Product Assignment in Flexible Multi-Lines The Case of Single Stage with Demand Splitting

Narayan Raman
Department of Business Administration
University of Illinois

Udatta. S. Palekar
Department of Mechanical
and Industrial Engineering
University of Illinois

Bureau of Economic and Business Research
College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

BEBR

FACULTY WORKING PAPER NO. 92-0177

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

November 1992

Product Assignment in Flexible Multi-Lines
The Case of Single Stage with Demand Splitting

Narayan Raman

Department of Business Administration

Udatta. S. Palekar

Department of Mechanical and
Industrial Engineering

Product Assignment in Flexible Multi-Lines
The Case of Single Stage with Demand Splitting

Narayan Raman

Department of Business Administration

University of Illinois at Urbana-Champaign

Champaign, Illinois

Udatta. S. Palekar

Department of Mechanical and Industrial Engineering

University of Illinois at Urbana-Champaign

Urbana, Illinois

November 1992

Product Assignment in Flexible Multi-Lines
The Case of Single Stage with Demand Splitting

Narayan Raman

Department of Business Administration

University of Illinois at Urbana-Champaign

Champaign, Illinois

Udatta S. Palekar

Department of Mechanical and Industrial Engineering

University of Illinois at Urbana-Champaign

Urbana, Illinois

November 1992

ABSTRACT

This study deals with the Flexible Multi-line Design problem in a serial manufacturing system. Such systems process a variety of products in large volumes with stable demand rates. These products have similar processing requirements in that they visit the various manufacturing stages in the same sequence. Each stage on any line comprises multiple identical CNC machines which perform a set of predetermined tasks on the products assigned to that line. Given the fixed cost of providing a line, and the fixed cost of each workcenter at each stage, the objective of the flexible multi-line design problem is to simultaneously determine the number of lines required as well as find the product-to-line allocation such that the total investment in lines and workcenters is minimized.

In this paper, we consider the special case of a single-stage system in which a product can be assigned to multiple lines. This special case arises as an important subproblem in the general multi-stage problem. However, it merits independent consideration for systems in which the *same* stage is the bottleneck for all products; for such systems, the multi-stage FMD problem reduces to a single-stage problem. In this paper we permit overlapping product partitions, the demand of any product can then be spread across several lines. We develop some characteristics of the optimal solution; in particular, we show that it must satisfy the *sequential assignment property* which renders it solvable in polynomial time using a dynamic programming algorithm. However, we develop an alternative, enumerative solution method that results in a much smaller average running time by making effective use of an imbedded greedy algorithm.

This study considers the problem of designing a flexible multi-line in a serial manufacturing system with multiple stages. Such systems process a variety of products in large volumes with stable demand rates. These products have similar processing requirements in that they visit the various manufacturing stages in the same sequence. Each stage on any line comprises multiple identical CNC machines which perform a set of predetermined tasks on the products assigned to that line. While these tasks require similar processing capabilities, the actual tasks done and their processing times are product-specific. The flexible CNC machines can switch from one product to another with negligible changeover time. The adjacent stages are tightly coupled with minimal buffer storage space in between. Each line is paced, and therefore, its cycle time is constrained by the maximum processing time across all stages required by any product assigned to it.

Given the fixed cost of providing a line, and the fixed cost of each workcenter at each stage, the objective of the flexible multi-line design (FMD) problem is to partition the set of products such that each subset is assigned to exactly one line, and the total investment in lines and workcenters is minimized. This problem is motivated by the manufacturing facility of one of the major auto companies that produces fuel-supply systems. This facility produces a number of different components that go through a number of forming operations during fabrication. At any given stage, the component is subjected to a specific type of forming operation. The tight coupling of the individual stages allows limited in-process buffer so that the entire line is forced to operate in a paced fashion. This problem also arises in printed circuit board manufacture (Farber et al. 1988). Indeed, the FMD problem arises naturally in many systems in the context of implementing a just-in-time approach within cellular manufacture. Given a set of products with their individual demands and processing requirements, FMD determines the optimal set of families, as well as the optimal configuration of the various cells that need to be formed. Additionally, it can be used at periodic intervals to evaluate the need for a system redesign in the face of changing product demands and processing needs.

The problems most closely related to the FMD problem are the mixed-model line balancing problem (Wester and Kilbridge 1964; Thomopolous 1967, 1970; MacAskill 1972; Dar-El 1978; Okamura and Yamashita (1979; and Yano and Rachamadugu 1991) and the line segmentation problem (Ahmadi and Matsuo 1991). Much of the previous work on mixed-model line balancing problem addresses the assignment of tasks required for assembling a number of products to operators stationed along an assembly line. The tasks are general enough in that they can be assigned to any operator on the line as long as the precedence relations among them are satisfied. It is easy to see that in such tandem systems, the cycle time and the overall output are constrained by the total processing time required at the bottleneck station. Consequently, the bulk of the research on this problem has considered the objective of smoothing workload assignments across all stations. Because of the variety of products assembled on this line, the amount of processing required at any station varies from one cycle to another, and workload balancing is based on the average processing time per cycle at each station. Work overloads are relieved by permitting limited operator movement upstream and downstream of the assigned station (Dar-El and Cucuy 1977, Dar-El 1978), or through the use of utility workers (Yano and Rachamadugu 1991). One of the major thrust of this research is on determining the appropriate sequence in which the various models should be processed at each station in order to minimize such overloads. Okamura and Yamashita (1979) address the objective of minimizing the maximum distance that any worker will have to move away from his workstation in order to complete all tasks assigned to him; as Yano and Rachamadugu (1991) note, this objective is similar to minimizing the maximum work overload at any station. Yano and Rachamadugu deal with the objective of minimizing the average work overload given that the overload at any station can be met through the use of utility workers.

An alternative line of research involving mixed-model lines addresses sequencing the various products for the objective of smoothing the rate of parts usage in assembling the final products. This problem was proposed by Monden (1983) in the context of just-in-time manufacture. Miltenberg (1989) considers the problem in which all final products require the same number and mix of parts. Under this assumption, smoothing part usage rate reduces to minimizing the sum of differences between the cumulative actual production and cumulative actual demands across all products. Miltenberg proposes nonlinear integer programming formulations, and proposes heuristic solution methods. Kubiak and Sethi (1991) relax Miltenberg's assumption, and also consider a more general form of the objective function; more importantly, they show that the resulting problem can be formulated as an assignment problem. Similar problems are studied by Miltenberg and Sinnamon (1989) and Inman and Bulfin (1991).

The FMD problem is similar to mixed-model line balancing in that it considers a paced flow line producing multiple products. In addition, the objective of minimizing total investment in lines and workcenters leads to workload balancing. However, these two problems differ in significant ways. First, the assignment of tasks to stations (stages) is not an issue here because any given task can be done only at a predetermined stage. Second, the stages are "manned" by *stationary* CNC machines. Consequently, there can be no variation in the time spent at any station from one cycle to another, and the sequence in which the different models are run is immaterial. Workload balance in our context is achieved purely by the formation of parallel lines and grouping products with similar processing times on a line. While there are economic incentives in having multiple lines in order to reduce idle time, the benefits of doing so need to be traded off against the fixed cost of providing the line.

Ahmadi and Matsuo (1991) consider the line segmentation problem (LSP); for a given number of machines at each stage, and a given partition of products into families such that each family is assigned to one line, the objective of LSP is to allocate machines at each stage to individual lines such that the overall makespan is minimized. They present several heuristics for solving LSP and show their efficacy with respect to valid lower bounds. The FMD problem differs from LSP in two important ways. First, LSP considers a *multi-model* situation in which the entire (daily) demand of any product is produced in one batch before the line changes over to produce the next product. In our *mixed-model* approach, each product is allowed to be produced as often as desired subject to the overall demand constraints. Second, FMD addresses a problem in which product-to-line allocation is done jointly with the determination of the number of lines and the number of workstations required at each stage for each line.

This paper is the first of two papers that together address our research on the FMD problem. In both papers, we consider the special case in which there is only one stage. This special case arises as an important subproblem while solving the general multi-stage problem. However, this case merits independent consideration for systems in which the *same* stage is the bottleneck for all products; for such systems, the multi-stage FMD problem reduces to a single-stage problem. Furthermore, we have encountered several systems that have only one stage. The two papers differ in that this paper considers overlapping product partitions; the demand of any product can then be spread across several lines. In the companion paper (Palekar and Raman 1992), we address the case in which each product is constrained to be produced on only one line.

