efficient technique for solving grouping problems than the standard GA. Falkenauer (1992) created GGA as a GA specifically for grouping problems. Since 1992, many researchers have proposed reasons for expecting GGA to perform more efficiently than GA on grouping problems. GGA was applied to problems such as ALBP (Falkenauer, 1992), bin packing, machine-part cell formation (e.g., Brown & Sumichrast, 2005; Tabitha, Evelyn, & Kellie, 2007), and registration area planning problem (e.g., as Tabitha, Vroblefski, & Nottingham, 2007). Chen, Hsaio, Chen, and Sun (2009) applied GGA to ALBP of sewing lines in garment industry. The computational results from their research reveal that GGA outperforms GA in the proposed problems.

In summary, real case applications for ALBPs are relatively few in literatures. Some academic studies may neglect significant factors affecting the decision-making on assembly line balancing in labor intensive industries, such as labor skill levels. Soft-computing-based methods like GA were widely employed to solve ALBPs. This research applies GGA to solve real case ALBP-I in garment manufacturing.

### **3. Problem Definition**

In garment industry, any garment manufacturer tries the best to finish the assembly work soon to increase on-time delivery and to increase machine/labor utilization to reduce production cost. Assembly line balancing is therefore a critical issue for the efficiency and competitiveness.

ALBP in sewing line assigns a given set of tasks to an ordered set of work stations and ensures that the precedence constraints of tasks are satisfied while the given performance measure is optimized. Assume that there are 10 tasks in a sewing line, Figure 2 depicts the precedence relationship between these tasks. Each task may have one or more predecessors, e.g., task 9 can be processed after the completion of tasks 5, 6 and 8. Workstations are tailored and configured for specific tasks, e.g., task 1 and 2 are grouped into a task set and assigned to workstation 1. These are the results generated from ALBP algorithms.

In a sewing line, each workstation consists of machines and operators of different types, and Figure 3 depicts the relationship between machine types, skill level, labor types and workstations. The operator of different labor types can operate different machine types. For example, the labor type X can operate machine types A, B and C, but labor type Y can operate machine types C and D only. The numbers of skill levels of labor types X and Y equal 3 and 2, respectively. Furthermore, tasks are grouped to a task set and assigned to a workstation in a sewing line. For example, task 3, 5 and 6 are grouped as task set b and assigned to workstation 2. In this example, a task can be done by machines of different types and also by operators of different labor types.

Before proceeding with the analysis of the problem, the assumptions of the model are listed below:

1. The processing time of each task is a constant.
2. Processing times, precedence relationships and machine types are known.

3. No task processing time is larger than the cycle time.
4. Task processing times are additive and independent of the task sequence.
5. The sewing line consists of a series of workstations and the number of workstations is not fixed.
6. Setup time of a task is included in processing time.
7. Task assignments to workstations should not violate precedence requirements.
8. Each task must be assigned to one of a specified set of machines.
9. One workstation has only one operator.
10. One workstation may have one to several machines but only one machine for each machine type.

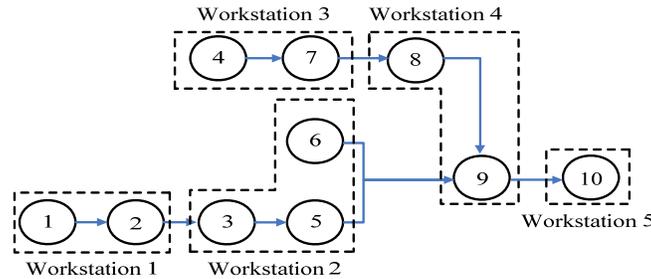


Figure 2: Precedence relationship for a sewing line.

