공학석사 학위논문

Integrated Mixed Model Assembly Line Balancing with Temporary Workers

임시 작업자를 활용한 혼합모델 조립라인의 통합적 균형화 연구

2015 년 2 월

서울대학교 대학원

산업공학과

김 동 욱

Integrated Mixed Model Assembly Line Balancing with Temporary Workers

지도 교수 문 일 경

이 논문을 공학석사 학위논문으로 제출함
2015년 1월

서울대학교 대학원

산업공학과

김 동 욱

김동욱의 공학석사 학위논문을 인준함
2014년 12월

위 원 장 ______ 홍 유 석 ______ (인)

부위원장 ______ 이 경 식 ______ (인)

위 원 ______ 문 일 경 ______ (인)

Abstract

Integrated Mixed Model Assembly Line Balancing with Temporary Workers

Dongwook Kim

M.S. in Industrial Engineering

The Graduate School

Seoul National University

This study extends a single-model assembly line balancing problem to an integrated mixed-model assembly line balancing problem by incorporating temporary unskilled workers, who enhance productivity. Three mathematical models are developed to minimize the sum of total workstation costs, salaries of all workers, and cycle times and potential work overload of a predetermined number of workstations. The proposed models are based on particular features of the real-world problem, such as simultaneous assignments of skilled and temporary unskilled workers as well as precedent restrictions among the tasks. Furthermore, a hybrid genetic algorithm that minimizes total operation costs is developed. Special genetic operators and heuristic algorithms are used to ensure feasibility of solutions and make the hybrid genetic algorithm efficient. Computational experiments demonstrate the superiority of the hybrid genetic algorithm over the mathematical models.

Keywords: Hybrid genetic algorithm, Integer programming, Line balancing, Mixed integer linear programming, Mixed-model assembly line

Student Number: 2013-21062

Contents

Chapter 1. Introduction 1
1.1 The assembly line 1
1.1.1 Characteristics of assembly line problem 1
1.1.2 Assembly line balancing problem 2
1.2 The mixed-model assembly line 3
1.2.1 Characteristics of mixed model assembly line problem 3
1.2.2 Mixed model assembly line balancing problem 4
1.3 Literature review 5
1.4 Contributions 9

Chapter 2. Mathematical Models 11
2.1 General features of mathematical models 11
2.2 Problem description 11
2.3 Model Ⅰ 14
2.4 Model Ⅱ 18
2.5 Model Ⅲ 20

Chapter 3. A Hybrid Genetic Algorithm 23
3.1 Chromosome representation 24
3.2 Objective and fitness function 26
3.3 Genetic operator 27
3.3.1 Selection 27
3.3.2 Crossover 28
3.3.3 Mutation 29
3.4 Terminating conditions and parameters 30

Chapter 4. Computational Experiments 31

4.1 Experiments for Model Ⅰ 39

4.2 Experiments for Model Ⅱ 42

4.3 Experiments for Model Ⅲ 46

4.4 Validation of a hybrid genetic algorithm 48

Chapter 5. Conclusions 50

Bibliography 51

Abstract 54

List of Tables

Table 1 Parameters used in a hybrid genetic algorithm......................................30
Table 2 Available sets and salaries of skilled workers (9 and 14 tasks).............33
Table 3 Available sets and salaries of skilled workers (26 and 46 tasks)...........34
Table 4 Demand rates of each example..35
Table 5 Reducible time and salaries of temporary worker (9 tasks) 35
Table 6 Reducible time and salaries of temporary worker (14 tasks) 36
Table 7 Reducible time and salaries of temporary worker (26 tasks) 37
Table 8 Reducible time and salaries of temporary worker (46 tasks) 38
Table 9 Cycle time and operation cost of workstation 40
Table 10 Result of each example solved through Model Ⅰ 41
Table 11 Solution of Example 1 of Model Ⅰ 42
Table 12 Upplier limit ratio of cycle time and number of workstations 43
Table 13 Result of each example solved through Model Ⅱ 44
Table 14 Solution of Example 10 of Model Ⅱ 45
Table 15 Upper limit of number of workers and workstations.............................46
Table 16 Result of each example solved through Model Ⅲ 47
Table 17 Solution of Example 11 of Model Ⅲ 48
Table 18 Comparison of Model Ⅰ and the proposed hybrid genetic algorithm..49

List of Figures

Figure 1 Different versions of an assembly line 2
Figure 2 Description of a baton touch zone 12
Figure 3 Flowchart of the proposed hybrid genetic algorithm..........................24
Figure 4 Chromosome structure..24
Figure 5 Example of initial solution creation...25
Figure 6 Two-cut point crossover of a gene for task assignment.....................28
Figure 7 One-cut point crossover of a gene for temporary worker assignment.. 29
Figure 8 Mutation of a gene for temporary worker assignment.......................30
Figure 9 Tasks precedent sequence diagrams for 9 tasks and 2 products 31
Figure 10 Tasks precedent sequence diagrams for 14 tasks and 4 products 32
Figure 11 Tasks precedent sequence diagrams for 26 tasks and 3 products 32
Figure 12 Tasks precedent sequence diagrams for 46 tasks and 1 products 33

Chapter 1. Introduction

1.1 The assembly line

1.1.1 Characteristics of assembly line problem

An assembly line is a flow-oriented production system typical of industrial sites that produce large quantities of standardized commodities and is even important for producers of low volume diverse or customized products. Many different types of assembly line systems and related problems characterize various industries. Assembly line problems can be classified based on the number of items produced in the line, the nature of operation times (whether deterministic or probabilistic), and the nature of flow (a straight or U type assembly line).

If only one product is manufactured, all work pieces are identical and the system is called “a single model assembly line.” The names of other assembly lines depend on the types of intermixed units. The different assembly line types, described by different geometric shapes, are characterized in Figure 1. A *multi-model assembly line* produces a sequence of batches containing units of only one item or a group of similar items with intermediate setup operations, whereas a *mixed-model assembly line* produces different items in an intermixed sequence.

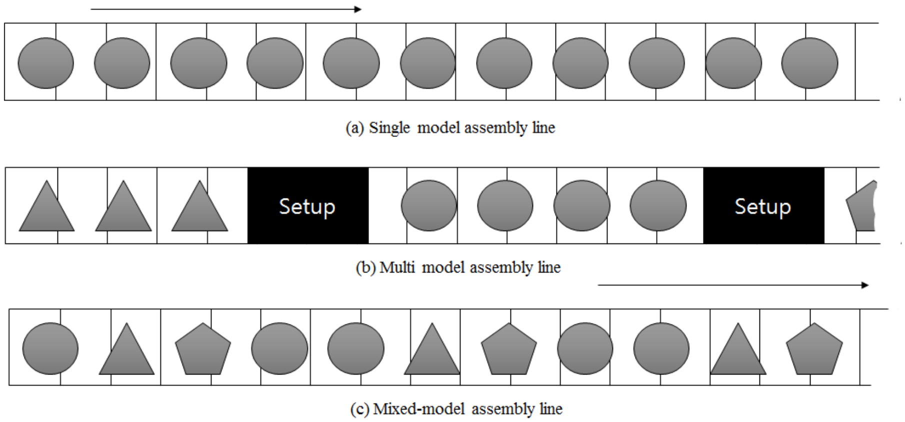

Figure 1 Different versions of an assembly line

Depending on assembly line types, single, mixed, and multi-model versions of assembly line problems should be applied and studied. Furthermore, as the level of competition increases, most manufacturing companies employ workers based on their experience or skill set while also trying to reduce production costs. Hence, different versions of assembly line problems have been widely researched during the latest period of industrial development.

1.1.2 The assembly line balancing problem

The assembly line balancing problem (ALBP) is used to search for the optimal assignment of assembly tasks to stations under precedent and additional practical constraints. Among the decisions that arise in managing modern industrial systems, those involving balancing an assembly line are key for implementing a cost-efficient production system because the installation of assembly line systems, which are highly automated, requires considerable investment. ALBPs are particularly important in medium-term production planning. ALBPs are typically used to minimize the number of workstations for a given cycle time and the length of operation times for a specific number of workstations.

Among the family of ALBPs, the best known involves the simple assembly line balancing problem (SALBP). Assembly line balancing research has traditionally focused on the SALBP that features some restricting assumptions. Although it may offer a limited reflection of complex real-world line balancing, the SALBP, nevertheless, captures the main aspects and is rightfully regarded as constituting the core problem of an ALB. In fact, varieties of more general problems are direct extensions of the SALB or at least require the solution of SALBPs in some form. Therefore, the SALBP is well suited to explain the basic principles of the ALBP, including the relevant terms.

1.2 The mixed-model assembly line

1.2.1 Characteristics of mixed model assembly line problem

Needs of customers become diverse and competitions among manufacturers tremendously increase. For example, German car manufacturer BMW offers a catalogue of optional features which results in 10^{32} different models to satisfy their customer (Meyr 2004). It illustrates that an efficient methodology for manufacturing multiple products is essential to survive in these days of limitless competitions. The installation of a mixed-model assembly line is one of the best responses to this challenging situation, and this methodology has already been widely adopted in industry because through it manufacturers can meet the diversified demands of their customers without maintaining large inventories or a string of assembly lines.

The mixed-model assembly line originated with Toyota, who developed the concept of mixed model production in the 1960s in response to problems created by line changeovers. The objective of the mixed model is to smooth demand on upstream work centers or suppliers and thereby reduce inventory. The system also eliminates difficult assembly line changeovers, and it can shorten setup times.