This paper is organized as follows. The problem formulation is given in §1. We develop some dominance properties in §2 that result in an efficient graph representation of the FMD problem. This representation is used in §3 to generate the optimal solution based on a dynamic programming approach. We also develop an alternative polynomial-time algorithm that makes repeated use of a greedy heuristic algorithm. Both the graph representation and the greedy algorithm play important roles in generating strong bounds and efficient solution methods for the no-demand-splitting case considered in the companion paper. We conclude in §5 with a summary of the main results of this paper.

1 PROBLEM DESCRIPTION

In this section, we present a mixed integer programming formulation of the flexible multiline design problem. However, first we give the notation used in the paper.

$\mathcal{N}$ = the set of products, and $|\mathcal{N}| = N$

F_1 = fixed cost per line

F_2 = fixed cost per machine

p_j = processing time of product j, $j \in \mathcal{N}$

$\mathcal{J}_i$ = set of products with processing times greater than or equal to i, $\{j | p_j \geq p_i, \ j \in \mathcal{N}\}$

d_j = per period demand of product j, $j \in \mathcal{N}$

A = available time per period on any machine

τ_l = cycle time of line l

$\mathcal{L}_l$ = the set of products assigned to line l

γ_j = $|\mathcal{L}_j|$

In any feasible solution, the cycle time τ_l of line l equals the processing time of its *pivot*, i. e., the product with the longest processing time that is assigned to that line. Let $\pi(l)$ denote the pivot of line l, $\lambda(j)$ denote the line for which product j is the pivot, and n_j denote the number of workcenters required at line $\lambda(j)$. Then, the cycle time of any line l with pivot j is $\tau_l = p_j$ and its capacity is An_j/p_j. We assume that $A \gg p_j, \ \forall j \in \mathcal{N}$ so that $\lfloor A/p_j \rfloor \approx A/p_j$.

The flexible multi-line design problem is stated as

FMD1

$$\text{Minimize} \quad Z_1 = \sum_{j=1}^{N} (F_1 y_j + F_2 n_j) \tag{1}$$

subject to

$$\sum_{j\in\mathcal{J}_i} x_{ji} = 1, ; i \in \mathcal{N} \tag{2}$$

$$p_j\left(\sum_{i=1}^{N} d_i x_{ji}\right) \le A n_j, ; j \in \mathcal{N} \tag{3}$$

$$x_{ji} \le y_j, ; i, j \in \mathcal{N} \tag{4}$$

$$x_{ji} \ge 0, ; i, j \in \mathcal{N} \tag{5}$$

$$y_j \in \{0,1\}; n_j \ge 0, ; \text{integer}, ; \; j \in \mathcal{N} \tag{6}$$

where x_{ji} is the fraction of product i's demand assigned to line $\lambda(j)$ and

$$y_j = \begin{cases} 1, & \text{if a line is opened with pivot } j \\ 0, & \text{otherwise.} \end{cases}$$

Equation (2) insures that the demand of each product is fully assigned, and a product is assigned only to lines with cycle times no less than the processing time of the product. Constraint (3) requires that all product-to-line assignments be capacity feasible. Constraint (4) insures that the fixed cost of opening a line is accounted for. Finally, constraints (5) and (6) specify the nature of the variables.

The total number of machines required on any line l with pivot j is

$$n_j = \left\lceil \frac{\sum_{i\in\mathcal{N}} d_i x_{ji} p_j}{A} \right\rceil,$$

and the total idle time on this line l is

$$A n_j - p_j\left(\sum_{i=1}^{N} d_i x_{ji}\right)$$

It is clear that the idle time on this line is reduced by assiging to it products which have processing times close to p_j. Thus, the FMD problem aims at balancing processing times as compared to workloads that is done in a mixed-model line balancing problem. It is also seen that the idle time is unaffected by the sequence in which the various products are processed.

2 DOMINANCE PROPERTIES

In this section, we develop dominance properties, and construct an efficient graph representation of problem **FMD1**.

Proposition 1. *There exists an optimal solution to* **FMD1** *with pivot set* $\mathcal{P} = \{j | j \in \mathcal{N}, y_j = 1\}$ *such that* $p_k \neq p_l$ *for* $k, l \in \mathcal{P}$, $k \neq l$.

PROOF: For any optimal solution σ to **FMD1** that does not have the above property, we construct an alternative solution σ' from σ by merging line $\lambda(k)$ with line $\lambda(l)$ while the assignments on other lines remain unchanged. Then

$$\begin{aligned} Z_2(\sigma) - Z_2(\sigma') &= 2F_1 + F_2 \left\{ \left\lceil \frac{p_l \sum_{t \in \mathcal{L}_l} d_t x_{tl}}{A} \right\rceil + \left\lceil \frac{p_k \sum_{t \in \mathcal{L}_k} d_t x_{tk}}{A} \right\rceil \right\} \\ &\quad - F_1 - F_2 \left\{ \left\lceil \frac{p_l \left(\sum_{t \in \mathcal{L}_l} d_t x_{tl} + \sum_{t \in \mathcal{L}_k} d_t x_{tk} \right)}{A} \right\rceil \right\} \\ &\geq 0 \end{aligned}$$

where the inequality follows from $F_1 \geq 0$, $p_k = p_l$, and the known inequality

$$\lceil a + b \rceil \leq \lceil a \rceil + \lceil b \rceil. \tag{7}$$

Hence, if σ is optimal, then so is σ' and the proof is complete. □

Proposition 2. *There exists an optimal solution to* **FMD1** *in which*

i) if i is not a pivot product, then it is assigned to exactly one line, i.e., $x_{ui} \in \{0, 1\}$ *for all* $i \in \mathcal{N} \backslash \mathcal{P}$ *and* $u \in \mathcal{P} \cap \mathcal{J}_i$.

ii) if i is a pivot product, then it is assigned to at most two lines.

PROOF As before, we show that any solution σ that is optimal to **FMD1** and that does not have the stated property can be modified to yield an alternative optimal solution that does so. Without loss of generality, we assume that σ satisfies proposition 1. Let L be the total number of lines in σ. Renumber these lines so that

$$\tau_1 > \tau_2 > \ldots > \tau_L. \tag{8}$$

Let $D_l = \sum_{i \in \mathcal{N}} d_i x_{\pi(l)i}$ denote the total quantity assigned to line l, $l = 1, 2, \ldots, L$. Construct another solution σ' from σ in the following manner. Rank all products in $\mathcal{N}$ in the nonincreasing order of their processing times. Starting from line 1, assign products from the top of this list, such that the total quantity assigned to line l is D_l. If this results in any product being partially assigned to a line, then allocate the remaining quantity to the subsequent line. Let τ'_l and $\mathcal{L}'_l$, respectively, denote the cycle time of line l and the set of products assigned to line l, $l = 1, 2, \ldots, L$.

Note that σ' satisfies property i) given above. Also note that

$$\tau'_1 = \tau_1. \tag{9}$$

Lemma 1. $\tau'_l \leq \tau_l, \forall l.$

PROOF: Consider the following disjunctive cases:

a) $\underline{min_{i \in \mathcal{L}'_l}\{p_i\} \leq \tau_{l+1}, \quad l = 1, 2, \ldots, L-1}$

From the construction of σ',

$$\tau'_{l+1} \leq min_{i \in \mathcal{L}'_l}\{p_i\} \leq \tau_{l+1}$$

for $l = 1, 2, \ldots, L-1$. The result follows from (9).

b) $\underline{min_{i \in \mathcal{L}'_t}\{p_i\} > \tau_{t+1}, \quad for some t \in \{1, 2, \ldots, L-1}$

Because $\tau_l = max_{q \in \mathcal{L}_l}\{p_q\}$ for any line l, it follows from (8) that

$$\tau_{t+1} \geq p_i, \quad \forall i \in \bigcup_{k=t+1}^{L} \mathcal{L}_k$$

Consequently, together with the fact that the total quantity allocated to each line is the same in both σ and σ', it must be true that in this case,

$$\bigcup_{k=1}^{t} \mathcal{L}_k = \bigcup_{k=1}^{t} \mathcal{L}'_k$$

and $\tau'_{t+1} = \tau_{t+1}$ to yield the desired result. □

Now

$$\begin{aligned} Z_2(\sigma') - Z_2(\sigma) &= \sum_{i \in \mathcal{N}} \sum_{l=1}^{L} \lceil \tau'_l D_l / A \rceil - \sum_{i \in \mathcal{N}} \sum_{l=1}^{N} \lceil \tau_l D_l / A \rceil \\ &\leq 0 \end{aligned}$$

and σ' is optimal. If it satisfies property ii) as well, the proof is complete. Otherwise, merge all those lines which have the same pivot to construct another solution σ'' that satisfies both i) and ii) and, from proposition 1, is optimal as well. □

In the rest of the paper as well, we assume that the products are numbered such that if $i < j, \quad i, j \in \mathcal{N}$, then $p_i \geq p_j$. We also assume that they satisfy proposition 1 which can now restated as

Remark 1. *If $F_1 \geq 0$, then in any optimal solution, $p_k > p_l$ for any $k, l \in \mathcal{P}$ such that $k > l$.*

We now the give the central result of this section.

