Technical Memorandum Number 747

Level Workforce Schedules for 2-stage Transfer Lines

by

George L.Vairaktarakis
Joseph Szmerekovsky

August 2001

Department of Operations
Weatherhead School of Management
Case Western Reserve University
10900 Euclid Avenue
Cleveland, Ohio 44106-7235

Level Workforce Schedules for 2-stage Transfer Lines

George Vairaktarakis and Joseph Szmerekovsky *

August 27, 2001

Abstract

In this article we define two different workforce leveling objectives for serial transfer lines. Each job is to be processed on each transfer station for c periods. The number of workers available for each operation of a job is known and it is equal to the number required to complete the operation in precisely c periods. Jobs transfer forward synchronously after every production cycle (i.e., c periods). The purpose of these leveling objectives is to produce job schedules where the cumulative number of workers needed in all the stations of the transfer line does not experience dramatic changes from one production cycle to the next. The objectives proposed are termed *maximin workforce size* ($W_{\min}$) and *range* (R). The former objective maximizes (over all possible schedules) the smallest workforce size required over the production horizon. The range objective yields a schedule for which the difference between the largest and smallest workforce requirement (over all production horizons) is the smallest possible. For $W_{\min}$ and the case of only 2 stations, we develop a fast polynomial algorithm. Finding the optimal range is proved to be strongly NP-complete even for 2 stations. For the 2-station case we propose an optimal algorithm for the range, which uses a very tight lower bound and an efficient procedure for finding complementary Hamiltonian cycles in bipartite graphs. The ensuing computational experiments show that the proposed algorithm is very efficient. Using these tools we examine the trade-off between the workforce size required to complete a set of jobs, and the fluctuations on the number of workers needed from cycle to cycle. Eventhough our analyses do not extend to transfer lines with more than 2 stations, the general case is likely to serve as a building block.

*Weatherhead School of Management, Dept. of Operations, Case Western Reserve University, 10900 Euclid Ave, Cleveland, OH 44106-7235, Email: gxv5@po.cwru.edu, Tel: (216) 368-5215, Fax: (216) 368-4776

1 Introduction

Several cells arranged in series and typically connected by a continuous material handling system form a *serial assembly line.* Such lines are designed to assemble component parts and perform any related operation necessary to produce a finished product. Consider an assembly line that consists of several stations arranged in tandem. Each job enters the same end of the assembly line and requires a series of operations in each of the assembly stations. A distinguishing characteristic of the assembly system studied is that it is *paced* or *synchronous.* This means that every job spends a fixed amount of time in each station, which is the same for all jobs and all stations. This amount of time is called *production cycle.*

Among the advantages of serial assembly systems are lower work-in-process inventories, reduced material handling costs, shorter production flow times, simplified production planning of materials and labor, improved visual control and fewer tooling changes. Overall performance often increases by lowering production costs and improving on-time delivery. Quality should improve as well although this might take other interventions beyond the layout change. Several researchers have considered the effect of serial assembly lines in manufacturing including Chen and Adam, 1991, Fry *et al.*, 1987, Goodrich, 1988, Hyer, 1984, etc.

Our research in this area has been motivated by fire-truck assembly operations. Here, the necessity for a common production cycle for all assembly stations (i.e., paced assembly) comes from the size of the trucks themselves. Since it is inefficient to move semifinished trucks from station to station in real time, such movements take place off-line at the end of the 16-hour production cycle. To improve productivity, management can control the order of processing trucks, and the number of workers assigned for each truck operation. For a given production cycle of c periods, the number of workers assigned to perform a particular operation is the smallest possible that can execute the operation in c periods.

A fire-truck consists of three main components; the body, the chassis and the engine. The chassis and the engine are purchased from an outside supplier, while the body and final assembly (of the three main components) take place in a paced assembly line physically located in two adjacent plants. The body related operations are performed in 8 distinct stations, and the progressively assembled body is moved from station to station on a cart. A final assembly station completes the assembly line for a total of $m = 9$ stations. The physical constraints for moving semi-finished bodies from station to station dictate the common production cycle of 16 hours (i.e., $c = 16$). Workers are assigned to work on stations for 8 hours each day, and the plant runs 2 shifts. At the end of the day (i.e., the 16-hour production cycle), semifinished units are moved to the next downstream station. The 16-hour production cycle allows some flexibility in deciding how many workers are needed in each of the two daily shifts.

This article extends the works of Lee and Vairaktarakis, 1997 and Vairaktarakis *et. al*, 2001. In Lee and Vairaktarakis, 1997 the objective is to find a job schedule that minimizes the total workforce size needed to perform a set of jobs on a serial synchronous assembly line like the one described above. The authors minimize the size of the workforce for a given partition of the assembly line into skills. In Vairaktarakis *et. al* the authors consider a paced job shop where jobs may visit a subset of the stations in different orders. In this protocol the length of the production schedule is no longer determined simply by the number of jobs (as in the case of a serial assembly line). The length of the schedule depends on both the workforce schedule and the workforce size. The authors minimize a linear function of the workforce size and the length of the workforce schedule.

Note that the workforce size objective may result to schedules in which the workforce requirements from period to period vary widely. Such schedules create distractions to both the management and the workforce. From an economic viewpoint, carrying large numbers of additional workers from cycle to cycle is equivalent to carrying excess work-in-process inventory. Workforce leveling objectives are of course known at the aggregate production level; see Vollmann *et al.*, 1997. Recently, leveling objectives were defined in Vairaktarakis and Cai, 2001 and the complexity of the associated decision problems was studied. In this article we develop solution methods for two of the several objectives presented in that article. The methodological difficulty of these problems limit our current study to lines with only 2 assembly stations.

To the best of our knowledge this is the first effort to develop algorithms for day-to-day leveling objectives on transfer assembly lines. Using these tools, in this article we make a first attempt to assess the value of level workforce schedules against minimum workforce schedules that are often used. Our study indicates that, workforce size schedules suffer a significant premium in disruptions caused by wildly changing demands on the number of workers from one day to the next. In comparison, level schedules smooth the daily workforce requirements by investing in a slightly larger workforce. We consider these insights as a main contribution of this article.

The general workforce planning literature includes works on assembly lines that produce a single item. Bartholdi, 1992 presented a case study in which workers perform different tasks of the same item while Pinto *et al.*, 1981 have developed branch and bound and heuristic procedures for the case that workers perform the same task on different items. The importance of decentralized workforce control is captured by Bartholdi and Eisenstein, 1996 where every worker of the assembly line follows a simple rule of what to do next. Ebeling and Lee, 1993 developed an optimization model to assess the effect of cross-training into firm profitability.

A different approach to workforce planning, is to treat workforce as a generic resource to be allocated and scheduled appropriately. Along these lines, Daniels and Mazzola, 1993 consider a flexible resource, the amount of which affects the processing time of an operation. In the context of workforce planning, this is analogous to assigning the right number of workers so as to achieve a predetermined processing time. The manufacturing environment considered in this research is a flowshop (and hence unpaced), and the authors present a tabu-search heuristic with near optimal performance. In a related paper, Daniels *et al.*, 1997 consider the flexible-resource scheduling problem on parallel identical machines.

