An Effective Scheduling Approach for Maximizing Polyimide Printing Weighted Throughput in Cell Assembly Factories

S. H. Chung, Y. T. Tai, and W. L. Pearn

***Abstract*—Polyimide printing (PI) is an important process operation but also often the cause of bottlenecks in capital-intensive cell assembly factories. Therefore, the development of an effective scheduling method to maximize throughput in this PI process is essential and difficult. In the polyimide printing scheduling problem (PISP), jobs are given weights and clustered by their product types, which must be processed on identical parallel PI machines. The setup times for two consecutive jobs between different product types in the PI machines are sequence-dependent. In this paper, the PISP is formulated as a mixed integer linear programming model. The PISP is also transformed into a multiple tour maximum collection problem (MTMCP), a well-known network problem which has been investigated extensively. Based on this transformation, one can therefore solve the PISP near-optimally using the efficient algorithm.**

***Index Terms*—cell assembly, multiple tour maximum collection problem, scheduling, sequence-dependent setup time.**

I. Introduction

In recent years, applications of thin-film transistor liquid crystal display (TFT-LCD) products have been increasing rapidly, for example, cellular phones, computer monitors, and LCD TVs. Not surprisingly, TFT-LCD manufacturing has attracted much attention. The TFT-LCD companies are being forced to increase their revenue and profits in order to maintain the necessary billion-dollar factories. Therefore, the development of an efficient and effective scheduling method to maximize the weighted throughput and to enhance the utilization of critical resources is essential and important.

The four major stages of TFT-LCD manufacture include the TFT array process, the color filter (CF) process, the cell assembly, and the module assembly, as depicted as Fig. 1. The TFT array and color filter processes are usually referred to as the "front-end," while the two assembly processes are referred to as the "back-end" of production. In the back-end operations, the cell assembly process prints polyimide films onto the cleaned TFT array and CF substrates using the Asahi–Kasei Photosensitive Resin (APR) plate. Then the surfaces are rubbed to orient the polyimide films in one direction. Next, spacers are sprayed onto the TFT substrate and sealants are dispensed onto the CF substrate to maintain a fixed distance between the TFT and CF substrates. Following this, the TFT substrate and the CF substrate are positioned and assembled with an extremely high level of accuracy, and the spaces between them are filled with liquid crystal. Finally, the assembled substrates are sealed and attached to polarizers. Fig. 2 shows the cell assembly process flow.

Polyimide (PI) printing usually causes a bottleneck in production because it involves the most expensive equipment (over three million U.S. dollars each) in cell assembly factories. PI machines print the polyimide films onto TFT/CF substrates uniformly using the APR plate, as depicted in Fig. 3. The polyimide films can help to position the direction of the liquid crystal. For the polyimide printing scheduling problem (PISP) being investigated in this paper, the jobs are stacked on the TFT/CF substrates in cassettes comprising 20 pieces each, which are clustered according to their product types and processed on identical

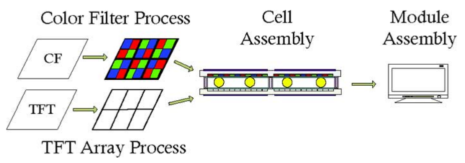

Fig. 1. TFT LCD process flow.

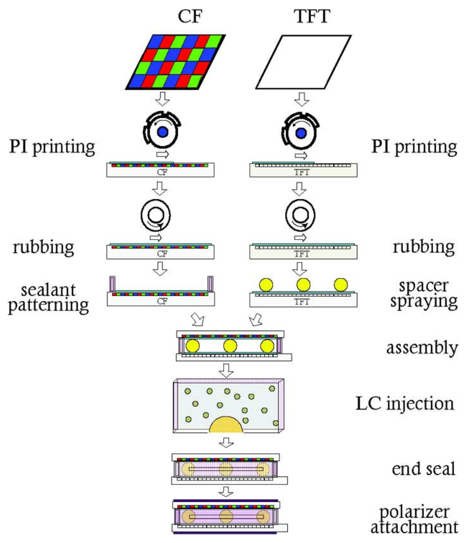

Fig. 2. Cell assembly process flow.

Manuscript received March 22, 2007; revised November 19, 2007 and June 16, 2008. First published June 02, 2009; current version published July 09, 2009. This work was recommended for publication by Associate Editor S. Mason upon evaluation of the reviewers comments.

