This article was downloaded by: *[Moon, Ilkyeong][Pusan National University Library]*
On: *24 July 2009*
Access details: *Access Details: [subscription number 911480638]*
Publisher *Taylor & Francis*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713696255

Integrated assembly line balancing with resource restrictions

Ilkyeong Moon [a]; Rasaratnam Logendran [b]; Jeonghun Lee [c]
[a] Department of Industrial Engineering, Pusan National University, Busan, Korea [b] School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, USA [c] Postal Technology Research Center, ETRI, Daejeon, Korea

First Published:January2009

To cite this Article Moon, Ilkyeong, Logendran, Rasaratnam and Lee, Jeonghun(2009)'Integrated assembly line balancing with resource restrictions',International Journal of Production Research,47:19,5525 — 5541

To link to this Article: DOI: 10.1080/00207540802089876

URL: http://dx.doi.org/10.1080/00207540802089876

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

International Journal of Production Research
Vol. 47, No. 19, 1 October 2009, 5525–5541

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Integrated assembly line balancing with resource restrictions

Ilkyeong Moon[a]*, Rasaratnam Logendran[b] and Jeonghun Lee[c]

[a]*Department of Industrial Engineering, Pusan National University, Busan, Korea;* [b]*School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, USA;* [c]*Postal Technology Research Center, ETRI, Daejeon, Korea*

(*Received 30 April 2007; final version received 20 March 2008*)

This paper documents a study carried out on the problem of designing an integrated assembly line when many workers with a variety of skills are employed. This study addresses the problem of selecting multi-functional workers with different salaries to match their skills and of assigning tasks to work stations when there are precedence restrictions among the tasks. The objective of this study is to minimise the total annual work station costs and the annual salary of the assigned workers within a predetermined cycle time. A mixed integer linear program is developed with a genetic algorithm in order to address the problem of resource restrictions related to integrated assembly line balancing. Numerical examples demonstrate the efficiency of the developed genetic algorithm.

Keywords: assembly line balancing; resource restrictions; genetic algorithm

1. Introduction

In recent years, since there has been an increased level of competition within the industry, it has become more critical to analyse production costs. Even in successful production systems such as the assembly line, possibilities still need to be found that will reduce production costs. As the final assembly is usually a worker-intensive production, the existing wage compensation system needs to be analysed.

Since the 1950s, the industry has considered the concept of assembly line balancing (ALB). An assembly line comprises a number of tasks, n, and a number of work stations, m, which are arranged in serial and parallel sequence, through which flows the progress work on a product. An ALB problem considers the assignment of n to m with various restrictions. One of the most widely researched objectives is the minimisation of cycle time for a given number of work stations or the minimisation of the number of work stations for a given cycle time. This objective considers the precedence constraints and incompatible relationships between tasks, given that the total operating time at each work station is not greater than a cycle time. In line with these objectives, the tasks are grouped into work stations. A grouping that satisfies the objective is called a 'balance'. In the last 50 years, many mathematical programming schemes and meta-heuristics have been introduced to the ALB problem. The problem of designing and balancing assembly lines is very difficult to solve optimally, and it has proven to be an NP-hard problem (Wee and Magazine 1982).

*Corresponding author. Email: ikmoon@pusan.ac.kr

ISSN 0020–7543 print/ISSN 1366–588X online
© 2009 Taylor & Francis
DOI: 10.1080/00207540802089876
http://www.informaworld.com

The problem of designing and balancing assembly lines has been extensively examined in the literature used and referred to in this study, and a number of analyses have been published. Faaland *et al.* (1992) considered assigning alternative resources and tasks to different stations. Their proposed solution included a precise procedure and two heuristic algorithms. Kim *et al.* (1998) developed a branch-and-bound algorithm for the two-sided assembly line balancing problem. Nearchou (2007) provided a differential evolution based approach for solving the multi-objective assembly line balancing problem. Vilarinho and Simaria (2006) developed an ant colony optimisation algorithm for the mixed-model assembly line both to minimise the number of work stations for a given cycle time and to balance the workload across the work stations. McMullen and Tarasewich (2006) applied ant colony optimisation to the assembly line balancing problem in which there are multiple objectives such as system utilisation, system design costs, etc. They showed that the multiple-objective approach outperforms the single-objective approach. Levitin *et al.* (2006) solved the problem on robotic assembly lines by using a genetic algorithm that they compared with a truncated branch-and-bound algorithm. As outlined by Becker and Scholl (2006), the problem of selecting the appropriate equipment selection requires the same criteria as that of the problem of selecting workers who have different task performance speeds. Bukchin and Rabinowitch (2006) considered the work station cost and task duplication cost and developed a branch-and-bound algorithm in order to solve the problem with mixed-model assembly line balancing. Akagi *et al.* (1983), Wilson (1986), and Lutz *et al.* (1994) assumed that the task performance speed is the same for all persons who are performing identical tasks. Dimitriadis (2006) studied an ALB problem that differs from the conventional problem, where groups of workers at multi-manned work stations simultaneously performed different assembly tasks on the same product and at the same work station. However, they did not consider each possible task for each worker, nor did they consider the cost of workers. Moreover, they applied a two-step solution method in which a group of workers on a work station are assigned to a task in the first step and an individual task is assigned to each worker in the second step. However, their method was not able to guarantee the attainment of an optimal solution.