The rest of the paper is organized as follows. In Section 2 we formally define the range and maximin workforce size objectives considered in this article and provide a graph theoretic representation for 2-station transfer lines. A polynomial algorithm is developed in Section 3 for the maximin workforce size objective. This algorithm is used in Section 4 within a search framework to obtain an optimal range schedule. In Section 5 these tools are used in a computational experiment to evaluate the relative merits of optimal range and workforce size schedules. Closing remarks are given in Section 6.

2 Problem Formulation

A given set J of simultaneously available jobs is to be processed on two stations. Every job J_i consists of two tasks. For convenience we use the pair of workforce requirements (W_{i1}, W_{i2}) to denote job J_i and each worker is capable of working on both stations. The first task must be processed at ST_1 for c periods and upon completion, the second task continues at ST_2 where it stays for another c periods. It is assumed that no station can handle more than one task at a time and no task can be interrupted once it has begun processing. The n jobs form a minimal product set (MPS) and are to be processed repeatedly. An MPS is the smallest combination of products satisfying the demand ratios. If we have $|P|$ products with integer demands $(d_1, d_2, \ldots, d_{|P|})$ respectively then the MPS will contain $(d_1/g, d_2/g, \ldots, d_{|P|}/g)$ units where g is the greatest common divisor of the demand volumes. Associated with the MPS is a *cyclic* schedule in which the units of each MPS are processed through the system in exactly the same order. The following notation will be used throughout this paper:

n: number of jobs

c: the time length of a production cycle

J_i: the i-th job of $J = \{J_1, J_2, \ldots, J_n\}$

ST_j: the j-th station of $ST = \{ST_1, ST_2\}$

W_{ij}: the workforce requirement of job J_i on station ST_j in order to complete the j-th task of

J_i in c periods, $i = 1, 2, \ldots, n$, $j = 1, 2$. It is assumed to be a positive integer.

c_k: the k-th production cycle.

W_k: the total (over the 2 stations) workforce size required during cycle c_k.

$\Delta_{k,k'} = |W_k - W_{k'}|$: represents that difference in the workforce requirements of production cycles c_k and $c_{k'}$.

Let us define the following binary decision variables:

$$x_{ij} := \begin{cases} 1 & \text{if job } J_i \text{ is scheduled at position } j; \\ 0 & \text{otherwise.} \end{cases}$$

Then, the number of workers required during the k-th production cycle is

$$W_k = \sum_{i=1}^{n} \sum_{j=1}^{2} W_{ij} x_{i,(k-j+1) \bmod n}, \quad 1 \leq k \leq n,$$

and hence

$$\Delta_{k,k'} = |W_k - W_{k'}| = |\sum_{i=1}^{n} \sum_{j=1}^{2} W_{ij} (x_{i,(k-j+1) \bmod n} - x_{i,(k'-j+1) \bmod n})|$$

where *mod* denotes the modulus operation and $n \bmod n = n$, $0 \bmod n = n$. A generic formulation for the level workforce planning problem (LW) on 2 stations is given below. Different objectives are captured by different functions $f(\vec{\Delta})$ where $\vec{\Delta}$ is a vector with elements $\Delta_{k,k'}$ for all $1 \leq k \neq k' \leq n$.

$$
\begin{aligned}
(\textbf{LW}) \quad MIN \quad & f(\vec{\Delta}) & \\
s.t. \quad & \textstyle\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \ldots, n & (1) \\
& \textstyle\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, \ldots, n & (2) \\
& x_{ij} \in \{0, 1\} \quad i, j \in \{1, \ldots, n\} & (3)
\end{aligned}
$$

Let $R_{\max}$ be the function:

$R_{\max} = \max_{1 \leq k \neq k' \leq n} \Delta_{k,k'}$: the *range* of workforce requirements over the n production cycles.

The following model is refered to as the *2-station workforce range* problem (2SRW) and amounts to finding a (cyclic) production schedule S for a given MPS $J_1, J_2, \ldots, J_n$, such that S minimizes the range of the workforce requirements over the n production cycles. Problem 2SRW minimizes the maximum among the $\Delta_{k,k'}$'s for $1 \leq k \neq k' \leq n$.

$$
\begin{aligned}
(\textbf{2SRW}) \quad MIN \quad & R & \\
s.t. \quad & \Delta_{k,k'} \leq R \quad k = 1, \ldots, n & (4) \\
& (1) - (3). &
\end{aligned}
$$

In 2SRW the same operations are repeated every n production cycles and in each of these cycles both stations of the assembly line are busy. Constraints (1)-(3) capture the assignment of jobs to production cycles.

Next, consider the leveling objective $W_{\min}$. In this, we want to maximize the minimum workforce requirement over the production horizon. Intuitively, such a schedule attempts to equalize the workforce requirements over all the cycles of the production horizon. The associated integer program for 2 stations is given below.

$$\textbf{(2SmW)} \quad W_{\min} = Max_x Min_k \quad \sum_{i=1}^{n} \sum_{j=1}^{2} W_{ij} x_{i,(k-j+1) \bmod n}$$
$$s.t. \qquad (1)-(3.)$$

We refer to this problem as the 2-station *maximin* workforce problem. The integer programs 2SRW and 2SmW extend easily to arbitrary number m of stations in the transfer line; simply compute the workforce requirements W_k during c_k over stations $j = 1, 2, \ldots, m$ rather than $j = 1, 2$ only. Additional objectives $f(\vec{\Delta})$ have been defined for problem LW in Vairaktarakis and Cai, 2001 who studied the complexity of the associated decision problems.

We observed earlier that 2SmW attempts to equalize the workforce requirements over all the cycles in the production horizon. Similar rationale holds for a schedule where the maximum workforce requirement is minimized over all production cycles. This problem has been studied extensively for any fixed number of stations in Lee and Vairaktarakis, 1997. The 2-station case will be used later in our analysis for 2SRW, and in our computational experiment for the evaluation of the merits of the various schedules.

In this article we develop methods for solving 2SmW and 2SRW. Using the tools developed, we draw insights on the differences between the 2 objectives. In our analyses we use a graph theoretic representation for 2SRW and 2SmW. This is described next.

2.1 Graph Theoretic Formulation for 2SRW and 2SmW

Let us start with a graph representation for 2SRW. Consider a bipartite graph $B = (V, U)$ where V and U consist of n nodes each, i.e. $V = \{v_1, v_2, \ldots, v_n\}$ and $U = \{u_1, u_2, \ldots, u_n\}$. With each $u_i \in U$ we associate the workforce requirement W_{i1}, and with each $v_i \in V$ we associate the requirement W_{i2}. To indicate job J_i in the graph B, we link v_i with u_i for $i = 1, 2, \ldots, n$. The resulting links form a matching M in B; we refer to this as the *job matching*. Associated with every permutation $J_{[1]}, J_{[2]}, \ldots, J_{[n]}$ of the jobs in J is a matching I whose edges link edges in M. We refer to I as a *sequence matching*. Let (u_l, v_i) be an edge in I. This means that the job J_l immediately precedes J_i. Hence, there is a production cycle (say c_k) during which W_{i1} workers are required in ST_1 and W_{l2} workers in ST_2. This means that, during c_k a total of $W_{i1} + W_{l2}$

workers is required.