Mixed-model production systems produce several types of a standardized commodity in an intermixed sequence. The models may differ from each other with respect to size, color, used material, or equipment such that their production requires different tasks, operation times, and precedent relations. As a consequence, perfect line balancing is almost impossible. Therefore, the line must be flexible with respect to the equipment as well as local cycle time violations. Unlike the single-model assembly line, which deals with deterministic operation times, in the mixed model, the cycle time is not defined by the maximum operation time needed at each station to perform tasks but the average time necessary to produce all the different models.

1.2.2 Mixed-model assembly line balancing problem

The principle of the mixed-model assembly line is quite simple. However, designing the process and system is quite difficult. Mixed-model assembly lines have two types of problems: line balancing and the sequencing of different models on the line. These two elements of the process collectively affect the performance of assembly lines. This study deals with the mixed-model ALBP.

One important assumption in the mixed-model ALBP is that the tasks common to several models should be performed at the same workstation. If this assumption is relaxed, the mixed-model ALBP is decomposed into N independent single-model ALBPs.

For modeling and solving a mixed ALBP, a researcher typically transforms it to a single-model ALBP with additional assumptions. Therefore, many studies are based on the assumption that a single diagram can be drawn to represent the combined precedent relationships among the different tasks associated with each model. This methodology has been widely used for decades because of its convenience and powerfulness. However, it may lead to considerable inefficiencies in line operations, even when an optimal solution is used as a result of applying this method.

1.3 Literature review

Bryton (1954) introduced the idea of line balancing in his graduate thesis. The first paper on the ALBP was published in 1955. Salveson (1955) provided the first mathematical attempt to solve it using a linear programming approach. Over the subsequent 60 years, this subject has been broadly studied.

A comprehensive review of single-model ALBPs and related solution procedures was provided by Scholl and Becker (2006), who classified them by the constraints and different objectives that characterized them. Recently, Sivasankaran and Shahabudeen (2014) presented a review of literature on assembly line balancing. They classified the ALBP into eight types based on three parameters: the number of model types (single or multiple), the nature of task times (deterministic or probabilistic), and the type of assembly line (straight or U-type).

Corominas et al. (2008) provided important research that informs this study. They considered the process of rebalancing the line at a Spanish motorcycle assembly plant. They suggested a binary linear programming solution to minimize the number of temporary workers required under a specified cycle time and different task groups. Moon et al. (2009) designed the integrated ALBP with resource restrictions. They considered multi-skilled workers as resources and formulated the problem using integer linear programming (ILP). A genetic algorithm (GA) for addressing the problem of resource restrictions related to integrated assembly line balancing was also developed. Kara et al. (2014) incorporated ergonomics and resource restrictions into assembly line balancing. They proposed a cost-oriented integrated solution by considering the psychological and physical strain of employees. They looked at multiple workers and their skills, equipment, working postures, and illumination level restrictions. In addition, the proposed model modified the existing constraints by means of real-life facts and introduced two new constraints indicated from the industry that had not been addressed in the literature.

Due to the complexity of the problem, in many cases, heuristic procedures are more promising than optimum-seeking algorithms. Therefore, numerous heuristic algorithms have been developed for ALBPs. Kim et al. (1996) developed GAs to solve ALBPs with five different objectives. Erel and Sarin (1998) examined and evaluated heuristic procedures critically and summarized them in sufficient detail to provide a state-of-the-art survey. Lapierre at al. (2006) presented a new tabu search algorithm and evaluated its performance. They tested the algorithm using a real industrial data set involving 162 tasks and 264 precedent constraints. No computational experiments have been performed on real industrial applications in most of the other research conducted.

The mixed-model assembly line has received significant interest in recent years and a number of related studies are currently underway. Thomopoulos (1967) was the first to model the mixed model ALBP by assuming that the variants of each task over all models are restricted to a single station. Under the application of this restriction, the balancing procedure is similar to that used for solving a single-model ALBP. Specifically, to avoid excessive capacity, the average cycle time of all models is used. As a result, the processing time of some models is longer than the cycle time. In other words, work overload may occur. Despite the difficulties with overload, Thomopoulos's paper triggered further study of a mixed-model assembly line problem.

Merengo et al. (1999) analyzed some typical problems of manual mixed-model assembly lines and introduced a complementary version of simple assembly line balancing for minimizing the rate of incomplete jobs with a secondary objective of smoothing workstation loads. Xiaobo et al. (1999) discussed Toyota's goal of sequencing mixed models on an assembly line with multiple workstations. The problem was formulated based on defining the ideal usage rate of a part as the requirement for the part- per-time period. A modified goal-chasing algorithm was proposed for solving this sequencing problem. Bukchin et al. (2002) designed a mixed-model assembly line in a make-to-order environment. Special characteristics of such mixed-model assembly lines in a make-to-order environment include relatively few workstations, a lack of mechanical conveyance, and employment of highly multi-skilled workers. Bukchin et al. presented a mathematical model by considering the different characteristics from traditional models.

Fattahi and Salehi (2009) considered the mixed-model ALBP to minimize the total utility and idle costs with variable launching intervals for sequencing the models. They also determined the sequence of products in the assembly line with variable launching intervals between the products. They developed a mathematical model as well as a simulated annealing algorithm for this problem. Rabbani et al. (2012) developed a mixed integer programming method in which two conflicting objectives, including minimizing the cycle time and number of workstations, were considered simultaneously. They also proposed a heuristic algorithm based on GAs. Tonelli et al. (2013) considered model flexibility in the planning problem of a real world assembly manufacturing system; model flexibility is an especially important concept in mixed-model assembly lines. They implemented an advanced planning system integrated with a mixed integer programming model, which is solved by a new iterative heuristic approach in which some important issues related to technological, organizational, and managerial constraints are considered. Akpinar and Baykasoglu (2014) studied a mixed-model ALBP with setups, and they developed a Bees algorithm, which is a relatively new methodology of swarm intelligence based on meta-heuristics.

1.4 Contributions

The ALBP has been a topic of research for several decades. The main contribution of this study is the development of a mathematical formulation and a hybrid genetic algorithm (HGA) for a mixed-model ALBP with a cost-oriented objective to reduce inefficiency in a manufacturing process by considering employment of temporary utility workers. A thorough review of the latest publications did not indicate that a mathematical model has been used to solve a mixed-model ALBP that accounts for the assignment of unskilled temporary workers, which makes up a key component of this study, to obtain balances that minimize potential work overloads. When workers of a station are not able to complete the assigned tasks before the work pieces leave the station, work overload occurs, causing inefficiency that should be minimized. Work overload can be ameliorated by the employment of temporary workers, or assembly the line should be stopped and generate opportunity costs.

This study deals with the problem of designing an integrated mixed-model assembly line with the simultaneous assignment of skilled and temporary, unskilled workers. Although Corominas et al. (2008) considered use of temporary workers, they did not account for integrated assembly line balancing. Their objective was to minimize the number of unskilled workers. Moon et al. (2009) studied an integrated ALBP, but they did not consider reducible task times through cooperation of permanent and temporary workers.

In Chapter 2, three mathematical models for the integrated mixed-model assembly line balancing situation with temporary workers are introduced. Each model has a different objective function; the minimize, respectively,

(1) total relevant cost,

(2) cycle time, and

(3) work overload.

In Section 3, an HGA is proposed to solve the realistic large problems. In Section 4, computational experiments are conducted for the developed mathematical model and the proposed algorithm models. Chapter 5 provides conclusions of this research.

Chapter 2. Mathematical Models

2.1 General features of mathematical models

If all decision variables must be integers, then the function is called an "integer programming problem." If only some of decision variables are required to be integers, then it is called a "mixed integer programming problem." All of the constraints and objective functions of mathematical models in this study are linear. All decision variables in Model Ⅰ and most decision variables in Models Ⅱ and Ⅲ are integers. Therefore, the mathematical models are developed by ILP and mixed ILP (MILP).

The single-model ALBP is known as an NP-hard problem (Scholl 1999). This study solved extended versions of single-model ALBPs. If there is only one type of product and no temporary workers, problems in this study can be transformed to single ALBPs. Therefore, the studied problems also become NP-Hard.

2.2 Problem description

This study tries to balance a mixed-model assembly line with temporary workers from an integrated point of view. Temporary workers can only perform a subset of all tasks required to manufacture an item, so they cannot be assigned alone to any task. However, they can reduce operation times of completing tasks by cooperating with skilled workers. As a result of this teamwork, the number of workstations needed for performing all tasks can be decreased. However, there is a trade-off between the salaries of temporary workers and the operation costs of workstations.

In this problem, the assembly lines have baton touch zones between workstations. A baton touch zone is a broad space between preceding and subsequent processes, as shown in Figure 2. It is widely used in manufacturing to support the sustainability of a system and compensate for variations in the production process. Variations always exist because an operation time for the same task can be different based on product type. In fact, total operation time for all tasks related to certain product types may exceed a cycle time. However, because of the baton touch zones, tasks of each workstation should be finished earlier than the time allowed. Hence, this model considers an upper limit in the total operation time of each workstation.

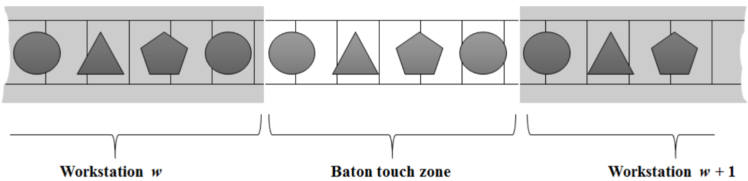

Figure 2 Description of a baton touch zone

Available skill sets for each worker can differ. Hence, assigning both tasks and workers should be optimized simultaneously. Salaries of workers and operation costs of workstation can be reduced under an efficiently designed workstation. The goal of line balancing is to minimize cost, and it is equivalent to minimizing the sum of the workstation costs and the salaries of skilled and temporary workers. This goal is the objective function of the mathematical model. To solve the industrial problem used in this study, the following assumptions were made:

(1) A skilled worker cannot be assigned to more than one workstation.