Table 1 defines the notations used in this research that focuses on a deterministic model denoted as an ALBP-I with labor skill levels. Problem formulation is shown in Eqs. (1)–(11). Eq. (1) presents minimization of the number of workstations module (MNWM). Eq. (2) Chiang (1998) proved that a nonlinear objective function that maximizes the sum of squares of processing times in workstations (SSPTW) can be used to minimize the number of workstations. Eq. (3) ensures that every task is assigned to exactly one workstation. Eq. (4) requires the limitation of the number of machine types that can be assigned to each workstation, i.e., the maximum operator skill level for each workstation. Eq. (5) guarantees that each task is assigned to one machine type at one workstation. Eq. (6) ensures the precedence requirements, i.e. a task is assigned to a workstation only if all its predecessors have been assigned to the same workstation or earlier workstations. Eq. (7) provides cycle time constraint at all workstations. Eq. (8) requires the maximum number of tasks in workstation. Eqs. (9) - (11) limit the binary situations of variables.

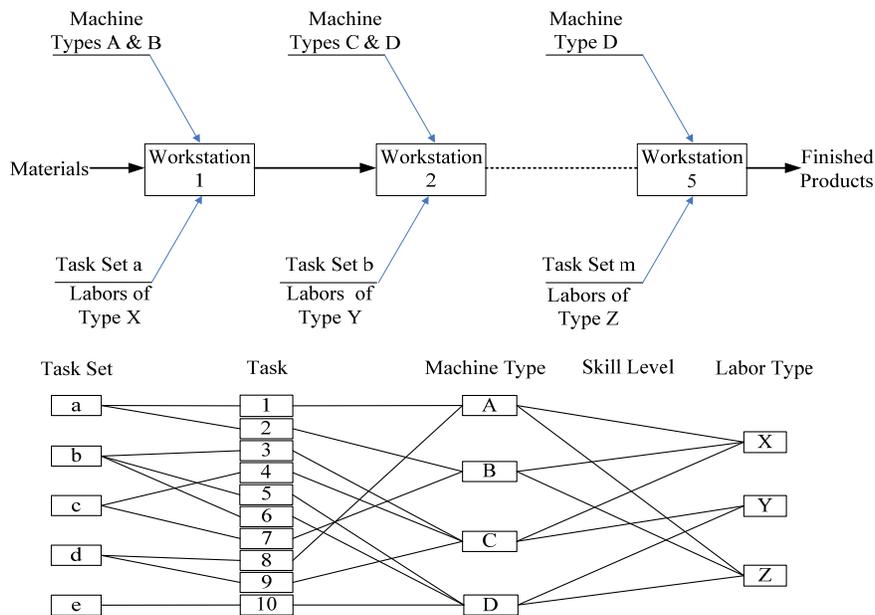


Figure 3: A sewing line with task, task set, machine type, skill level, and labor type.

Table 1: Notation list.

i	Index of task, where $i = 1, 2, \dots, N$
j	Index of workstation, where $j = 1, 2, \dots, W$
k	Index of machine type, where $k = 1, 2, \dots, M$
N	Total number of tasks
W	Total number of workstations
M	Total number of machine types
R	Number of machine types in workstation
Q	Limitation of maximum number of tasks in workstation
$t_i$	Standard processing time of task i
$t_{ij}$	The time to finish the i-th task in workstation j
CT	Cycle time (the average time between two successive products coming out of the end of the line)
$N_j$	Number of tasks assigned to workstation j
$X_{ij}$	1: if task i is assigned to workstation j; 0: otherwise
$Y_{jk}$	1: if machine type k is allocated to workstation j; 0: otherwise
$Z_{ijk}$	1: if task i is assigned to machine type k at workstation j; 0: otherwise

$$\text{Minimize } f_1 = W \quad (1)$$

$$\text{Maximize } SSPTW = \sum_{j=1}^W \left( \sum_{i \in I_j} t_i \right)^2 \quad (2)$$

$$\sum_{j=1}^W X_{ij} = 1 \quad (3)$$

$$\sum_{k=1}^M Y_{jk} \leq R \quad (4)$$

$$\sum_{i=1}^N \sum_{j=1}^W Z_{ijk} = 1 \quad (5)$$

$$\sum_j j X_{aj} \leq \sum_j j X_{bj} \quad \text{if task a precedes task b} \quad (6)$$