Example: Consider the 7-job problem with W_{ij} given in Table 1. The bipartite graph B corresponding to the sequence $J_1J_2J_3J_4J_5J_6J_7$ is given in Figure 1. The dotted edges represent the sequence matching I and the solid edges represent the job matching M.

Table 1: A 7-job example

i \ j	1	2
1	2	13
2	2	12
3	5	4
4	6	5
5	10	3
6	12	9
7	12	1

INSERT FIGURE 1 HERE

Evidently, for every permutation of the n jobs, the graph $I \cup M$ forms a Hamiltonian cycle; i.e. a cycle that spans all the nodes of B. This Hamiltonian cycle consists of $2n$ edges that alternate between the sets I and M. Since the job matching M is given, the matching I complements M into a Hamiltonian cycle. For this reason, we refer to $I \cup M$ as a *complementary Hamiltonian cycle* or CHC. Let

$$w_{il} = W_{i1} + W_{l2} \quad for \quad every \quad edge \quad (W_{i1}, W_{l2}) \quad of \quad B(V, U).$$

Then, 2SRW can be cast as a minimum cost CHC problem where cost is measured as the difference between the largest and smallest edge costs w_{il} among edges in I. Below we summarize our observations.

Proposition 1 *The sequence matching I corresponds to a permutation of the jobs in J if and only if $I \cup M$ is a complementary Hamiltonian cycle.*

Example: Consider the CHC given in Figure 1. The workforce requirements for this cyclic schedule for every production cycle are given in Table 2. Evidently, the total number of workers required by this schedule is 21, the minimum number of workers required over all production cycles is 3, and hence the range is $R = \Delta_{1,7} = 18$.

The CHC representation for problem 2SRW extends naturally to problem 2SmW. The only difference is that the matching I of an optimal CHC $I \cup M$ should be such that the minimum

Table 2: An example schedule

Job\ Cycle	1	2	3	4	5	6	7	
J_1	**2**	**13**						
J_2		**2**	**12**					
J_3			**5**	**4**				
J_4				**6**	**5**			
J_5					**10**	**3**		
J_6						**12**	**9**	
J_7	**1**						**12**	
W_k	3	15	17	10	15	15	21	Max = 21
								Min = 3

edge cost w_{il} is the largest possible. In what follows we develop solution algorithms for 2SmW and 2SRW. We start our analysis with problem 2SmW because the algorithm developed for this is used to obtain an optimal range schedule for 2SRW.

3 An Optimal Algorithm for 2SmW

In this section we develop an $\mathcal{O}(n \log n)$ algorithm that solves the 2-station workforce problem with the objective to maximize the minimum workforce requirement over the production horizon.

$$(\mathbf{2SmW}) \quad W_{\min} = Max_x Min_k \quad \sum_{i=1}^{n} \sum_{j=1}^{2} W_{ij} x_{i,(k-j+1) \text{ mod } n}$$
$$s.t. \qquad (1) - (3.)$$

In the CHC representation of Section 2.1 for problem 2SmW, recall that matching I of an optimal CHC $I \cup M$ should be such that the minimum edge cost w_{il} is the largest possible. Consider the relaxation of 2SmW where we want to find an unconstrained perfect matching I between the nodes of V and U so that the minimum edge cost is maximized. In other words, there is no requirement that $I \cup M$ forms a CHC. Then, the maximin edge cost of I is a lower bound to the optimal solution of 2SmW. To present this result, let us redefine $V = \{v_i : 1 \leq i \leq n\}$ and $U = \{u_i : 1 \leq i \leq n\}$ so that

$$W_{11} \leq W_{21} \ldots \leq W_{n1}, \quad and \quad W_{12} \geq W_{22} \geq \ldots \geq W_{n2}.$$

From now on we will refer to nodes in $V \cup U$ by the associated W_{ij} value. E.g., we use (W_{i1}, W_{j2}) to denote edge $(u_i, v_j) \in E(B)$. We can state the following result.

Lemma 1 *Let $W_{\min}$ be the optimal solution for 2SmW. Then,*

$$W_{\min} \leq \min_{1\leq i\leq n} \{W_{i1} + W_{i2}\}.$$

Proof: Let I_U be an unconstrained matching between the integers $W_{11} \leq W_{21} \ldots \leq W_{n1}$ and $W_{12} \geq W_{22} \geq \ldots \geq W_{n2}$. Let W_U be the smallest cost among edges in I_U. If $W_{\min}$ is the smallest edge cost of an optimal (and hence constrained) matching (i.e., assignment) for 2SmW, we have $W_{\min} \leq W_U$. To complete the proof it suffices to show that

$$W_U = \min_{1\leq i\leq n} \{W_{i1} + W_{i2}\}.$$

Indeed, consider two arbitrary pairs W_{i1}, W_{j1} and W_{k2}, W_{l2}. Without loss of generality assume that $W_{i1} \leq W_{j1}$ and $W_{k2} \geq W_{l2}$. There are 2 possible matchings between these 2 pairs; matching $I_1 = \{(W_{i1}, W_{k2}), (W_{j1}, W_{l2})\}$ and matching $I_2 = \{(W_{i1}, W_{l2}), (W_{j1}, W_{k2})\}$. It is easy to verify that

$$\min\{W_{i1} + W_{k2}, W_{j1} + W_{l2}\} \leq \min\{W_{i1} + W_{l2}, W_{j1} + W_{k2}\}.$$

Hence, for any 2 arbitrary pairs, it is always optimal to order W_{i1}-values in nondecreasing order and W_{i2}-values in nonincreasing order. Now, start with an arbitrary ordering of W_{i1}- and W_{i2}-values, and apply this argument to every 4 values W_{i1}, W_{j1}, W_{k2}, and W_{l2} that violate the condition $W_{i1} \leq W_{j1}$ or $W_{k2} \geq W_{l2}$. After no more than $\binom{n}{4}$ iterations, the W_{i1}'s will be in nondecreasing order and the W_{i2}'s in nonincreasing order. Hence, $W_U = \min_{1\leq i\leq n}\{W_{i1} + W_{i2}\}$. This completes the proof of the lemma. □

Consider the unconstrained maximin matching

$$I_0 = \{(W_{i1}, W_{i2}) : 1 \leq i \leq n\}$$

of Lemma 1. Generally, $I_0 \cup M$ is a union of subcycles that alternate between edges in I_0 and edges in M. In the event that $I_0 \cup M$ forms a CHC, we have $W_{min} = \min_i\{W_{i1} + W_{i2}\}$. Let $C_1, C_2, \ldots, C_r$ be the subcycles of $I_0 \cup M$. The process of merging these subcycles into a CHC is known in the literature as *patching* and has been used extensively to solve variants of the Traveling Salesperson problem; see Gilmore and Gomory, 1964, Karp, 1979 and Vairaktarakis and Solow, 2001. A main result from this literature is described next.