This study extends the ALB problem to one that is integrated and that has resource restrictions. Multi-functional workers are defined as the restrictions and these workers are paid different salaries depending on their skills and task precedence in the assembly line. Figure 1 illustrates the optimisation method for the integrated ALB, which has restrictions on task precedence and assignment of multi-functional workers. In contrast to the optimisation procedure used by other researchers in the literature used for this study, the procedure for this study considers the method as an integrated optimisation. In this present research, the assignment of tasks to the work stations and the assignment of multi-functional workers to the tasks are simultaneously optimised in order to minimise the associated overall costs.

The proposed mathematical model and genetic algorithm represent an integrated optimisation. The solution for different relevant costs is obtained using the method in which the operation procedure is manipulated by minimising the sum of the total annual work station costs and the annual salary of the workers. In this context, the study considers the assignment of a task to each work station and the assignment of a multi-functional worker to each task. Each work station and multi-functional worker incurs an annual fixed operating cost and annual salary, respectively. The objective of examining this problem is to minimise the total annual work station cost as well as the annual salary of the workers. The mathematical model and genetic algorithm minimise the total relevant

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

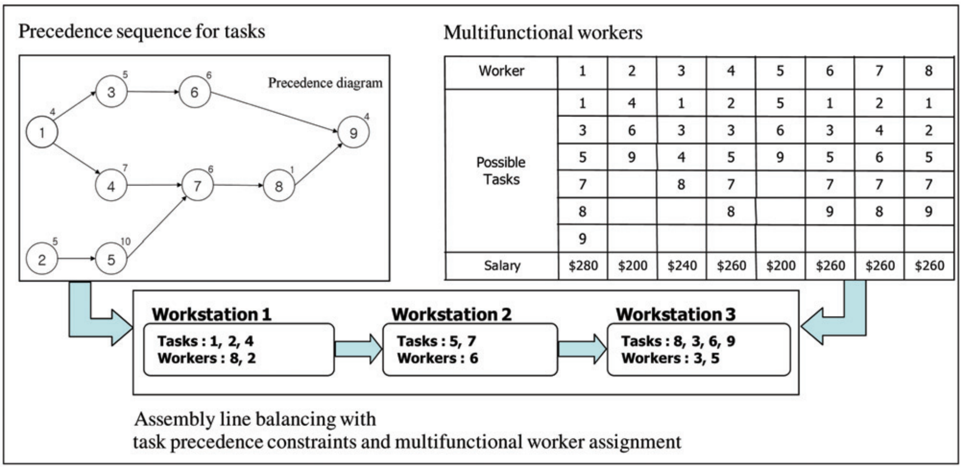


Figure 1. Optimising processes for the integrated ALB with resource restrictions.

costs and provide the operating sequence of the tasks, the work station index, and the assignment of multi-functional workers.

In the next section, the paper outlines the development of a mixed integer linear program to be used for the integrated ALB problem. The genetic algorithm is developed in Section 3. The paper then presents computational experiments for the developed algorithms in Section 4. The final section provides a number of brief concluding remarks.

2. Mathematical model

For the integrated assembly line balancing which has resource restrictions, the following assumptions were made.

(1) The workers are multi-functional with different salaries.
(2) The workers can be assigned to only one work station.
(3) The workers can be assigned to tasks depending on their skills.
(4) Multiple workers can be assigned to a single work station.
(5) The precedence constraints determine the sequence in which the tasks can be processed.

Given the previous problem statement, an initial approach used for solving the problem is to formulate the problem as a mixed integer linear program. The model used for solving the problem is explained below:

Notations

i Index of tasks ($i = 1, 2, \ldots, I$).
s Index of work stations ($s = 1, 2, \ldots, S$).
w Worker index ($w = 1, 2, \ldots, W$).
C Cycle time.
t_i Operating time for task i.
FC Annual operating cost of a work station.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

LC_w Annual salary of worker w.
$P_{(ij)}$ Set of pairs of tasks (i, j) where there is an immediate precedence relationship between them.
M Big M (a very large number).
A_w Set of available tasks that can be assigned to worker w.

Decision variables

F Number of work stations to be used in the assembly line.

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

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

Objective function

$$Min\, FC \cdot F + \sum_{w=1}^{W} LC_w \left(\sum_{s=1}^{s} Y_{sw} \right) \tag{1}$$

Subject to

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

$$\sum_{s=1}^{S} Y_{sw} \leq 1 \quad \forall w, \tag{3}$$

$$\sum_{s=1}^{S} \sum_{w=1}^{W} (s \cdot X_{isw} - s \cdot X_{jsw}) \leq 0 \quad \forall (i,j) \in P_{(ij)}, \tag{4}$$

$$\sum_{i=1}^{I} \sum_{w=1}^{W} t_i \cdot X_{isw} \leq C \quad \forall s, \tag{5}$$

