Integrated Assembly Line Balancing with Skilled and Unskilled Workers

Ilkyeong Moon[1,*], Sanghoon Shin[2], and Dongwook Kim[1]

[1] Department of Industrial Engineering, Seoul National University, Seoul 151-744, Korea
ikmoon@snu.ac.kr
[2] Postal Technology Research Center, ETRI, Daejeon, Korea

Abstract. In this paper, we extend the general assembly line balancing problem by designing an integrated assembly line and addressing the number of workstations and simultaneous assignments of skilled and unskilled workers. We develop a mixed integer program that minimizes the sum of total annual workstation costs and annual salaries of skilled and unskilled workers within a predetermined cycle time. Because this problem is NP-hard, we also develop a genetic algorithm to obtain efficient solutions for large problems. Numerical experiments demonstrate the efficiency of the random key-encoded genetic algorithm.

Keywords: assembly line balancing, genetic algorithm, skilled/unskilled workers.

1 Introduction

Most manufacturing companies employ workers based on their experience or skill set while also trying to reduce production costs. Therefore, assembly line problems have been widely researched during the latest period of industrial development. In his graduate thesis, Bryton (1954) introduced the idea of line balancing, and many studies have undertaken this subject over the subsequent 60 years. Various new assembly line balancing problem types such as two-sided, parallel, mixed-models and others, have emerged as have effective solution algorithms. Hoffmann [5] presented optimal line balances by operation on a matrix of zeros and ones called a precedence matrix.

Because the installation of an assembly line requires large investments, assembly line balancing problems can be distinguished based on objectives addressed through such as cost- or profit-oriented models. One of the cost-oriented models involves selecting equipment specific to the tasks assignments to workstations. The manufacturing of a product depends on resources such as equipment and manpower as well as processes. The term assembly line design problem is frequently used where these decisions are related. Pinto et al. [7] considered a model that combines the balancing problem with process alternatives that reduce task time. Bukchin and Tzur [2]

* Corresponding author.

B. Grabot et al. (Eds.): APMS 2014, Part I, IFIP AICT 438, pp. 459–466, 2014.

suggested a model of equipment alternatives that minimizes the total equipment costs for a given cycle time. The cost-oriented model can be extended by considering.

Since Salveson [8] formulated the assembly line balancing problem (ALBP) mathematically, numerous heuristic methods and procedures have been introduced. Corominas et al. [4] considered the process of rebalancing the line at a motorcycle assembly plant. They suggested a mathematical model based on a binary linear program to minimize the number of unskilled workers required. Chong et al. [3] compared a heuristics-treated initial population and a randomly generated initial population of genetic algorithm for solving ALBP. Moon et al. [6] introduced the integrated assembly line balancing problem with resource restrictions. They considered multi-skilled workers as resources and formulated a mixed integer linear program.

In this paper, we deal with the problem of designing an integrated assembly line with simultaneous assignment of skilled and unskilled workers. Although Corominas et al. [4] considered labor, they did not consider integrated assembly line balancing at the same time. Their objective was to minimize the number of unskilled workers for each workstation. Moon et al. [6] studied an integrated assembly line balancing problem, but they did not consider reducible task times through cooperation, a problem that arises when limited resources affect the operation time for every task. The selection of both skilled workers, whose salaries depend on their competencies, and unskilled workers, who can reduce operation time to perform tasks, is addressed in this study. In addition, the assignment of tasks to workstations with precedence restrictions is also considered as the resource issues are evaluated.

2 Mathematical Model

For the integrated assembly line balancing with skilled/unskilled workers, the following assumptions were made:

(1) The skilled workers are multi-skilled with different salaries.
(2) The skilled workers must be assigned to one workstation.
(3) The skilled workers can be assigned to tasks depending on their skills.
(4) The unskilled workers cannot be assigned alone to any workstations.
(5) The task time can be reduced by assigning an unskilled worker to a task.
(6) An unskilled worker can be assigned to only one task.
(7) A skilled worker can be assigned to a single workstation.
(8) The precedence constraints determine the sequence in which the tasks can be processed.
(9) There is a limitation for assigning tasks at a station.