Let $G(C)$ be the graph with node set $\mathcal{C} = \{C_1, C_2, \ldots, C_r\}$ and edge set

$$E(G) = \{(C_i, C_j)\ :\ \exists\ e = (W_{k1}, W_{l2}) \in B(V, U)\ s.t.\ C_i\ (C_j)\ traverses\ W_{k1}\ (W_{l2})\}.$$

Then we can state the following known result.

Lemma 2 *Given arbitrary bipartite graph $B(V,U)$ and matching M, then if A CHC exists , graph G is connected.*

Lemma 2 is a necessary (but not sufficient) condition that applies to arbitrary bipartite graphs. The graph $B(V,U)$ associated with 2SmW is complete and every edge (W_{i1}, W_{l2}) has an associated cost w_{il}. This means that G is a complete graph on r nodes, each edge $(C_i, C_j) \in E(G)$ has an associated cost w_{ij}, and the optimal solution $I^\star \cup M$ of 2SmW yields a spanning subgraph in G. In particular, for our application, Lemma 2 can be restated as follows.

Corollary 1 *Let $I^* \cup M$ be an optimal CHC for problem 2SmW and $G'(C) \subseteq G(C)$ such that*

$$E(G') = \{(C_i, C_j) \ : \ \exists \ e = (W_{k1}, W_{l2}) \in I^* \ s.t. \ C_i \ (C_j) \ traverses \ W_{k1} \ (W_{l2})\}.$$

Then, G' is a maximin spanning subgraph of G.

By maximin subgraph we mean that G' consists of edges in $E(G)$ whose minimum edge cost is the largest possible. Motivated by Corollary 1 we construct next a maximin (cost) spanning tree $\mathcal{T}$ of G. Then, we present an algorithm O-2SmW that uses the edges of $\mathcal{T}$ to optimally merge the subcycles $C_1, C_2, \ldots, C_r$ into a single subcycle. Consider the maximin tree $\mathcal{T}$ of the subgraph $G_{\mathcal{T}}(C) \subseteq G(C)$ with

$$E(G_{\mathcal{T}}) = \{(C_i, C_j) \ : \ \exists \ e = (W_{k1}, W_{k+1,2}) s.t. \ C_i \ (C_j) \ traverses \ W_{k1} \ (W_{k+1,2})\}.$$

In other words, rather than trying to find an arbitrary maximin spanning subgraph of G, it suffices to find a spanning tree of the subgraph $G_{\mathcal{T}}$. Note that the edges of $G_{\mathcal{T}}$ correspond to edges in $B(V,U)$ of the form $(W_{k1}, W_{k+1,2})$. If the pairs (W_{k1}, W_{k2}) and $(W_{k+1,1}, W_{k+1,2})$ are traversed by different subcycles C_i, C_j respectively, then $(C_i, C_j) \in E(G_{\mathcal{T}})$. There are precisely $n-1$ edges $(W_{k1}, W_{k+1,2}) \in E(B)$ and hence $|E(G_{\mathcal{T}})| \leq n-1$.

Example: In Figure 2(a) we show the subcycles C_1, C_2, C_3, C_4, C_5 produced by the matching $I_0 \cup M$ corresponding to the example in Table 1. Note that the W_{i1}-values are ordered in nonincreasing order, the W_{i2}'s in nondecreasing, the solid edges correspond to the job matching M, and the dotted edges correspond to I_0. Graph $G_{\mathcal{T}}$ is shown in Figure 2(b) together with a maximin spanning tree (solid edges). Note that the the 5 edges on $G_{\mathcal{T}}$ correspond to the 5 edges (W_{i1}, W_{i2}) of $E(B)$ that have their endpoints on different subcycles in Figure 2(a). For instance, edge (C_1, C_2) of $G_{\mathcal{T}}$ corresponds to edge (W_{11}, W_{22}) in Figure 2(a). Evidently, the maximin edge cost of $\mathcal{T}$ is 10. Graph G is simply the complete graph on nodes C_1, C_2, C_3, C_4, C_5.

INSERT FIGURE 2 HERE

The following algorithm uses the edges $(W_{k1}, W_{k+1,2})$ of $B(V,U)$ that correspond to edges in $\mathcal{T}$, to merge the subcycles of $I_0 \cup M$ into a CHC.

Algorithm O-2SmW

Input : A Bipartite subgraph $B(U,V)$ and perfect matching M of B

Output : A maximin CHC $I \cup M$

Begin

Order $V = \{v_i\}_{i=1}^n$ in nondecreasing order of W_{i1}'s and $U = \{u_i\}_{i=1}^n$ in nonincreasing order of W_{i2}'s

Let $I := \{(W_{i1}, W_{i2})\}_{i=1}^n$

Let $C_1, C_2, \ldots, C_r$ be the subcycles of $I \cup M$

If $r = 1$ **then stop** (a CHC is found)

Let $\mathcal{T}$ be a maximin cost spanning tree of $G_{\mathcal{T}}$ and let

$\{(W_{i_j,1}, W_{i_j+1,2}), (W_{i_j+1,1}, W_{i_j+2,2}), \ldots, (W_{i_j+r_j-1,1}, W_{i_j+r_j,2})\}_{j=1}^l$ be the corresponding edges in $E(B)$.

For $j := 1$ **to** l **do begin**

[1] $I_j := \{(W_{i1}, W_{i+1,2})\}_{i=i_j}^{i_j+r_j-1} + \{(W_{i_j+r_j,1}, W_{i_j,2})\}$

[2] $I := I - \{(W_{i1}, W_{i2})\}_{i=i_j}^{i_j+r_j} + I_j$

end

End

Evidently, we have written the edges of $E(B)$ associated with $\mathcal{T}$ as a union of subsets in $\{(W_{i_j,1}, W_{i_j+1,2}), (W_{i_j+1,1}, W_{i_j+2,2}), \ldots, (W_{i_j+r_j-1,1}, W_{i_j+r_j,2})\}$ for $j = 1, 2, \ldots, l$. Each such subset contains a maximal subset of consecutive edges of the form $(W_{k1}, W_{k+1,2})$. This, together with edge $(W_{i_j+r_j,1}, W_{i_j,2})$ form in line [1] a *chain* I_j as in Figure 3. The for-loop in O-2SmW uses these chains to merge the subcycles of $I_0 \cup M$ into a CHC.