(2) Skilled workers should be assigned to tasks depending on their available task sets.

(3) Temporary workers cannot be assigned alone to any task.

(4) The operation time of tasks can be reduced by assigning a temporary worker to a task.

(5) At most, one temporary worker can be assigned to each task to reduce operation time.

(6) Operation and reducible times for production are fixed and known.

(7) Demands for all models during the planning period are fixed and known.

(8) The tasks common to several models are performed at the same workstation.

(9) Precedent constraints determine the sequence in which the tasks can be processed.

(10) There is a limit on the number of workers at each workstation.

Using these assumptions, this study presents three mathematical models for an integrated mixed-model ALBP with temporary workers.

2.3 Model Ⅰ – Minimization of total cost

Model Ⅰ minimizes the total costs, including those of operating workstations and the salaries of skilled and temporary workers, for a given cycle time. The following notation is used to explain Model Ⅰ:

Indices:

i, j	tasks (i, j = 1, 2, …, I)
k	product type (k = 1, 2, …, K)
s	workstations (s = 1, 2, …, S)
w	skilled workers (w =1,2, …, W)

Parameters:

C	cycle time
C^u	upper limit of total operation time of each workstation
K_i	number of different products that requires task i
D_k	demand of product k
o_{ik}	operation time for task i for product k when performed by a skilled worker
r_{ik}	reducible time for task i for product k when performed by a skilled worker and a temporary worker together
$P_{(i,j,k)}$	set of task pairs i, j for product k such that there is an immediate precedent relation between them
T_w	set of available tasks that can be assigned to skilled workers w
OC	operation costs of a workstation
SS_w	salary for skilled worker w
SU	salary for a temporary worker
n	upper bound of number of workers that can be assigned to a workstation
M	a sufficiently large number

The following two additional parameters are calculated with given parameters, and Equations (1) and (2) are used for calculation.

o_i	aggregated operation time for task i under demand rates when performed by a skilled worker
r_i	aggregated reducible time for task i under demand rates when performed by a skilled worker and a temporary worker together

$$o_i = \frac{\sum_{k=1}^{K} D_k \cdot o_{ik}}{\sum_{k=1}^{K} D_k} \tag{1}$$

$$r_i = \frac{\sum_{k=1}^{K} D_k \cdot r_{ik}}{\sum_{k=1}^{K} D_k} \tag{2}$$

Decision Variables:

F	Number of workstations to be used in the assembly line
X_{iksw}	$\begin{cases} 1, \text{ if task } i \text{ for product } k \text{ is performed by skilled worker } w \text{ at workstation } s \\ 0, \text{otherwise} \end{cases}$
Y_{sw}	$\begin{cases} 1, \text{ if skilled worker } w \text{ is assigned to workstation } s \\ 0, \text{otherwise} \end{cases}$
Z_{is}	$\begin{cases} 1, \text{ if a temporary worker is assigned to task } i \text{ at workstation s} \\ 0, \text{otherwise} \end{cases}$
A_{is}	$\begin{cases} 1, \text{ if task } i \text{ assigned to workstation s} \\ 0, \text{otherwise} \end{cases}$

Objective function

$$Min \quad OC \cdot F + \sum_{w=1}^{W} SS_w \left(\sum_{s=1}^{S} Y_{sw} \right) + SU \cdot \sum_{i=1}^{I} \sum_{s=1}^{S} Z_{is} \tag{3}$$

Subject to

$$\sum_{k=1}^{K}\sum_{s=1}^{S}\sum_{w=1}^{W} X_{iksw} = K_i \quad \forall i \tag{4}$$

$$\sum_{k=1}^{K}\sum_{w=1}^{W} X_{iksw} = K_i \cdot A_{is} \quad \forall i, s \tag{5}$$

$$\sum_{s=1}^{S}\sum_{w=1}^{W} X_{iksw} \le M \cdot o_{ik} \quad \forall i, k \tag{6}$$

$$\sum_{i=1, i \ne T_w}^{I}\sum_{k=1}^{K}\sum_{s=1}^{S} X_{iksw} = 0 \qquad \forall w \tag{7}$$

$$\sum_{s=1}^{S}\sum_{w=1}^{W} (s \cdot X_{iksw} - s \cdot X_{jksw}) \le 0 \qquad \forall (i, j, k) \in P_{(i,j,k)} \tag{8}$$

$$Z_{is} \le \sum_{k=1}^{K}\sum_{w=1}^{W} X_{iksw} \qquad \forall i, s \tag{9}$$

$$\sum_{i=1}^{I}\left(\sum_{k=1}^{K}\sum_{w=1}^{W} o_i \cdot X_{iksw} - r_i \cdot Z_{is}\right) \le K \cdot C \quad \forall s \tag{10}$$

$$\sum_{i=1}^{I}\left(\sum_{w=1}^{W} o_{ik} \cdot X_{iksw} - r_{ik} \cdot Z_{is}\right) \le C^u \quad \forall k, s \tag{11}$$

$$\sum_{s=1}^{S} Y_{sw} \le 1 \qquad \forall w \tag{12}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K} X_{iksw} \le M \cdot Y_{sw} \quad \forall s, w \tag{13}$$

$$\sum_{w=1}^{W} Y_{sw} + \sum_{i=1}^{I} Z_{is} \le n \quad \forall s \tag{14}$$

$$\sum_{w=1}^{W} X_{iksw} \leq 1 \qquad \forall i,k,s \tag{15}$$

$$\sum_{s=1}^{S} s \cdot X_{iksw} \leq F \qquad \forall i,k,w \tag{16}$$

$$X_{iksw},\ \ Y_{sw},\ \ Z_{is},\ \ A_{is} \in \{0, 1\} \tag{17}$$

The objective function (3) minimizes the sum of the total workstation costs and the salaries of skilled and temporary workers. Constraints (4) ensure that every task should be performed by one skilled worker at one workstation. Constraint (5) indicates that the same task for different products should be assigned to the same workstation. Constraints (6) prevent inconsistency of X_{iksw}. X_{iksw} can be greater than zero when o_{ik} is not zero. Zero operation times mean that the task is not required for the model. Therefore, X_{iksw} can be 1 if task i is necessary to manufacture product k. Constraints (7) prevent skilled worker w from being assigned to a workstation with a task that the worker cannot complete according to the skilled worker's available task set T_w. Constraints (8) ensure that the precedent relationships between tasks are considered for product k. Constraints (9) ensure that a temporary worker can be assigned to task i only when a skilled worker is also assigned to task i. Constraints (10) represent the total operation time under conditions of a demand rate of each workstation that is smaller than a specific time. Constraints (11) represent the total maximum operation time to manufacture product k at each workstation as smaller than the upper limit given to the time. Constraints (12) guarantee that a skilled worker is assigned to exactly one workstation. Constraints (13) suggest that a task is assigned to a skilled worker at the workstation; X_{iksw} can be 1 when a skilled worker is assigned to a workstation. Constraints (14) restrict the number of total workers to one workstation to prevent overcrowding. Constraints (15) ensure that the same task for different product types is assigned to the same worker. Constraints (16) are used to decide the total number of workstations needed. Constraints (17) demonstrate the binary nature of decision variables.

2.4 Model Ⅱ – Minimization of a cycle time

Model Ⅱ is similar to Model Ⅰ. It is used to minimize a cycle time for a specific number of workstations. In this model, F is not a decision variable. Instead, the cycle times, C, of workstation s are decision variables. The ratio of total operation time to cycle time at each workstation U is considered in this model. Some constraints are also changed in Model Ⅱ. Constraint (10) in Model Ⅰ is replaced by Constraint (10-2) in Model Ⅱ. Likewise Constraint (11) is replaced with (11-2). The following notation is used to explain Model Ⅱ:

Indices:

i , j	tasks (i , j = 1, 2, …, I)
k	product type (k = 1, 2, …, K)
s	workstations (s = 1, 2, …, S)
w	skilled workers (w =1,2, …, W)

Parameters:

F	Number of workstations used in the assembly line
K_i	number of different products that requires task i

D_k	demand of product k
o_{ik}	operation time for task i for product k when performed by a skilled worker
r_{ik}	reducible time for task i for product k when performed by a skilled worker and a temporary worker together
$P_{(i,j,k)}$	set of task pairs i, j for product k such that there is an immediate precedent relation between them
T_w	set of available tasks that can be assigned to skilled workers w
OC	operation costs of a workstation
SS_w	salary for skilled worker w
SU	salary for a temporary worker
U	Ratio of total operation time to cycle time of each workstation
n	upper bound of number of workers that can be assigned to a workstation
M	a sufficiently large number

The following two additional parameters are calculated with given parameters as in Model Ⅰ.

o_i	aggregated operation time for task i under demand rates when performed by a skilled worker
r_i	aggregated reducible time for task i under demand rates when performed by a skilled worker and a temporary worker together

Decision Variables:

