A Mathematical Programming Model of the Assembly Line Balancing Problem

Dennis L. Bricker
Shuw-Hwey Juang

Department of Industrial Engineering
The University of Iowa

Working Paper
November 1993

Abstract: *(to be completed later)*

Introduction

An assembly line consists of a series of M work stations, along which a product moves. The product remains at each work station an amount of time C called the cycle time, during which one or more tasks are performed, each with a known performance time. The assignment of the tasks to the work stations should be done in a "balanced" way, i.e., the total of the performance times of the tasks should be roughly the same at each work station. Restricting the assignment of a task to a work station is the requirement that certain other tasks (predecessors) must be assigned to the same or an earlier work station.

There are two common problems which are of interest. The assembly line balancing problem of type 1 (ALB1) is to minimize the number of work stations, given a desired cycle time. The assembly line balancing problem of type 2 (ALB2), on the other hand, assumes the number of work stations to be specified and attempts to minimize the cycle time (or equivalently, maximize the production rate) of the assembly line.

Given:

$I = \{1,2,... N\}$ = set of tasks to be assigned to work stations

T_i = time required to complete task i, $i \quad I$

P_i = set of indices of the predecessors of task i

$J = \{1,2,... M\}$ = set of work stations

C = cycle time

Values must be selected for the decision variables

λ_{ij} = 1 if task i is assigned to station j,
0 otherwise

for all i   I and j   J, and

Y_j = 1 if work station j is used,
0 otherwise

for each j   J. Given any upper bound M on the number of work stations, the ALB1 objective, then, is

$$\text{Minimize} \sum_{j=1}^{M} Y_j \tag{0}$$

The constraints of ALB1 are of several varieties:

The sum of the performance times assigned to a station, if it is used, cannot exceed the cycle time:

$$\sum_{i \quad I} T_i\lambda_{ij} \le CY_j \qquad j \quad J \tag{1}$$

Each task must be assigned to exactly one station:

$$\sum_{j \quad J} \lambda_{ij} = 1 \qquad i \quad I \tag{2}$$

There are several ways to express the precedence constraints, e.g.,

$$\lambda_{ij} \le \sum_{k=1}^{j} \lambda_{pk} \qquad i \quad I,\ p \quad P_i,\ \text{and } j \quad J \tag{3}$$

due to Bowman [ ], and

$$\sum_{j \quad J} j\lambda_{pj} \le \sum_{j \quad J} j\lambda_{ij} \qquad i \quad I \text{ and } p \quad P_i \tag{4}$$

due to Patterson and Albracht [ ]. We will state the precedence constraints as "paired precedence cuts" (cf. Wilson, [ ]). Consider p P_i, i.e., a task p which must be assigned either to the same station as task i, or else to an earlier station. For each work station j, we wish to eliminate the possibility that task p is assigned to a station later than j <u>and</u> task i is assigned to a station j or earlier. That is, we require that <u>either</u> task p is assigned to station j or earlier, <u>or</u> task i is assigned to station j or later (or both). This is accomplished by the constraint

$$\sum_{k=1}^{j} \lambda_{pk} + \sum_{k=j+1}^{M} \lambda_{ik} \ge 1 \qquad j \quad J \tag{5}$$

We can require that station j not be used unless all preceding work stations are used:

$$Y_j \le Y_{j-1} \qquad j \quad J\backslash\{1\} \tag{6}$$

Finally,

$$Y_j \quad \{0,1\} \qquad j \quad J, \tag{7}$$

$$\lambda_{ij} \quad \{0,1\} \qquad i \quad I \text{ and } j \quad J \tag{8}$$

The Type 1 assembly line balancing problem, then, is defined by the objective (0) and the constraints (1,2,5,6,7,8).