Proposition 3. *(Sequential Assignment Property) There exists an optimal solution to* **FMD1** *with the property that if $x_{ji} > 1$, then $x_{jq} = 1$ for $q = j+1, j+2, \ldots, i-1$.*

Proof: As before, let σ be an optimal solution to **FMD1** that does not have this property. Then there has to be at least one product $t, j < t \leq i-1$ that is produced on line $\pi(k), k \neq j$. Note that $k \leq t < i$. If $k < j$, then construct σ' from σ by shifting $\delta = d_t x_{kt}$, the demand of t currently allocated to line $\lambda(k)$ to line $\lambda(j)$, and replacing these units on line $\lambda(k)$ by products currently assigned to line $\lambda(j)$ considered in the increasing order of their index starting with j. If $\delta < d_j x_{jj}$, then j continues to remain a pivot, otherwise its entire demand is absorbed by line $\lambda(k)$ and j is replaced as a pivot by some $q, q > j$, with $p_q \leq p_j$. In either case, $Z_2(\sigma') \leq Z_2(\sigma)$, and therefore, σ' is optimal.

If $k > j$, then construct σ' by shifting $\delta = \sum_{q=t+1}^{i} d_q$ units of demand corresponding to products $t+1$ through i from line $\lambda(j)$ to line $\lambda(k)$, and replace these units on line $\lambda(j)$ with products currently assigned to line $\lambda(k)$ considered in the increasing order of their index starting with k. As before, it follows that $Z_2(\sigma') \leq Z_2(\sigma)$, and therefore, σ'' is optimal. Repeating these steps whenever required yields the solution σ' that is optimal to **FMD1** and that satisfies the condition stated in the proposition. □

Hereafter, we deal only with those solutions that satisfy the sequential assignment property. An immediate consequence of the above propositions is that in an optimal solution, if $j, \quad j > 1$ is a pivot in an optimal solution, then it is assigned to at most two lines, namely $\lambda(j) - 1$ and $\lambda(j)$, i.e., $x_{qj} > 0$, only if $q \in \{\pi(\lambda(j)-1), j\}$. Furthermore, $x_{ju} = 1$ for all u, $u = j+1, j+2, \ldots, \pi(\lambda(j)+1)-1$.

Problem **FMD1** can now be represented on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ shown in Figure 1. In this graph, node V_{ij}, which is depicted as ij in the figure, represents an assignment in which product j is produced on line with pivot i. Note that node V_{ij} is feasible only if $p_j \leq p_i$; hence, the upper triangular nature of this graph. Let E_{ij}^{uv} denote the arc leading from V_{ij} to V_{uv}. [As we discuss later, each arc joining two nodes in Figure 1 actually represents a set of arcs.] We can append a dummy sink node T at the end to denote an artificial product. The optimal solution to **FMD1** then corresponds to the shortest path from V_{11} to T.

INSERT FIGURE 1 HERE

We partition arc-set $\mathcal{E}$ into disjoint subsets $\mathcal{H}$, $\mathcal{B}$ and $\mathcal{F}$ where $\mathcal{H}$ is the set of *horizontal* arcs *(h-arcs)* of the form $E_{ij}^{i,j+1}$, while $\mathcal{B}$ is the set of *backward* arcs *(b-arcs)* of the form E_{ij}^{uv} where $u < i$. $\mathcal{F}$ comprises the *forward* arcs *(f-arcs)*.

From the sequential assignment property (SAP), it follows that any path that includes a b-arc is not dominant. Furthermore, in an optimal solution, a pivot must be (at least partially) assigned to its own line (otherwise it is being produced at a higher than required cycle time). Consequently, if V_{ij} lies in an optimal path, then so must V_{ii}. Together with SAP, this implies that V_{ij} is reachable only via node V_{ii} along the path comprising the h-arcs $E_{ii}^{i,i+1} - E_{i,i+1}^{i,i+2} - \ldots - E_{i,j-1}^{ij}$. Therefore, we need consider only those f-arcs that are incident on a pivot, i.e., arcs of the form $E_{ij}^{j+1,j+1}$.

Clearly, product 1 must be the pivot for line 1 in any feasible solution. Consider a path Π in which i and j are adjacent pivots, $j > i$; i.e, Π passes through $V_{11}, \ldots, V_{1,i-1}, V_{ii}, \ldots, V_{i,j-1}, V_{jj}, \ldots$. Let M_{jk} denote the number of machines required on line $\lambda(j)$ corresponding to node V_{jk}. Then the number of machines required at line 1 in Π is

$$n_1 = M_{1,i-1} = \left\lceil \frac{p_1 \sum_{u=1}^{i-1} d_u}{A} \right\rceil .$$

The capacity remaining, hereafter the *remnant*, at line 1 after this assignment is

$$r_{1,i-1} = \frac{AM_{1,i-1}}{p_1} - \sum_{u=1}^{i-1} d_u .$$

Clearly, if $r_{1,i-1} > 0$, then it is optimal to use this capacity for (partially) meeting the demand of product i, so that $x_{1i} > 0$, and in general, the remnant available at any line will be used for producing the pivot product of the next line. Consequently, the number of machines required at line (j) corresponding to node V_{jk} is

$$M_{jk} = \left\lceil \frac{p_j \left(\sum_{u=j}^{k} d_u - r_{i,j-1} \right)}{A} \right\rceil$$

and

$$r_{jk} = \frac{AM_{jk}}{p_j} - \left(\sum_{u=j}^{k} d_u - r_{i,j-1} \right). \tag{10}$$

Note that $r_{jk} < A/p_j$, hence, it is strictly less than one machine's capacity on line $\lambda(j)$, and because $p_j \geq p_k$, for $k > j$, it is less than one machine's capacity on subsequent lines as well.

We now determine the cost of each arc in $\mathcal{G}$. The cost of f-arc $E_{jk}^{k+1,k+1}$ is

$$c_{jk}^{k+1,k+1} = F_1 + F_2 \left\lceil \frac{p_{k+1}\left(d_{k+1} - r_{jk}\right)}{A} \right\rceil \tag{11}$$

From (10), it follows that r_{jk} and therefore, $c_{jk}^{k+1,k+1}$ depend upon i, the pivot for the line immediately preceding $\lambda(j)$. This in turn implies by induction that they depend upon the path selected to reach V_{jk}. While this suggests that the number of arcs in $\mathcal{G}$ is exponential in the number of products, note that the costs of all f-arcs incident upon any node differ by no more than F_2. For example, there are k f-arcs, of the form $E_{1k}^{k+1,k+1}, E_{2k}^{k+1,k+1}, \ldots, E_{kk}^{k+1,k+1}$ that are incident upon $V_{k+1,k+1}$. However, because, $r_{uk}, r_{vk} < A/p_{k+1}$, for for any $u, v \leq k$, we have

$$|c_{uk}^{k+1,k+1} - c_{vk}^{k+1,k+1}| < F_2 \tag{12}$$

This indicates that if two paths reaching the same node differ in cost by F_2 or more, then the longer path is dominated. It is clearly also true that if these paths had the same cost, then the path with the smaller remnant is dominated. Therefore, the set of undominated paths $\Psi_{k+1,k+1}$ reaching node $V_{k+1,k+1}$ comprises only those arcs that are within F_2 of each other in cost but have distinct r_{jk} values. Because there can be no more that A/p_{k+1} such values, $\psi_{k+1,k+1} = |\Psi_{k+1,k+1}| \leq A/p_{k+1}$. And in general, for any node V_{jj}, we have $\psi_{jj} = |\Psi_{jj}| \leq A/min_{i \in \mathcal{N}}\{p_i\} = \psi$. Thus each f-arc shown in Figure 1 represents a set of arcs whose cardinality is less than or equal to ψ and whose costs differ by at most F_2.