$$\sum_{i=1}^{I} X_{isw} \leq M \cdot Y_{sw} \quad \forall s, w, \tag{6}$$

$$\sum_{i \notin A_w}^{i} \sum_{s=1}^{S} X_{isw} = 0 \quad \forall w, \tag{7}$$

$$\sum_{s=1}^{S} s \cdot X_{isw} \leq F \quad \forall i, w, \tag{8}$$

$$X_{isw}, Y_{sw} \in \{0, 1\} \tag{9}$$

The objective function (1) is to minimise the sum of the total annual work station cost and the annual salary of the workers. Constraints (2) ensure that each worker at each work station performs every task. Constraints (3) restrict one worker to be assigned to exactly one work station. Constraints (4) ensure the fulfilment of the precedence relationships between the precedence task sets $P_{(ij)}$. If task i is the immediate predecessor of task j, then it must be assigned with a higher operating index than task j. Constraints (5) express that the total operating time of each work station should not exceed the cycle time. Constraints (6) define the task assignment to a worker at the work station. In this constraint, Y_{sw} guarantees that if worker w is assigned to a work station, then X_{isw} can be any value. Constraints (7) ensure that a worker cannot be assigned to a work station with a task that the worker cannot perform according to the worker's available task set A_w. Constraints (8) determine the total number of work stations to be used. Constraints (9) express the binary nature of the two kinds of decision variables.

In addition, the integrated ALB problem is extended to an integrated ALB problem that has resource restrictions by considering the balance between a worker's maximum operation time and a worker's normal operation time. It is assumed that the total worker's operation time cannot be longer than the predetermined upper limit. This assumption considers the workload smoothness and the worker's operating balance. An additional notation is defined as follows:

C_w Predetermined maximum operation time for worker w.

The additional constraints for the worker's maximum operation time can be represented as follows:

$$\sum_{i=1}^{I} t_i \cdot X_{isw} \leq C_w \quad \forall s, w \tag{10}$$

The constraints (10) determine that the total operating time for each worker cannot be longer than the predetermined maximum operation time.

3. Genetic algorithm

Many optimisation problems within the industrial engineering field, particularly those of manufacturing systems, are very complex by nature and are considerably difficult to solve by using conventional optimisation techniques. Since the 1960s, there has been an increasing interest in imitating humans for solving such difficult optimisation problems. The simulation of the natural evolutionary process of human beings results in stochastic optimisation techniques called 'evolutionary algorithms'. When applied to difficult real-world problems, these evolutionary algorithms can often outperform conventional optimisation methods. Currently, there are four main avenues for this research: genetic algorithms (GA), genetic programming, evolutionary programming, and evolution strategies. Among these, the genetic algorithm is currently the most widely known type of evolutionary algorithm (Gen and Cheng 1997).

A comprehensive description of GAs can be found in Goldberg (1989). In a study by Rubinovitz and Levitin (1995), the application of GAs for a simple, single-model assembly line balancing (SALB) is described and the reasons for choosing this approach are discussed. One of the advantages of GAs with regard to the SALB problem is the ease in which they are able to handle different evaluation functions. Consequently, this approach

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

has been further explored by Kim *et al.* (1996) mainly to cope with the multiple objectives of an assembly line.

A GA, in contrast to conventional search techniques, begins with an initial set of random solutions called a 'population'. Each individual in the population is called a 'chromosome' which represents a solution to the problem. The chromosomes evolve through successive iterations called 'generations'. During each generation, the chromosomes are evaluated by recording a number of fitness measures. By combining parts of two or more parental chromosomes from the current generation, the offspring is formed. This is carried out by using the crossover operator and/or by modifying a chromosome using the mutation operator. A new generation is selected according to the fitness values of the parents and their offspring. Poor chromosomes are then weeded out in order to maintain a constant population size (Mitchell 1998). It is generally accepted that any GA must have the four basic components, namely representation and initialisation; objective and fitness function; reproduction, crossover, and mutation; and terminating conditions and parameters, in order to solve a problem, and that they must have different characteristics depending on the problem being studied. Figure 2 shows the structure of the GA used in this study.

The overall strategies used in this study for each of these components, including the chromosome style, are explained in the following subsections.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

3.1 *Representation and initialisation*

The choice of solution representation affects the method of transformation and evaluation. In this study, a chromosome set is used, which consists of two chromosome parts. The first chromosome part represents the sequence for the tasks and the work station assignment. The second chromosome part represents the assignment of multi-functional workers to each work station. The length of the sequence chromosome for the tasks is equal to the number of tasks. Each gene represents a sequence for a task operation. The length of the assignment chromosome is also equal to the number of tasks. Each gene represents the assignment of multi-functional workers to tasks. It should be noted that a worker could be assigned to more than one task at the same work station. Figure 3 shows an example of the nine tasks performed by the eight multi-functional workers. The sequence of the tasks is 1-2-4-5-7-8-3-6-9. Workers 8 and 2 at work station 1 perform tasks 1, 2, and 4. Worker 6 at work station 2 performs tasks 5 and 7. Workers 3 and 5 at work station 3 perform tasks 8, 3, 6, and 9.

Since the ALB problem in this study considers a large number of tasks and precedence restrictions and the solution algorithm should seek to assign the multi-functional workers to each task, the algorithm's efficiency is highly dependent on the initial solution. Therefore, by using a simple heuristic, a feasible initial solution is created. This heuristic consists of a sorting procedure based on task precedence restrictions and the worker's available task set. The steps for the simple heuristic are as follows:

Step 1: Arrange the tasks among the precedence restrictions.

Step 2: Calculate the operating time of the cumulative task by using the predetermined cycle time.