C_s Cycle time of workstation s

$$X_{iksw} \quad \begin{cases} 1, \text{ if task } i \text{ for product } k \text{ is performed by skilled worker } w \text{ at workstation } s \\ 0, \text{otherwise} \end{cases}$$

$$Y_{sw} \quad \begin{cases} 1, \text{ if skilled worker } w \text{ is assigned to workstation } s \\ 0, \text{otherwise} \end{cases}$$

$$Z_{is} \quad \begin{cases} 1, \text{ if a temporary worker is assigned to task } i \text{ at workstation s} \\ 0, \text{otherwise} \end{cases}$$

$$A_{is} \quad \begin{cases} 1, \text{ if task } i \text{ assigned to workstation s} \\ 0, \text{otherwise} \end{cases}$$

Objective function

$$Min \ \max_{s} C_s \tag{3-2}$$

Modified constraints from Model Ⅰ:

$$\sum_{i=1}^{I}\left(\sum_{k=1}^{K}\sum_{w=1}^{W} o_i \cdot X_{iksw} - r_i \cdot Z_{is}\right) \le K \cdot C_s \quad \forall s \tag{10-2}$$

$$\sum_{i=1}^{I}\left(\sum_{w=1}^{W} o_{ik} \cdot X_{iksw} - r_{ik} \cdot Z_{is}\right) \le U \cdot C_s \quad \forall k, s \tag{11-2}$$

2.5 Model Ⅲ – Minimization of work overload

Model Ⅲ is similar to Model Ⅱ. It is used to minimize the work overload at a predetermined number of workstations. The cycle times of workstations for each product should be defined as decision variables. The ratio of total operation time to cycle time at each workstation U is not used in Model III. Furthermore, Constraint (10-2) used in Model Ⅱ is replaced by Constraint (10-3) in Model Ⅲ. Likewise, Constraint (11-2) is replaced by (11-3). The following notation is used to explain Model Ⅲ:

Indices:

i, j	tasks (i, j = 1, 2, …, I)
k	product type(k = 1, 2, …, K)
s	workstations(s = 1, 2, …, S)
w	skilled workers(w =1,2, …, W)

Parameters:

F	Number of workstations used in the assembly line
K_i	number of different products which requires task i
D_k	demand of product k
o_{ik}	operation time for task i for product k when performed by a skilled worker
r_{ik}	reducible time for task i for product k when performed by a skilled worker and a temporary worker together
$P_{(i,j,k)}$	set of task pairs i, j for product k such that there is an immediate precedence relation between them
T_w	set of available tasks that can be assigned to skilled workers w
OC	operation costs of a workstation
SS_w	salary for skilled worker w
SU	salary for a temporary worker
n	upper bound of number of workers that can be assigned to a workstation
M	a sufficiently large number

The following two additional parameters are calculated and utilized along with the parameters described in Model Ⅰ:

o_i	aggregated operation time for task i considering demand rates when performed by a skilled worker
r_i	aggregated reducible time for task i considering demand rates when performed by a skilled worker and a temporary worker together

Decision Variables:

C_s	Cycle time of workstation s
C_{ks}	Cycle time of workstation s for product k
X_{iksw}	$\begin{cases} 1, & \text{if task } i \text{ for product } k \text{ is performed by skilled worker } w \text{ at workstation } s \\ 0, & \text{otherwise} \end{cases}$
Y_{sw}	$\begin{cases} 1, & \text{if skilled worker } w \text{ is assigned to workstation } s \\ 0, & \text{otherwise} \end{cases}$

Z_{is} $\begin{cases} 1, \text{ if a temporary worker is assigned to task } i \text{ at workstation s} \\ 0, \text{otherwise} \end{cases}$

A_{is} $\begin{cases} 1, \text{ if task } i \text{ assigned to workstation s} \\ 0, \text{otherwise} \end{cases}$

Objective function

$$Min \sum_{k=1}^{K}\sum_{s=1}^{S}\left(C_{ks} - C_s\right)^{+} \tag{3-3}$$

Modified constraints from Model Ⅱ:

$$\sum_{i=1}^{I}\left(\sum_{k=1}^{K}\sum_{w=1}^{W} o_i \cdot X_{iksw} - r_i \cdot Z_{is}\right) \le K \cdot C_s \quad \forall s \tag{10-3}$$

$$\sum_{i=1}^{I}\left(\sum_{w=1}^{W} o_{ik} \cdot X_{iksw} - r_{ik} \cdot Z_{is}\right) \le C_{ks} \quad \forall k, s \tag{11-3}$$

Chapter 3. A Hybrid Genetic Algorithm

A GA is a population-based search and optimization method inspired by Darwin's theory of evolution. The two main concepts of evolution, natural selection and genetic dynamics, play a key role in this method. A GA and basic principles were first introduced by Holland (1975) and have been widely used in various areas during the last four decades. One of the advantages of the GA is its usability with other techniques to improve search performance. An algorithm incorporating a search method within a GA is called a "hybrid genetic algorithm" (HGA). In this study, an HGA for Model Ⅰ is proposed to solve large and real-world problems. The suggested HGA structure is shown in Figure 3.

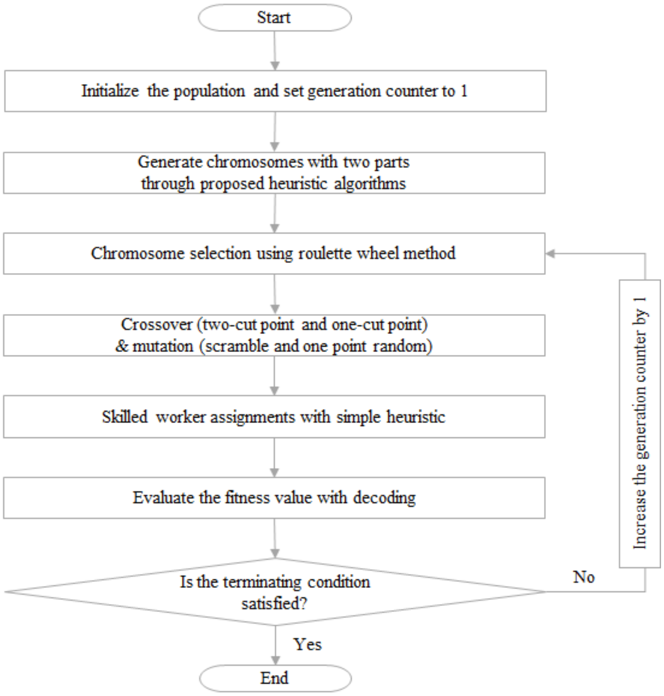

Figure 3 Flowchart of the proposed hybrid genetic algorithm

3.1 Chromosome representation

The proper representation of a solution is most important in the development of a GA. Traditionally, chromosomes are represented as simple binary strings; however, this representation is not well suited for solving complex problems such as those associated with mixed-model assembly line balancing. Therefore, this study features a chromosome that consists of two parts: one that represent task precedents and another that stands for temporary worker assignments. See Figure 4.

Task	2	1	4	3	5	7	6	8	9
Temporary worker	0	1	1	0	1	0	1	1	0

Figure 4 Chromosome structure

In this study, one part in a chromosome offers solutions for assigning tasks to workstations while the other part which has a binary string offers solutions for assignments of temporary workers through a heuristic procedure. The efficiency of the algorithm greatly depends on the initial solution. Therefore, a simple heuristic is proposed to create a feasible initial solution. This heuristic consists of a sorting procedure based on task precedent restrictions. An example of initialization through use of a heuristic algorithm is shown in Figure 5.

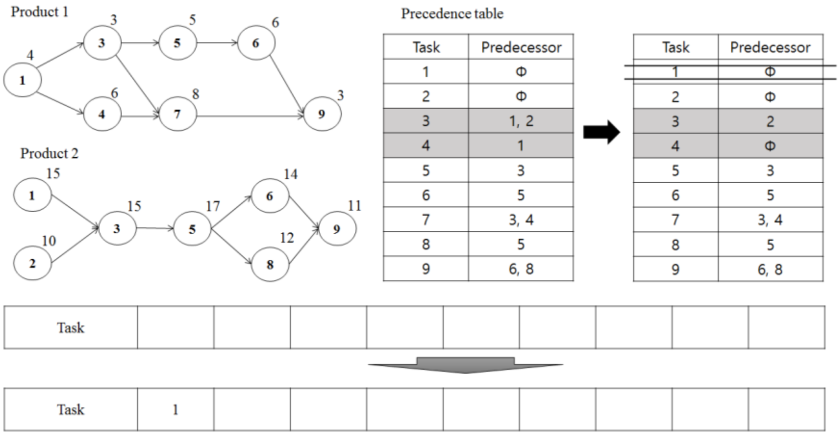

Figure 5 Example of initial solution creation

The steps for the heuristic are as follows:

Step 1. Find all tasks with predecessor Φ and assign one of them which is used the most in the first gene that has no value yet assigned to it.

Step 2. Update the table by deleting the selected task in the predecessor table.

Step 3. Go back to Step 1 until all tasks are assigned.

After finishing the task assignment, temporary workers are assigned in another part of chromosome. For the temporary worker assignment, another simple heuristic is proposed based on tradeoffs between operation costs of a workstation and the salary of a temporary worker. The steps for the heuristic are as follows:

Step 1. Calculate a critical value *cv*, $cv = \frac{SU}{OC} \cdot C$.

Step 2. Assign temporary workers to all tasks with reducible time r_i larger than *cv*.

3.2 Objective and fitness function

The fitness function plays a role similar to that of the environment in evolution. Each individual in the population represents a potential solution to the problem. A fitness function is computed for each string in the population and the string with the minimum fitness function value is selected. Equation (1) is used as a fitness function in the proposed GA. To calculate fitness value, information about assignment of skilled workers is needed. Hence, a simple heuristic that sorts on the basis of skilled workers' available task sets is proposed to assign skilled workers to each workstation. The steps for the heuristic are as follows:

Step 1. Calculate the cumulative operation time of assigned tasks in the chromosome and divide workstations using the predetermined cycle time and its upper limit.