The remnant at node $V_{k+1,k+1}$ is

$$r_{k+1,k+1} = \frac{A\lceil M_{k+1,k+1}\rceil}{p_{k+1}} - d_{k+1} + r_{jk}. \tag{13}$$

Note that $r_{k+1,k+1}$ depends upon the arc selected to reach this node, and it can take A/p_{k+1} values.

Now consider the h-arc $E_{jk}^{j,k+1}$. The marginal increase in the number of machines required on line (j) for producing the $(k+1)th$ product, given that products $j+1, j+2, \ldots, k$ are assigned to this line, is

$$\begin{aligned} M_{j,k+1} - M_{jk} &= \left\lceil \frac{p_j \left(\sum_{u=j}^{k+1} d_u - r_{i,j-1} \right)}{A} \right\rceil - \left\lceil \frac{p_j \left(\sum_{u=j}^{k} d_u - r_{i,j-1} \right)}{A} \right\rceil \\ &= \left\lceil \frac{p_j \left(d_{k+1} - r_{jk} \right)}{A} \right\rceil, \end{aligned}$$

where i is the pivot for the line immediately preceding $\lambda(j)$. The cost of h-arc $E_{jk}^{j,k+1}$ is

$$c_{jk}^{j,k+1} = F_2 \left\lceil \frac{p_j \left(d_{k+1} - r_{jk} \right)}{A} \right\rceil \tag{14}$$

From the above discussion, it follows that this cost also depends upon the path selected to reach V_{jk}, and h-arc $E_{jk}^{j,k+1}$ in effect represents a set of undominated arcs whose cardinality is no more than ψ.

3 Solution Algorithms for FMD1

FMD1 can be formulated as a dynamic program in which the stages are the consecutively numbered products, and the states at a given stage i are given by the combination of the pivots, $j, \;\; j \leq i$, to which product i can be assigned, and the remnant available at the pivot. Define $g^*(t)$ to be the minimum cost of reaching stage $t, t = 2, \ldots, T$. The resulting shortest path problem can be stated as

$$Z^* = \min \;\; g^*(T)$$

where

$$g^*(t) = min_{\substack{j \leq t \\ s \leq \psi}} \{g_{jt}^s\}, \tag{15}$$

and

$$g_{jt}^s = \begin{cases} g_{j,t-1}^s + c_{j,t-1}^{jt}, & \text{if } j < t \\ g_{k_s,t-1}^s + c_{k_s,t-1}^{jt}, & \text{if } j = t \end{cases}$$

where,

$$k_s = arg \;\; min_{\substack{i \leq t \\ u \leq \psi}} \left\{ g_{i,t-1}^u + c_{i,t-1}^{jt} | r_{jt} = s \right\}$$

This dynamic program requires evaluating $O(\psi N^2)$ arcs at each stage; hence, it requires an overall computational effort of $O(\psi N^3)$. While this approach solves **FMD1** in polynomial time, note that ψ can be a large number. We now present an alternative algorithm for solving **FMD1** which is likely to be more efficient computationally for most real problems. In addition, this algorithm makes use of a heuristic that we use extensively later for solving problem **P1**.

The proposed algorithm is based on a controlled enumeration of a sequence of upper bounds. First, we discuss the heuristic method that is used for deriving these upper bounds.

3.1 A Greedy Algorithm

Consider a policy that ignores remnant differences at a given pivot. Under this policy, the f-arc leading to any pivot node V_{jj} is selected myopically on the basis of the total cost incurred in reaching that node, and ties are broken in favor of the f-arc that results in the largest remnant at V_{jj}. [From (13), it follows that this tie-breaking rule will select the node with the maximum remnant at stage $j-1$.] Define $g^G(t)$ to be the minimum cost of reaching stage $t, t = 2, \ldots, T$. The resulting shortest path problem can be stated as

$$Z^G = \ min \ \ g^G(T)$$

where

$$g^G(t) = min_{j \leq t}\{g^G_{jt}\}, \tag{16}$$

and

$$g^G_{jt} = \begin{cases} g^G_{j,t-1} + c^{jt}_{j,t-1}, & \text{if } j < t \\ g^G_{h_{(t-1)},t-1} + c^{jt}_{h_{(t-1)},t-1}, & \text{if } j = t \end{cases}$$

where,

$$h_{(t-1)} = arg \ \ max_{i<t}\left\{r_{i,t-1} | g_{i,t-1} + c^{jt}_{i,t-1} = min_{l<t}\left\{g_{l,t-1} + c^{jt}_{l,t-1}\right\}\right\}. \tag{17}$$

Essentially, $V_{h_{(t-1)},t-1}$ is the node at stage $t-1$ on the shortest path from V_{11} to V_{tt}; ties are broken in favor of the node with the larger remnant. Note that because it ignores remnants, any pair of adjacent nodes in the graph considered by this dynamic program is connected by only one arc; it can, therefore, be solved in $O(N^2)$ computation time. Hereafter, we refer to this solution method as **Greedy**, and this graph as $\mathcal{G}^G$. Clearly, because $\mathcal{G}^G$ considers only a subset of arcs in $\mathcal{E}$, the **Greedy** solution is only an upper bound on the optimal solution to **FMD1**. The following result, however, indicates that these two solutions differ by no more than the cost of one machine.

Proposition 4. *Let Z^* be the optimal solution value to* **FMD1**, *and let Z^G denote the solution value of* **Greedy**. *Then, $Z^G < Z^* + F_2$.*

PROOF: Let the set of pivots in the optimal path Π^* between V_{11} and T be $\mathcal{P}^* = \{j_1, j_2, \ldots, j_L\}$. Consider two subgraphs of $\mathcal{G}$ as follows. The first subgraph is $\mathcal{G}^G$. Let Π^G denote the path followed by the **Greedy** solution in this subgraph. Construct the other subgraph $\mathcal{G}^*$ from $\mathcal{G}$ as follows. In order to reach any pivot node $V_{j_l j_l}$, $j_l \in \mathcal{P}^*$, $l \geq 2$, select arc $E^{j_l j_l}_{j_{l-1}, j_l - 1}$. For reaching any other pivot node, select the f-arc with the minimum cost as given by (17). In other words, Π^* is the solution to the dynamic program

$$Z^* = \ min \ \ g^*(T)$$

where

$$g^*(t) = min_{j \leq t}\{g^*_{jt}\},$$

and

$$g^*_{jt} \ = \ \begin{cases} g^*_{j,t-1} + c^{jt}_{j,t-1}, & \text{if } j < t \\ g^*_{k,t-1} + c^{jt}_{k,t-1}, & \text{if } j = t,\ j \notin \mathcal{P}^* \\ g^*_{q,t-1} + c^{jt}_{q,t-1}, & \text{if } j = t,\ j \in \mathcal{P}^* \end{cases}$$

where q is the pivot immediately preceding j in $\mathcal{P}^*$.

Lemma 2. *Let $z^*(t)$ denote the cost of reaching stage t, along path Π^* in $\mathcal{G}^*$. Then, $g^G(t) < z^*(t) + F_2$, for $t = 1, 2, \ldots, N$.*

PROOF: Let superscripts 'G' and '$*$' distinguish the variables under the subgraphs $\mathcal{G}^G$ and $\mathcal{G}^*$, respectively. If these two subgraphs are identical, then the result holds trivially. Note that these two subgraphs can differ only if the f-arc leading to node V_{qq} (say) is not the same as the f-arc selected by Π^*. Note that $q \geq 3$, because there is only one f-arc leading to node V_{22}. Let $t = \tau$ be the first stage where the f-arcs differ. Then

$$g^G(t) = g^*(t) \leq z^*(t), \quad 1 \leq t \leq \tau - 1, \tag{18}$$

and from (17),

$$g^G(\tau) \leq g^*(\tau) \leq z^*(\tau) \tag{19}$$

as well. As shown in Figure 2, let the f-arc used for reaching $V_{\tau\tau}$ in $\mathcal{G}^G$ ($\mathcal{G}^*$) be $E_{u,\tau-1}^{\tau\tau}$ ($E_{v,\tau-1}^{\tau\tau}$), and let $r_{\tau\tau}^G$ ($r_{\tau\tau}^G$) be the corresponding remnant available at $V_{\tau\tau}$. Then it must be true that $r_{\tau\tau}^G < r_{\tau\tau}^*$, since otherwise $E_{v,\tau-1}^{\tau\tau}$ is dominated by $E_{u,\tau-1}^{\tau\tau}$.