INSERT FIGURE 3 HERE

Example: For our example problem O-2SmL yields the sequence $S = J_1J_4J_3J_5J_7J_6J_2$ depicted in Figure 2(c). There are 2 chains associated with the edges $(W_{k1}, W_{k+1,2})$ in $\mathcal{T}$. They are $I_1 = \{(W_{11}, W_{22}), (W_{21}, W_{32}), (W_{31}, W_{42}), (W_{41}, W_{12})\}$ and $I_2 = \{(W_{61}, W_{72}), (W_{71}, W_{62})\}$. Then, $I = I_1 + I_2 + \{(W_{i1}, W_{i2})\}$. Evidently, the minimum workforce requirement in S is $W_{\min} = 10$. Correctness of O-2SmW is proved next.

Theorem 1 *Algorithm O-2SmW correctly solves problem 2SmW in* $\mathcal{O}(n \log n)$ *time.*

Proof: The matching I produced in line [2] of O-2SmW uses 3 kinds of edges: i) edges in I_0, ii) edges of $E(B)$ that correspond to edges of the maximin tree $\mathcal{T}$, and edges of the form $(W_{i_j+r_j,1}, W_{i_j,2})$ for $j = 1, 2, \ldots, l$. First, we will show that these edges indeed produce a CHC.

Then, if W_I is the smallest cost among the edges of I, we will show that

$$W_I = min\{ \min_{(C_i,C_j)\in T} w_{ij}, \min_{1\le i\le n}\{W_{i1}+W_{i2}\}\}.$$

Corollary 1 states that

$$W_{\min} \le \min_{(C_i,C_j)\in T} w_{ij}.$$

This inequality together with Lemma 1 would imply that $W_{\min} \le W_I$ and since $I \cup M$ is a CHC, $W_{\min} = W_I$ and hence algorithm O-2SmW must yield an optimal solution.

Let us first show that $I \cup M$ is a CHC. Consider the chain of edges in I:

$$I_j = \{(W_{i_j,1}, W_{i_j+1,2}), (W_{i_j+1,1}, W_{i_j+2,2}), \ldots, (W_{i_j+r_j-1,1}, W_{i_j+r_j,2}), (W_{i_j+r_j,1}, W_{i_j,2})\}.$$

for $j = 1, 2, \ldots, l$. In Figure 3 it is clear that the chain I_j merges together all the subcycles that traverse a node from the chain. Recall that, the edges of $\mathcal{T}$ form a spanning tree of G and that the corresponding edges of $E(B)$ are edges of I. Hence, the chains $I_1, I_2, \ldots, I_l$ merge all subcycles of $I_0 \cup M$ into a single subcycle.

To complete the proof let us compute W_I. To do this, note that the edge $(W_{i_j+r_j,1}, W_{i_j,2}) \in I_j$ has cost $W_{i_j+r_j,1} + W_{i_j,2} \ge W_{i_j+r_j,1} + W_{i_j+r_j,2}$ because $W_{i_j,2} \ge W_{i_j+r_j,2}$ from the ordering of the W_{i2}-values. Hence, the fact $W_{i_j+r_j,1} + W_{i_j+r_j,2} \ge \min_{1\le i\le n}\{W_{i1}+W_{i2}\}$ together with Lemma 1 imply that the edge $(W_{i_j+r_j,1}, W_{i_j,2}) \in I_j$ never determines W_I. Therefore, W_I must be attained either by some edge in $I_0 \cap I$, or an edge in $I_j - \{(W_{i_j+r_j,1}, W_{i_j,2})\}$ for some $j \in \{1, 2, \ldots, l\}$.

If W_I is attained by an edge in $I_0 \cap I$, then $W_I = \min_{1\le i\le n}\{W_{i1}+W_{i2}\}$ and $\min_{(C_i,C_j)\in T} w_{ij} \ge \min_{1\le i\le n}\{W_{i1}+W_{i2}\}$ in which case W_I equals the smaller of the 2 minimums. Otherwise, W_I is attained by an edge in $I_j - \{(W_{i_j+r_j,1}, W_{i_j,2})\}$ for some $j = 1, 2, \ldots, l$. These edges correspond to edges in $\mathcal{T}$ and hence the smallest cost among edges in $\mathcal{T}$ is the same as the smallest cost among edges in $\cup_{j=1}^{l} I_j$. Equivalently, in this case $W_I = \min_{(C_i,C_j)\in T} w_{ij}$. This completes the correctness part of the theorem.

To verify the complexity of O-2SmW, note that the total effort spent in the for-loop is $\mathcal{O}(n)$ because every node in $B(U,V)$ is visited at most twice. Identifying the subcycles $C_1, C_2, \ldots, C_r$ requires a single traversal of the graph $I_0 \cup M$. Since this is a union of subcycles, this also takes $\mathcal{O}(n)$ time. Hence, the complexity of O-2SmW is determined by the sorting of the W_{i1}- and W_{i2}-values which requires $\mathcal{O}(n \log n)$ time. This completes the proof of the theorem. □

In the next section we use O-2SmW to develop a lower bound on the optimal range R^* for 2SRW. Then, we solve a sequence of complementary Hamiltonian cycle subproblems.

4 An Optimal Algorithm for Problem 2SRW

In this section we develop an efficient algorithm for 2SRW. The complexity of problem 2SRW has been considered in Vairaktarakis and Cai, 2001.

Theorem 2 *(Vairaktarakis and Cai, 2001)*

Problem 2SRW is $\mathcal{NP}$-complete in the strong sense.

In light of Theorem 2, the best way to address the problem is via enumerative methods. Such methods are significantly expedited with the use of lower bounds. An efficient lower bound is developed next.

4.1 A Lower Bound

As we saw in Section 2 problem 2SRW is equivalent to $\min_x \max_{1 \leq k < k' \leq n} \Delta_{k,k'}$ subject to constraints (1)-(3). Let $S^\star$ be an optimal schedule for 2SRW with range value $R^\star$ equal to

$$R^\star = W_{k_0} - W_{k'_0} \quad for \quad some \quad 1 \leq k_0 \neq k'_0 \leq n.$$

Then, W_{k_0}, $W_{k'_0}$ are the largest and smallest number of workers respectively required by $S^\star$ over the n production cycles. Consider the subproblems 2SmW and

$$(\textbf{2SMW}) \quad W_{\max} = Min_{x_{ij}} Max_k \quad \sum_{i=1}^{n} \sum_{j=1}^{2} W_{ij} x_{i,(k-j+1) \bmod n}$$
$$s.t. \qquad (1)-(3).$$

Problem 2SMW finds a schedule with the smallest possible value for the maximum workforce size over all periods of the production horizon. Hence,

$$W_{k_0} \geq W_{\max}.$$

Problem 2SmW finds a schedule with the largest possible value for the minimum workforce size over the production horizon. Hence,

$$W'_{k_0} \leq W_{\min}.$$

Combining these two inequalities we obtain the following lower bound.