$$\sum_{i \in I_j} t_i \cdot X_{ij} \leq CT \quad (7)$$

$$N_j \leq Q \quad (8)$$

$$X_{ij} \in \{0,1\} \quad (9)$$

$$Y_{jk} \in \{0,1\} \quad (10)$$

$$Z_{ijk} \in \{0,1\} \quad (11)$$

## 4. Methodology

This section presents the implementation of GGA for ALBP-I with different labor skill levels. As depicted in Figure 4, the proposed algorithm consists of three modules-data load module, minimization of the number of workstations module and data output module. First, task processing times of different machine types, task precedence relationships, machine and labor types and constraints are collected in data load module. Second, minimization of the number of workstations module is activated to balance the assembly line by using GGA with SSPTW as the objective function.

Finally, a balanced assembly line is generated and performance indices can be collected in data output module. GGA for minimization of the number of workstations module is discussed in following sections. Section 4.1 presents the chromosome encoding. Section 4.2 describes the genetic operators for GGA. Section 4.3 depicts the steps for the initial solution of the problem.

### 4.1. Encoding

The following notation is used for chromosome representation:  $G_{N_1}G_{N_2} \dots G_{N_N}G_{P_1}G_{P_1}G_{P_2} \dots G_{P_P}G_{H_1}G_{H_2} \dots G_{H_W}|1$   
 $2 \dots W, (W+1) (W+2) \dots (W+P), (W+P+1) (W+P+2) \dots (W+P+PN)$ . The chromosome consists of three parts, where  $G_{N_i}$  indicates that task i is assigned to workstation j ( $j = 1, 2, \dots, W$ ),  $G_{P_u}$  indicates that the operator type u ( $u = 1, 2, \dots, P$ ) is assigned to workstation j, and  $G_{H_j}$  indicates that the number of operators are assigned to workstation j.

The chromosome encoding of the proposed GGA can be explained by the following example. Suppose that there are ten tasks to be assigned to four workstations. There are two labor types, and the number of operators in a workstation is limited from one to three. The encoding is expressed as  $G_{N_1}G_{N_2} \dots G_{N_{10}}G_{P_1}G_{P_2} \dots G_{P_4}G_{H_1}G_{H_2} \dots G_{H_4}|$

123456789,  $G_{Ni} \in \{1,2,3,4\}$ ,  $G_{Pu} \in \{5,6\}$ ,  $G_{Hj} \in \{7,8,9\}$ . The example chromosome is expressed as  $\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 2 & 3 & 1 & 3 & 1 & 4 & 4 & 5 & 6 & 6 & 5 & 7 & 9 & 8 & 7 \end{matrix} | 1234$ . The main chromosome consists of eighteen genes with numerical values to indicate workstation assignments, labor type assignment and number of operators' assignment in workstations. It would assign task one, two, six and eight into workstation 1, task three and four into workstation 2, task five and seven into workstation 3, task nine and ten into workstation 4. Labor types assigned in workstation 1, 2, 3, and 4 are labor type 5 - 4 = 1, 6 - 4 = 2, 6 - 4 = 2 and 5 - 4 = 1, respectively. Number of operators in workstation 1, 2, 3, and 4 are 7 - 6 = 1, 9 - 6 = 3, 8 - 6 = 2 and 7 - 6 = 1, respectively.

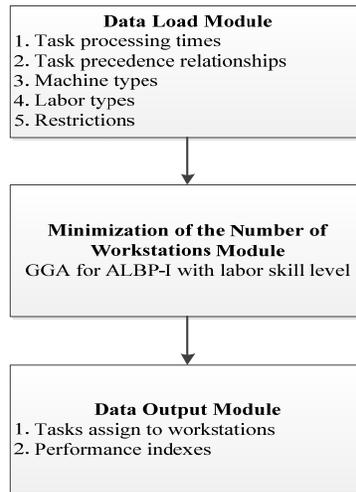


Figure 4: The framework for line balancing algorithm.

#### 4.2. Genetic operators

A GGA is an iterative procedure. Either heuristic procedures or random creations can be used to generate feasible strings that form the initial population. Tasan and Tunali (2008) noted that most of the researchers generated the initial populations randomly, and the sizes of the population setting from recent GA researches are between 20 and 100. In this research, 50 and 100 individuals are used, and the individuals of initial population are randomly generated.