S. H. Chung and W. L. Pearn are with the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan.

Y. T. Tai is with the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan, and also with the Department of Information Management, Kainan University, Taoyuan33857, Taiwan (e-mail: yttai@mail.knu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEPM.2009.2019340

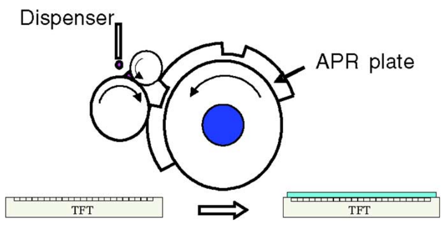

Fig. 3. Side view of the APR plate in a PI machine.

parallel PI machines. Furthermore, the job processing time may vary depending on product type as different product types usually are printed using a specific type of APR plate. Setup times for two consecutive jobs of different product types on the same machine are sequence dependent.

The two major performance criteria for PI printing are the number of moves and the achievement of daily production targets for each product type. A move is defined as the completion of one substrate plate processing on a PI machine. To prevent shop floor operators from simplifying the production line, which results in the production of only one product type, a shop floor scheduler sets daily production target volumes to ensure an efficient production flow for downstream equipment and to produce the required monthly product mix. Initially, a job is given a weight which is determined by the factors such as job's profit, customer importance, and market liquidity. However, the initial weighting can be adjusted in order to meet daily production targets and to reflect the target-related moves by being given a higher weight. In general, jobs clustered in the same product type are given the same weights initially. When jobs are associated with daily production target volumes, they are given higher weights than the other jobs of the same product type in order to schedule them appropriately. In this paper, it is assumed that the weight of each job is usually determined by experts and this process is beyond the scope of this paper.

Since the polyimide printing scheduling problem (PISP) involves constraints on job-dependent processing time, weight, daily production target, machine capacity, and sequentially dependent setup time, it is more difficult to solve than the classical parallel machine scheduling problems considered by So [1] and Tovia *et al.* [2]. In this paper, the criterion of maximum weighted throughput has been used to determine the appropriate subset of jobs for the PISP. Using the three-field notation of Graham *et al.* [3], this scheduling problem is described as $Pm/s_{ii'}, d_{ij} = d/\sum w_{ij}U_{ij}$, where w_{ij} is the weight of job j with product type i and U_{ij} is a 0–1 variable indicating whether the job j with product type i is completed or not in the schedule. In the third field of this notation, minimizing the weighted number of tardy jobs can be viewed as maximizing weighted throughput. In this paper, the PISP is first formulated as a mixed integer linear programming (MILP) model. This model considers different job weights, sequence dependent setup times, and finite machine capacities. Also, it should be noted that the multiple knapsack problem is the special case of PISP mainly because the former unlike the latter, does not consider sequence dependent setup times; that is, PISP is NP-hard in the strong sense. To overcome the limitation of the MILP model; specifically, its inability to solve large-scale problems within a reasonable computational time, an efficient transformation which theoretically converts the PISP into the multiple tour maximum collection problem (MTMCP) is presented here. That is, MTMCP, also referred to as the team orienteering problem (TOP), is a well-known network problem which has been investigated extensively [4]–[8]. The transformation allows the use of the existing algorithm to solve the PISP efficiently.

This paper is organized as follows. Section II presents a brief review of the related literatures. Section III provides the mathematical formulation for the scheduling problem. Section IV shows the network transformation, and Section V gives an illustrative example to demonstrate the transformation. This paper also presents a heuristic algorithm to solve this example and to develop a procedure to identify an upper bound. In addition, real-world applications have been provided in Section VI. Exact solutions and upper bounds are used here as convenient reference points for evaluating the accuracy and effectiveness of the heuristic algorithm. Finally, Section VII provides the conclusion.

II. Literature Review

During the last decade, there have been many researchers who have investigated the identical parallel machine scheduling problem which is dependent on the completion time of all jobs. The objectives usually involve (see Cheng and Sin [9] for a comprehensive survey) completion time-based [10], due-date based [11], [12], and flow-time based [13], [14] performance measures, etc. However, So [1] points out that determining the best schedule to process all the jobs currently on hand is not practical. Instead, he suggests choosing one subset of jobs for which to construct daily work schedules according to the existing capacity and demand. However, the PISP we investigated in this paper selects the appropriate job set to maximize weighted throughput by considering setup times and job weights.