Lemma 3

$$LB = W_{\max} - W_{\min} \leq R^*.$$

An optimal algorithm 2SMW is presented in Lee and Vairaktarakis, 1997. The complexity of 2SMW is $\mathcal{O}(n \log n)$. Algorithm O-2SmW requires $\mathcal{O}(n \log n)$ time as well (Theorem 1). Hence, this is also the time required to compute LB. The following theorem provides a range of values for W_{k_0} and W'_{k_0}.

Theorem 3 *Let $R^\star = W_{k_0} - W_{k_0'}$ for some $1 \leq k_0 \neq k_0' \leq n$ Then,*

$$W_{\max} \leq W_{k_0} \leq W_{\min} + R, \quad \textit{and} \quad W_{\max} - R \leq W_{k_0'} \leq W_{\min},$$

where $R = \min\{R_M, R_m\}$ and R_M, R_m are the range values associated with the solutions obtained by 2SMW and 2SmW respectively.

Proof: First we show $W_{\max} \leq W_{k_0} \leq W_{\min} + R$. We have already seen the left part of this inequality. Since $R^\star = W_{k_0} - W_{k_0'}$ we have $W_{k_0} = R^\star + W_{k_0'}$. The solutions of 2SMW and 2SmW are not necessarily optimal for 2SRW and hence $R \geq R^\star$. Also, we have seen that $W_{k_0'} \leq W_{\min}$. Hence, $W_{k_0} \leq W_{\min} + R$. Proving $W_{\max} - R \leq W_{k_0'} \leq W_{\min}$ is analogous. This completes the proof of the theorem. □

Example: As an example, consider the problem in Table 1. In Section 3 we saw that the optimal sequence is $S_m = J_1J_4J_3J_5J_7J_6J_2$ and $W_{min} = 10$. The maximum worforce requirement for S_m is 19 and hence $R_m = 9$. Applying the algorithm of Lee and Vairaktarakis for 2SMW on the instance of Table 1 yields the sequence $S_M = J_1J_2J_3J_5J_7J_6J_4$ and $W_{\max} = 17$. The minimum worforce requirement for S_M is 7 and hence $R_M = 10$. Therefore, $R = \min\{9, 10\} = 9$ is an upper bound on R^*. Then, according to Theorem 3, $W_{k_0} \in [17, 19]$ and $W_{k_0'} \in [8, 10]$.

In the next subsection we use the ranges provided in Theorem 3 within a 2-dimensional search framework to solve problem 2SRW optimally.

4.2 Solution Algorithm

One way to identify an optimal solution for 2SRW is the following. Associated to every edge (W_{i1}, W_{l2}) in $B(V, U)$ is the edge weight $w_{il} = W_{i1} + W_{l2}$. For given bounds W_U and W_L for the largest and smallest number of workers required over the n production cycles, let $B(W_L, W_U)$ be the restriction of $B(V, U)$ on edges with costs w_{il} such that

$$W_L \leq w_{il} \leq W_U.$$

Then, according to Proposition 1 there exists a solution for 2SRW with $W_L \leq W_k \leq W_U$ for every $k = 1, 2, \ldots, n$ if and only if there exists a complementary Hamiltonian cycle for $B(W_L, W_U)$. Iterating over all possible values for W_U and W_L can therefore produce an optimal solution for 2SRW.

Theorem 3 provides a tight range of values for each of W_{k_0}, $W_{k_0'}$. Using these ranges one can employ a 2-dimensional search to compute the optimal range value R^*. For each pair of trial values, say W_U and W_L, we can solve the CHC problem on the graph $B(W_L, W_U)$. The CHC problem is shown to be strongly NP-complete; see Vairaktarakis and Solow, 2001. However, using the algorithm of Vairaktarakis and Solow, 2001 we can solve problems of up to 250 jobs in less than 1 second. Hence, a 2-dimensional search combined with the CHC algorithm is computationally feasible. Depending on how the 2-dimensional search is designed, the resulting computational requirements may be significantly different. We propose 2 alternative search schemes.

In the first scheme we let W_L take on every integer value in $[(W_{\max} - R)^+, W_{\min}]$ where x^+ is the positive part of integer x. For every such W_L we perform bisection search on the range $[W_{\max}, W_{\min} + R]$ to identify the smallest integer W_U for which a CHC exists for the subset of $E(B)$ where $W_L \leq w_{il} \leq W_U$ for every edge (W_{i1}, W_{l2}). The number of integers in $[(W_{\max} - R)^+, W_{\min}]$ and $[W_{\max}, W_{\min} + R]$ is no more than $2 \max_{il} w_{il}$ because $W_{\max}, W_{\min}, R \leq \max_{il} w_{il}$. Hence, the complexity of this approach results to $\mathcal{O}(w_{\max} \log w_{\max})$ CHC subproblems where $w_{\max} = \max_{il} w_{il}$. This approach does not depend on the number of jobs; just the maximum number of workers over all operations. When the number of workers is relatively small, the approach is expected to perform satisfactorily.

An alternative search that involves polynomially many CHC subproblems is the following. Let $Z_1 < Z_2 < \ldots < Z_r$ be the distinct w_{il} values in the range $[(W_{\max} - R)^+, W_{\min}]$. Also, let $Z'_1 < Z'_2 < \ldots < Z'_s$ be the distinct w_{il} values in the range $[W_{\max}, W_{\min} + R]$. Since there are at most n^2 w_{il}-values (because $w_{il} = W_{i1} + W_{l2}$), we have $r, s \leq n^2$. For each Z-value perform bisection search to identify the smallest Z'-value for which the associated CHC problem is feasible. This approach requires $\mathcal{O}(n^2 \log n)$ time and hence it is strongly polynomial. For large values of n this approach is expected to be inefficient.

Example: As we saw, Theorem 3 implies that $W_{k_0} \in [17, 19]$ and $W'_{k_0} \in [8, 10]$ for our example. Note that all integers in $[17, 19]$ and $[8, 10]$ are possible $W_{i1} + W_{l2}$ sums. Depending on our choice of 2-dimensional search, different (W_L, W_U) pairs are tested. In both cases, when $W_{k_0} = 17$ we get that the optimal $W_{k'_0}$ is 10 and hence $R = 7$. The sequence obtained is $R_S = J_1 J_2 J_3 J_4 J_5 J_7 J_6$. This sequence requires no more than 17 workers, and no production cycle requires less than 10 workers. This is the best possible R-value over all combinations, so $R^* = 7$. In this particular example the optimal sequence accieves both $W_{\min}$ and $W_{\max}$. In general, this is not possible. Then, it is important to know how well the optimal range schedule performs with respect to the maximin and minimax schedules, and vice-versa. This is done empirically in the next section.

5 Computational Experiments

In our computational experiments we tested the computational efficiency of O-2SRW and the value of the resulting schedule as compared with those obtained by O-2SmW and O-2SMW.

