periodica polytechnica

Social and Management Sciences
19/1 (2011) 43–50
doi: 10.3311/pp.so.2011-1.06
web: http://www.pp.bme.hu/so

RESEARCH ARTICLE

Formulation of simple workforce skill constraints in assembly line balancing models

Tamás Koltai / *Viola* Tatay

Received 2011-01-07

Abstract

The objective of assembly line balancing (ALB) is to minimize the number of workstations organized to perform tasks with precedence constraints. An important element of proper assignment of tasks to workstations is the consideration of skill requirement of tasks, and skill level of workers. Some tasks may require special skills, some workers might not be able to perform complicated tasks, and some workers might be specialized for specific tasks.

This paper provides a general framework to model skill requirements and skill conditions for assembly line balancing models. Three types of skill constraints are defined. Low skill constraints determine workstations for workers who are able to perform only some simple tasks. High skill constraints consider tasks which require higher than average skills of workers. Finally, exclusive skill constraints consider situations where a group of workers is specialized in a subset of tasks.

The paper summarizes the mathematical description of the different skill constraints, and shows how simple assembly line balancing models can be completed with skill considerations. The mathematical characteristics of the resulting models are discussed, and some sample problems are solved to illustrate the results of each specific skill situation.

Keywords

Assembly line balancing · worker skill · production management · mathematical programming

Tamás Koltai

Dept of Management and Corporate Economics, BME, 1117 Magyar tudósok körútja 2. Budapest, Hungary
e-mail: koltai@mvt.bme.hu

Viola Tatay

Dept of Management and Corporate Economics, BME, 1117 Magyar tudósok körútja 2. Budapest, Hungary
e-mail: tatay@mvt.bme.hu

1 Introduction

Assembly line balancing (ALB) problems occur where several indivisible work elements (tasks) are to be grouped into (work)stations along a continuous production line. Workers may work at each station, and in case of efficient allocation of tasks to workstations, the number of workers and consequently the cost of operation can be decreased. ALB problems are typical for example in automobile, in bicycle and in electronic industry (assembling refrigerators, televisions, etc.), but the operation of some service systems is also very similar to assembly line operations [3, 10].

Tasks cannot be allocated to the stations arbitrarily. Cycle time constraints, precedence relations – generally visualized by a precedence graph –, zoning conditions, technological and logical requirements may influence the optimal assignment. Even considering these restrictions many feasible solutions may exist for the allocation of tasks to stations and optimization models can be used to find the best task assignment. The assembly line problem is NP hard; therefore, a considerable amount of research effort is made to reduce the computation work. As a consequence of the development of computer and information technology, however, even large ALB problems can be solved in a reasonable time frame.

Early research in this area focused on the simple assembly line balancing problem (SALBP) with its restrictive characteristics such as deterministic task times, no assignment restrictions other than the precedence constraints, serial line layout, etc [2, 10]. Extended forms of the SALBP consider for example the possibility of U-shaped lines, parallel stations, and stochastic task times. These models are referred in the literature as general assembly line balancing problems (GALBP). GALBPs may be closer to practical problems; however, their solution procedures, in most cases, are based on SALBP algorithms [10]. Depending on the management objective of assembly line balancing the two most frequently used versions of SALBPs are the following,

– When management objective is related to operating cost reduction the ALB model minimizes the number of workstations (workers) for a given cycle time. The related problems are referred in the literature as SALBP-1.

– When management objective is related to production quantity the ALB model minimizes the cycle time for a given number of workstation. The related problems are referred in the literature as SALBP-2.

SALBPs can be formulated as mathematical programming models. The first analytical formulation of ALB was given by Bryton [5] and the first linear programming problem that might have infeasible solutions because of split tasks was given by Salveson [9]. Bowman was the first to suggest integer programming (IP) models to solve the classical ALB problem [4]. White modified Bowman's IP model and defined 0-1 decision variables for the problem [12]. Since ALB models are NP hard the research focused on reducing the number of variables and constraints in order to reduce the complexity of the models (see for example [1, 10, 11]).

Today mathematical programming models of practical size ALB problems can be solved by optimization software very efficiently. Therefore, the focus of research should be shifted to practice driven model formulation and to the investigation of new areas of application [3]. In this paper we try to make the ALB problem more attractive to practical applications by showing how skill constraints of workers can be considered.