From the sequential assignment property, Π^* must next visit either node $V_{\tau+1,\tau+1}$ or node $V_{\tau,\tau+1}$. First consider the case in which Π^* passes through $V_{\tau+1,\tau+1}$. If the remnant difference $r_{\tau\tau}^* - r_{\tau\tau}^G$ results in the saving of a machine along arc $E_{\tau,\tau}^{\tau+1,\tau+1}$ ($\mathcal{G}^*$) with respect to arc $E_{\tau,\tau}^{\tau+1,\tau+1}$ ($\mathcal{G}^G$), then

$$c_{\tau,\tau}^{\tau+1,\tau+1}(\mathcal{G}^*) = c_{\tau,\tau}^{\tau+1,\tau+1}(\mathcal{G}^G) - F_2. \tag{20}$$

Consequently,

$$g_{\tau+1,\tau+1}^G \leq z^*(\tau+1) + F_2 \tag{21}$$

From (19) and (21), it follows that

$$g^G(\tau+1) \leq g_{\tau+1,\tau+1}^G \leq z^*(\tau+1) + F_2 \tag{22}$$

However, in this case,

$$r_{\tau+1,\tau+1}^G > r_{\tau+1,\tau+1}^* \tag{23}$$

and, from (13), that

$$c_{\tau+1,\tau+1}^{\tau+2,\tau+2}(\mathcal{G}^G) \leq c_{\tau+1,\tau+1}^{\tau+2,\tau+2}(\mathcal{G}^*). \tag{24}$$

Therefore,

$$g_{\tau+2,\tau+2}^G \leq g_{\tau+1,\tau+1}^G + c_{\tau+1,\tau+1}^{\tau+2,\tau+2}(\mathcal{G}^G) < z^*(\tau+1) + F_2 + c_{\tau+1,\tau+1}^{\tau+2,\tau+2}(\mathcal{G}^*). \tag{25}$$

and

$$g_{\tau+1,\tau+2}^G \leq g_{\tau+1,\tau+1}^G + c_{\tau+1,\tau+1}^{\tau+1,\tau+2}(\mathcal{G}^G) < z^*(\tau+1) + F_2 + c_{\tau+1,\tau+1}^{\tau+1,\tau+2}(\mathcal{G}^*). \tag{26}$$

This implies that

$$g^G(\tau+2) \leq g_{\tau+1,\tau+2}^G < z^*(\tau+2) + F_2 \tag{27}$$

On the other hand, if the remnant difference $r_{\tau\tau}^* - r_{\tau\tau}^G$ does not result in any machine saving, then it is carried forward to node $V_{\tau+1,\tau+1}$, and

$$g^G(\tau+2) \leq g_{\tau+1,\tau+2}^G < z^*(\tau+2) \tag{28}$$

A similar argument can be used to show that the above results hold also for the case in which Π^* next visits node $V_{\tau,\tau+1}$. By induction, as long as $r_{jt}^G \geq r_{jt}^*$ for any node V_{jt} on path Π^*, if follows that

$$g^G(t) < z^*(t) + F_2.$$

But $r_{jt}^G < r_{jt}^*$ at any node $V_{j\tau'}$ is possible only if a machine is saved at that node in $\mathcal{G}^G$ with respect to Π^*. In that case,

$$g^G(\tau') \leq g_{j\tau'}^G < z^*(\tau')$$

Similar to the above argument, we can then consider the two cases in which Π^* passes through either node $V_{\tau'+1,\tau'+1}$ or node $V_{\tau',\tau'+1}$ to show that

$$g^G(t) < z^*(t) + F_2$$

for all $t > \tau'$. This completes the proof of the lemma. □

The result stated in the proposition follows immediately from the lemma if we substitute N for t, and note that the cost of all arcs leading to node T is zero. □

The following result is derived similar to Proposition 4.

Corollary 1. *Let Z_{ij}^* and Z_{ij}^G be the optimal cost and the cost under* **Greedy** *to reach node V_{ik} from V_{11}. Then, $Z_{ij}^* > Z_{ij}^G - F_2$.*

3.2 An Exact Algorithm

We construct an improvement algorithm for solving **FMD1** exactly that combines corollary 1 with the **Greedy** solution. Suppose that the optimal solution is given by path Π^* with pivots $j_1^*, j_2^*, \ldots, j_L^*$ and solution value Z^*. Let the **Greedy** solution is given by path Π^G with pivots $j_1^G, j_2^G, \ldots, j_L^G$, with solution value Z^G. Consider subgraph $\mathcal{G}^G$. Assume that $\Pi^G \neq \Pi^*$. Note that

$$Z^G \leq g_{j_L^*,N}^G < Z^* + F_2 \tag{29}$$

where the last inequality follows from corollary 1. From $Z^* \leq Z^G$, we then have

$$g_{j_L^*,N}^G - Z^G < F_2 \tag{30}$$

Let $\Gamma_j \triangleq \left\{V_{ij} | g_{ij}^G < g^G(j) + F_2\right\}$ be the set of nodes at stage $j, j = 1, 2, \ldots, N$, that can be reached from V_{11} at a cost not exceeding F_2 of the minimum cost of reaching stage j. Then it must be true that $V_{j_L^* N} \in \Gamma_N$.

Suppose that $\Pi^* \neq \Pi^G$. Consider a node $V_{kN} \in \Gamma_N$. Denote the (fractional) number of machines required at line (k) corresponding to node V_{kN} by m_{kN}; i.e., $M_{kN} = \lceil m_{kN} \rceil$. Let $m_{kN} = \iota_{kN} + f_{kN}$ where ι_{kN} (f_{kN}) is the integer (fractional) part of m_{kN}. If $k = j_L^*$, then the path Π_k^* from V_{11} to V_{kk} in the optimal solution must be different from the shortest path Π_k^G between these two nodes in $\mathcal{G}^G$, and hence, these two paths must differ in at least one f-arc. Traversing $\mathcal{G}^G$ backwards, let $j+1$ be the first stage where Π^* and Π^G differ. Suppose that the f-arcs leading to $V_{j+1,j+1}$ in paths Π^* and Π^G are $E_{ij}^{j+1,j+1}$ and $E_{h_j,j}^{j+1,j+1}$, respectively. Clearly, $V_{ij} \in \Gamma_j$. Let $\Delta_{ij} = r_{ij} - r_{h_j}$ be the remnant difference achieved at $V_{j+1,j+1}$, and $\kappa_{ij} = g_{ij}^G - g_{h_j,j}^G$ be the cost penalty incurred if arc $E_{ij}^{j+1,j+1}$ is selected instead of arc $E_{h_j,j}^{j+1,j+1}$.

If $\Delta_{ij} p_k / A \geq f_{kN}$, then the additional remnant provided by arc $E_{ij}^{j+1,j+1}$ is large enough to absorb the fractional part of the machine required at line (k). Consequently, this switch saves a machine, and the cost of the resulting solution is $g_{kN}^G + \kappa_{ij} - F_2$. After completing the switch, the fractional part of machine remaining at (k) is $1 - (\Delta_{ij} p_k / A - f_{kN})$. On the other hand, if $\Delta_{ij} p_k / A < f_{kN}$, then no machine saving is effected; the fractional part of machine required at line (k) is $f_{kN} - \Delta_{ij} p_k / A$, and the cost of the resulting solution is $g_{kN}^G + \kappa_{ij}$. A vertex v is fathomed if its solution value $\phi_v \geq UB + F_2$. Note that in this case, any completion of that vertex can be no better than the incumbent solution.