Using these assumptions, we present a mixed integer linear program for an integrated assembly line balancing problem with skilled and unskilled workers. The following notation is used to explain the model:

Notation

i, j	indices of tasks (i,j = 1, 2, …, I, J)
s	index for workstations (s = 1, 2, …, S)
W	index for skilled workers (w=1,2, …, W)
C	cycle time
o_i	operating time for task i when performed by a skilled worker
γ_i	reducible time for task i when performed by a skilled worker and a unskilled worker together
$P_{(i,j)}$	set of task pairs (i,j) such that there is an immediate precedence relation between them
A_w	set of available tasks that can be assigned to skilled workers
FC	operating costs of a workstation
LC_w	salary for skilled worker w
LA	salary for unskilled worker
n	upper bound of number of tasks that can be assigned to a workstation
M	a sufficiently large number

Decision Variables

F Number of workstations to be used in the assembly line

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

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

$$Z_{isw} \begin{cases} 1, \text{ if task } i \text{ is performed by an unskilled worker with skilled worker } w \\ \quad \text{at workstation } s \\ 0, \text{otherwise} \end{cases}$$

Objective function

$$Min \quad FC \cdot F + \sum_{w=1}^{W} LC_w \left(\sum_{s=1}^{S} Y_{sw} \right) + LA \cdot \sum_{i=1}^{I} \sum_{s=1}^{S} \sum_{w=1}^{W} Z_{isw} \tag{1}$$

Subject to

$$\sum_{s=1}^{S} \sum_{w=1}^{W} X_{isw} = 1 \quad \forall i \tag{2}$$

$$Z_{isw} \le X_{isw} \qquad \forall i, s, w \tag{3}$$

$$\sum_{s=1}^{S} Y_{sw} \le 1 \qquad \forall w \tag{4}$$

$$\sum_{s=1}^{S} \sum_{w=1}^{W} (s \cdot X_{isw} - s \cdot X_{jsw}) \le 0 \qquad \forall (i, j) \in P_{(ij)} \tag{5}$$

$$\sum_{i=1}^{I} \sum_{w=1}^{W} (o_i \cdot X_{isw} - r_i \cdot Z_{isw}) \le C \qquad \forall s \tag{6}$$

$$\sum_{i=1}^{I} X_{isw} \le M \cdot Y_{sw} \quad \forall s, w \tag{7}$$

$$\sum_{i=1, i \neq A_w}^{I} \sum_{s=1}^{S} X_{isw} = 0 \quad \forall w \tag{8}$$

$$\sum_{i=1}^{I} \sum_{w=1}^{W} (X_{isw} + Z_{isw}) \le n \quad \forall s \tag{9}$$

$$\sum_{s=1}^{S} s \cdot X_{isw} \le F \quad \forall i, w \tag{10}$$

$$X_{isw}, Y_{sw}, Z_{isw} \in \{0,1\} \tag{11}$$

The objective function (1) is to minimize the sum of the total annual workstation costs and the annual salaries of the skilled and unskilled workers. Constraints (2) ensure that every task is performed by one skilled worker at exactly one workstation. Constraints (3) restrict that an unskilled worker might be assigned to task i when a skilled worker is assigned to task i. Constraints (4) restrict that a skilled worker is assigned to exactly one workstation. Constraints (5) ensure the implementation of the precedence relationships between the precedence task sets identified by $P_{(i,j)}$; for example, if task i is the immediate predecessor of task j, it must be assigned a higher operating index than task j. Constraints (6) represent that the total maximum operating time of each workstation within a given cycle time. Constraints (7) define the task assignment at the workstation with a worker. $X_{(isw)}$ can be greater than zero when skilled worker w is assigned to a workstation. Constraints (8) guarantee that a skilled worker cannot be assigned to a workstation with a task that the worker cannot perform according to the skilled worker's available task set A_w. Constraints (9) restrict the number of skilled and unskilled workers to the predetermined number for the workspace. Constraints (10) determine the total number of workstations to be used. Constraints (11) express the binary nature of the variables.

3 Genetic Algorithm

We developed a random key-encoded genetic algorithm with a procedure that slightly differs from the general one. The suggested genetic algorithm is summarized in following sentences.

Step 1. Generate an initial population of randomly constructed chromosomes with random keys for task and unskilled worker assignments that represent solutions to the problem.

Step 2. For the task arrangement chromosome, the offspring is generated using the convex hull crossover.

Step 3. Subject the offspring to mutation based-on probability to create slight changes in offspring structure. This mutation procedure can avoid premature convergence to a local optimum.

Step 4. Reorder the offspring to obtain fitness values, which are used for measuring the quality of each chromosome in comparison with others.

Step 5. Apply a selection procedure. The chromosome with the best fitness value joins the population and the one with the poorest fitness value is removed

Step 6. Terminate the algorithm when predetermined generations were reached or when the best individual does not improve more than 0.01% for predetermined generations.