Step 2. Arrange skilled worker candidates for each workstation.

Step 3. Select the workstation with the fewest candidates. If a tie occurs, select a workstation arbitrarily.

Step 4. Select the worker with the lowest salary.

Step 5. If the index of skilled worker candidates is empty for any workstation, go back to Step 4 and select the worker with the next lowest salary.

Step 6. Finish the algorithm until skilled workers are assigned to all workstations.

3.3 Genetic operator

Genetic operators are used to produce the next generation of chromosomes. The method of the genetic operators—selection, crossover, and mutation—can change the next population to encourage genetic diversity.

3.3.1 Selection

Selection is the stage in which individual chromosomes are chosen from a population. In this algorithm, parents are chosen by a roulette wheel method. In this approach, the fitness of each chromosome is calculated and is used to associate a probability of selection with each individual chromosome. Usually a roulette wheel is divided into proportions such that when it is rotated a ball lands in one of the possible partitions. Likewise, with this method, a solution is selected from a range of possibilities such that there is a chance that some weaker solutions may survive through the selection process. This inclusion of the weaker solutions proves advantageous as it may include some components useful in the recombination process.

3.3.2 Crossover

The crossover operation is a diversification mechanism that enables the GA to generate a solution in previously unvisited areas of the gene pool. In this algorithm, a two-cut crossover operator is used for the task assignment and the one-cut point crossover operator is used for the temporary worker assignment. A direct swap method is impossible for part of the task assignment because of precedent constraints between tasks. Hence, a special two-cut crossover method is applied as shown in Figure 6. This crossover method is proposed by Leu et al. (1994) to guarantee that the resulting offspring is always feasible.

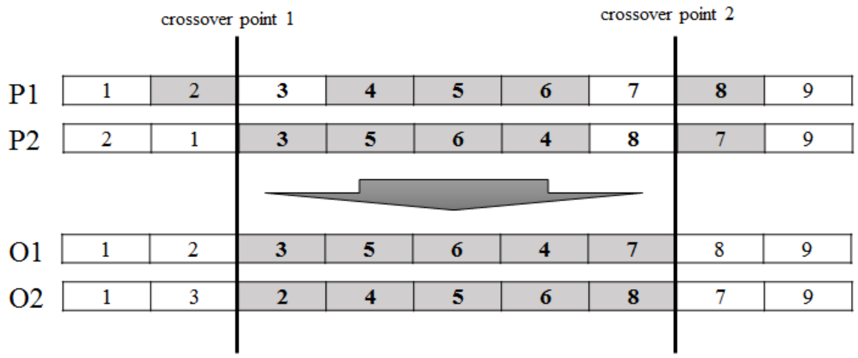

Figure 6 Two-cut point crossover of a gene for task assignment

For the temporary worker assignment, the position of the cut point is randomly generated from the range [1, (length of the chromosome – 1)]. We generated the offspring simply by exchanging the parts of their parents as shown in Figure 7.

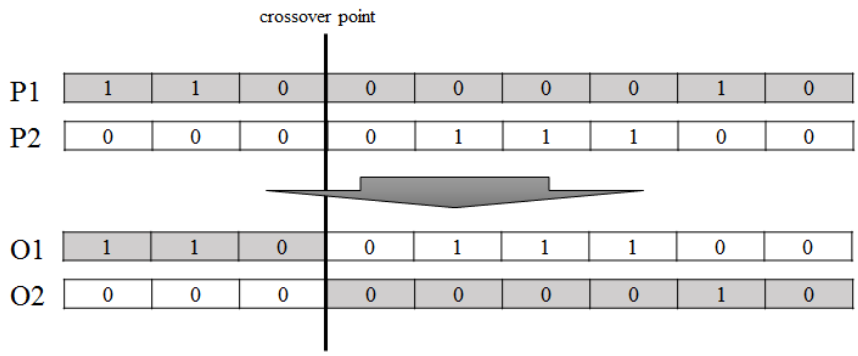

Figure 7 One-cut point crossover of a gene for temporary worker assignment

3.3.3 Mutation

In a GA, mutation produces a random change in a chromosome. The main difference between mutation and crossover is that the mutation operators affect one chromosome, that is, they are unary, while crossovers are binary operators. Mutation plays the crucial role of replacing genes of chromosomes during evolution so that they can maintain diversity in the population. In the algorithm employed in this study, a *scramble mutation* operator is used for the task assignment and a random point mutation operator is used for the temporary worker assignment. Leu et al. (1994) proposed the scramble mutation to move the search out of a local neighborhood and avoid the possibility of finding a local optimum in a manner that ensures feasibility. The random point mutation involves changing the value on a randomly selected gene to another value from the range [0, length of the chromosome]. Figure 8 shows a random point mutation for the temporary worker assignment.

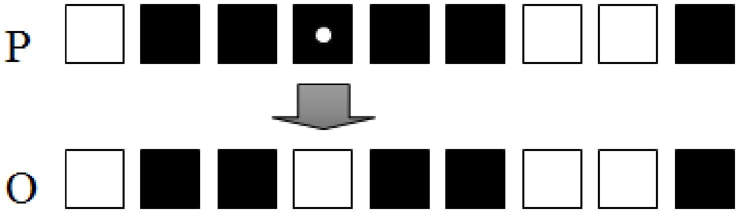

Figure 8 Mutation of a gene for temporary worker assignment

3.4 Terminating conditions and parameters

The terminating condition was determined when 100,000 generations were produced or when the fitness value of the best chromosome was not improved over 3,000 generations. For the proposed GA, the crossover and mutation operators were employed with the parameters listed in Table 1. A pilot test was conducted to find appropriate parameter values.

Table 1 Parameters used in a hybrid genetic algorithm

Parameters	Value
Population size	150
Two-cut crossover probability	0.6
One-cut point crossover probability	0.6
Scramble mutation probability	0.7
One point mutation probability	0.7

Chapter 4. Computational Experiments

The mathematical models were solved with FICO Xpress-IVE version 7.3 with the time limited to 259,200 seconds. HGAs were run in C++ language. Numerical experiments were conducted with an Intel(R) Xeon(R) 3.5GHz processor with 16GB RAM on the Microsoft Windows Server 2008 R2 operating system. Although the specifications of the computer used in the experiments were good, these models were not able to solve problems as large as those with 26 tasks, 3 product types, and 14 workers within the time limit. However, HGAs can find solutions in a reasonable computation time even for large size problems. Task precedent sequence diagrams of 9, 14, 26, and 46 tasks are shown in Figures 9, 10, 11, and 12. Salaries of skilled workers and their available skill sets are shown in Tables 2 and 3. Demand rates of each example are arranged in Table 3. Reducible times for tasks of each product type are organized in Tables 4, 5, 6, 7, and 8.