Backtracking along each $V_{ij} \in \Gamma_j$ until V_{11} is reached, and pricing each vertex in the manner shown above will eventually lead to the evaluation of all candidate f-arcs that constitute the difference between Π^G and Π_k^*. Π_k^* is clearly the best among all candidate paths that have been enumerated. Repeating this exercise for each $V_{jN} \in \Gamma_N$ will clearly determine the optimal path Π_k^*. We now give a detailed description of the algorithm.

Initial Solution and Pre-Processing

Step 1

i) Solve **FMD1** using **Greedy** with solution value Z^G. Set the current upper bound $UB = Z^G$. Record the **Greedy** solution as the current incumbent.

ii) For $j = 2, 3, \ldots, N$, and $i \leq j$, determine r_{ij} and compute

$$\Delta_{ij} = r_{ij} - r_{h_j j}; \quad \text{and} \quad \kappa_{ij} = g_{ij}^G - g_{h_j j}^G$$

where h_j is defined by (17).

iii) Determine Γ_N. Go to Step 2.

Branching, Updating and Fathoming

Step 2

Construct a search tree $\mathcal{S}$ rooted at T by generating a vertex at level 1 in $\mathcal{S}$ corresponding to each node in Γ_N. For each such vertex v that corresponds to (say) node V_{jN} in $\mathcal{G}^G$, set

$$\theta_v = f_{jN}, \quad \delta_v = p_j/A, \quad \text{and} \quad \phi_v = g_{jN}^G.$$

Determine Γ_j and generate vertices at the second level corresponding to nodes in Γ_j, and similarly generate vertices at other levels in $\mathcal{S}$ such that the descendants of any unfathomed vertex that corresponds to (say) node V_{kk} are vertices corresponding to nodes in Γ_{k-1}. For any vertex v at level 2 or below that is a descendant of vertex u, set

$$\delta_v = \delta_u, \quad \text{and} \quad z_v = \Delta_{ik}\delta_v$$

where v and u correspond, respectively, to nodes V_{ik} and $V_{k+1,k+1}$ in $\mathcal{G}^G$.

If $z_v \geq \theta_u$, then set $\phi_v = \phi_u + \kappa_{ik} - F_2$. If $\phi_v < UB$, then set $UB = \phi_v$, and compute $\theta_v = 1 - (z_v - \theta_u)$. Record v as the incumbent.

If $z_v < \theta_u$, then set $\phi_v = \phi_u + \kappa_{ik}$. Fathom v if $\phi_v \geq UB + F_2$. Else, set $\theta_v = \theta_u - z_v$.

When the entire tree is generated, go to Step 3.

Generation of the Optimal Solution

Step 3

At the end of the procedure, let v be the incumbent vertex, which corresponds to (say) node V_{ij}. Trace the path leading from v to the root vertex in $\mathcal{S}$, and find the nodes in G corresponding to each vertex in this path. These nodes determine the corresponding path in $\mathcal{G}^G$ from V_{ij} to T. Find the shortest path from V_{11} to V_{ij} using **Greedy** to complete the solution.

3.3 An Example Problem

We illustrate the above algorithm with the following 5-product example $F_1 = 50; F_2 = 100; A = 800; d_1 = 60; d_2 = 280; d_3 = 180; d_4 = 1000; d_5 = 1900; p_1 = 5.0; p_2 = 2.5; p_3 = 2.0; p_4 = 1.0; p_5 = 0.45$. At the end of step 1, the **Greedy** solution is $V_{11} - V_{22} - V_{23} - V_{44} - V_{55}$ with a value $z^G = 800$. The g_{jt}^G and κ_{jt} values are shown in Table 1, while the r_{jt} and the Δ_{jt} values are given in Table 2.

INSERT TABLES 1 AND 2 HERE

From Table 1, it can be seen that

$$\Gamma_5 = \{V_{45}, V_{55}\}\,; \Gamma_4 = \{V_{44}\}\,; \Gamma_3 = \{V_{13}, V_{23}, V_{33}\}\,; \Gamma_2 = \{V_{11}, V_{22}\}\,; \Gamma_1 = \{V_{11}\}$$

The enumeration tree is shown in Figure 3, and the details for this tree are given in Table 3. Paths corresponding to the **Greedy** and the optimal solutions are shown on graph $\mathcal{G}^G$ in Figure 4. The optimal path is $V_{11} - V_{22} - V_{33} - V_{44} - V_{55}$ with a value $z^* = 750$. The arcs shared by both paths are shown with double lines while the arcs exclusive to the optimal path are shown in thin bold lines.

4 Conclusion

This paper addresses the flexible multi-line design problem in a single-stage manufacturing system. For a given fixed cost of providing a line, and the fixed cost of each workcenter, the objective of the flexible multi-line design problem is to simultaneously determine the number of lines required as well as find the product-to-line allocation such that the total investment in lines and workcenters is minimized.

In this paper, we consider the case in which a product can be assigned to multiple lines. We show that in this case, the optimal solution must satisfy the *sequential assignment property*, i. e., the products assigned to any line must be consecutively ordered in their processing times. We give a dynamic programming algorithm that solves the problem in polynomial time. We also construct an alternative, enumerative algorithm that results in a much smaller average running time by making use of an imbedded greedy algorithm.

REFERENCES

1. Ahmadi, R. H. and H. Matsuo (1991), "The Line Segmentation Problem," *Operations Research*, Vol. 39, 42–55.

2. Dar-El E. M. (1978), "Mixed-Model Assembly Line Sequencing Problems," *Omega*, Vol. 6, 317–323.

3. Dar-El, E. M. and S. Cucuy (1977), "Optimal Mixed-Model Sequencing for Balanced Assembly Lines," *Omega*, Vol. 5, 333- -341.

4. Farber, M., H. Luss and C.-S. Yu (1988), "Assembly Line Design Tools Line Balancing and Line Layout," Working Paper, AT&T Bell Laboratories, Holmdel, NJ.

5. Inman, R. and R. Bulfin (1991), "Sequencing JIT Mixed-Model Assembly Lines," *Management Science*, Vol. 37, 901–904.

6. Kubiak, W. and S. P. Sethi (1991), "A Note on "Level Schedules for Mixed-Model Assembly Lines in Just-in-Time Production Systems"," *Management Science*, Vol. 37, 121–122.

7. MacAskill, J. L. C. (1972), "Production Line Balances for Mixed Model Lines," *Management Science*, Vol. 19, 423–434.

8. Miltenberg, J. (1989), "Level Schedules for Mixed-Model Assembly Lines in Just-in-Time Production Systems," *Management Science*, Vol. 35, 192–207.

9. Miltenberg, J. and G. Sinnamon (1989), "Scheduling Mixed-Model, Multilevel Assembly Lines in Just-in-Time Production Systems," *International Journal of Production Research*, Vol. 27.

10. Monden, Y. (1983), *Toyota Production System*, Industrial Engineering and Management Press, Institute of Industrial Engineers, Atlanta, GA.

11. Okamura, K. and H. Yamashita (1979), "A Heuristic Algorithm for the Assembly Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor, " *International Journal of Production Research*, Vol. 17, 233-247.

12. Palekar, U. S. and N. Raman (1992), "The Product Assignment Problem in Flexible Multi-Lines: The Single Stage Case," Working Paper # 92–0120, Bureau of Economic and Business Research, Univ. of Illinois at Urbana-Champaign, IL.

13. Thomopolous, N. T. (1967), "Line Balancing-Sequencing for Mixed Model Assembly," *Management Science*, Vol. 14, 69–75.

14. Thomopolous, N. T. (1970), "Mixed Model Line Balancing with Smoothed Station Assignments", *Management Science*, Vol. 16, 593–603.

15. Wester, L. and M. Kilbridge (1964), "The Assembly Line Mixed Model Sequencing Problem," in *Proceedings of the Third International Conference on Operations Research*, Paris, France.

16. Yano, C. A. and R. V. Rachamadugu (1991), "Sequencing to Minimize Work Overload in Assembly Lines with Product Options," *Management Science*, Vol. 37, 572–586.