The code for O-2SRW used the code for solving the CHC problem in bipartite graphs that was used in Vairaktarakis and Solow, 2001. Algorithm O-2SMW was reproduced from Lee and Vairaktarakis, 1997. All code was written in C++ and experiments where performed on a PC running at 266 MHz. We considered problem sizes that are multiples of 50, up to $n = 250$ jobs. Fifty problems were solved for each value of n. In all problems, the W_{ij}-values were drawn randomly from the discrete uniform distribution on $[5, 30]$.

We measured the computational efficiency of O-2SmW and O-2SRW in CPU seconds. As expected, algorithm O-2SmW yielded the optimal solution in negligible time for all problems considered including those with 250 jobs. For this reason we do not report these times. The CPU times reported in Table 3 are averages over the 50 problems tested for every value of n. These times do not include the time required to find $W_{\min}$ and $W_{\max}$, however, as we mentioned these times are negligible (less than one hundrendth of a second). Evidently, the CPU times for O-2SRW are in the order of a few seconds and increase slowly with n. Hence, we deem O-2SRW as very efficient.

Table 3: Comparative performance of workforce range and size

n	CPU	$\frac{R(W_{\max})-R^*}{R^*} \times 100\%$	$\frac{W_{\min}-W_{\min}(R)}{W_{\min}} \times 100\%$	$\frac{W_{\max}(R)-W_{\max}}{W_{\max}} \times 100\%$
50	0.346	3.45	0.57	0.51
100	0.922	4.14	0.62	0.54
150	1.870	4.84	0.64	0.57
200	3.285	5.45	0.66	0.58
250	4.981	6.11	0.66	0.58

To evaluate the comparative performance of the schedules obtained by O-2SmW, O-2SMW and O-2SRW against the objectives $W_{\min}$, $W_{\max}$ and R, we defined the following 3 statistics. Let

$W_{\min}(R)$: the minimum workforce requirement of the schedule obtained by O-2SRW,

$W_{\max}(R)$: the maximum workforce requirement of the schedule obtained by O-2SRW,

$R(W_{\max})$: the range value of the schedule obtained by O-2SMW.

We define the following statistics.
$\frac{R(W_{\max})-R^*}{R^*} \times 100\%$: the relative percentage difference of the range value associated with S_M, from R^*.
$\frac{W_{\min}-W_{\min}(R)}{W_{\min}} \times 100\%$: the relative percentage difference of the minimum workforce requirement associated with S_R, from $W_{\min}$.

$\frac{W_{\max}(R)-W_{\max}}{W_{\max}} \times 100\%$: the relative percentage difference of the maximum workforce requirement associated with S_R, from $W_{\max}$.

These 3 statistics were computed for every problem. In Table 3 we report the average values (per statistic) over the 50 problems tested for each n value. Let S_M, S_m and S_R be the optimal schedules for objectives $W_{\max}$, $W_{\min}$ and R. The first statistic captures the range performance of schedule S_M. Evidently, it deteriorates linearly as n increases, and is in excess of 6% for 250 jobs. In other words, minimizing the number of workers needed to complete all jobs in a 2-station transfer line carries a 6% range premium over the optimal range schedule.

The second statistic captures the performance of S_R with respect to the minimum workforce requirement. As evidenced in Table 3, S_R is within 0.7% of $W_{\min}$. Problem size does not seem to affect this performance. These figures indicate that the range schedule is nearly optimal even if it is evaluated against the $W_{\min}$ objective. The third statistic is used to evaluate S_R against $W_{\max}$. Again, irrespective of problem size, S_R is within 0.6% of the optimal workforce size. Combining the last 2 observations we conclude that the range schedule is nearly optimal for the other 2 objectives.

The main insight from the above observations is that, using the optimal range schedule not only minimizes workforce size disruptions across cycles, but also achieves this performance with a minimal increase in workforce size. Cast differently, a 0.5% increase in workforce can protect a transfer assembly line from unnecessary disruptions in the size of the workforce from one day to the next.

6 Conclusion

In this article we developed tools for the range and maximin workforce leveling objectives for 2-station transfer lines. We used these tools to develop insights on the relative value of the various schedules. Our study is confined to 2 stations for methodological reasons. For three or more stations, the graph theoretic representation does not extend to an easily manageable construct. Theorem 3 still holds but problems 3SMW and 3SmW are already strongly NP-complete subproblems. Hence, integer programming approaches seem to be the likely candidate for more than 2 stations. Completely different approaches are needed. Our future research in this area will focus on such approaches. The observations made in this article on the near optimality of range schedules when evaluated on workforce size, are expected to be even more pronounced for more than 2 stations. Intuitively, the larger the number of stations, the greater the contribution of individual stations to the range. Hence, a schedule that does not optimize the range may have a horrible range performance. These observations indicate the need for further research. Consideration of objective functions other than R and $W_{\min}$ is also desirable.

References

[1] Bartholdi J.J. III (1992). Balancing 2-Sided Assembly Lines (A Case Study). Working Paper.

[2] Bartholdi J.J. III, D.D. Eisenstein (1996). A Production Line That Balances Itself. *Operations Research* 44(1):21-34.

[3] Chen F.F. and E.E. Adam, Jr. (1991). The Impact of Flexible Manufacturing Systems on Productivity and Quality. *IEEE Transactions on Engineering Management* 38:33-45.

[4] Daniels R.L. and J.B. Mazzola (1993). A Tabu-Search Heuristic for the Flexible-Resource Flow Shop Scheduling Problem. *Management Science* 41:207-230.

[5] Daniels R.L., B.J. Hoopes and J.B. Mazzola (1997). An Analysis of Heuristics for the Parallel-Machine Flexible-Resource Scheduling Problem. *Annals of Operations Research* 70:439-472.

[6] Ebeling A.C. and C.-Y. Lee (1993). Cross-training effectiveness and profitability. To appear in *International Journal of Production Research.*

[7] Fry T.D., M.G. Wilson and M. Breen (1987). A Successful Implementation of Group Technology and Cell Manufacturing. *Production and Inventory Management* 28:4-6.

[8] Goodrich T.H. (1988). Just-in-Time with an Emphasis on Group Technology. *Manufacturing Systems* 6:78-79.

[9] Gilmore P. C. and R. E. Gomory. Sequencing a One State-Variable Machine: A Solvable Case of the Traveling Salesman Problem, *Operations Research*, 12:655-679, 1964.

[10] Hyer N.L. (1984). The Potential of Group Technology for U.S. Manufacturing. *Journal of Operations Management* 4:183-202.

[11] Karp R.M. A Patching Algorithm for the Nonsymmetric Traveling Salesman Problem, *SIAM Journal on Computing*, 8:561-573. 1979.

[12] Lee C.-Y. and G. Vairaktarakis. Workforce Planning in Mixed Model Transfer Lines, *Operations Research*, 45(4):553-567, 1997.

[13] Pinto P.A., D.G. Dannenbring and B.M. Khumawala (1981). Branch and Bound and Heuristic Procedures for Assembly Line Balancing With Parallel Stations. *International Journal of Production Research* 19:565-576.