There are relatively few papers, as can be seen in Table I, which addressed the scheduling problem with a throughput objective. Hwang and Chang [15] and Tovia *et al.* [2] have focused on the semiconductor manufacturing process and assumed that setup times are negligible. Hwang and Chang [15] have designed a Lagrangian relaxation-based hierarchical production scheduling engines to maximize total number of weighted moves in semiconductor manufacturing. Tovia *et al.* [2] have presented a mathematical programming model and a rule-based heuristic approach to solve a throughput maximization problem in a semiconductor packaging factory. Furthermore, Baptiste *et al.* [16] and Fung *et al.* [17] have considered the preemptive scheduling problem with equal length processing and negligible setup time. However, setup times and nonpreemptive scheduling characteristics of the polyimide (PI) printing process are essential and significant in cell assembly factories.

Allahverdi *et al.* [20] conducted an extensively survey for the scheduling problems which involve setup times. They pointed out that So [1] is the only one who has considered a total reward objective for parallel machine scheduling problems prior to 1999. So [1] considered an analogous version of the PISP and presented three heuristics to solve the problem approximately. He tackled a problem that existed in a minor setup time between

TABLE I
LITERATURE RELATED TO WITH A THROUGHPUT MAXIMIZATION CRITERION

Authors and Refs.	Setup time	Preemption	Performance criterion (max.)	Shop type
Hwang and Chang [15]	negligible	non-preemptive	total weighted moves	Job shop
Tovia *et al.* [2]	negligible	non-preemptive	throughput	parallel machines
Baptiste *et al.* [16]	negligible	preemptive	total weighted throughput	single machine
Fung *et al.* [17]	negligible	preemptive	total weight	single machine
So [1]	sequence independent batch setup time	non-preemptive	total reward	parallel machines
Hiraishi *et al.* [18]	sequence independent setup time	non-preemptive	weighted number of just-in-time jobs	parallel machines
Rojanasoonthon *et al.* [19]	sequence dependent setup time	non-preemptive	weighted number of scheduled jobs	parallel machines

jobs of the same family and a major setup time between jobs from different groups. According to the classification presented by Allahverdi *et al.* [20], what So [1] considered classified to the sequence independent batch setup times, which is the special case of sequence dependent setup time. Since 1999, with respect to the total weight of completed jobs, Hiraishi *et al.* [18] and Rojanasoonthon *et al.* [19] have maximized the weighted number of just-in-time jobs and the weighted number of scheduled jobs, respectively, as solutions to the scheduling problems they investigated. However, Hiraishi *et al.* [18] in their consideration of JIT job and Rojanasoonthon *et al.* [19] in their examination of strict order of priority in job completion schedule, both limit and reduce the throughput of parallel machines.

The amount of available literature which investigates TFT-LCD factories is limited. Jeong *et al.* [21] presented mathematical models and proposed two heuristic algorithms to minimize flow time and to maximize the fulfillment of production demands in cell assembly processes. Shin and Leon [22] discussed the liquid crystal display module stage scheduling problem. They provided two heuristics based on the MULTI-FIT method and tabu search to minimize total tardiness and number of family setups. This paper investigates the weighted throughput criterion of a TFT-LCD which, till now, has been ignored.

III. INTEGER PROGRAMMING FORMULATION

We formulate the MILP model for PISP and define a product types set $H = \{H_i | i = 0, 1, 2, \ldots, I\}$, containing $I+1$ product types. Each H_i contains a subset $H_i = \{h_{ij} | j = 1, 2, \ldots, J_i\}$, representing the job set, which could be processed on the machine set $M = \{m_k | k = 1, 2, \ldots, K\}$ containing K identical machines. Term J_i is the total job numbers of product type i. We denote that H_0 contains $J_0 = K$ jobs and denotes the idle status of the K parallel machines. Associated with each piece of one job in product type i is a unit processing time denoted by p_i. Let n_{ij} be the job size for each job. Thus, the job processing time can be represented as $n_{ij}p_i$. Associated with each job h_{ij} is a nonnegative weight denoted by w_{ij}. Let *Cap* be the limited machine capacity for each machine. The "minute" is used as the unit for machine capacity. Moreover, a setup time is incurred with different product types. When job h_{ij} immediately succeeds job $h_{i'j'}$ on machine m_k, a setup time $s_{ii'}$ happens. The setup time is a sequence-dependent setup times. Before the mathematical model is presented, the notations of decision variables used in the formulation are listed in the equation at the bottom of the page.