Step 3: Assign a worker to the tasks if the worker can be assigned such tasks. The low skilled worker (who has a lower index for possible tasks) will be assigned first.

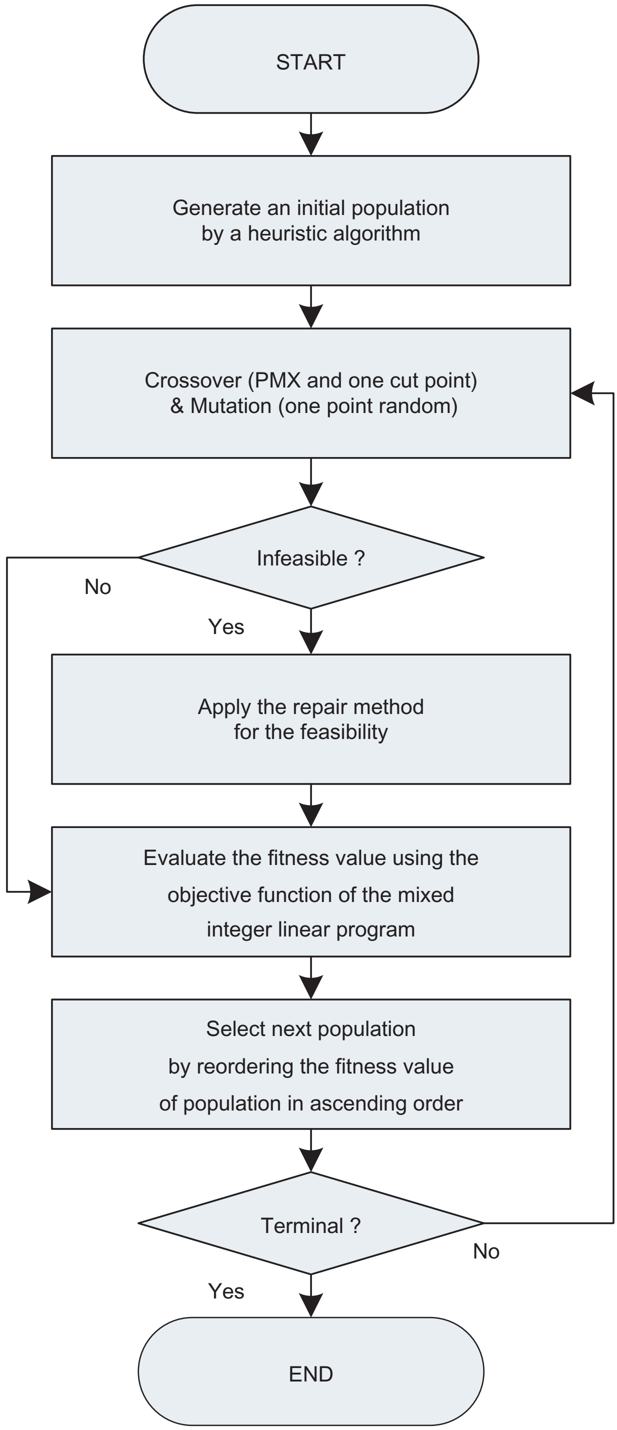
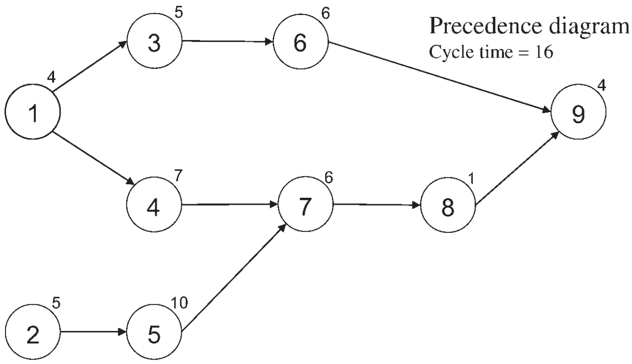


Figure 2. Flowchart of the proposed genetic algorithm.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009



Representation and Initialisation

	Workstation 1 (Cycle time = 16)			Workstation 2 (Cycle time = 16)		Workstation 3 (Cycle time = 16)			
Operating Sequence	1	2	3	4	5	6	7	8	9
Tasks	1	2	4	5	7	8	3	6	9
Workers	8	8	2	6	6	3	3	5	5

Figure 3. Representation and initialisation for a sample problem.

Step 4: Check the index of each worker at each work station. If the index of a worker overlaps with other work stations, go to Step 3. (The workers must be assigned to a single work station.) Otherwise, return the result to the GA.

3.2 *Objective and fitness function*

A fitness function is computed for each string in the population. The objective of this is to determine the string that has a minimum fitness function value. For a given string, the fitness function is computed for each chromosome in the population. The objective is to minimise the sum of the total annual work station costs and the annual salary of the workers function, which is the same objective function, Equation (1), as that of the mixed integer linear program in Section 2.