3.1 Representation Chromosomes

The proper representation of a solution plays a key role in the development of a genetic algorithm. A string that consists of real integers is a solution (a chromosome). Traditionally, chromosomes are simple binary strings; however, this simple representation is not well suited for reproduction and other procedures. In this study, we use a chromosome that consists of two parts to represent task precedence and unskilled worker assignments. One of these offers solutions for assigning the tasks to the workstations through a random key, while the other offers solutions for unskilled worker assignments through randomly generated binaries as shown in Figure 1.

We used a random key to the task assignment for maintaining precedence relationships. If tasks do not have any predecessors, then their random keys are compared for selection and assigned to a workstation. For example, because Tasks 1 and 2 do not have any predecessors, as shown in Figure 1, they are candidates for selection. However, Task 2 is selected for a workstation assignment because the random key of Task 2 is larger than that of Task 1. After selection, the value of the random key is assigned a negative value to prevent it from being selected again.

For the skilled worker assignment, we used a simple heuristic that sorts on the basis of task precedence restrictions and the skilled workers' available task sets. The steps for the simple heuristic are as follows:

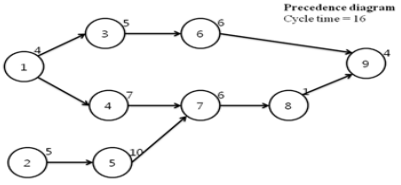
Step 1. Calculate operating time of cumulative tasks by observing the chromosome and divide workstations using the predetermined cycle time.

Step 2. Arrange the 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 workstations, find the appropriate combination of skilled workers.



Precedence table

task	Predecessor
1	Φ
2	Φ
3	1
4	1
5	2
6	3
7	4,5
8	7
9	6,8

Representation chromosome

Task	1	2	3	4	5	6	7	8	9
Random key	0.93	1.62	4.27	5.62	7.52	8.11	3.62	6.21	9.73
Temporary worker	0	1	1	0	1	0	1	1	0

Fig. 1. Representation and initialization for a sample problem

3.2 Objective and Fitness Functions

The evaluation function in a genetic algorithm plays a role similar to that of the environment in natural evaluation. 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 objective is to find the string with the minimum fitness function value. For a given string, one can locate the minimum relevant cost. Equation (1) is used as a fitness function in the proposed genetic algorithm.

3.3 Reproduction, Crossover, and Mutation

The genetic operators such as crossover and mutation produce a second generation population of solutions. The method of the genetic operators -reproduction, crossover, and mutation- can change the set of the next population.

Reproduction is the biological process of producing new offspring from parents. In this study, parents are chosen by a rank mechanism. Unlike other reproduction approaches, a ranking approach offers a smoother selection probability curve. This prevents organisms from being dominated at early points in evolution. The crossover operation is a diversification mechanism that enables the genetic algorithm to examine unvisited regions and generate a solution. In this paper, we show the convex hull crossover operator for the task assignment and the one-cut point crossover operator for the unskilled worker assignment. We select two relative chromosomes for the crossover. Figure 2 shows an example to illustrate it. For the unskilled worker assignment, the position of the cut point is randomly generated from the range [1, (length of the chromosome – 1)]. We generate the offspring simply by exchanging the appropriate parts of their parents.

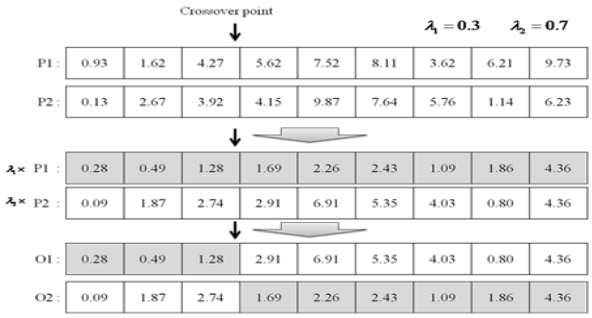


Fig. 2. Example of the convex hull crossover procedure

In a genetic algorithm, mutation is a background operator that produces a spontaneous random change in various chromosomes. Mutation plays the crucial role of replacing genes lost from the population during evolution so that they can maintain diverse populations, thereby preventing the population from being too rigid to adapt to a dynamic environment. In this paper, we used a swap mutation operator for the task assignment and a random point mutation operator for the unskilled worker assignment. The random point mutation selects a random point from the range [1, (length of the chromosome – 1)] where the gene is subsequently changed.