The selection in GAs, based on the natural law of the survival of the fittest among string creatures, is the process in which strings are selected for the next generation in terms of their fitness. The selection policy employed herein is the rank-based roulette-wheel selection, where the probability of selection is proportional to the relative solution quality as measured by the fitness function. The fitness function of MNWM was chosen as the objective function listed in Eq. (12).  $FV_i$  in Eq. (12) denotes the fitness function of  $i$ -th chromosome of MNWM and  $SSPTW_i$  means the objective function of  $i$ -th chromosome of MNWM.

$$FV_i \text{ of } MNWM = SSPTW_i \quad (12)$$

When a solution chromosome for the ALBP is created through crossover, it has the role to combine pieces of information from different individuals in the population. In this study, two parent chromosomes have been selected by rank-based. A pair of crossover points is randomly selected for both parents. The five-step crossover operator from Falkenauer's GGA is applied in this study (Brown & Sumichrast, 2005; Falkenauer, 1992; Lit et al., 2000).

Mutation operator is to reorder the chromosome structure. The mutation operator explored in this research consists of random selection of two points to determine the two changing points of the selected parent. The number of generations is the most straightforward stopping criteria for a GGA.

#### 4.3. Initial solution

A heuristic based procedure is proposed to create initial population of MNWM, as depicted in Figure 5 with the following steps.

- Step 1: Read in list of tasks including processing time and machine type.  
 Step 2: Select a task by precedence relationships.  
 Step 3: List the workstations in ascending order of its available processing time (APT). Determine if there are any workstations whose APT are greater than zero. If yes, these workstations are kept in order of APT and are considered as available workstations (AWs). Select an AW. If there exists no AW, create a new one and link it to the AW list.  
 Step 4: If the machine type of the selected task is the same as the machine type of the equipment in the selected AW, go to Step 5; otherwise, go to step 3.  
 Step 5: If the processing time  $T_i$  of the selected task  $i$  is less than the APT of the selected AW, go to Step 6; otherwise, go to step 3.  
 Step 6: If the number of tasks of the selected AW exceeds the limitation of the number of tasks in a workstation, go to Step 3; otherwise, go to step 7.  
 Step 7: Assign the selected task to the selected AW whose inclusion does not violate all the hard constraints in Equations (3) to (11) and update APT in the AW.  
 Step 8: If there exist unassigned tasks, return to Step 2; otherwise, stop. Repeat the above steps until all tasks are assigned to workstations.

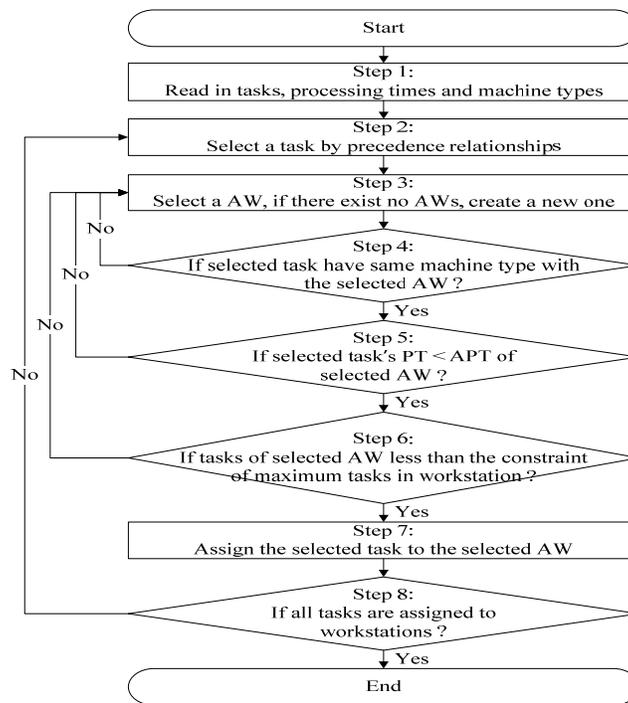


Figure 5: Steps of initial solution

## 5. Experimental Results

### 5.1. Experimental problems

Data from the sewing department of a garment factory producing sportswear in south China are collected for the performance evaluation of GGA. Table 2 shows two sets of sewing line ALBP, one is simple problem including 34 tasks assigned to 20 workstations, and the other is complex problem including 176 tasks assigned to 54 workstations. The heuristic-based GGA is coded in Evolver using Excel.