There are very few papers which are dedicated to the consideration of workforce skill in assembly line modeling. Most papers add some work force skill constraints when a special practical case is solved. For example, Corominas et al. [6] model temporary and permanent workers in a motor-cycle assembly process, and these two worker groups are able to perform different set of tasks. Johnson [7] considers some very simple skill constraints in a paper dedicated mostly to some mathematical questions of the optimization process. There is no paper, however, which generalizes skill constraints and provides method to formulate skill conditions routinely for practical situations

The paper is structured as follows. First, formulation of the basic ALB models used in the paper is provided. Next, skill constraints are generalized and the mathematical description of the different simple skill conditions is given. The analyses suggested in the paper are illustrated with a small sample problem, and the effect of the different skill constraint types is analyzed.

2 Formulation of the basic ALB problems

Notation used in the paper is summarized in Tab. 1. Tasks are numbered in increasing order. The number i assigned to a task is called the task index. We refer to a task either by its name or by its task index. The index set of all tasks is denoted by O. Those tasks which are not succeeded by any other task are called *last* tasks. The index set of last tasks is denoted by L.

Workstations are also numbered in increasing order. The first workstation is numbered 1 and the last workstation is numbered N. The number j assigned to a workstation is called the workstation index. Workstations are referred in the paper by the workstation index. An assumption must be made about the possible number of stations prior to task assignment. The number of stations used in the model is J. That is, J is the number of stations used in the mathematical model, and N is the number of stations used in the actual line.

In this paper we use the following integer linear programming formulation of SALBP-1,

$$\min N; \tag{1}$$

$$\sum_{i=1}^{I} t_i x_{ij} \leq T_c, \quad j = 1, \ldots, J; \tag{2}$$

$$\sum_{j=1}^{J} x_{ij} = 1, \quad i = 1, \ldots, I; \tag{3}$$

$$\sum_{j=1}^{J} j \cdot \left(x_{qj} - x_{pj}\right) \geq 0, \quad (p, q) \in R; \tag{4}$$

$$N \geq \sum_{j=1}^{J} \left(j \cdot x_{ij}\right), \quad i \in L; \tag{5}$$

$$x_{ij} = 0, \quad j < LJ_i \text{ and } j > UJ_i, \quad i = 1, \ldots, I. \tag{6}$$

The objective of the model is to minimize the number of stations used in the actual system; that is, to minimize the largest index belonging to a station with task assignment. The right-hand-side of constraint (5) determines the index of those workstations which perform last tasks. The highest such index must be minimized. If each of these indices is smaller than or equal to N, and $N$is minimized, then the index of the final workstation, and consequently the number of workstations, is minimized.

Cycle time constraints are expressed by constraints (2). For each workstation the sum of task times of the assigned tasks is not allowed to exceed the cycle time. As a consequence of constraints (3) each task is assigned to one of the workstations. Constraints (4) express the precedence constraints. If task p must be performed before task q, the difference in the bracket is equal to -1, 0 or 1 for each workstation. Since task $p$must be assigned to an earlier or to the same workstation as task q; the weighted sum of these differences is always greater than or equal to 0, if the weights are the indices of the corresponding workstations.

Finally the number of variables is reduced by constraints (6). Some tasks can not be assigned to very *early* workstations because of preceding tasks. The earliest workstations which can be used by task i is determined by LJ_i. LJ_i is a lower limit of the feasible station indices of task i, and its value is calculated as follows,

$$LJ_i = \left\lceil \frac{t_i + \sum_{k \in P_i} t_k}{T_c} \right\rceil, \tag{7}$$

where $\lceil x \rceil$ is the smallest integer value not smaller than x.

Tab. 1. Summary of notation

Indices:
i = index of tasks ($i = 1, \dots, I$),
k = index of tasks in a subset of tasks,
j = index of workstations ($j = 1, \dots, J$).
Parameters:
I = number of tasks,
J = number of workstations in the mathematical model,
N = number of workstations applied,
R = set of pair of indices which belong to tasks with precedence relations, that is, $(p, q) \in R$, if task p immediate precedes task q,
t_i = time necessary to perform task i (task time),
s_j = time necessary to perform all tasks at station j (station time),
T_c = cycle time of the assembly line,
$UB(T_c)$ = estimated upper bound of the cycle time,
T = total available time for production,
LJ_i = the earliest workstation which can be used as a consequence of preceding tasks of task i,
UJ_i = the latest workstations which can be used by task i as a consequence of succeeding tasks of task i,
Q = production quantity,
c_j = capacity utilization of station j,
W = limit on special workers,
z = sufficiently high number.
Sets:
O = index set of all tasks,
L = set of final tasks, that is, $i \in L$ if task i does not precede any other tasks,
P_i = index set of those tasks which must be finished before task i is started,
S_i = index set of those tasks which can not be started before task i is finished,
S = index set of special tasks,
$\overline{S}$ = index set of non-special tasks (complementary set of S).
Decision variables:
$x_{ij} = 0-1$ decision variable; if $x_{ij} = 1$, then task i is assigned to station j, otherwise $x_{ij} = 0$,
$l_j = 0-1$ decision variable; if $l_j = 1$, then low skilled worker is applied at workstation j, otherwise $l_j = 0$,
$h_j = 0-1$ decision variable; if $h_j = 1$, then high skilled worker is applied at workstation j, otherwise $h_j = 0$,
$e_j = 0-1$ decision variable; if $e_j = 1$, then special worker is applied at workstation j, otherwise $e_j = 0$.