TABLE 1

Values of g_{jt}^G and κ_{jt} in the Example Problem

j	g_{jt}^G at $t=$ 1	2	3	4	5	κ_{jt} at $t=$ 1	2	3	4	5
1	150	350	450	1050	2250	0	50	50	500	1450
2	-	300	400	700	1300	-	0	0	150	500
3	-	-	450	650	1150	-	-	50	100	350
4	-	-	-	550	850	-	-	-	0	50
5	-	-	-	-	800	-	-	-	-	0
$g^G(t)$	150	300	400	550	800					
h_t	1	2	2	4	5					

TABLE 2

Values of r_{jt} and Δ_{jt} in the Example Problem

j	r_{jt} at $t=$ 1	2	3	4	5	Δ_{jt} at $t=$ 1	2	3	4	5
1	100	140	120	80	100	0	0	-160	0	-1634
2	-	140	280	240	260	-	0	0	160	-1474
3	-	-	360	160	260	-	-	80	80	-1474
4	-	-	-	80	580	-	-	-	0	-1154
5	-	-	-	-	1734	-	-	-	-	0

TABLE 3

Details of the Enumeration Tree

Vertex v	*Corresponding Node in* $\mathcal{G}^G$	θ_v	δ_v	ϕ_v	z_v	*Remarks*
0	T					
1	V_{45}	0.275	0.001250	850		
2	V_{55}	0.025	0.000562	800		Incumbent, $UB = 800$
3	V_{44}	0.025	0.000562	800	0.000	
4	V_{13}	0.475	0.001250	900	-0.200	Fathomed, $\phi_v \geq UB + F_2$
5	V_{23}	0.275	0.001250	850	0.000	
6	V_{33}	0.175	0.001250	900	0.100	Fathomed, $\phi_v \geq UB + F_2$
7	V_{13}	0.034	0.000562	850	-0.090	
8	V_{23}	0.025	0.000562	800	0.000	
9	V_{33}	0.980	0.000562	750	0.045	Current incumbent Revised $UB = 750$
10	V_{12}	0.275	0.001250	900	0.000	Fathomed, $\phi_v \geq UB + F_2$
11	V_{22}	0.275	0.001250	850	0.000	Fathomed, $\phi_v \geq UB + F_2$
12	V_{12}	0.034	0.000562	900	0.000	Fathomed, $\phi_v \geq UB + F_2$
13	V_{22}	0.034	0.000562	850	0.000	Fathomed, $\phi_v \geq UB + F_2$
14	V_{12}	0.025	0.000562	850	0.000	Fathomed, $\phi_v \geq UB + F_2$
15	V_{22}	0.025	0.000562	800	0.000	Fathomed, $\phi_v \geq UB + F_2$
16	V_{12}	0.980	0.000562	800	0.000	Fathomed, $\phi_v \geq UB + F_2$
17	V_{22}	0.980	0.000562	750	0.000	