Change of Variable

As in [Bricker], we define the linear transformation

$$\delta_{ij} = \sum_{k=j+1}^{M} \lambda_{ik} \qquad i \quad I \ \&\ j \quad J\backslash\{M\} \tag{9}$$

and

$$\delta_{i0} \equiv 1 \ \&\ \delta_{iM} \equiv 0 \qquad i \quad I, \tag{10}$$

which has the inverse transformation

$$\lambda_{ij} = \delta_{i,j-1} - \delta_{ij} \qquad i \quad I \ \&\ j \quad J \tag{11}$$

This new set of variables may be defined as

$$\delta_{ij} = 1 \text{ if task i is } \underline{\text{not}} \text{ assigned to one of the first j work stations,}$$
$$0 \ \text{ otherwise}$$

The nonnegativity of λ_{ij}, then, implies

$$\delta_{i,j-1} - \delta_{ij} \geq 0, \text{ i.e.,} 1 = \delta_{i0} \geq \delta_{i1} \geq \delta_{i2} \geq \cdots \delta_{i,M-1} \geq \delta_{iM} = 0 \tag{12}$$

and constraint (5), rewritten as

$$\left(1 - \sum_{k=j+1}^{M} \lambda_{pk}\right) + \sum_{k=j+1}^{M} \lambda_{ik} \geq 1 \qquad j \quad J\backslash\{M\}$$

transforms into

$$\left(1-\delta_{pj}\right) + \delta_{ij} \geq 1, \text{ i.e., } \ \delta_{ij} \geq \delta_{pj} \qquad j \quad J\backslash\{M\},\ i \quad I, \ \&\ p \quad P_i \tag{13}$$

Constraint (1) becomes

$$\sum_{i \quad I} T_i\left(\delta_{i,j-1} - \delta_{ij}\right) \leq CY_j \qquad j \quad J \tag{14}$$

Finally, then, our mathematical programming model of the type 1 assembly line balancing problem is

$$\text{Minimize} \sum_{j=1}^{M} Y_j \tag{0}$$

subject to

$$\delta_{i,j-1} \geq \delta_{ij} \qquad i \quad I \ \&\ j \quad J \tag{12}$$

$$\delta_{ij} \geq \delta_{pj} \qquad j \quad J\backslash\{M\},\ i \quad I, \ \&\ p \quad P_i \tag{13}$$

$$\sum_{i \quad I} T_i\left(\delta_{i,j-1} - \delta_{ij}\right) \leq CY_j \qquad j \quad J \tag{14}$$

$$Y_j \leq Y_{j-1} \qquad j \quad J\backslash\{1\} \tag{6}$$

$$Y_j \quad \{0,1\} \qquad j \quad J, \tag{7}$$

$$\delta_{i0} \equiv 1, \delta_{iM} \equiv 0, \ \ \delta_{ij} \quad \{0,1\} \qquad i \quad I \ \&\ j \quad J\backslash\{M\} \tag{15}$$

Lagrangian Relaxation

Suppose that we introduce Lagrangian multipliers $\mu_j \geq 0$, j J, and relax constraints (14). The Lagrangian function is

$$\sum_{j\in J} Y_j - \sum_{j\in J}\mu_j\left[CY_j - \sum_{i\in I} T_i\left(\delta_{i,j-1} - \delta_{ij}\right)\right] = \sum_{j\in J}\left(1-C\mu_j\right)Y_j + \sum_{i\in I}\sum_{j=0}^{M-1} T_i\left(\mu_{j+1}-\mu_j\right)\delta_{ij}$$

and the Lagrangian relaxation is therefore

$$\Phi(\mu) = \text{Minimum} \sum_{j\in J}\left(1-C\mu_j\right)Y_j + \sum_{i\in I}\sum_{j=0}^{M-1} T_i\left(\mu_{j+1}-\mu_j\right)\delta_{ij} \qquad (16)$$

subject to (6), (7), (12), (13), and (15)

This problem is separable in Y_j and δ_{ij}, i.e., $\Phi(\mu) = \Phi_Y(\mu) + \Phi_\delta(\mu)$, where

$$\Phi_Y(\mu) = \text{Minimum}\left\{\sum_{j\in J}\left(1-C\mu_j\right)Y_j \mid Y_j\le Y_{j-1}, Y_j\in\{0,1\}, j\in J\right\} \qquad (17)$$

$$\Phi_\delta(\mu) = \text{Min}\left\{\sum_{i\in I}\sum_{j=0}^{M-1} T_i\left(\mu_{j+1}-\mu_j\right)\delta_{ij} \mid \delta_{i,j-1}\ge\delta_{ij}, \delta_{ij}\ge\delta_{pj}, \delta_{ij}\in\{0,1\}\ \ i\in I, j\in J, \&\ p\in P_i\right\} \qquad (18)$$

The evaluation of $\Phi_Y(\mu)$ is easy, with the minimum attained at

$$Y_j=1 \text{ for } j \le \operatorname{argmin}\left\{k - C\sum_{j=1}^{k}\mu_j\right\} \qquad (19)$$

which is the solution of the LP relaxation of $\Phi_Y(\mu)$. Hence, $\Phi_Y(\mu)$ exhibits the "Integrality Property".