Some tasks can not be assigned to very *late* workstations because of succeeding tasks. The latest workstation which can be used by task i is determined by UJ_i. UJ_i is an upper limit of the feasible station indices of task i, and its value is calculated as follows,

$$UJ_i = J + 1 - \left\lceil \frac{t_i + \sum_{k \in S_i} t_k}{T_c} \right\rceil. \tag{8}$$

(1)-(8) is a linear programming model with one integer and several 0-1 variables. The required number of variables can be determined using the LJ_i and UJ_i value with the following formula,

$$\sum_{j=1}^{J} (UJ_i + 1 - LJ_i). \tag{9}$$

We note that model (1)-(8) is slightly different from the models used in the literature. Most models formulate the problem for a single last task, that is, only one index belongs to L (see for example [11]). If several final tasks exist (see the following sample problem) then a dummy task is used which directly succeeds the real final tasks. This dummy task increases the number of 0-1 variables; because in that case $I+1$ task must be assigned to J workstations. In formulation (1) to (8), however, instead of the dummy task, the index of the final workstation is used. This way only one new integer variable (variable N) is required.

SALBP-2 minimizes the cycle time for a given number of workstations (N), that is, the objective function is as follows,

$$\min T_c. \tag{10}$$

Cycle time constraint (2), constraints for the performance of each operation (3) and precedence constraints (4) are the same as in SALBP-1. That is, SALBP-2 is determined by objective function (10) and constraints (2)-(4). The limit on the number of variables in this case can be determined by using an estimate of the upper bound of the cycle time ($UB(T_c)$).

The earliest workstations which can be used by task i is now the following,

$$LJ_i = \left\lceil \frac{t_i + \sum_{k \in P_i} t_k}{UB(T_c)} \right\rceil. \tag{11}$$

The latest workstation which can be used by task i is now calculated as follows,

$$UJ_i = J + 1 - \left\lceil \frac{t_i + \sum\limits_{k \in S_i} t_k}{UB(T_c)} \right\rceil. \qquad (12)$$

Consequently SALBM-1 is defined by constraints (1)-(8) and SALBM-2 is defined by constraints (2)-(4), (6) and (10)-(12). These models are summarized in the first row of Table 2.

In the following sections the basic SALBP-1 and SALBP-2 models will be completed with constraints expressing work force skill requirements.

3 Consideration of workforce skill

Frequently, a set of tasks performed at an assembly line require special skills of workers, and a set of workers working at an assembly line may have special or limited skills. This must be considered when tasks are assigned to workstations. It is assumed, that S is the index set of special tasks. This subset of tasks has special characteristics. Those set of tasks, which does not have this characteristics is the complementary set of special tasks. The index set of this complementary set of tasks is denoted by $\overline{S}$. The index set of all tasks and the index set of special tasks relates as follows,

$$S \cup \overline{S} = O.$$

In this section, three types of worker skill requirement will be examined.

- *A limited number of low skilled workers* must be applied at the assembly line. We call the constraints describing this situation low skill constraints (LSC). It is assumed, that two types of workers are applied. Regular workers are able to perform any operations, and low skilled workers are able to perform only the special operations. In this case, set S contains the index set of those tasks which can be performed by low skilled workers.
- *A limited number of tasks require high skilled workers.* We call the constraints describing this situation high skill constraints (HSC). It is assumed, that two types of workers are applied. High skilled workers are able to perform any tasks, but regular workers are not able to perform special tasks. In this case, set S contains the index set of those tasks which can be performed only by high skilled workers.
- *A limited number of tasks can be performed only by special workers.* We call the constraints describing this situation exclusive skill constraints (ESC). It is assumed, that two types of workers are applied. Special workers are able to perform only the special tasks, and regular workers are able to perform only the regular tasks. In this case, set S contains the index set of those tasks which require special skills.