3.3 *Reproduction, crossover, and mutation*

A simple GA that yields sound results for many practical problems is composed of three operators. These are reproduction, crossover, and mutation. Reproduction is a process in which individual strings are copied according to their fitness function values. The reproduction procedure may create a new population for the subsequent generation based on the fitness scores. The reproduction operator may be implemented in a number of ways. This study produced offspring for the sequence chromosome part using a partially matched crossover (PMX) because each gene represents the sequences for tasks. The operation of this crossover has several stages. A random cut point is chosen in both parents P1 and P2. The segment to the right-hand side of the cut point, in both strings, functions as a partial mapping of the cells that are to be exchanged in

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

P1 in order to generate the offspring. In order to implement this crossover, after the selection of the cut point, a pair of cells is chosen in a predetermined location for both segments. This pair of cells is then exchanged in the first parent. This process is repeated for all the cells in the segment. Therefore, a cell in the segment of the first parent, and a cell at the same location in the second parent, will define which cells in the first parent need to be exchanged in order to generate the offspring. For the assignment chromosome part, the one-cut-point crossover was used in this study. Two parental chromosomes were selected for the crossover. A general mutation operator was used, which generates random numbers and replaces each gene. A random point from the range [1, (length of the chromosome – 1)] is generated for each gene in the chromosome. Subsequently, the gene is mutated at that point. A direct use of standard operators in the ALB problem can rarely ensure offspring feasibility. If it results in an infeasible offspring, we use a repair method to overcome this shortcoming. The procedure of the repair method is the same as that used for generating a feasible initial solution using a proposed simple heuristic, as outlined in Sub-section 3.1.

3.4 *Terminating condition and parameters*

The terminating condition was determined when 300,000 generations were reached or when the best individual did not improve more than 0.01% for 5,000 generations. For the parameters of the GA operators, the abovementioned crossover operator and mutation operator were employed with the parameters as listed in Table 1. The values of the parameters were decided after a pilot test.

4. Computational experiments

The mixed integer linear program was implemented and solved using LINGO on a Pentium-IV 3.0 GHz PC with 1 GB RAM on the Microsoft Windows XP operating system. As expected, this model requires an unacceptable amount of computational time in order to find an optimal solution. The mathematical model was compared with the GA for small sized problems in order to test the validity of the GA. The GA was implemented using Microsoft Office Excel 2003 and VBA (Visual Basic for Applications) on the same machine as that of the MILP. The precedence diagram shown in Figure 4 and Table 2 demonstrates the available operating task sets of the multi-functional workers and their salaries.

Tables 3 and 4 show a comparison of the results obtained from the mixed integer program and the GA. Both results have the same objective value. Due to the characteristics of the NP-hard problem, the computational time of the mixed integer

Table 1. Parameters of GA operators.

	Population size	Crossover probability	Mutation probability
Task chromosome part	100	0.5	0.3
Worker chromosome part	100	0.5	0.4