3.4 Termination

The terminating condition was determined when 50,000 generations are produced or when individual fitness was not bettered by more than 0.1% over 2,000 generations. For the parameters of the proposed genetic algorithm, 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 of genetic operators

Parameters	*Value*
Population size	100
Convex hull crossover probability	0.7
One-cut point crossover probability	0.6
Swap mutation probability	0.4
One point mutation probability	0.4

4 Computational Experiments

The mixed integer linear program was implemented and solved using LINGO version 10.0. The proposed random key-encoded genetic algorithm was developed using C# with .Net framework version 4.0. Both numerical experiments were conducted on a Pentium IV 3.2GHz processor with 2GB RAM on the Microsoft Windows XP

Table 2. Comparison of mathematical model and proposed genetic algorithm

No	Examples	Mathematical Model		Genetic Algorithm	
		Objective	Time(sec)	Objective	Time(sec)
1	9 Tasks 8 Workers (1)	390,000	105	390,000	30
2	9 Tasks 8 Workers (2)	406,000	110	406,000	33
3	9 Tasks 8 Workers (3)	358,000	201	358,000	25
4	11 Tasks 8 Workers (1)	523,000	190	523,000	21
5	11 Tasks 8 Workers (2)	473,000	85	473,000	63
6	11 Tasks 8 Workers (3)	476,000	187	476,000	52
7	12 Tasks 8 Workers (1)	578,000	153	578,000	68
8	12 Tasks 8 Workers (2)	509,000	221	509,000	96
9	12 Tasks 8 Workers (3)	572,000	427	572,000	111
13	21 Tasks 15 Workers (1)	746,000	2,807	746,000	155
14	21 Tasks 15 Workers (2)	746,000	3,848	746,000	109
15	21 Tasks 15 Workers (3)	-	-	756,000	91
16	32 Tasks 15 Workers (1)	-	-	1,191,000	434
17	32 Tasks 15 Workers (2)	-	-	1,118,000	598
18	32 Tasks 15 Workers (3)	-	-	1,148,000	572
19	61 Tasks 29 Workers (1)	-	-	3,188,000	1,533

operating system. The mathematical model requires significant time to find an optimal solution. Moreover, this model was not able to solve problems as large as those with 32 tasks. To validate results, the mathematical model was compared with the proposed genetic algorithm for small problems.

The experiment was conducted for large problems: 61 tasks and 29 workers. The predetermined cycle time is 350 minutes and the annual operating cost for each workstation is $180,000/year. This assembly line is composed of 12 workstations, 23 skilled workers, and 23 unskilled workers. The total operating cost, which combines expenses of workstations as well as those of skilled and unskilled workers, is $3,188,000. The average computational time of 10 evaluations is 1,533 seconds. In addition, we solved 19 different examples. The result of the mathematical model and proposed genetic algorithm is shown in Table 2.

5 Conclusions

Moon et al. [6] proposed the concept of integrated assembly line balancing problem, and this study extends the idea by considering skilled and unskilled worker assignments. The skilled workers have multiple competencies and commensurate salaries, and unskilled workers, with lower salaries, are assigned to help them. We developed a mathematical model as an MILP for the integrated assembly line balancing problem with skilled and unskilled workers based on a task precedence relationship. Furthermore, we developed an efficient genetic algorithm based on random key-encoded chromosomes. The proposed genetic algorithm overcame the computational burden of the MILP. Therefore, the genetic algorithm has been shown to be a more helpful alternative than an MILP for designing a cost-effective assembly line with suitable worker assignments.

Acknowledgments. This work was supported by the BK21 Plus Program (Center for Sustainable and Innovative Industrial Systems) funded by the Ministry of Education, Korea.

References

1. Bryton, B.: Balancing of a continuous production line. M.S. Thesis, Northwestern University, Evanston, IL (1954)
2. Bukchin, J., Tzur, M.: Design of flexible assembly line to minimize equipment cost. IIE Transactions 32, 585–598 (2000)
3. Chong, K.E., Omar, M.K., Bakar, N.A.: Solving Assembly Line Balancing Problem using Genetic Algorithm with Heuristics-Treated Initial Population. In: Proceedings of the World Congress on Engineering 2008, July 2-4, London, U.K. (2008)
4. Corominas, A., Pastor, R., Plans, J.: Balancing assembly line with skilled and unskilled workers. The International Journal of Management Science 36, 1126–1132 (2008)
5. Hoffman, T.R.: Assembly line balancing with precedence matrix. Management Science 9, 551–562 (1963)
6. Moon, I.K., Logendran, R., Lee, J.H.: Integrated assembly line balancing with resource restrictions. International Journal of Production Research 47, 5525–5541 (2008)
7. Pinto, P.A., Dannenbring, D.G., Khumawala, B.M.: Assembly line balancing with processing alternatives: an application. Management Science 29, 817–830 (1983)
8. Salveson, M.E.: The assembly line balancing problem. The Journal of Industrial Engineering 6(3), 18–25 (1955)