3.1 A limited number of low skilled workers must be applied at the assembly line (LSC)

This case is found in practice when there are simple tasks, which do not require any advanced knowledge, and can be performed by any worker. This set of tasks is called special tasks and they belong to set S. The rest of the tasks are regular tasks and belong to $\overline{S}$. It is assumed, that a limited number of low skilled workers are already employed, therefore workstations for them must be organized. Assembly line balancing must consider the two following conditions when tasks are assigned to workstations in this case:

1 Only those tasks can be assigned to the workstations of low skilled workers which are elements of S.

2 Any task can be assigned to the workstations of regular workers.

Let binary variable l_j indicate which workstation applies low skilled workers, that is

$$l_j = \begin{cases} 1 & \text{if low skilled worker is assigned to workstation j,} \\ 0 & \text{otherwise.} \end{cases}$$

If l_j is equal to 1, then a low skilled worker is assigned to workstation j. In this case only those tasks can be assigned to workstation j which can be performed by low skilled workers, that is, those tasks, which are elements of S. If l_j is equal to 0, then workstation j does not have low skilled workers. Any task can be assigned to this workstation, because regular workers can perform any of the tasks. Conditions for this case are expressed as follows,

$$\sum_{i \in \overline{S}} x_{ij} \le z(1 - l_j), \quad j = 1, \ldots, J, \qquad (13)$$

where z is a sufficiently high number (higher than J). If workstation j has any regular task, then the left hand side of (13) is greater than 0, and consequently the right hand side must be higher than 0 as well. This is possible only if l_j=0, that is, low skilled workers can not be applied at workstation j. If workstation j has only special tasks, then the left hand side of (13) is equal to 0, and consequently l_j can be equal to 0 or 1. In this case low skilled workers and regular workers can be assigned to workstation j as well. Constraints (13) do not determine which workstation with special tasks will have low skilled workers; only exclude the application of low skilled workers from workstations with regular tasks.

According to (13) l_j can be equal to 1 even if tasks are not assigned to workstation j at all. For example, in SALBM-1 J workstations are assumed when the model is set up, and finally N workstations are applied according to the optimal solution. Consequently, J-N workstations are not applied, and therefore tasks are not assigned to them. In SALBM-2, the minimal cycle time can be obtained by using less than the given number of workstations (N), consequently workstations without tasks may also occur. The following condition excludes this illogical situation,

$$\sum_{i=1}^{n} x_{ij} \geq l_j, \quad j = 1, \ldots, J. \tag{14}$$

According to (14) if tasks are not assigned to workstation j then l_j is equal to 0, and consequently low skilled workers can not be applied.

Finally a given number (W) of low skilled workers must be applied at the assembly line, that is,

$$\sum_{j=1}^{m} l_j \geq W. \tag{15}$$

3.2 A limited number of tasks require high skilled workers (HSC)

This case is found in practice when there are complicated tasks, which require special skills, and can be performed by special, qualified workers. Those tasks which require special skills belong to S. The rest of the tasks do not require special skill and/or special qualification of the workers. These regular tasks belong to $\overline{S}$. It is assumed, that there is an upper limit on the skilled workers. Assembly line balancing must consider the two following conditions when tasks are assigned to workstations in this case:

1 Any task can be assigned to the workstations of high skilled workers.

2 Only those tasks can be assigned to the workstations of regular workers, which are not elements of S.

Low skill constraints (LSC) can easily be transformed into high skill constraints (HSC). Special workers in HSC can be considered as regular workers in LSC. There is a lower limit on the special workers in LSC, which can be substituted by an upper limit on the regular workers. This upper limit is the limit on the high skilled workers in HSC.

Let 0-1 variable h_j indicate which workstation applies high skilled workers, that is

$$h_j = \begin{cases} 1 & \text{if high skilled worker is assigned to workstation j,} \\ 0 & \text{otherwise.} \end{cases}$$

If tasks belonging to S are assigned to workstation j, then special worker must be assigned to it, therefore h_j must be equal to 1. Workstations without tasks from set S can have special or regular workers. This necessarily mean, that a workstation j with h_j=0 can only have tasks belonging to set $\overline{S}$. Conditions for this case are expressed as follows,

$$\sum_{i \in S} x_{ij} \leq z h_j, \quad j = 1, \ldots, J. \tag{16}$$

where, z is a sufficiently high number. If workstation j has any special task, then the left hand side of (16) is higher than 0, and consequently the right hand side must be higher than 0 as well. This is possible only if h_j=1, that is, special worker is required. If workstation j has only regular tasks, then the left hand side of (16) is equal to 0, and consequently h_j can be equal to 0 or 1. In this case high skilled workers or regular workers can be assigned to workstation j. Constraints (16) do not determine which workstation with only regular tasks will have regular workers; only exclude the application of regular workers from workstations with special tasks.