Table 2: Description of simple and complex problems.

Item	Simple problem	Complex problem
Number of tasks	34	176
Machine type	7	5
Cycle time	60	177
Number of precedence requirement constraints	42	181

### 5.2. Parameter setting

A 2\*3\*3\*3 experimental design for complex problem is presented to evaluate the performance of parameters for proposed GGA. The control factors are population size, crossover rate, mutation rate and operator skill level as listed in Table 3. The experiment is performed in the basic condition of operator skill level, operator efficiency, and number of operators in workstations is set to one. There are 54 experiment sets and each experiment set is run by two times to calculate SSPTW. Results are analyzed using descriptive statistics and analysis of variance (ANOVA) to identify significant main and interaction effects. All ANOVAs are run by MINITAB. The significance level for all statistical analyses is 0.05.

Table 3: Experimental design of parameter setting for GGA.

Factor	Level
population size	50, 100
crossover rate	0.3, 0.5, 0.7
mutation rate	0.001, 0.005, 0.01
operator skill level	1, 2, 3

### 5.3. Experiments for simple problem

As shown in Table 4, computational results demonstrate that crossover rate, mutation rate, operator skill level and interaction between factors of crossover rate and mutation rate significantly influence SSPTW. Figure 6 shows the main effects plot of SSPTW for simple problem, and the best parameter setting of population size, crossover rate, mutation rate and Operator skill level are 50, 0.3, 0.01 and 3, respectively. The termination criterion for further experiments is therefore set to 35,000 generations.

Table 4: ANOVA table of SSPTW for simple problem. (A: population size, B: crossover rate, C: mutation rate, D: operator skill level)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
A	1	54675	54675	54675	2.98	0.088
B	2	243633	243633	121817	6.64	0.002
C	2	322142	322142	161071	8.77	0.000
D	2	1943164699	1943164699	971582349	52927.55	0.000
A*B	2	77312	77312	38656	2.23	0.114
A*C	2	92522	92522	46261	2.52	0.086
A*D	2	81020	81020	40510	2.34	0.103
B*C	4	246651	246651	61663	3.36	0.013
B*D	4	163068	163068	40767	2.22	0.073
C*D	4	146493	146493	36623	2.00	0.102
Error	86	1578688	1578688	18357		
Total	107	1946012571				

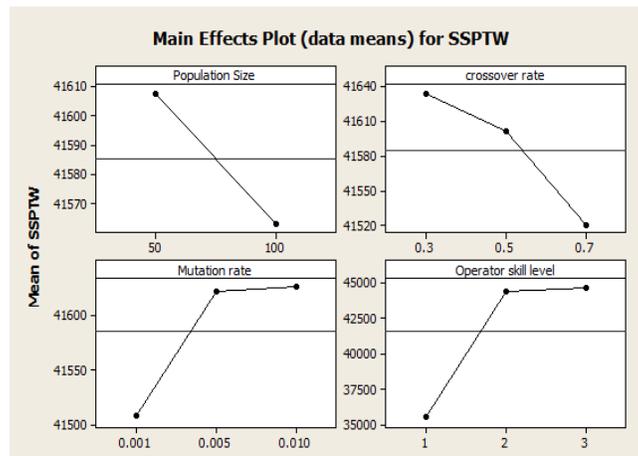


Figure 6: Main effects plot of the SSPTW for simple problem

#### 5.4. Experiments for complex problem

As shown in Table 5, computational results of MNWM illustrate that mutation rate and operator skill level significantly influence the SSPTW measure, and the interactions between factors of population size cross mutation rate, mating rate cross number of staff skills, mutation rate cross number of staff skills are also significant to the SSPTW measure. Figure 7 shows the main effects plot of SSPTW for complex problem, and the best parameter setting of population size, crossover rate, mutation rate and Operator skill level are 50, 0.5, 0.01 and 3, respectively. The termination criterion for further experiments is therefore set to 50,000 generations.