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

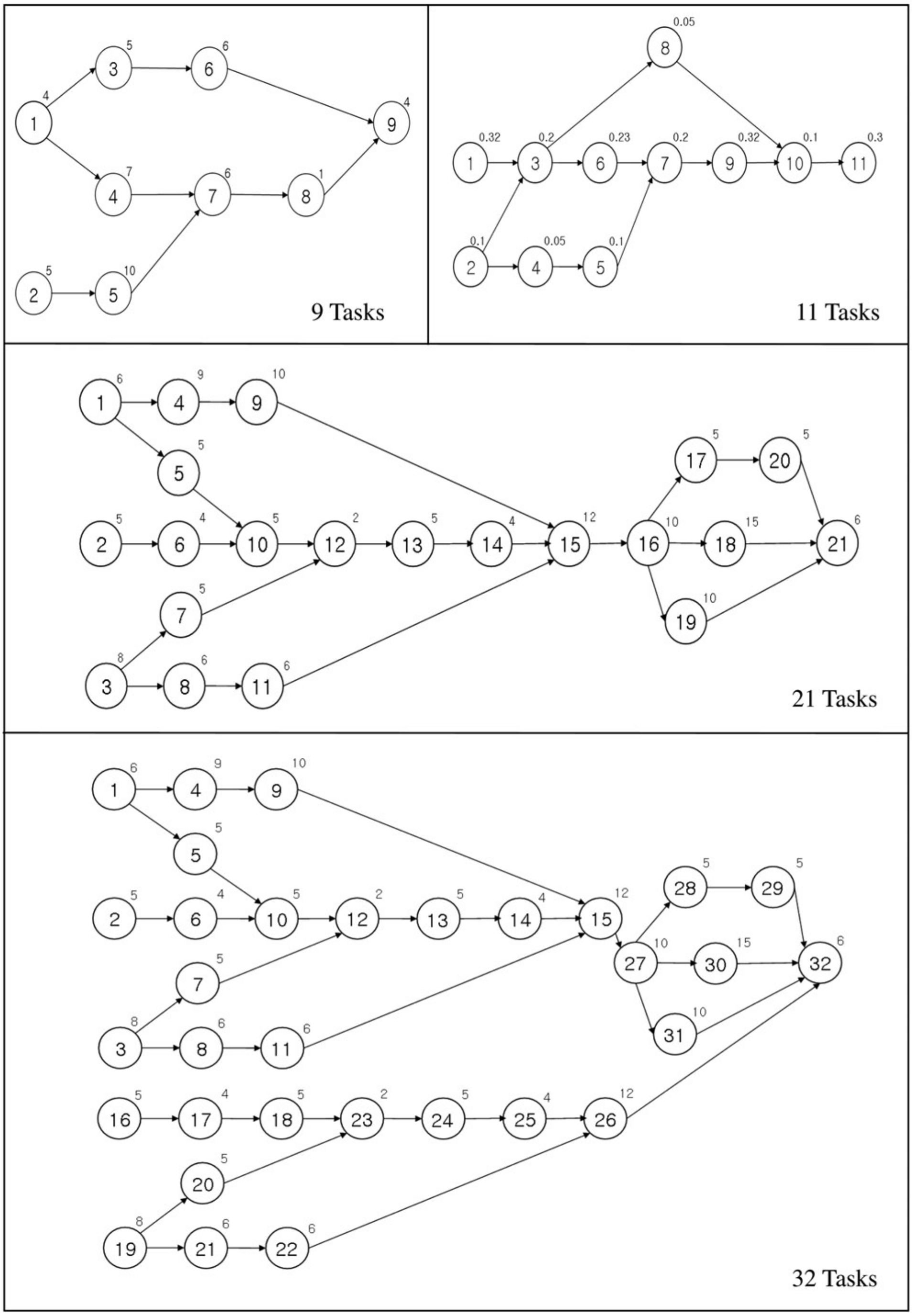


Figure 4. Precedence sequence diagrams.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Table 2. Multi-functional workers and their salaries (32 tasks and 15 workers).

Workers	Possible tasks						Salary
1	11	14	16	19	20	23	$23,000
2	2	6	13	17	21	26	$23,000
3	4	9	16	19	24		$20,000
4	3	7	8	13	30		$20,000
5	10	16	20	27	28	32	$23,000
6	15	17	18	20	25		$20,000
7	1	6	13	21	28		$20,000
8	21	24	25	30	32		$20,000
9	1	11	14	21	27	30	$23,000
10	8	17	24	29	30	31	$23,000
11	3	12	14	16	17		$20,000
12	12	16	22	28	31		$20,000
13	3	5	10	18	24	32	$23,000
14	3	13	22	26	32		$20,000
15	5	10	12	13	22	26	$23,000

Table 3. Comparison result of mathematical model and GA(1).

		Mathematical model		Genetic algorithm	
Examples	Predetermined cycle time	Assigned workstations	Assigned workers	Assigned workstations	Assigned workers
9 tasks and 8 workers	16 minutes	3	5	3	5
11 tasks and 8 workers	0.55 minutes	4	7	4	7
21 tasks and 15 workers	36 minutes	4	9	4	9
32 tasks and 15 workers	32 minutes	7	11	7	11

Table 4. Comparison result of mathematical model and GA(2).

	Mathematical model			Genetic algorithm		
Examples	Computation time*	Objective value	Remark	Computation time*	Objective value	Remark
9 tasks and 8 workers	10 minutes	$416,000	Optimal	19 seconds	$416,000	Optimal
11 tasks and 8 workers	25 minutes	$560,000	Optimal	27 seconds	$560,000	Optimal
21 tasks and 15 workers	2 hours 17 minutes	$748,000	Optimal	10 minutes	$748,000	Optimal
32 tasks and 15 workers	18 hours 55 minutes	$1,078,000	Optimal	15 minutes	$1,078,000	Optimal

Note: *Average of 10 evaluations.

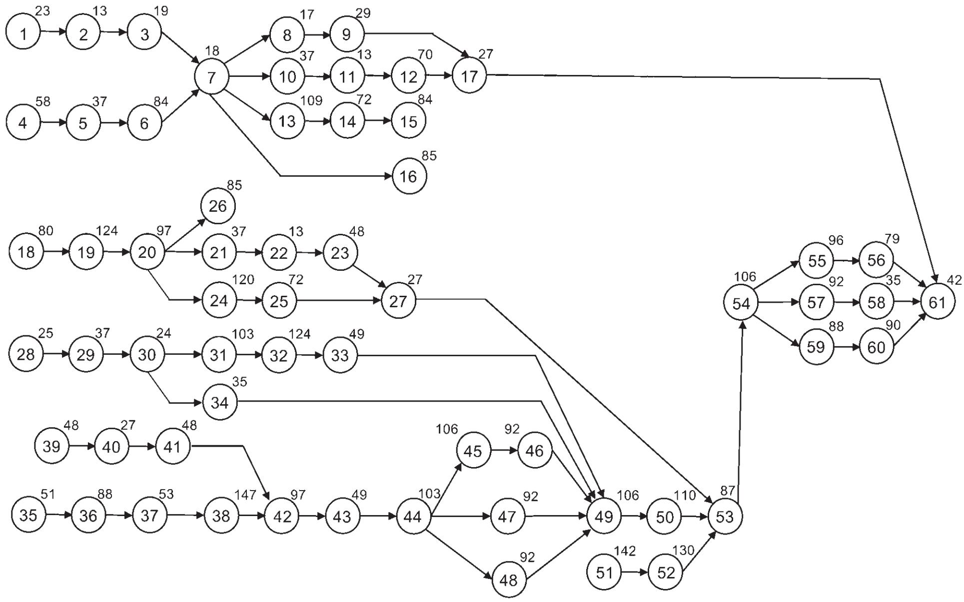


Figure 5. Precedence sequence diagram for 61 tasks.

Table 5. Multi-functional workers and their salaries.