According to (16) h_j can be equal to 1 even if tasks are not assigned to workstation j at all. For example, in SALBM-1 J workstations are assumed when the model is set up, and finally N workstations are applied according to the optimal solution. Consequently, J-N workstations are not applied, and therefore tasks are not assigned to them. In SALBM-2, the minimal cycle time can be obtained by using less than the given number of workstations (N), consequently workstations without tasks may also occur. The following conditions exclude this illogical situation,

$$\sum_{i=1}^{n} x_{ij} \geq h_j, \quad j = 1, \ldots, J. \tag{17}$$

According to (18) if tasks are not assigned to workstation j then h_j is equal to 0, and consequently low skilled workers can not be applied.

Limited number of high skilled workers is available at the assembly line, that is,

$$\sum_{j=1}^{m} h_j \leq W. \tag{18}$$

3.3 A limited number of tasks can be performed only by special workers (ESC)

This case is found in practice when there are special tasks, which require special qualification of workers. The workers with the required qualifications can only perform these special tasks. The tasks which require special skills belong to set S. The rest of the tasks do not require special skill and/or special qualification of the workers, however special workers do not perform these tasks. These regular tasks belong to $\overline{S}$. Assembly line balancing must consider the two following conditions when tasks are assigned to workstations in this case:

1 Only special tasks can be assigned to the workstations of special workers.

2 Only regular tasks can be assigned to the workstations of regular workers.

Let binary variable e_j indicate which workstation applies special workers, that is

$$e_j = \begin{cases} 1 & \text{if special worker is assigned to workstation j,} \\ 0 & \text{otherwise.} \end{cases}$$

Special and regular tasks can not be mixed on workstations, and we have to know which workstation requires special workers and which workstation requires regular workers.

If $e_j = 0$, then special worker can not be assigned to workstation j, that is,

$$\sum_{i \in S} x_{ij} \leq z e_j, \quad j = 1, \ldots, J. \tag{19}$$

If $e_j = 0$, than the right hand side of (19) is equal to 0, and consequently the left hand side is also equal to 0. That is, each x_{ij} for $i \in S$ is equal to 0 at the corresponding workstation (j) and consequently, special tasks are not assigned to workstation j.

If $e_j = 1$, than regular worker can not be assigned to workstation j, that is,

$$\sum_{i \in \overline{S}} x_{ij} \leq z(1 - e_j), \quad j = 1, \ldots, J. \tag{20}$$

If $e_j = 1$, than the right hand side of (20) is equal to 0, and consequently the left hand side is also equal to 0. That is, each x_{ij} for $i \in \overline{S}$ is equal to 0 at the corresponding workstation (j) and consequently, regular tasks are not assigned to workstation j.

If (19) and (20) are simultaneously satisfied, then the different groups of tasks are separated on the workstations, and the proper worker skill is applied.

3.4 Summary of the suggested worker skill models

Table 2 summarizes simple assembly line balancing models and the corresponding worker skill constraints. The basic models are presented in the first row of the table. SALBM-1 is an integer linear programming model and it is given in the first column. SALBM-2 is a 0-1 linear programming model and it is given in the second column. Note, that if there is only a single final task in SALBM-1, than the right hand side of (5) can be directly minimized, and consequently there is no need for integer variable N. In this case SALBM-1 is also a 0-1 linear programming model. The workstation index limits (LJ_i and UJ_i) are calculated with the (7), (8) and (11), (12) expressions respectively.

For SALBM-1, and SALBM-2 the corresponding worker skill constraints are given in the LSC, HSC and ESC rows. Note that the set of special tasks (S) must be defined differently for each constraint type. Set S contains simple tasks for low skilled workers in LSC, contains complicated tasks for high skilled workers in HSC, and finally contains tasks which can be performed exclusively by specialized workers in ESC.

Based on the skill constraints defined in Tab. 2 several skill conditions can be combined in the same model. In practice it is possible, that there are low skilled workers, highly qualified workers, and specialized workers as well, and task assignment to workstations must consider all these three conditions. In this case all three constraint types must be added to the basic assembly line balancing model, and l_j, h_j and e_j are decision variables of the same model. Furthermore S must be defined for each skill constraints.

In practice, frequently complicated combination of the basic skill conditions can be found. For example several skill levels in hierarchically increasing and/or decreasing orders or several set of tasks with exclusive specialist may exist. In these cases a skill level index must be added to set S and to the l_j, h_j or e_j variables, and the set of special tasks must be defined accordingly. In this paper we do not discuss in details these complicated cases. That is the reason of the application of the term "simple skill conditions" in the title of the paper.