For any vector of Lagrangian multipliers, $\mu\ge 0$, the optimal value of the Lagrangian relaxation, $\Phi(\mu)$, is a lower bound of the solution of the assembly line balancing problem. The Lagrangian dual problem, namely

$$\underset{\mu\ge 0}{\text{Maximize}}\ \Phi(\mu) \qquad (18)$$

is the problem of searching for the Lagrangian multipliers which yield the greatest lower bound. Depending upon the size of the duality gap, this lower bound might prove useful in a branch-and-bound algorithm for obtaining the optimal solution of the assembly line balancing problem.

Network Structure of the Relaxation

Note that constraints (12) and (13) have at most two nonzero coefficients, namely +1 and/or -1, after $\delta_{i0}=1$ and $\delta_{iM}=0$ have been substituted. In the dual of the LP relaxation of $\Phi_\delta(\mu)$, then, each column will also exhibit this property. By adding a redundant row, obtained by negating the sum of all the original rows, one obtains a matrix having exactly two nonzero elements (+1 and -1), which is a characteristic of a node-arc incidence matrix. Therefore, the dual of the LP relaxation is equivalent to a minimum cost network flow problem. This has several important implications:

a) For each $\mu\ge 0$, the evaluation of $\Phi_\delta(\mu)$ is relatively easy, using any of several efficient network algorithms.

b) The optimal values of δ_{ij}, which are the dual variables for the network problem, are integer-valued, because of the total unimodularity of the node-arc incidence matrix. That is, $\Phi_\delta(\mu)$ also exhibits the "Integrality Property".

c) Hence the Lagrangian relaxation $\Phi(\mu)$ exhibits the "Integrality Property", which implies that the maximum value of $\Phi(\mu)$ is equal to the optimal value of the LP relaxation of the problem (cf. [Fisher]).

Although the greatest lower bound provided by $\Phi(\mu)$ can be no better than that provided by the value of the LP relaxation of (0,12,13,14,15), the search for the optimal μ may be more efficient than applying the simplex algorithm to the LP relaxation, due to the size of the problem.

The network has $M \times N + 1$ nodes, one for each pair (i,j), i I and j J, plus the node (0) introduced by the redundant constraint. The supply (demand if negative) of commodity at each node (i,j) is $T_i(\mu_{j+1}-\mu_j)$, and for node (0), the quantity $(1-\Sigma_j\ \mu_j)$ which, as noted previously, can be assumed to be positive.

The arcs of the network are as follows:

For each i I and j=2,3,...M, there will be an arc from node (i,j-1) to node (i,j). There will be (M-1) × N arcs of this type.

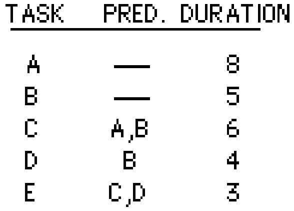
For each i I, p P_i, and j=2,3,...M, there will be an arc from (i,j-1) to (p,j). The total number of arcs, then, is

$$M \times (N+1) \times \sum_{i\quad I} |P_i|$$

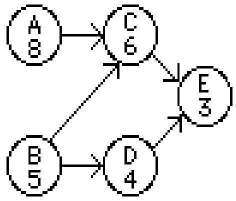
arcs, where $|P_i|$ is the cardinality of the set P_i. Suppose, for example, that there are M=100 tasks to be assigned to N=9 work stations, and that the number of precedence restrictions $\Sigma|P_i|$ is 200. Then the network will have 1000 nodes and 200,000 arcs. While this may seem large, it is very efficiently solved by state-of-the-art implementations of the primal simplex method.