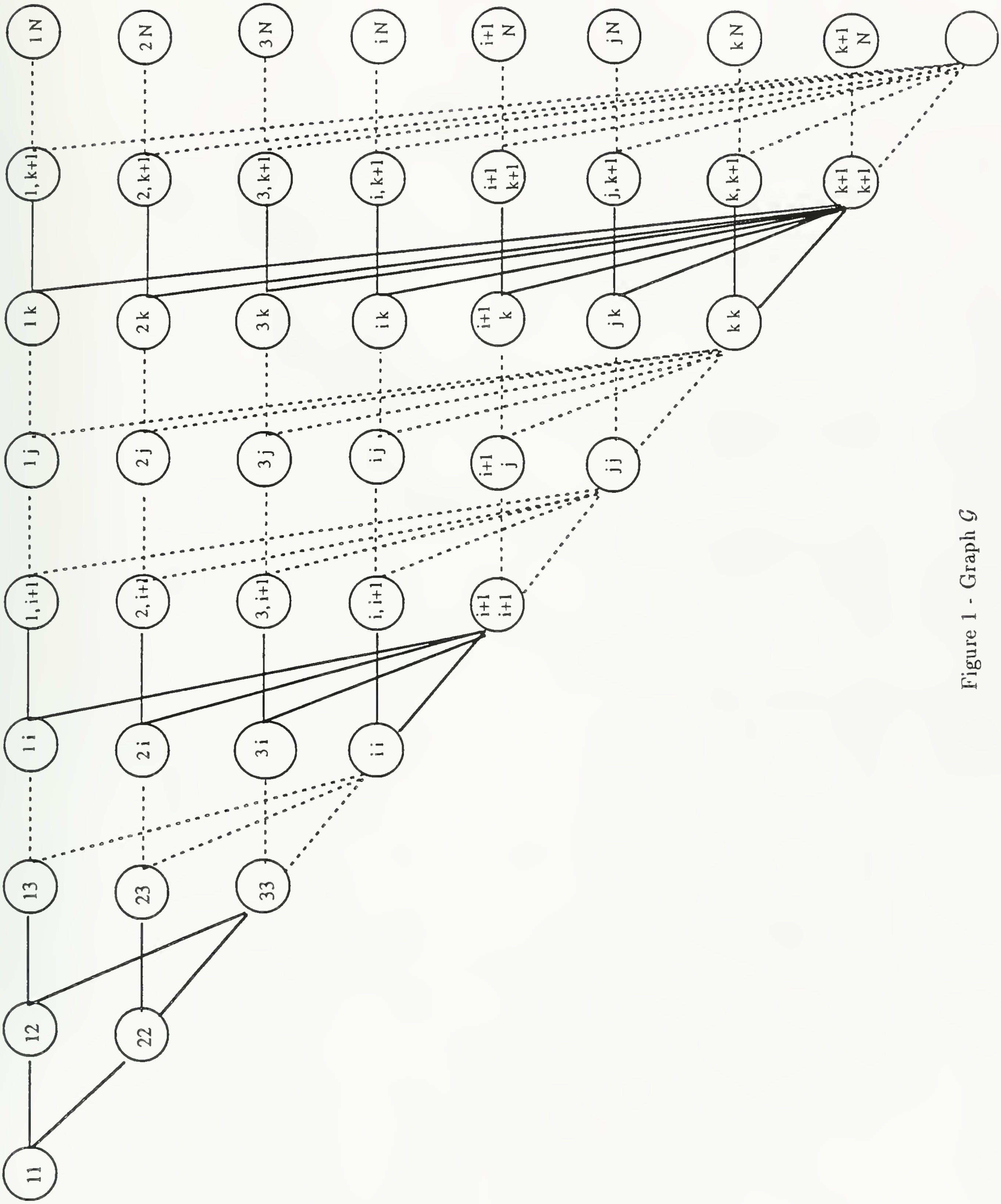

Figure 1 - Graph $\mathcal{G}$

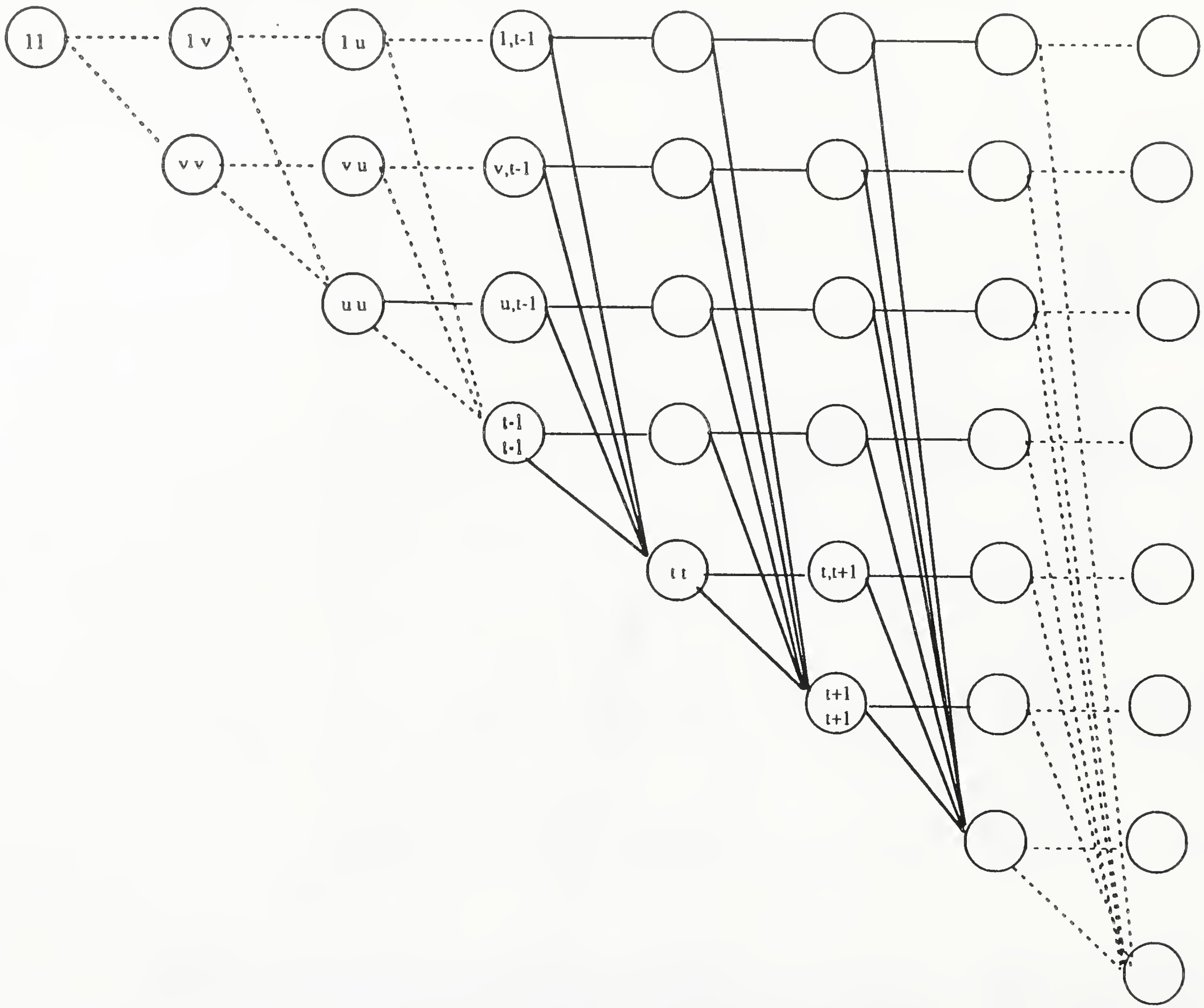

Figure 2 - Proof of Proposition 4

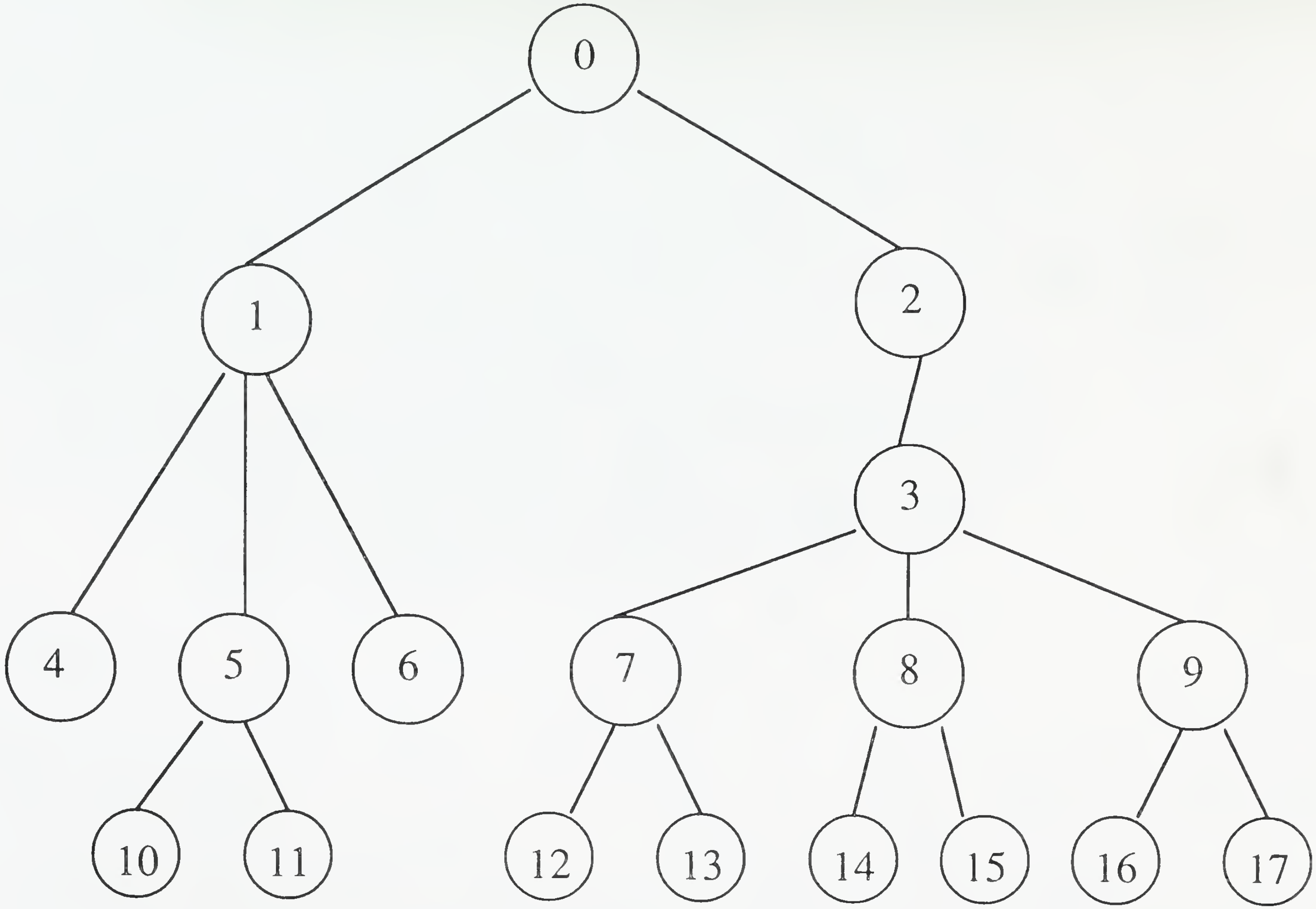

Figure 3 - Enumeration Tree for the Example Problem

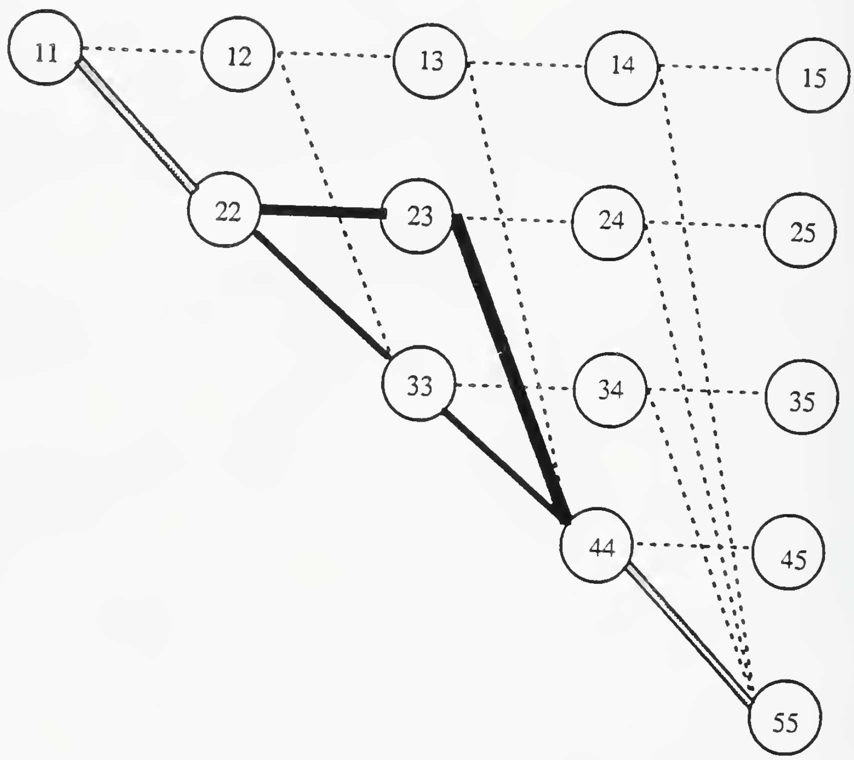

Figure 4 - Paths Π^G and Π^* in the Example Problem