Finally, in Tab. 2 skill constraints are added to SALBM-1 and to SALBM-2, that is the number of workstation (line utilization) or the cycle time is minimized. The proposed models can easily incorporate other objective functions which express the different labor cost of differently skilled workers.

4 Calculation results

To illustrate the performance of the models in Tab. 2 let us consider the example of Bowman [4]. Fig. 1 shows the precedence relations of eight tasks with the corresponding task indices and task times. It can be seen that two final tasks exist, that is, $7 \in L$ and $8 \in L$.

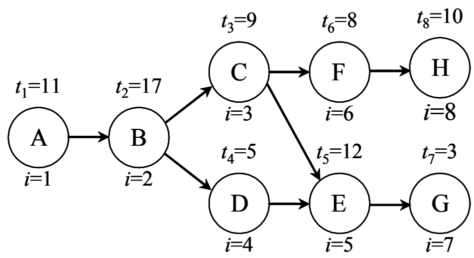

Fig. 1. Precedence diagram of the sample problem

In case of SALBP-1 the number of workstations must be minimized. It is assumed that the line works in one seven hour shift, that is, the total time available for production (T) is equal to 25 200 seconds ($7 \cdot 60 \cdot 60$). The required daily production quantity (Q) is 1260 units, thus the cycle time is equal to 20 seconds ($7 \cdot 60 \cdot 60/1260$).

The *maximum* number of workstations necessary for production is equal to the number of tasks ($J = 8$). In this *worst case* each task is performed at different workstations, and 64 binary variables (x_{ij}) and one integer variable (N) is required. However, by calculating the LJ_i and UJ_i values for $T_c = 20$ seconds, the number of binary variables can be reduced to 45.

The optimal solution of this sample problem and the effect of several skill constraints on the solution are summarized it Table 3.

The optimum value of the objective function of the SALBM-1 (first model in Table 3) shows that the minimum number of workstations necessary to process the required quantity is 5. The optimal values of the x_{ij} binary variables determine the optimal task assignment. The "Optimal task assignment" section of Table 3 is the translation of the optimal values of the binary decision variables x_{ij} into task assignment. The SALBM-1 row

Tab. 2. Summary of ALB models and constraints

	Minimization of workstation number (SALBM-1)		Minimization of cycle time (SALBM-2)	
Basic models	$\min N$	$\min T_c$		
	$\sum_{i=1}^{I} t_i x_{ij} \leq T_c,$	$j = 1, \ldots, J$	$\sum_{i=1}^{I} t_i x_{ij} \leq T_c,$	$j = 1, \ldots, J$
	$\sum_{j=1}^{J} x_{ij} = 1,$	$i = 1, \ldots, I$	$\sum_{j=1}^{J} x_{ij} = 1,$	$i = 1, \ldots, I$
	$\sum_{j=1}^{J} j(x_{qj} - x_{pj}) \geq 0,$	$(p, q) \in R$	$\sum_{j=1}^{J} j(x_{qj} - x_{pj}) \geq 0,$	$(p, q) \in R$
	$N \geq \sum_{j=1}^{J} (j x_{ij}),$	$i \in L$		
	$x_{ij} = 0,$	$j < LJ_i,$	$x_{ij} = 0,$	$j < LJ_i,$
		$j > UJ_i,$		$j > UJ_i,$
		$i = 1, \ldots, I$		$i = 1, \ldots, I$
Low skill constraints (LSC)	$\sum_{i \in \overline{S}} x_{ij} \leq z(1 - l_j),$	$j = 1, \ldots, J$	$\sum_{i \in \overline{S}} x_{ij} \leq z(1 - l_j),$	$j = 1, \ldots, J$
	$\sum_{i=1}^{n} x_{ij} \geq l_j, \quad j = 1, \ldots, J$	$\sum_{i=1}^{n} x_{ij} \geq l_j,$	$j = 1, \ldots, J$	
	$\sum_{j=1}^{m} l_j \geq W$	$\sum_{j=1}^{m} l_j \geq W$		
High skill constraints (HSC)	$\sum_{i \in S} x_{ij} \leq z h_j,$	$j = 1, \ldots, J$	$\sum_{i \in S} x_{ij} \leq z h_j,$	$j = 1, \ldots, J$
	$\sum_{i=1}^{n} x_{ij} \geq h_j,$	$j = 1, \ldots, J$	$\sum_{i=1}^{n} x_{ij} \geq h_j,$	$j = 1, \ldots, J$
	$\sum_{j=1}^{m} h_j \leq W$	$\sum_{j=1}^{m} h_j \leq W$		
Exclusive skill constraints (ESC)	$\sum_{i \in S} x_{ij} \leq z e_j,$	$j = 1, \ldots, J$	$\sum_{i \in S} x_{ij} \leq z e_j,$	$j = 1, \ldots, J$
	$\sum_{i \in \overline{S}} x_{ij} \leq z(1 - e_j),$	$j = 1, \ldots, J$	$\sum_{i \in \overline{S}} x_{ij} \leq z(1 - e_j),$	$j = 1, \ldots, J$