Table 5: ANOVA table of SSPTW for complex problem. (A: population size, B: crossover rate, C: mutation rate, D: operator skill level)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
A	1	4795195	4795195	4795195	0.11	0.737
B	2	246696436	246696436	123348218	2.91	0.060
C	2	997117146	997117146	498558573	11.76	0.000
D	2	1.85090E+11	1.85090E+11	92545072958	2182.97	0.000
A*B	2	224072621	224072621	112036311	2.64	0.077
A*C	2	803866742	803866742	401933371	9.48	0.000
A*D	2	95553154	95553154	47776577	1.13	0.329
B*C	4	84863846	84863846	21215962	0.50	0.735
B*D	4	1227536353	1227536353	306884088	7.24	0.000
C*D	4	553623507	553623507	138405877	3.26	0.015
Error	82	3476320852	3476320852	42394157		
Total	107	1.92805E+11				

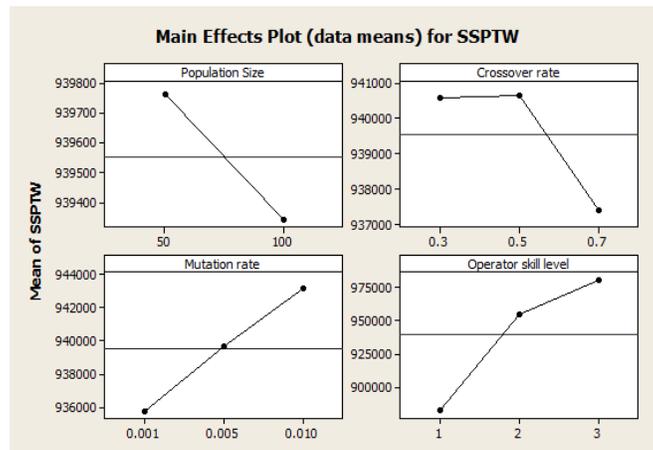


Figure 7: Main effects plot of the SSPTW for complex problem

#### 6. Conclusions

A line balancing algorithm is developed for sewing line in garment industry. The proposed algorithm applies GGA to minimize the number of workstations. Simple problem's best parameter setting of population size, crossover rate, mutation rate and Operator skill level are 50, 0.3, 0.01 and 3, respectively. Complex problem's best parameter setting of population size is 50, crossover rate is 0.5, mutation rate is 0.01 and operator skill is 3.

This study can be extended in several directions. An extension would be considering multiple operators in workstation and operator skill efficiency by different skill levels. Multiple objectives integrating line efficiency, smoothness index, cost, and reliability is also of interests. This study deals with ALBP of single model and the modification of the proposed algorithm of mixed-model and multi-model is also the extension of further researches.

#### Acknowledgements

The research was supported in part by National Science Council of Taiwan, R.O.C. (Contract No. NSC101-2221-E-007-134 -MY3).