Example

Consider a very small example, with 3 work stations and 5 tasks:



TASK	PRED.	DURATION
A	—	8
B	—	5
C	A,B	6
D	B	4
E	C,D	3

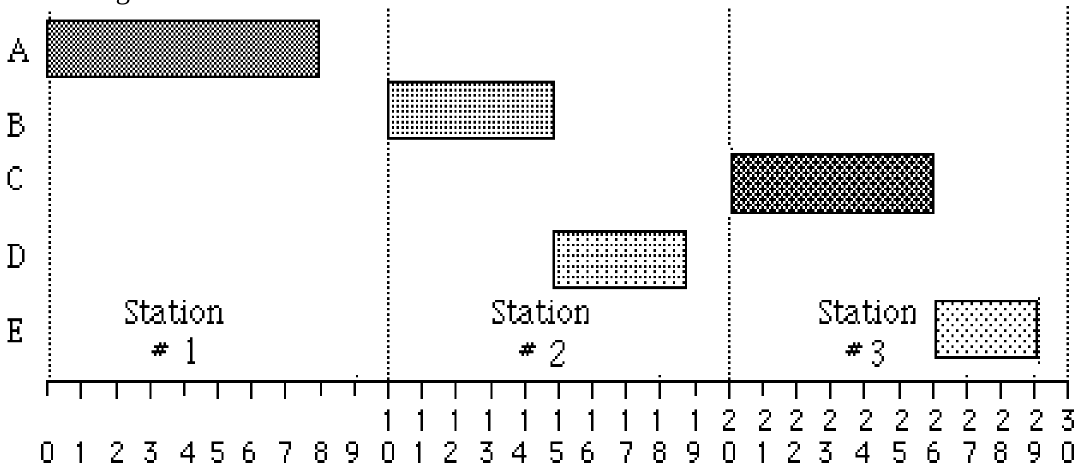


The solutions of the problem (0), (1), (2), (3), (7), and (8) and its LP relaxation are, respectively,

N=3, $Y_1=Y_2=Y_3=1,Y_4=0$
$\lambda_{A1}=1$, $\lambda_{A2}=\lambda_{A3}=\lambda_{A4}=0$,
$\lambda_{B1}=0,\lambda_{B2}=1$, $\lambda_{B3}=\lambda_{B4}=0$,
$\lambda_{C1}=\lambda_{C2}=0,\lambda_{C3}=1,\lambda_{C4}=0$
$\lambda_{D1}=0,\lambda_{D2}=1,\lambda_{D3}=\lambda_{D4}=0$,
$\lambda_{E1}=\lambda_{E2}=0,\lambda_{E3}=1$, $\lambda_{E4}=0$

N=2.6, $Y_1=Y_2=Y_3=Y_4=0.65$
$\lambda_{A1}=\lambda_{A2}=\lambda_{A3}=\lambda_{A4}={}^1/_4$,
$\lambda_{B1}={}^1/_4,\lambda_{B2}={}^3/_4$, $\lambda_{B3}=\lambda_{B4}=0$,
$\lambda_{C1}={}^1/_4,\lambda_{C2}=0,\lambda_{C3}={}^3/_4,\lambda_{C4}=0$
$\lambda_{D1}={}^1/_4,\lambda_{D2}=\lambda_{D3}=0,\lambda_{D4}={}^3/_4$,
$\lambda_{E1}=\lambda_{E2}={}^1/_4,\lambda_{E3}=0$, $\lambda_{E4}={}^1/_2$