Tab. 3. Summary of calculation results

	Models	T_c	N	W	Set S	Optimal task assignment $j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
1.	SALBM-1	20	5	-	-	A	B	C,F	H	D,E,G	-
2.	SALBM-1+LSC	20	5	1	C,G	A	B	*C*	D,E,G	F,H	-
3.	SALBM-1+LSC	20	6	2	C,G	A	B	*C*	D,E	*G*	F,H
4.	SALBM-1+HSC	20	Inf.	1	C,G	-	-	-	-	-	-
5.	SALBM-1+HSC	20	5	2	C,G	A	B	*C, F*	*D, E, G*	H	-
6.	SALBM-1+ESC	20	6	-	C,G	A	B	*C*	D,E	*G*	F,H
7.	SALBM-1+ESC	20	5	-	C,F	A	B	*C, F*	D,E,G	H	-
8.	SALBM-2	17	5	-	-	A	B	C,F	D,E	G,H	-
9.	SALBM-2+LSC	20	5	1	C,G	A	B	*C*	D,E,G	F,H	-
10.	SALBM-2+LSC	28	5	2	C,G	A,B	*C*	D,E	*G*	F,H	-
11.	SALBM-2+HSC	24	5	1	C,G	A	B	D	*C, E, G*	F,H	-
12.	SALBM-2+HSC	17	5	2	C,G	A	B	*C, F*	D,E	*G, H*	-
13.	SALBM-2+ESC	28	5	-	C,G	A,B	*C*	D,F,H	E	*G*	-
14.	SALBM-2+ESC	17	5	-	C,F	A	B	*C, F*	D,E	G,H	-

shows, that cycle time in the example is 20 seconds and 5 workstations are required. According to the optimal assignment tasks A and B are assigned to the first two workstations, tasks C and F are assigned to the third workstation, task H is alone on the forth workstation, and finally tasks D, E and G are assigned to the fifth workstation.

The optimal solution of the SALBM-2 can also be seen in Table 3 (model eight). In this case it is assumed, that the assembly line consists of 5 workstations, and the cycle time is minimized. The optimum value of the objective function is 17, that is, the minimal value of the cycle time is 17 seconds. It can be seen that the optimal task assignment is now slightly different to the optimal solution of the SALBM-1.

Table 3 shows the optimal solutions of the basic SALBMs when workforce skill constraints are applied. The first column of the table indicates which constraints are added to the basic models. The "Set S" column lists the special tasks of the problem, and column "W" contains the limit on special workers. The cells with italic letters indicate the workstations with special workers.

In the second model of Table 3 tasks C and G are simple tasks, which can be performed by low skilled workers and at least 1 special worker must be applied. The optimal assignment has changed now compared to the basic model (SALBM-1). One low skilled worker is applied at workstation 3, as indicated by the cell with italic letters of the table. Since task G can be performed by regular workers as well, it is assigned to workstation 4 together with regular tasks. Workforce constraints in this case did not increase the required number of workstations.

In the third model of Table 3 tasks C and G are the simple tasks again, but now at least 2 special workers must be applied. Since there are two special tasks, and two special workers must be applied, the only possible solution is to assign the two special tasks to two different workstations with the two low skilled workers. Optimal assignment shows, that workstation 3 and 5 have special workers, and now, 6 workstations are necessary to produce the required production quantity.

In the fourth model of Table 3 tasks C and G are special tasks for high skilled workers, that is, these tasks can be performed only by the one available special worker. The model has no feasible solution with this condition. The total time of these two tasks is 12 seconds, which is less than the required cycle time (20 seconds), however, precedence constraint and cycle time constraints together impede the assignment of these special tasks to the same workstation. Consequently, with only 1 special worker there is no feasible solution.

With two special workers, however, the problem has feasible solution, as it is indicated by model five. In this case tasks C and G are assigned to two different workstations with the two high skilled workers. Since high skilled workers can perform regular tasks as well workstations 3 and 4 with special workers can have regular tasks, and 5 workstations are necessary to produce the required production quantity.