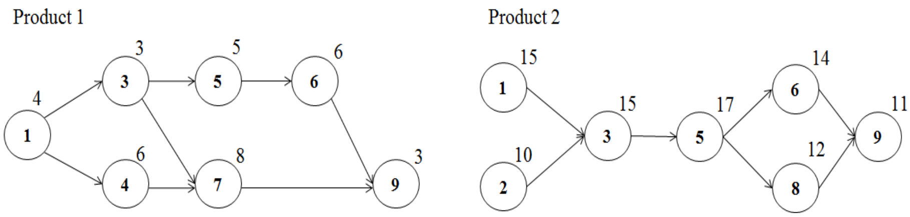

Figure 9 Tasks precedent sequence diagrams for 9 tasks and 2 products

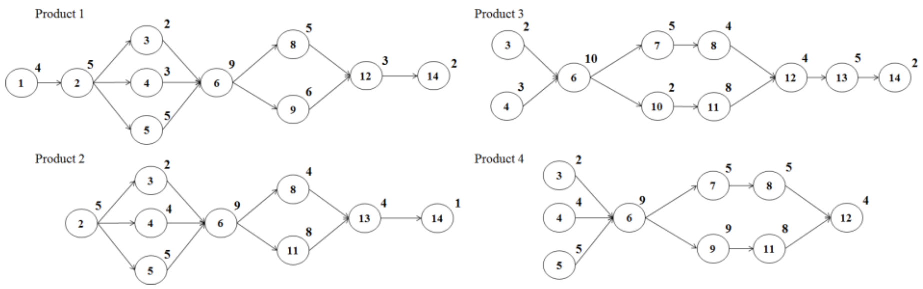

Figure 10 Tasks precedent sequence diagrams for 14 tasks and 4 products

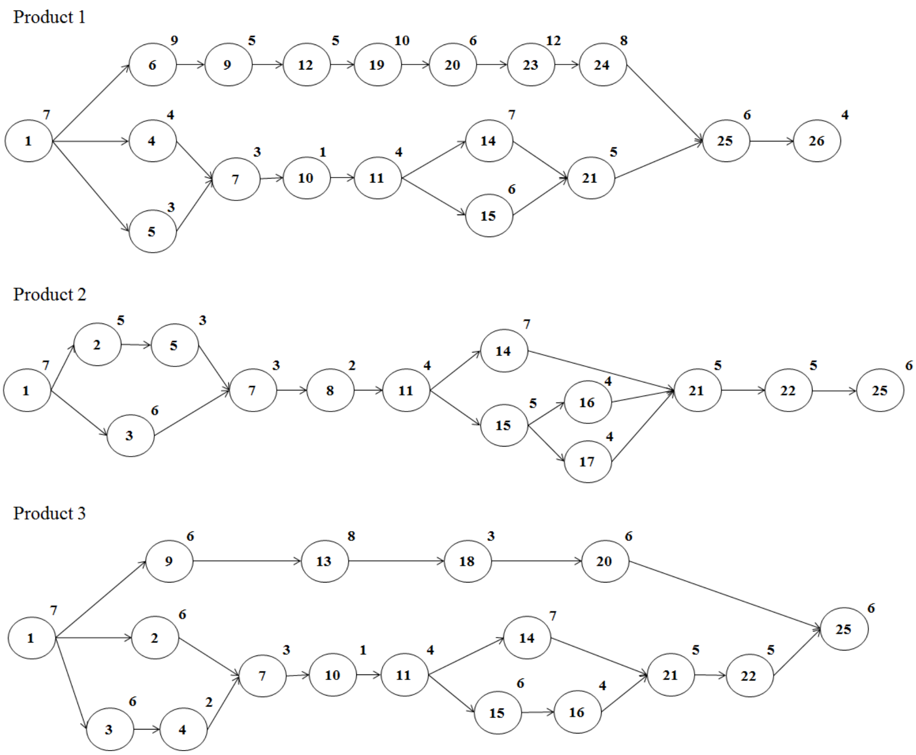

Figure 11 Tasks precedent sequence diagrams for 26 tasks and 3 products

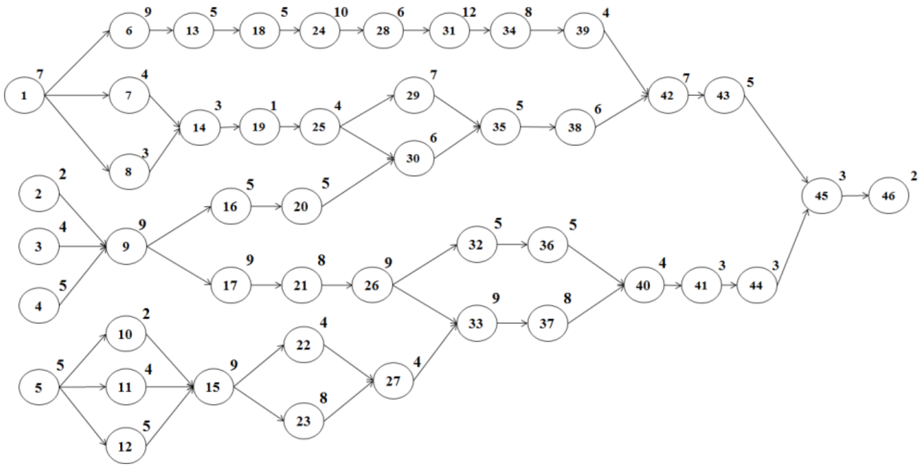

Figure 12 Tasks precedent sequence diagrams for 46 tasks and 1 product

Table 2 Available sets and salaries of skilled workers (9 and 14 tasks)

Tasks	Worker	Possible Tasks	Salary
9	1	3 6 8	$3,800
	2	4 5 6 9	$4,000
	3	1 5 7 9	$3,700
	4	2 3 6 7 8	$4,000
	5	1 5 8	$3,000
	6	1 2 7	$3,000
14	1	1 3 5 7 8 10 13	$5,000
	2	1 2 3 5 7 10 13	$4,800
	3	2 3 4 5 8 9 10	$4,700
	4	1 4 6 7 8	$3,800
	5	1 3 7 10 12 14	$4,500
	6	2 3 8 9 10	$4,300
	7	1 5 8 9 10 14	$4,400
	8	1 3 4 6 8 9 10 11	$5,400
	9	6 7 8 11 12 13 14	$5,000
	10	5 8 9 11 12 13 14	$4,900

Table 3 Available sets and salaries of skilled workers (26 and 46 tasks)

Tasks	Worker	Possible Tasks	Salary
26	1	1 3 6 8 10 14 18 19 20 25	$5,700
	2	6 8 19 20 25	$3,800
	3	1 8 10 13 18 19 25	$4,000
	4	1 7 8 10 11 14 19 25	$4,400
	5	2 4 8 18 19 25	$39,00
	6	1 3 5 8 10 11 15 18 19 23 25	$6,000
	7	2 3 5 8 10 11 13 14 15 16 25	$61,00
	8	1 5 8 9 12 14 15 21 25	$4,300
	9	1 3 5 6 7 8 9 10 12 15 17 25	$6,300
	10	1 3 5 6 7 8 9 10 12 15 17 25	$6,300
	11	20 21 22 23 24 25 26	$4,000
	12	1 3 5 7 9 11 13 15 17 19 21 23 25	$7,000
	13	2 4 6 8 10 12 14 16 18 20 22 24 26	$7,000
	14	All tasks (manager)	$9,000
46	1	1 7 9 16 26 27 35	$5,000
	2	7 13 25 30 39 41 45	$5,000
	3	9 15 16 21 24 28 29 34	$5,500
	4	1 2 6 9 16 19 42	$5,000
	5	2 16 17 32 39 40 45	$5,000
	6	6 8 13 21 30 35 40	$5,000
	7	8 15 16 25 30 42 43	$5,000
	8	4 5 11 12 24 27 28 29	$5,600
	9	3 5 13 18 23 40 45	$5,000
	10	10 17 18 19 23 27 31 35	$5,700
	11	8 18 21 25 39 40 44	$5,000
	12	7 8 11 16 35 41 44	$5,000
	13	6 9 18 19 25 31 33 40	$5,500
	14	2 14 22 27 33 36 37	$5,000
	15	14 18 20 23 28 32 33 36	$5,900
	16	4 8 13 27 32 34 39 40	$6,000

	17	4 8 18 19 39 41 44	$5,000
	18	8 13 14 23 32 36 38 39	$5,800
	19	1 5 6 14 20 22 39	$5,000
	20	1 2 7 9 10 38 42 45	$5,900
	21	2 6 18 21 30 33 34	$5,000
	22	2 19 25 26 27 28 43 46	$6,100
	23	2 13 14 22 27 30 38 46	$5,900
	24	2 16 19 31 34 35 37	$5,000
	25	5 11 14 27 29 36 44 46	$5,500

Table 4 Demand rates of each example

Number of tasks	Demand of each product				
	Product 1	Product 2	Product 3	Product 4	Sum
9	50	50			100
14	35	30	20	15	100
26	35	35	30		100
46	100				100

Table 5 Reducible time and salaries of temporary worker (9 tasks)

Number of tasks	Task	Reducible time		Salaries of temporary workers
		Product 1	Product 2	
9	1	1	3	$1,500
	2	0	3	
	3	1	5	
	4	2	0	
	5	3	4	
	6	2	4	
	7	4	0	
	8	0	3	
	9	1	3	

Table 6 Reducible time and salaries of temporary worker (14 tasks)

Number of tasks	task	Reducible time				Salaries of temporary workers
		Product 1	Product 2	Product 3	Product 4	
14	1	1	0	0	0	$1,500
	2	2	2	0	0	
	3	1	0	0	1	
	4	1	1	1	1	
	5	2	2	0	1	
	6	3	3	5	4	
	7	0	0	2	2	
	8	1	1	1	1	
	9	3	0	0	5	
	10	0	0	1	0	
	11	0	3	3	3	
	12	1	0	1	1	
	13	0	1	2	0	
	14	0	0	1	0	

Table 7 Reducible time and salaries of temporary worker (26 tasks)

Number of tasks	task	Reducible time			Salaries of temporary workers
		Product 1	Product 2	Product 3	
26	1	2	2	2	$1,500
	2	0	1	2	
	3	0	2	2	
	4	1	0	1	
	5	1	1	0	
	6	3	0	0	
	7	1	1	1	
	8	0	1	0	
	9	1	0	1	
	10	0	0	0	
	11	1	1	1	
	12	2	0	0	
	13	0	0	3	
	14	3	3	3	
	15	2	1	2	
	16	0	1	1	
	17	0	1	0	
	18	0	0	1	
	19	5	0	0	
	20	3	0	2	
	21	2	2	2	
	22	0	1	2	
	23	5	0	0	
	24	3	0	0	
	25	1	1	1	
	26	1	0	0	

Table 8 Reducible time and salaries of temporary worker (46 tasks)

Number of tasks	task	Reducible time	task	Reducible time	Salaries of temporary workers
46	1	3	24	6	$1,500
	2	1	25	2	
	3	1	26	4	
	4	2	27	1	
	5	2	28	2	
	6	4	29	3	
	7	1	30	2	
	8	1	31	6	
	9	3	32	2	
	10	1	33	3	
	11	1	34	3	
	12	2	35	2	
	13	2	36	1	
	14	1	37	1	
	15	5	38	3	
	16	2	39	3	
	17	6	40	2	
	18	2	41	2	
	19	1	42	1	
	20	2	43	3	
	21	3	44	2	
	22	2	45	1	
	23	3	46	1	

4.1 Experiments for Model Ⅰ

In this section, experiments for 26 examples were conducted with Model Ⅰ, which is used to minimize the total cost for a given cycle time. Operation costs of a workstation and a cycle time are necessary to implement the model. They are given in Table 9. The upper limit of the operation time is also shown in Table 9. In this model, the maximum number of workers allowed in one workstation is fixed at 5 for all examples. Other cost coefficients and necessary parameters are described in Section 2.3. Results of the examples are summarized in Table 10.