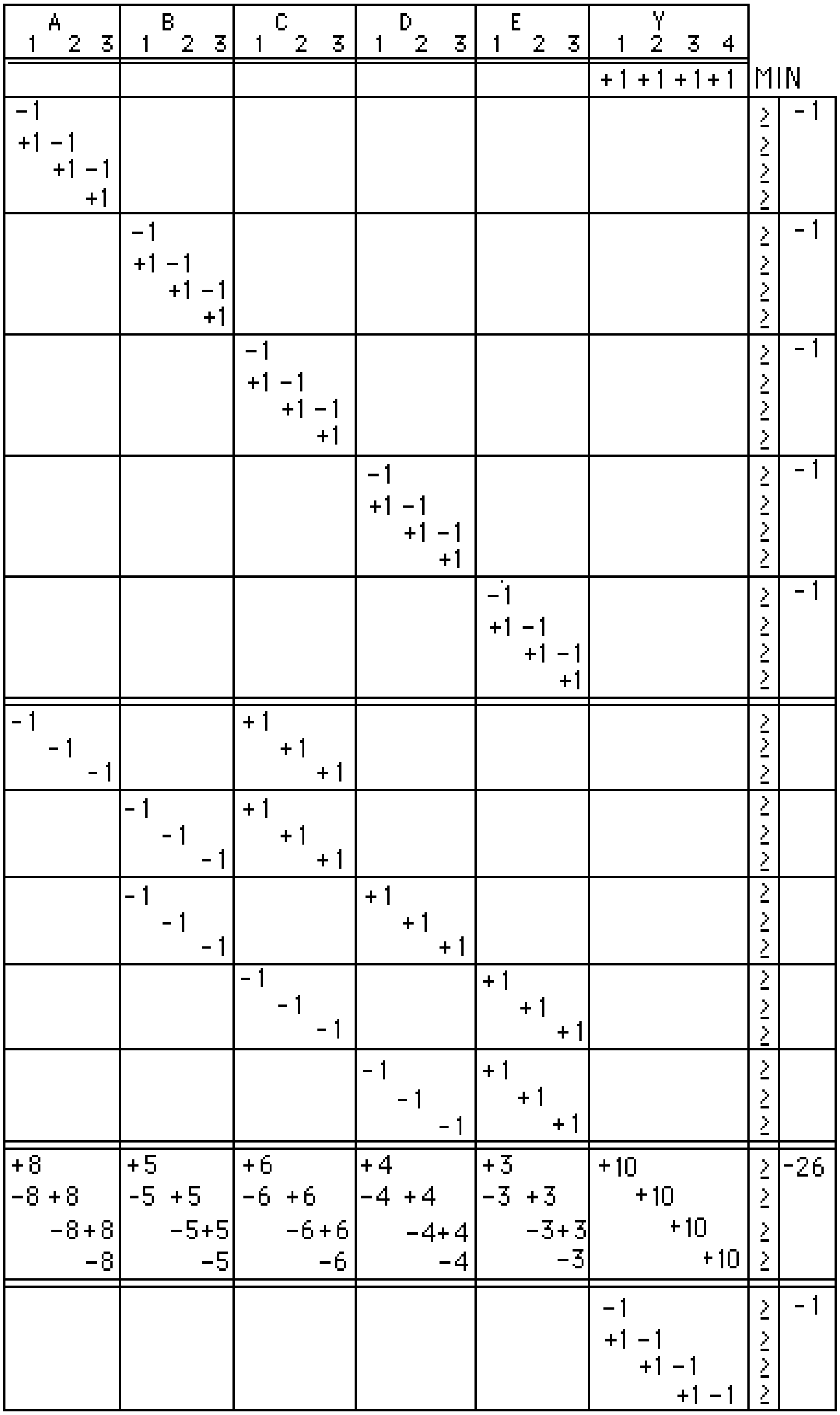
Eleven branches and 84 pivots were required by LINDO to solve the integer problem. A Gantt chart of the integer solution is shown below.



If constraint (5) replaces constraint (3), the LP relaxation has the solution

N=2.6, $Y_1=Y_2=1,Y_3=Y_4=0.3$
$\lambda_{A1}=0.125,\lambda_{A2}=0.660714,\lambda_{A3}=0.214286,\lambda_{A4}=0$,
$\lambda_{B1}=1,\lambda_{B2}=\lambda_{B3}=\lambda_{B4}==0$,
$\lambda_{C1}=0,\lambda_{C2}=0.785714,\lambda_{C3}=0.214286,\lambda_{C4}=0$
$\lambda_{D1}=1,\lambda_{D2}=\lambda_{D3}=\lambda_{D4}=0$,
$\lambda_{E1}=\lambda_{E2}=\lambda_{E3}=0$, $\lambda_{E4}=1$

The integer solution is the same, but required only four branches and 49 pivots. Notice that only two of the activities are split between stations, compared to all the activities in the previous LP relaxation. The tableau for the LP relaxation of (12) through (15) is