## References

- Agpak, K., & Gokcen, H. (2005). Assembly line balancing: Two resource constrained cases. *International Journal of Production Economics*, 96, 129–140.
- Balin, S. (2011). Non-identical parallel machine scheduling using genetic algorithm. *Expert Systems with Applications*, 38, 6814–6821.
- Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168, 694–715.
- Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, 183, 674–693.
- Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111, 509–528.
- Brown, E. C., & Sumichrast, R. T. (2005). Evaluating performance advantages of grouping genetic algorithms. *Engineering Applications of Artificial Intelligence*, 18, 1–12.
- Chan, F. T. S., Au, K. C., & Chan, P. L. Y. (2006). A decision support system for production scheduling in an ion plating cell. *Expert Systems with Applications*, 30, 727–738.
- Chan, F. T. S., Chung, S. H., & Chan, P. L. Y. (2005). An adaptive genetic algorithm with dominated genes for distributed scheduling problems. *Expert Systems with Applications*, 29, 364–371.
- Chen, J. C., Hsaio, M. H., Chen, C. C., & Sun, C. J. (2009). A grouping genetic algorithm for the assembly line balancing problem of sewing lines in garment industry. In *Proceedings of the eighth international conference on machine learning and cybernetics* (pp. 2811–2816).
- Chen, K. J., & Ji, P. (2007). A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with a frozen interval. *Expert Systems with Applications*, 33, 1004–1010.
- Chen, J. S., Pan, J. C. H., & Lin, C. M. (2008). A hybrid genetic algorithm for the reentrant flow-shop scheduling problem. *Expert Systems with Applications*, 34, 570–577.
- Erel, E., & Sarin, S. C. (1998). A survey of the assembly line balancing procedures. *Production Planning and Control*, 9, 414–434.
- Falkenauer, E. (1992). The grouping genetic algorithm-widening the scope of the GAs. *JORBEL- Belgian Journal of Operations Research, Statistics and Computer Science*, 33, 79–102.
- Ghosh, S., & Gagnon, R. J. (1989). A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, 27, 637–670.
- Guo, Z. X., Wong, W. K., Leung, S. Y. S., Fan, J. T., & Chan, S. F. (2006). Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the garment industry. *Computers & Industrial Engineering*, 50, 202–219.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications in biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press.
- Hsu, H. M., Hsiung, Y., Chen, Y. Z., & Wua, M. C. (2009). A GA methodology for the scheduling of yarn-dyed textile production. *Expert Systems with Applications*, 36, 12095–12103.
- Musharavati, F., & Hamouda, A. S. M. (2011). Modified genetic algorithms for manufacturing process planning in multiple parts manufacturing lines. *Expert Systems with Applications*, 38, 10770–10779.
- Song, B. L., Wong, W. K., Fan, J. T., & Chan, S. F. (2006). A recursive operator allocation approach for assembly line-balancing optimization problem with the consideration of operator efficiency. *Computers & Industrial Engineering*, 51, 585–608.
- Tabitha, J. L., Evelyn, B. C., & Kellie, K. B. (2007). A hybrid grouping genetic algorithm for the cell formation problem. *Computers & Operations Research*, 34, 2059–2079.
- Tabitha, J., Vroblefski, M., & Nottingham, Q. (2007). A hybrid grouping genetic algorithm for the registration area planning problem. *Computer Communications*, 30, 2180–2190.
- Tasan, S. O., & Tunali, S. (2008). A review of the current applications of genetic algorithms in assembly line balancing. *Journal of Intelligent Manufacturing*, 19(1), 49–69.

## Biography

**James C. Chen** is Professor in the Department of Industrial Engineering and Engineering Management at National Tsing-Hua University (NTHU), Taiwan. He received a B.S. in Industrial Engineering from NTHU, Taiwan, an M.S.

in Manufacturing Systems Engineering, and a Ph.D. in Industrial Engineering, both from the University of Wisconsin- Madison. His research interests include advanced planning and scheduling, lean production, supply chain management, business process reengineering, and project management. Dr. Chen was awarded Dr. Yi-Chi Mei Scholarship at NTHU in 1983, IBM Manufacturing Research Graduate Fellowship 1991-92, Outstanding Teaching Awards at CYCU, Distinguished University-Industry Collaborative Research Award at Chung Hua Picture Tubes (CPT) in 2009, Distinguished Industrial Engineer Award from Chinese Institute of Industrial Engineers (CIIE) in 2011, Outstanding Research Award at NTUST in 2011, and Feature Person: Enjoying the International University-Industry Collaboration, Engineering Science and Technology Bulletin, NSC in 2011.

**Yi-Jhen Lin** is a graduate student in the Department of Industrial Engineering and Engineering Management at National Tsing Hua University. He received a B.S. from Department of Industrial and Systems Engineering at Chung Yuan Christian University in 2012. His research interest is line balancing.