Some obvious managerial insights can be confirmed through the results of these experiments. The shorter the cycle time, the more costs are incurred. If a baton touch zone is large, the assembly line can be balanced in a more flexible manner and cost savings are delivered. In one of the most interesting findings, few temporary workers are assigned when operation costs of workstations are inexpensive. Managers need not reduce operation times of tasks in a workstation to meet a given cycle time. Instead, they just need to assign fewer tasks to a workstation. Therefore, installation of additional workstations is a better choice than employing temporary workers. Evidence of the success of this cost-reduction strategy is illustrated in Table 11, which shows the solution of Example 1.

Table 9 Cycle time and operation cost of workstation

Example number	Number of tasks	Cycle time	Upper limit of operation time	Operation cost of a workstation
1	9	30min	45 min	$30,000
2	9	30 min	36 min	
3	9	25 min	37.5 min	
4	9	25 min	30 min	
5	9	20 min	30 min	
6	9	20 min	24 min	
7	9	30min	45 min	$3,000
8	9	30 min	36 min	
9	9	25 min	37.5 min	
10	9	25 min	30 min	
11	9	20 min	30 min	
12	9	20 min	24 min	
13	14	20 min	30 min	$30,000
14	14	20 min	24 min	
15	14	15 min	22.5 min	
16	14	15 min	18 min	
17	14	12 min	18 min	
18	14	12 min	14.4 min	
19	14	20 min	30 min	$3,000
20	14	20 min	24 min	
21	14	15 min	22.5 min	
22	14	15 min	18 min	
23	14	12 min	18 min	
24	14	12 min	14.4 min	
25	26	30 min	36 min	$30,000
26	46	25 min	30 min	$30,000

Table 10 Result of each example solved through Model Ⅰ

Example number	Number of assigned skilled workers	Number of assigned temporary workers	Number of assigned workstations	Objective value
1	5	3	2	$82,200
2	5	1	3	$109,000
3	5	0	3	$107,800
4	5	3	3	$112,000
5	5	3	3	$113,000
6	5	3	4	$142,000
7	5	0	3	$26,500
8	5	1	3	$28,000
9	5	0	3	$26,800
10	5	0	4	$29,500
11	5	1	4	$31,000
12	5	3	4	$34,000
13	3	1	2	$74,900
14	3	3	2	$77,900
15	3	3	3	$109,700
16	4	4	3	$113,800
17	5	4	3	$118,100
18	4	5	4	$146,000
19	3	1	2	$20,900
20	3	3	2	$23,900
21	3	3	3	$28,700
22	4	0	4	$30,900
23	4	2	4	$33,500
24	4	5	4	$38,000
25	Not found			
26	Not found			

Table 11 Solution of Example 1 of Model Ⅰ

Task sequence	i	k	s	o_{ik}	r_{ik}	o_i	r_i	Cumulative workstation time	w	Temporary worker
1	1	1	1	4	1	9.5	2	7.5	3	O
2		2		15	3					
3	2	2		10	3	5	3	12.5	4	
4	3	1		3	1	9	3	18.5	4	O
5		2		15	5					
6	5	1		5	3	11	3.5	26	3	O
7		2		17	4					
8	4	1	2	6	2	3	2	3	2	
9	6	1		6	2	8	3	11	2	
10		2		10	4					
11	7	1		8	4	4	4	15	6	
12	8	2		12	3	6	3	21	5	
13	9	1		3	1	7	2	28	2	
14		2		11	3					

4.2 Experiments for Model Ⅱ

In this section, experiments for examples were undertaken with Model Ⅱ, which minimizes the cycle time for a specified number of workstations. Table 12 shows the necessary parameters: the number of workstations that can be installed, the number of workers allowed in a workstation, and the ratio of total operation time to the cycle time of each workstation. Results are summarized in Table 13. Table 14 shows the solution of Example 10.

Table 12 Upper limit ratio of cycle time and number of workstations

Example number	Number of tasks	Upper limit of number of workers allowed	Ratio of total operation time to cycle time	Number of workstations can be installed
1	9	3	1.5	2
2	9			3
3	9		1.2	2
4	9			3
5	9	4	1.5	2
6	9			3
7	9		1.2	2
8	9			3
9	9	5	1.5	2
10	9			3
11	9		1.2	2
12	9			3
13	14	3	1.5	2
14	14			3
15	14		1.2	2
16	14			3
17	14	4	1.5	2
18	14			3
19	14		1.2	2
20	14			3
21	14	5	1.5	2
22	14			3
23	14		1.2	2
24	14			3
25	26	5	1.2	8
26	46	5	1.2	8

Table 13 Result of each example solved through Model Ⅱ

Example number	Number of assigned skilled workers	Number of assigned temporary workers	Cycle time
1	4	1	33.33 min
2	5	4	22.00 min
3	5	1	41.67 min
4	6	3	27.50 min
5	5	2	30.67 min
6	6	4	20.00 min
7	5	3	38.33 min
8	5	6	25.00 min
9	5	3	28.67 min
10	6	8	18.00 min
11	6	4	35.83 min
12	5	6	22.50 min
13	4	2	18.74 min
14	5	4	12.00 min
15	3	2	20.83 min
16	4	5	14.17 min
17	4	4	18.38 min
18	6	6	11.48 min
19	4	4	19.25 min
20	4	8	13.10 min
21	5	5	18.05 min
22	7	8	11.30 min
23	5	5	18.93 min
24	5	10	13.05 min
25	Not found		
26	Not found		

Table 14 Solution of Example 10 of Model Ⅱ

Task sequence	i	k	s	o_{ik}	r_{ik}	o_i	r_i	Cumulative workstation time	w	Temporary worker
1	1	1	1	4	1	9.5	2	7.5	6	O
2		2		15	3					
3	2	2		10	3	5	1.5	11	6	O
4	4	1		6	2	3	1	14	2	
5	3	1	2	3	1	9	3	6	4	O
6		2		15	5					
7	5	1		5	3	11	3.5	13.5	5	O
8		2		17	4					
9	7	1		8	4	4	2	15.5	4	O
10	6	1	3	6	2	10	3	7	1	O
11		2		14	4					
12	8	2		12	3	6	1.5	11.5	1	O
13	9	1		3	1	7	2	16.5	3	O
14		2		11	3					

In Table 14, the largest cumulative aggregated operation time of workstations is 16.5 minutes. However, if Product 2 is being manufactured, the operation time required for workstation 3 is at least 27 minutes. The upper limits of the operation time should be smaller than 1.5 times that of the cycle time. Hence, the optimal cycle time in this example is 18 minutes.

4.3 Experiments for Model Ⅲ

In this section, experiments for examples were undertaken using Model Ⅲ, which minimizes the total work overload for a specified maximum number of workstation. Table 15 shows necessary parameters: the number of workstations that can be installed and the number of workers allowed in each workstation. Results of solutions to examples solved with this model are summarized in Table 16. Table 17 shows the solution of Example 11.

Table 15 Upper limit of number of workers and workstations

Example number	Number of tasks	Upper limit of number of workers allowed	Maximum number of workstations can be installed
1	9	2	2
2	9		3
3	9		4
4	9	3	2
5	9		3
6	9		4
7	9	4	2
8	9		3
9	9		4
10	9	5	2
11	9		3
12	9		4
13	14	2	2
14	14		3
15	14		4
16	14	3	2

17	14		3
18	14		4
19	14	4	2
20	14		3
21	14		4
22	14	5	2
23	14		3
24	14		4
25	26	5	8
26	46	5	8

Table 16 Result of each example solved through Model Ⅲ

Example number	Number of assigned skilled workers	Number of assigned temporary workers	Total work overload
1	3	1	36.25 min
2	4	2	32.75 min
3	5	3	30.25 min
4	3	3	30.75 min
5	5	4	28.00 min
6	6	6	23.75 min
7	4	4	28.00 min
8	5	6	23.75 min
9	6	9	21.75 min
10	4	6	23.75 min
11	6	7	21.75 min
12	6	7	21.75 min
13	3	1	36.10 min
14	3	3	32.75 min
15	4	4	29.68 min
16	3	3	21.34 min
17	4	5	18.53 min
18	6	6	18.31 min

19	3	5	14.95 min
20	4	7	13.11 min
21	6	8	12.90 min
22	3	7	11.40 min
23	4	9	10.16 min
24	6	9	10.16 min
25	Not found		
26	Not found		

Table 17 Solution of Example 11 of Model Ⅲ

<table>
<tr><th>Task sequence</th><th>i</th><th>k</th><th>s</th><th>o_{ik}</th><th>r_{ik}</th><th>o_i</th><th>r_i</th><th>Cumulative workstation time</th><th>w</th><th>Temporary worker</th></tr>
<tr><td>1</td><td rowspan="2">1</td><td>1</td><td rowspan="3">1</td><td>4</td><td>1</td><td rowspan="2">9.5</td><td rowspan="2">2</td><td rowspan="2">7.5</td><td rowspan="2">6</td><td rowspan="2">O</td></tr>
<tr><td>2</td><td>2</td><td>15</td><td>3</td></tr>
<tr><td>3</td><td>2</td><td>2</td><td>10</td><td>3</td><td>5</td><td>1.5</td><td>11</td><td>6</td><td>O</td></tr>
<tr><td>4</td><td rowspan="2">3</td><td>1</td><td rowspan="5">2</td><td>3</td><td>1</td><td rowspan="2">9</td><td rowspan="2">3</td><td rowspan="2">6</td><td rowspan="2">4</td><td rowspan="2">O</td></tr>
<tr><td>5</td><td>2</td><td>15</td><td>5</td></tr>
<tr><td>6</td><td>4</td><td>1</td><td>6</td><td>2</td><td>3</td><td>1</td><td>9</td><td>2</td><td></td></tr>
<tr><td>7</td><td rowspan="2">5</td><td>1</td><td>5</td><td>3</td><td rowspan="2">11</td><td rowspan="2">3.5</td><td rowspan="2">16.5</td><td rowspan="2">5</td><td rowspan="2">O</td></tr>
<tr><td>8</td><td>2</td><td>17</td><td>4</td></tr>
<tr><td>9</td><td rowspan="2">6</td><td>1</td><td rowspan="6">3</td><td>6</td><td>2</td><td rowspan="2">10</td><td rowspan="2">3</td><td rowspan="2">7</td><td>1</td><td rowspan="2">O</td></tr>
<tr><td>10</td><td>2</td><td>14</td><td>4</td><td>1</td></tr>
<tr><td>11</td><td>8</td><td>2</td><td>12</td><td>3</td><td>6</td><td>1.5</td><td>15.5</td><td>1</td><td>O</td></tr>
<tr><td>12</td><td>7</td><td>1</td><td>8</td><td>4</td><td>4</td><td>2</td><td>11</td><td>3</td><td></td></tr>
<tr><td>13</td><td rowspan="2">9</td><td>1</td><td>3</td><td>1</td><td rowspan="2">7</td><td rowspan="2">2</td><td rowspan="2">20.5</td><td>3</td><td rowspan="2">O</td></tr>
<tr><td>14</td><td>2</td><td>11</td><td>3</td><td>3</td></tr>
</table>