Workers	Possible tasks												Salary
1	1	3	5	6	9	14	19	26	33	54			\$36,000
2	10	11	12	17	22	26	33	40	56	59			\$36,000
3	1	5	19	23	35	40	45	52	61				\$33,000
4	2	15	22	32	40	44	48	56	60				\$33,000
5	7	20	23	29	36	37	53	56	59				\$33,000
6	15	18	28	34	43	44	45	48	51	54			\$36,000
7	2	5	9	14	20	21	42	56	61				\$33,000
8	1	10	13	22	28	31	32	33	34	39	48		\$39,000
9	4	15	26	39	40	41	42	46	50	54	59		\$39,000
10	8	9	10	12	16	22	32	41	52				\$33,000
11	2	24	25	28	39	44	48	52	54	55	57	58	\$42,000
12	18	23	28	38	42	53	56	58	59	60			\$39,000
13	12	23	29	30	34	38	44	51	55				\$33,000
14	2	5	15	24	28	32	37	58	61				\$33,000
15	4	11	18	20	28	36	40	46	57				\$33,000
16	5	9	15	20	31	41	50	52	53	60			\$39,000
17	4	7	13	35	47	51	54	55	57	58			\$39,000
18	6	19	27	30	35	40	51	56	61				\$33,000
19	17	25	34	36	44	46	47	49	52				\$33,000
20	1	2	3	7	13	14	23	27	44	50	53		\$33,000

linear program increases exponentially as the number of tasks and the number of workers increase.

An experiment was carried out on a larger sized problem that had 61 tasks and 20 workers. The precedence diagram for 61 tasks is shown in Figure 5. A modified example

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Table 6. Result of the assignment of workers and tasks at the workstations.

Operating sequence	Index for task	Index for work station	Operating time	Cumulative operating time	Assigned worker
1	28	1	25	25	15
2	18		80	105	15
3	4		58	163	15
4	35		51	214	3
5	19		124	338	3
6	20	2	97	97	7
7	21		37	134	7
8	24		85	219	11
9	25		72	291	11
10	26	3	85	85	1
11	5		37	122	1
12	6		84	206	1
13	36		88	294	5
14	37		53	347	5
15	38	4	147	147	13
16	51		142	289	13
17	29		37	326	13
18	30		24	350	13
19	34	5	35	35	8
20	31		103	138	8
21	32		124	262	8
22	33		49	311	8
23	22		13	324	8
24	23	6	48	48	20
25	27		27	75	20
26	1		23	98	20
27	2		13	111	20
28	3		19	130	20
29	7		18	148	20
30	13		109	257	20
31	14		72	329	20
32	16	7	85	85	10
33	8		17	102	10
34	9		29	131	10
35	10		37	168	2
36	11		13	181	2
37	12		70	251	2
38	17		27	278	2
39	15	8	84	84	4
40	39		48	132	9
41	40		27	159	9
42	41		48	207	9
43	42		97	304	9
44	43	9	49	49	6
45	44		103	152	6
46	48		92	244	6
47	45		106	350	6
48	46	10	92	92	19
49	47		92	184	19
50	49		106	290	19

(*continued*)

Table 6. Continued.

Operating sequence	Index for task	Index for work station	Operating time	Cumulative operating time	Assigned worker
51	50	11	110	110	16
52	52		130	240	16
53	53		87	327	16
54	54	12	106	106	17
55	57		92	198	17
56	58		35	233	17
57	55		96	329	17
58	56	13	79	79	12
59	59		88	167	12
60	60		90	257	12
61	61		42	299	14

Table 7. Comparison result of Kim *et al.* (1991) and GA.

	Kim *et al.* (1991)	Genetic algorithm
Number of workstations	8	7
Balance efficiency* (%)	90.73%	95.52%

Note: *Balance efficiency = [total operating time/(cycle time * number of workstations)] * 100.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

was used of the large-sized truck assembly line that was taken from the problem identified in Kim *et al.* (1991). Table 5 shows the available operating task sets of the multi-functional workers and their salaries. The predetermined cycle time is 350 minutes. The annual operating cost for each work station is \$180,000/year. The result of this example is shown in Table 6. This assembly line is composed of 13 work stations and 19 workers. The sum of the total annual operating cost of a work station and the annual salary of a worker is \$3,020,000. The average computational time of 10 evaluations is 10 minutes.

Since this problem differs from that defined by Kim *et al.* (1991), a direct comparison is not possible. However, to overcome this limitation, the number of work stations and balance efficiency were compared with the same predetermined cycle time (600 minutes), as shown in Table 7.

Compared with the results documented by Kim *et al.* (1991), the number of work stations was reduced from eight to seven and the balance efficiency enhanced by 4.79%. Consequently, the proposed GA shows a significant improvement over that by Kim *et al.* (1991) since it simultaneously considers the assignments of tasks to work stations with workers to tasks. In addition, in the extended model for balancing workers' operation time, the examples were modified and 30 problems were solved. The mixed integer linear program (MILP) and the GA produced the same optimal value for all 30 problems (Table 8).

Table 8. Comparison result of mathematical model and GA.