In the sixth model tasks C and G are special tasks again, but now those workers which are able to perform these tasks can not perform other tasks. In the optimal assignment these tasks are assigned to two different workstations, and the total number of workstations is equal to 6. Note that it would be better to assign these special tasks to the same workstations, but precedence constraints and cycle time constraints exclude this possibility.

If tasks C and F are the special tasks with exclusive constraint, then these tasks can be assigned to the same workstation, as it is shown by model seven in Table 3.

The second part of Table 3 shows how the results of SALBM-2 changes if workforce skill constraints are added to the basic model. We applied the same skill constraints as before. Each SALBM-2 in the table is solved with five workstations (M=5), consequently, skill constraints influence the optimal cycle time.

Table 3 shows, that skill constraints in most cases (models eight, ten, eleven and thirteen) lead to increased cycle time. The most restrictive constraints are found at models ten and thirteen, when cycle time increased to 30 seconds.

5 Conclusions

In this paper we showed how basic assembly line balancing models can be completed with simple workforce skill constraints. First, the two basic models, that is, the workstation minimization model and the cycle time minimization model are presented. Next, in order to generalize workforce skill constraints we classified the basic cases into three categories. Low skill constraints are applied for simple tasks, high skill constraints are applied for complicated tasks, and exclusive skill constraints are applied for tasks requiring specialists. Finally, skill constraints are formulated and integrated into the basic simple assembly line balancing models.

A simple example is used to show, how skill constraints influence the optimal solution of SALBMs. Since skill constraints impose further restriction, generally the objective function deteriorates compared to the optimal solution of the basic models. It can be concluded, however, that special attention must be given to the design of the manufacturing process. When the grouping possibilities of special tasks are considered in the design phase of the production process, the restrictive effect of skill constraints can be reduced.

Finally, it must be noted, that skill constraints presented in this paper can be further generalized by considering different levels of skill for high, low, and exclusive skill situations. This generalization is not presented in this paper, which explains why we used the term "simple skill constraints" in the paper. The formulation and application of complex skill constraints are topics of further research.

References

1 **Baybars Í**, *A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem*, Management Science(32),, posted on 1986, 909–932, DOI 10.1287/mnsc.32.8.909, (to appear in print).

2 **Becker C, Scholl A**, *A Survey on Problems and Methods in Generalized Assembly Line Balancing*, European Journal of Operational Research(168),, posted on 2006, 694–715, DOI 10.1016/j.ejor.2004.07.023, (to appear in print).

3 **Boysen N, Fliedner M, Scholl A**, *Assembly Line Balancing*: *Which Model to Use When?*, International Journal of Production Economics(111),, posted on 2008, 509–528, DOI 10.1016/j.ijpe.2007.02.026, (to appear in print).

4 **Bowman E H**, *Assembly Line Balancing by Linear Programming*, (1960), 385–389.

5 **Bryton B**, *Balancing of a Continuous Production Line*, Northwestern University, Evanston, ILL., 1954. Unpublished M.S. Thesis.

6 **Corominas A, Pastor F, Plans J**, *Balancing Assembly Line with Skilled and Unskilled Workers*, Omega, **36**, (2008), 1126–1132, DOI 10.1016/j.omega.2006.03.003.

7 **Johnson R V**, *A Branch and Bound Algorithm for Assembly Line Balancing Problems with Formulation Irregularities*, Management Science(29),, posted on 1983, 1309–1324, DOI 10.1287/mnsc.29.11.1309, (to appear in print).

8 **Patterson J H, Albracht J J**, *Assembly-Line Balancing*: *Zero-One Programming with Fibonacci Search*, Operations Research(23),, posted on 1975, 166-174, DOI 10.1287/opre.23.1.166, (to appear in print).

9 **Salveson M E**, *The Assembly Line Balancing Problem*, Journal of Industrial Engineering(6), (1955), 18–25.

10 **Scholl A, Becker C**, *State-of-art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing*, European Journal of Operational Research(168),, posted on 2006, 666–693, DOI 10.1016/j.ejor.2004.07.022, (to appear in print).

11 **Thangavelu S R, Shetty C M**, *Assembly Line Balancing by Zero-One Integer Programming*, AIIE Transactions(3),, posted on 1971, 61–68, DOI 10.1080/05695557108974787, (to appear in print). Equation 1.

12 **White W W**, *Comments on a Paper by Bowman*, Operations Research(9),, posted on 1961, 274–276, DOI 10.1287/opre.9.2.274, (to appear in print).