4.4 Validation of a hybrid genetic algorithm

To validate the proposed algorithms, mathematical model Ⅰ was compared with the proposed GA. The results are presented in Table 18.

Table 18 Comparison of Model Ⅰ and the proposed hybrid genetic algorithm

Example number	Mathematical model Ⅰ		Hybrid Genetic Algorithm		Gap
	Objective value	Remark	Objective value	Remark	
1	$82,200	Optimal	$82,300		0.12%
2	$109,000	Optimal	$112,000		2.75%
3	$107,800	Optimal	$109,000		1.11%
4	$112,000	Optimal	$112,000		0.44%
5	$113,000	Optimal	$113,500		0.44%
6	$142,000	Optimal	$142,200		0.14%
7	$26,500	Optimal	$26,500	Optimal	0%
8	$28,000	Optimal	$31,000		10.71%
9	$26,800	Optimal	$28,000		4.48%
10	$29,500	Optimal	$29,500	Optimal	0%
11	$31,000	Optimal	$32,000		3.22%
12	$34,000	Optimal	$34,200		0.59%
13	$74,900	Optimal	$75,000		0.13%
14	$77,900	Optimal	$78,000		0.13%
15	$109,700	Optimal	$112,600		2.64%
16	$113,800	Optimal	$116,000		1.93%
17	$118,100	Optimal	$118,500		0.34%
18	$146,000	Optimal	$146,400		0.27%
19	$20,900	Optimal	$21,000		0.48%
20	$23,900	Optimal	$24,000		0.34%
21	$28,700	Optimal	$31,100		8.36%
22	$30,900	Optimal	$35,600		15.21%
23	$33,500	Optimal	$33,600		0.30%
24	$38,000	Optimal	$38,400		1.05%
25	Not found		$128,200		
26	Not found		$496,000		

Chapter 5. Conclusions

Moon et al. (2009) introduced the integrated ALBP, and this study extends their idea to the mixed-model assembly line and considers simultaneous assignments of skilled and temporary workers. In the production line described herein the skilled workers have multiple competencies and commensurate salaries and temporary workers are assigned to help them. Three mathematical models were developed by ILP and MILP for the integrated mixed-model ALBP that features temporary workers. The models were designed to minimize total costs, including those associated with operation of workstations and the salaries of skilled and temporary workers for a given cycle time, to minimize the maximum cycle time of workstations for a specified number of workstations, and to minimize the total work overload for a specific number of workstations. Computational experiments of each model are conducted to reveal managerial insights.

Furthermore, an efficient HGA was created to solve the problems. Effective heuristic algorithms were combined with the GA, and the efficiency of solution processes and the quality of solutions were taken into account. The proposed algorithm overcame the computational burden of the mathematical models. Therefore, the GA was shown to be a more helpful means than the mathematical models for designing a cost-effective assembly line with worker assignments for large problems in the real world.

Bibliography

Akpinar, S. and Baykasoglu, A., Modeling and solving mixed-model assembly line balancing problem with setups. Part II: A multiple colony hybrid bees algorithm, *Journal of Manufacturing Systems*, In Press, 2014.

Bryton, B., Balancing of a continuous production line. M.S. Thesis, Northwestern University, Evanston, IL, 1954.

Bukchin, J., Dar-El, E.M., and Rubinovitz, J., Mixed-model assembly line design in a make-to-order environment. *Computers and Industrial Engineering*, Vol. 41, p.405–421, 2002.

Corominas, A., Pastor, R., Plans, J., Balancing assembly line with skilled and unskilled workers. *Omega*, Vol. 36, p.126-1132, 2008.

Erel, E. and Sarin, S.C., A survey of the assembly line balancing procedures. *Production Planning and Control*, Vol. 9, No. 5, p.414-434, 1998

Fattahi, P. and Salehi, M., Sequencing the mixed-model assembly line to minimize the total utility and idle costs with variable launching interval. *International Journal of Advanced Manufacturing Technology*, Vol. 45, p.987–998, 2009.

Holland, J., Adaptation in Natural and Artificial Systems, The University of Michigan, 1975.

Kara, Y., Atasagun, Y., Gokcen, H., Hezer, S., and Demirel, N., An integrated model to incorporate ergonomics and resource restrictions into assembly line balancing. *International Journal of Computer Integrated Manufacturing*, Vol. 27, No.11, p.997-1007, 2014.

Kim, Y.K., Kim, Y.J., and Kim, Y.H., Genetic algorithms for assembly line balancing with various objectives. *Computers & Industrial Engineering*, Vol. 30, No. 3, p.397-409, 1996.

Lapierre, S.D., Ruzi, A., and Socarino, P., Balancing assembly lines with tabu search. *European Journal of Operational Research*, Vol. 168, p.826-837, 2006.

Leu, Y.Y., Matheson, L.A. and Rees, L.P., Assembly line balancing using genetic algorithms with heuristic generated initial populations and multiple criteria. *Decision Sciences*, Vol. 15, p.581–606, 1994.

Merengo, C., Nava, F., Pozetti, A., Balancing and sequencing manual mixed-model assembly lines. *International Journal of Production Research*, Vol. 37, p.2835–2860, 1999.

Meyr, H., Supply chain planning in the German automotive industry. *OR Spectrum*, Vol. 26, p.447–470, 2004.

Monden, Y., Toyota Production System, 2011

Moon, I.K., Logendran, R., and Lee, J.H., Integrated assembly line balancing with resource restrictions. *International Journal of Production Research*, Vol. 47, p.5525-5541, 2008.

Rabbani, M., Moghaddam, M., and Manavizadeh, N., Balancing of mixed-model two-sided assembly lines with multiple U-shaped layout, *International Journal of Advanced Manufacturing Technology*, Vol. 59, p.1191-1210, 2012.

Salveson, M.E., The assembly line balancing problem. *Journal of Industrial Engineering*, Vol. 6, No.3, p.18-25, 1955.

Scholl, A., Balancing and sequencing of assembly Lines, Physica-Verlag, 1999.

Thomopoulos, N.T., Line balancing-sequencing for mixed-model assembly. *Management Science*, Vol. 14, No. 2, p.59–75, 1967.

Tonelli, F., Paolucci, M., Anghinolfi, D., and Taticchi, P., Production planning of mixed-model assembly lines: a heuristic mixed integer programming based approach, *Production Planning & Control*, Vol. 24, No. 1, p.110-127, 2013.

Scholl, A. and Becker, C., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, Vol. 168, p.666-693, 2006.

Sivasankaran, P. and Shahabudeen, P., Literature review of assembly line balancing problems. *The International Journal of Advanced Manufacturing Technology*, Vol. 73, p.1665-1694, 2014.

Xiaobo, Z., Zhou, Z., and Asres, A., A note on Toyota's goal of sequencing mixed models on an assembly line. *Computers & Industrial Engineering*, Vol. 36, p.57-65, 1999.

초　록

이 논문은 단일 제품을 조립하는 일반적인 조립라인 균형화 문제를 복수 제품들을 동시에 조립할 수 있는 혼합 모델 조립라인으로 확장하였으며, 임시 작업자를 고용하여 조립라인을 효율화할 수 있도록 하였다. 이를 고려한 세 가지 버전의 수학적 모형들을 개발하였다. 각 모형의 목표는 모든 직원의 임금과 작업장 비용을 합친 총 비용을 최소화하는 것, 작업장 수가 주어진 상황에서 사이클 시간을 최소화하는 것, 그리고 정해진 작업장 안에서 업무 과부하를 최소화하는 것이다. 제안된 모형들은 숙련된 작업자와 임시 작업자를 동시에 할당하는 사안과 작업들 사이의 선행관계 등 실제 현장에서 적용되는 실용적 특성들을 고려하고 있다. 뿐만 아니라, 총 비용을 최소화할 수 있는 복합유전알고리즘도 개발되었다. 해의 타당성을 보장하고 복합유전알고리즘의 우수성을 높이기 위해 특별한 유전연산자들과 발견적 기법이 사용되었다. 수치실험들을 통해서 복합유전알고리즘의 우수성을 입증하기 위하여 수학적 모형과 비교하였다.

주요어: 라인 균형화, 복합 유전알고리즘, 정수 계획법, 혼합 모델 조립 라인, 혼합정수계획법

학　번: 2013-21062