Example		Result		
Number of tasks	Number of workers	Number of workstations	Objective value	Remark
9	11	4	$416,000	Optimal
9	13	4	$475,000	Optimal
9	14	4	$525,000	Optimal
9	15	4	$550,000	Optimal
9	18	5	$567,000	Optimal
11	16	4	$570,000	Optimal
11	16	4	$579,000	Optimal
11	17	4	$590,000	Optimal
11	17	4	$585,000	Optimal
11	18	4	$595,000	Optimal
11	19	5	$610,000	Optimal
15	19	4	$615000	Optimal
15	19	4	$620,000	Optimal
15	19	4	$630,000	Optimal
15	20	5	$650,000	Optimal
20	25	5	$715,000	Optimal
20	25	5	$726,000	Optimal
20	25	5	$730,000	Optimal
20	30	5	$770,000	Optimal
20	30	5	$775,000	Optimal
20	30	5	$780,000	Optimal
22	20	6	$815,000	Optimal
22	20	6	$820,000	Optimal
22	20	6	$817,000	Optimal
22	30	7	$920,000	Optimal
25	30	7	$816,000	Optimal
25	30	7	$830,000	Optimal
25	30	7	$835,000	Optimal
25	30	7	$850,000	Optimal
32	15	8	$1,175,000	Optimal

5. Conclusions

This paper reports the challenges and findings from a comprehensive research investigation aimed at solving the ALB problem in which multi-functional workers have different salaries depending on their skills. Two MILPs were formulated. Firstly, a MILP was developed for the integrated ALB for simultaneously determining the tasks sequence and workers assignment. Secondly, by considering the balancing of the workers' operation time, the MILP was extended. The GAs for the two proposed ALB problems were also developed. The GA found optimal solutions for the small and medium-sized test problems. The GA identifies an implementable solution more rapidly than the mathematical programming, thus solving complex, large problems very efficiently. The algorithm will help the designer to choose a suitable alternative for assembly line design and to be able to construct a flexible assembly line operating system with assigned multi-functional workers.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

Several further research directions are suggested. The first is to investigate how the GA operator finding methods affect the quality of the GA solution. The second is to compare the performance of the GA in this study with that of other meta-heuristics such as simulated annealing, ant colony optimisation, and tabu search. Finally, while the GA significantly reduced the computational time, it might be possible to incorporate mechanisms that can reduce this even further.

Acknowledgements

The authors are grateful to the careful and constructive reviews from anonymous referees. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (The Regional Research Universities Program/Research Center for Logistics Information Technology).

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009

References

Akagi, F., Osaki, H., and Kikuchi, S., 1983. A method for assembly line balancing with more than one worker in each station. *International Journal of Production Research*, 21, 755–770.

Becker, C. and Scholl, A., 2006. A survey on problems and methods in generalised assembly line balancing. *European Journal of Operational Research*, 168, 694–715.

Bukchin, Y. and Rabinowitch, I., 2006. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimising stations and task duplication costs. *European Journal of Operational Research*, 174, 492–508.

Dimitriadis, S., 2006. Assembly line balancing and group working: a heuristic procedure for workers'group operating on the same product and work station. *Computers & Operations Research*, 33, 2757–2774.

Faaland, B., *et al.*, 1992. Assembly line balancing with resource dependent task times. *Decision Sciences*, 23, 343–364.

Gen, M. and Cheng, R., 1997. *Genetic algorithm and engineering design*. New York: Wiley.

Goldberg, D., 1989. *Genetic algorithm in search, optimisation and machine learning*. New York: Addison-Wesley.

Kim, Y., Kim, Y., and Kim, Y., 1996. Genetic algorithms for assembly line balancing with various objectives. *Computers and Industrial Engineering*, 30, 397–409.

Kim, Y., Kwon, S., and Cho, M., 1991. A heuristic method for assembly line balancing of large-sized product. *Journal of the Korean Institute of Industrial Engineers*, 17, 51–62.

Kim, Y., Lee, T., and Shin, T., 1998. Two-sided assembly line balancing using a branch-and-bound method. *Journal of the Korean Institute of Industrial Engineers*, 24, 417–430.

Levitin, G., Rubinovitz, J., and Shnits, B., 2006. A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research*, 168, 811–825.

Lutz, C., Davis, K., and Turner, C., 1994. Development of operator assignment schedules: a DSS approach. *Omega*, 22, 57–67.

McMullen, P.R. and Tarasewich, P., 2006. Multi-objective assembly line balancing via a modified ant colony optimisation technique. *International Journal of Production Research*, 44, 27–42.

Mitchell, M., 1998. *Introduction to genetic algorithms*. England: MIT Press.

Nearchou, A.C., 2007. Multi-objective balancing of assembly lines by population heuristics. *International Journal of Production Research*, 46, 2275–2297.

Rubinovitz, J. and Levitin, G., 1995. Genetic algorithm for assembly line balancing. *International Journal of Production Economics*, 41, 343–354.

Vilarinho, P.M. and Simaria, A.S., 2006. ANTBAL: an ant colony optimisation algorithm for balancing mixed-model assembly lines with parallel work stations. *International Journal of Production Research*, 44, 291–303.

Wee, T. and Magazine, M., 1982. Assembly line balancing as generalised bin packing. *Operations Research Letters*, 1, 56–58.

Wilson, J., 1986. Formulation of a problem involving assembly lines with multiple manning of work stations. *International Journal of Production Research*, 24, 59–63.

Downloaded By: [Moon, Ilkyeong][Pusan National University Library] At: 00:39 24 July 2009