[14] Vairaktarakis G. and X.Q. Cai. Complexity of Workforce Scheduling in Transfer Lines, *Journal of Global Optimization*, submitted, 2001.

[15] Vairaktarakis G., X.Q. Cai and C-Y Lee. Workforce Planning in Synchronous Production Systems, *European Journal of Operations Research*, Forthcoming, 2001.

[16] Vairaktarakis G and D. Solow. Properties of Complementary Hamiltonian Cycles on Bipartite Graphs, Weatherhead School of Management, Dept. of Operations, Case Western Reserve University, Research Report, 2001.

[17] Vairaktarakis G and D. Solow. An Efficient Algorithm for the Complementary Hamiltonian Cycle Problem on Bipartite Graphs, *INFORMS Journal on Computing*, submitted, 2001.

[18] Vollmann, T.E, Berry, W.L. and Whybark, D.C, *Manufacturing Planning and Control Systems*. Irwin/MGraw-Hill, U.S.A., 1997.

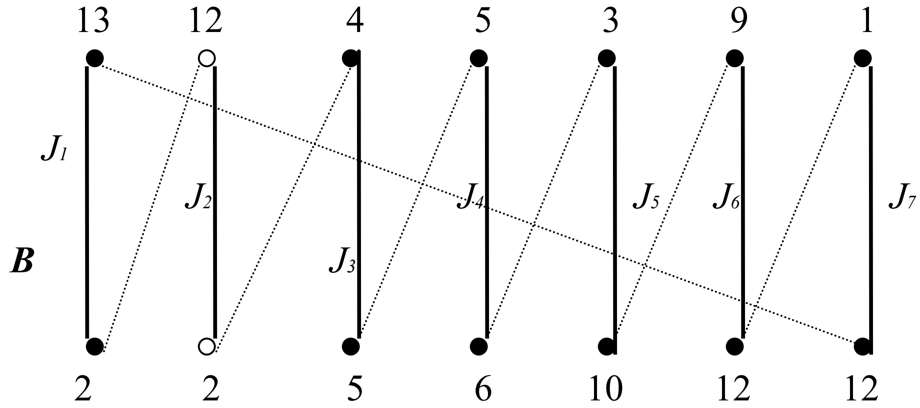

Figure 1: Representation of 2SLW using bipartite graphs

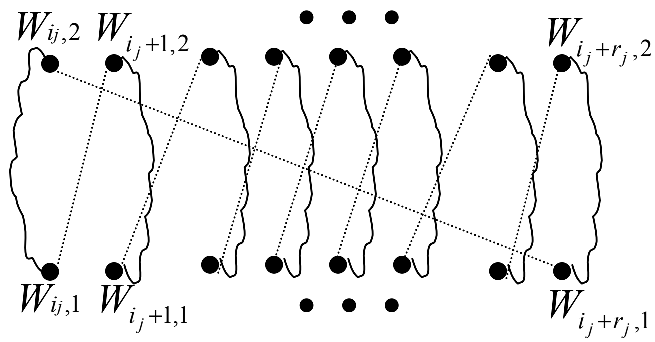

Figure 3: The chain Ij

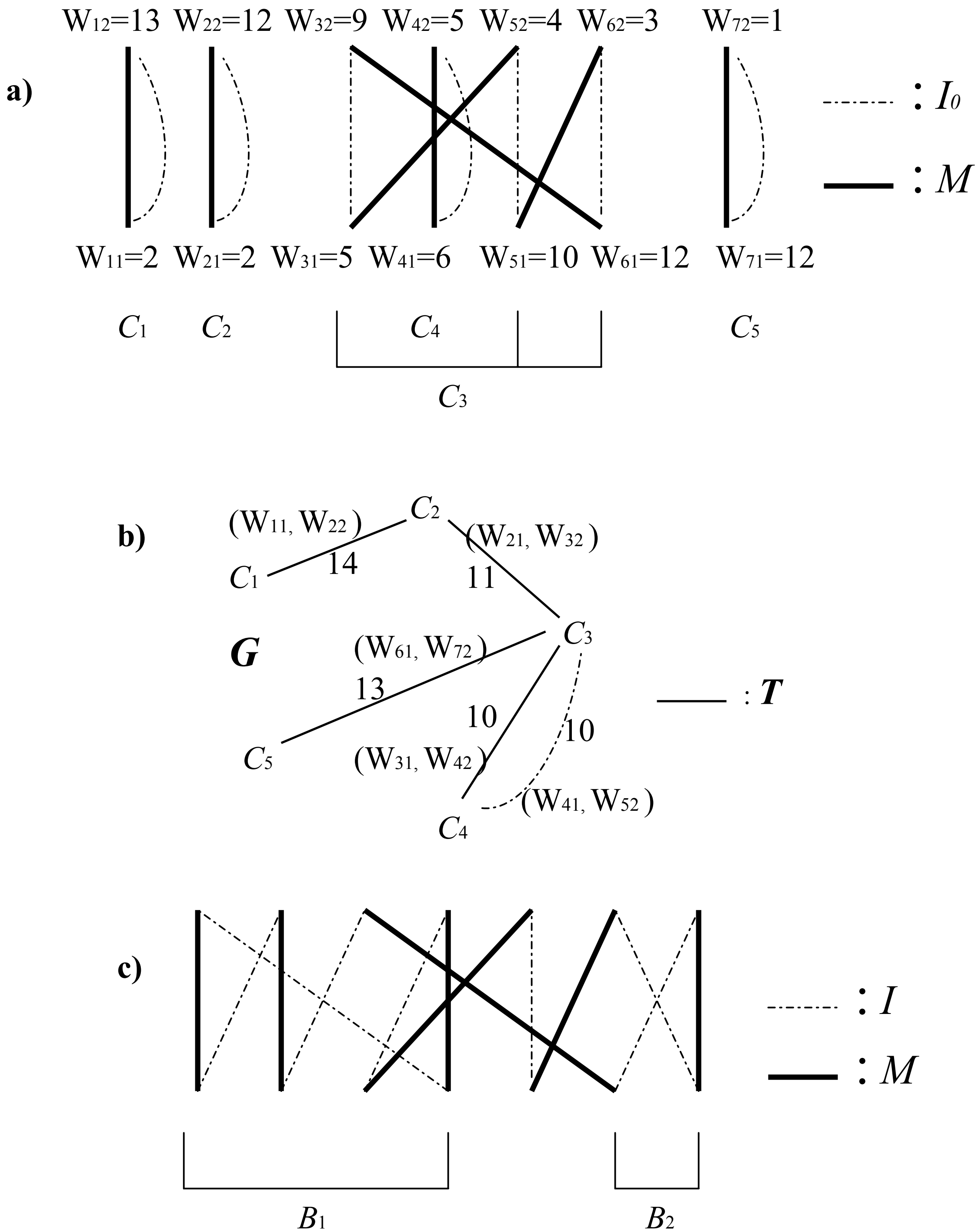

Figure 2: Example and solution algorithm for problem 2SmW