A			B			C			D			E			Y					
1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	4		
															+1	+1	+1	+1	MIN	
-1																			≥	-1
+1	-1																		≥	
	+1	-1																	≥	
		+1																	≥	
			-1																≥	-1
			+1	-1															≥	
				+1	-1														≥	
					+1														≥	
						-1													≥	-1
						+1	-1												≥	
							+1	-1											≥	
								+1											≥	
									-1										≥	-1
									+1	-1									≥	
										+1	-1								≥	
											+1								≥	
												-1							≥	-1
												+1	-1						≥	
													+1	-1					≥	
														+1					≥	
-1						+1													≥	
	-1						+1												≥	
		-1						+1											≥	
			-1			+1													≥	
				-1			+1												≥	
					-1			+1											≥	
			-1						+1										≥	
				-1						+1									≥	
					-1						+1								≥	
						-1						+1							≥	
							-1						+1						≥	
								-1						+1					≥	
									-1			+1							≥	
										-1			+1						≥	
											-1			+1					≥	
+8			+5			+6			+4			+3			+10				≥	-26
-8	+8		-5	+5		-6	+6		-4	+4		-3	+3			+10			≥	
	-8	+8		-5	+5		-6	+6		-4	+4		-3	+3			+10		≥	
		-8			-5			-6			-4			-3				+10	≥	
															-1				≥	-1
															+1	-1			≥	
																+1	-1		≥	
																	+1	-1	≥	

where δ_{i0}=1 and δ_{i3}=0 have been substituted, redundant constraints have been eliminated, and blank entries represent zeroes. The solutions of this LP relaxation and of the integer LP are

$C = 8\ {}^{2}\!/_{3}$,

$\delta_{A1}=1,\ \delta_{A2}=0.20833,$

$\delta_{B1}=\delta_{B2}=0,$

$\delta_{C1}=0.3888,\ \delta_{C2}=0,$

$\delta_{D1}=\delta_{D2}=1,$

$\delta_{E1}=\delta_{E2}=1,$

$C = 9$,

$\delta_{A1}=\delta_{A2}=0,$

$\delta_{B1}=1,\ \delta_{B2}=0,$

$\delta_{C1}=\delta_{C2}=1,$

$\delta_{D1}=1,\ \delta_{D2}=0,$

$\delta_{E1}=\delta_{E2}=1,$

-- to be revised --

It is interesting to note the disparity between the integer solution and the solution obtained by rounding the continuous solution.

The LP solution, in which tasks may be split between stations, is

-- to be added --

The LP relaxation of the objective (0) and constraints (1), (2), (4), & (6) has the solution

$N=2.6,\ Y_1=Y_1=Y_1=Y_1=0.65$

$\lambda_{A1}=\lambda_{A2}=\lambda_{A2}=\lambda_{A2}={}^{1}\!/_{4},$

$\lambda_{B1}={}^{1}\!/_{4},\lambda_{B2}={}^{3}\!/_{4},\ \lambda_{B3}=\lambda_{B4}=0,$

$\lambda_{C1}={}^{1}\!/_{4},\lambda_{C2}=0,\lambda_{C3}={}^{3}\!/_{4},\lambda_{C4}=0$

$\lambda_{D1}={}^{1}\!/_{4},\lambda_{D2}=\lambda_{D3}=0,\lambda_{D4}={}^{3}\!/_{4},$

$\lambda_{E1}=\lambda_{E2}={}^{1}\!/_{4},\lambda_{E3}=0,\ \ \lambda_{E3}={}^{1}\!/_{2}$

-- to be revised --

In this LP solution, also, the tasks are split among the work stations:

-- to be added --

The dual of the LP relaxation of the above problem has the tableau

-- to be added --

Note the similarity of the above dual tableau to a node-arc incidence matrix, i.e., the (+1,-1) pair of coefficients in the columns. Suppose that we relax the constraints (14) by introducing Lagrangian multipliers μ_1 , μ_2, and μ_3. Then the LP relaxation of the remaining problem, with slack variables introduced to obtain equality constraints, will have columns having either a (+1,-1) pair, or a +1, or a -1 as the nonzero elements. Summing the equations and negating both sides will therefore yield a node-arc incidence matrix (having only (+1,-1) nonzero pairs in each column), drawn below:

-- to be added --

Here, the node labeled "S" corresponds to the redundant row which was added.

References

Bowman, E. H. (1960). Assembly Line Balancing by Linear Programming. Operations Research, 8, 385-389.

Bricker, D. L. (1977). Reformulation of Special Ordered Sets for Implicit Enumeration Algorithms with Applications in Nonconvex Separable Programming. AIIE Transactions, 9, 195-203.

Fisher, Marshall (1981). Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science, 27, 1-18.

Patterson, J. H., & Albracht, J. J. (1975). Assembly Line Balancing: 0-1 Programming with Fibonacci Search. Operations Research, 23, 166-174.

Wilson, J. M. (1990). Generating Cuts in Integer Programming with Families of Special Ordered Sets. European Journal of Operational Research, 46, 101-108.