Decision Variables: The definition of $F_k = \{h_{ij} | x_{ijk} = 1; i = 0, 1, 2, \ldots, I, j = 1, 2, \ldots, J_i\}$ is the set of jobs scheduled to be processed on machine m_k. The mathematical formulation of the PISP is shown as follows:

$$\text{Maximize } Z = \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{j=1}^{J_i} x_{ijk} w_{ij} \tag{1}$$

subject to

$$\sum_{k=1}^{K} x_{ijk} \leq 1, \text{ for all } i, j, \tag{2}$$

$$x_{0kk} = 1, \text{ for all } k \tag{3}$$

Capacity Constraints:

$$\sum_{i=0}^{I} \sum_{j=1}^{J_i} x_{ijk} n_{ij} p_i + \sum_{i=0}^{I} \sum_{j=1}^{J_i} \left(\sum_{i'=0}^{I} \sum_{j'=1}^{J_{i'}} z_{iji'j'k} s_{ii'} \right) \leq Cap, \text{ for all } k \tag{4}$$

Precedence Constraints:

$$(y_{iji'j'k} + y_{i'j'ijk}) - Q(x_{ijk} + x_{i'j'k} - 2) \geq 1, \quad \text{for all } i, j, k \tag{5}$$

$$(y_{iji'j'k} + y_{i'j'ijk}) + Q(x_{ijk} + x_{i'j'k} - 2) \leq 1, \quad \text{for all } i, j, k \tag{6}$$

$$(y_{iji'j'k} + y_{i'j'ijk}) - Q(x_{ijk} + x_{i'j'k}) \leq 0, \quad \text{for all } i, j, k \tag{7}$$

$$(y_{iji'j'k} + y_{i'j'ijk}) - Q(x_{i'j'k} - x_{ijk} + 1) \leq 0, \quad \text{for all } i, j, k \tag{8}$$

$$x_{ijk} = \begin{cases} 1, & \text{if job } h_{ij} \text{ is scheduled to be processed on machine } m_k \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{iji'j'k} = \begin{cases} 1 & \text{if job } h_{i'j'} \text{ is scheduled following job } h_{ij} \text{ on machine } m_k \\ 0, & \text{otherwise.} \end{cases}$$

$$z_{iji'j'k} = \begin{cases} 1, & \text{if job } h_{i'j'} \text{ is scheduled immediately following job } h_{ij} \text{ on machine } m_k \\ 0, & \text{otherwise.} \end{cases}$$

$$(y_{iji'j'k} + y_{i'j'ijk}) - Q(x_{ijk} - x_{i'j'k} + 1) \leq 0, \quad \text{for all } i, j, k \tag{9}$$

$$y_{iji*j*k} - Q(y_{iji'j'k} + y_{i'j'i*j*k} - 2) \geq 1, \quad \text{for all } i, j, k \tag{10}$$

$$y_{iji'j'k} \geq z_{iji'j'k}, \quad \text{for all} i, j, k \tag{11}$$

$$\sum_{i=0}^{I}\sum_{j=1}^{J_i} x_{ijk} - \sum_{h_{ij} \neq h_{i'j'}} z_{iji'j'k} = 1, \qquad \text{for all } k \tag{12}$$

$$\sum_{h_{ij} \neq h_{i'j'}} z_{iji'j'k} \leq 1, \quad \text{for all } i, j, k \tag{13}$$

$$\sum_{h_{ij} \neq h_{i'j'}} z_{i'j'ijk} \leq 1, \quad \text{for all } i, j, k \tag{14}$$

Binary Variables:

$$x_{ijk} \in \{0, 1\}, \text{for all } i, j, k \tag{15}$$

$$y_{iji'j'k} \in \{0, 1\}, \text{ for all } i, j, k \tag{16}$$

$$z_{iji'j'k} \in \{0, 1\}, \text{ for all } i, j, k. \tag{17}$$

The objective function (1) states that the total weighted throughput is to be maximized over all the machines. Constraint (2) ensures that each job is scheduled on at most one machine. Constraint (3) guarantees that only one pseudo-job h_{0k} is scheduled on a machine. For instance, the idle status of machine m_1 is assigned pseudo-job h_{01}; thus, $x_{011} = 1$. The advantage of the pseudo-jobs is that they represent the setup times from the idle status to the processing status of first job in order to assist the functioning of the precedence variables. Constraint (4) is the capacity constraint, which forces the summation of the processing times and setup times incurred by the scheduled jobs (involving the pseudo-job) for each machine within the available capacity. In cell assembly factories, jobs are stacked on the TFT/CF substrates in cassettes of 20 pieces. Therefore, the n_{ij} is set to 20 throughout this paper. Due to the manufacturing process being nonpreemptive, the production quantity of job h_{ij} is $x_{ijk}n_{ij}$, and is dependent on whether the job h_{ij} is scheduled on the machine $m_k(x_{ijk} = 1)$ or not $(x_{ijk} = 0)$. Thus, the total production quantity of PISP for each machine is equal to $\sum_{i=0}^{I}\sum_{j=1}^{J_i} x_{ijk}n_{ij}$. Further, the summation of the job processing times on each machine can be represented as $\sum_{i=0}^{I}\sum_{j=1}^{J_i} x_{ijk}n_{ij}p_i$. Constraints (5)–(14) are the precedence constraints; however, constraints (11)–(14) are the directly related ones. Constraints (5) and (6) guarantee that one job should precede another $(y_{iji'j'k} + y_{i'j'ijk} = 1)$ when job h_{ij} and job $h_{i'j'}$ are scheduled on the same machine $(x_{ijk} = 1$ and $x_{i'j'k} = 1)$. Term Q represents a sufficiently large constant, so that constraints (5) and (6) are satisfied for $x_{ijk} + x_{i'j'k} < 2$. Pearn *et al.* [23] suggested that the Q can be chosen as $\sum_{i=1}^{I}\sum_{j=1}^{J_i}(n_{ij}p_i) + \max_{i'}\{s_{ii'}\}$. Moreover, constraint (7) ensures that the precedence variables $(y_{iji'j'k} + y_{i'j'ijk} \leq 0)$ should be equal to 0, when any two jobs h_{ij} and $h_{i'j'}$ are not scheduled on machine m_k $(x_{ijk} + x_{i'j'k} = 0)$. Constraint (8) indicates the situation that one job h_{ij} is scheduled on machine m_k and job $h_{i'j'}$ is scheduled on another machine ($x_{ijk} = 1$ and $x_{i'j'k} = 0$), then the precedence variables ($y_{iji'j'k}$ and $y_{i'j'ijk}$) should be set to zero. Constraint (9) guarantees the situation that the precedence variables ($y_{iji'j'k}$ and $y_{i'j'ijk}$) should be equal to 0 when one job $h_{i'j'}$ is scheduled on machine m_k and job h_{ij} is scheduled on another machine ($x_{ijk} = 0$ and $x_{i'j'k} = 1$). Constraint (10) ensures that job h_{ij} precedes job h_{i*j*} $(y_{iji*j*k} = 1)$ when job h_{ij} precedes job $h_{i'j'}$ and job $h_{i'j'}$ precedes job h_{i*j*} $(y_{i'j'i*j*k} = 1)$.

Constraint (11) is a contingent constraint. That is, if the job h_{ij} could precede job $h_{i'j'}$ directly $(z_{iji'j'k} = 1)$, then the job h_{ij} should precede job $h_{i'j'}$ $(y_{iji'j'k} = 1)$. Constraint (12) indicates that there should exist $n - 1$ direct-precedence variables, which are set to one, on the schedule with n jobs. Constraint (13) ensures that at most one job $h_{i'j'}$ can be scheduled behind job h_{ij} directly for all the jobs, which are scheduled on the same machine m_k. Similarly, constraint (14) ensures that at most one job $h_{i'j'}$ can be preceded job h_{ij} directly for all the jobs, which are scheduled on the same machine m_k. Constraints (15)–(17) indicate that $x_{ijk}, y_{iji'j'k}$ and $z_{iji'j'k}$ are binary integer variables. The total number of variables is $N_I K(2N_I - 1)$, and the total number of constraints is $N_I^3 K + (5/2)N_I^2 K + N_I - (7/2)N_I K + 3K$, where $N_I = \sum_{i=0}^{I} J_i$.

IV. Network Transformation

In this section, we consider a transformation, which converts the PISP into the multiple tour maximum collection problem (MTMCP), which has been investigated extensively [4]–[8]. The first published reference to the MTMCP is in a paper which is concerned with the recruitment of high school athletes for a university football team [4]. From prior experience, the coach involved finds it is impossible to visit all of the surrounding high schools. Therefore, in order to maximize the recruiting results within limited time, the coach decides to visit a best subset of these high schools which represents those that have achieved high potential based on past performance. As a result of this approach, the following options are possible for each tour taken by the coach: 1) no school is visited; 2) one athlete is visited at one school; 3) multiple athletes are visited at the same school; 4) multiple athletes are visited at multiple schools. The transformation steps, which convert the PISP into the MTMCP, are presented as follows.

Step 1) (Network construction) Construct a network $G = [N, E]$ with N representing the set of nodes and E representing the set of edges. In the network $G = [N, E]$, each edge (ℓ, v) in E is associated with distance $d_{\ell v}$ $(\ell, v \in N)$. The node set N consists of the following subsets: N_M and N_{H_i}. The subset $N_M = \{m_k | k = 1, 2, \ldots, K\}$ is the set of depot nodes representing the K machines. Here, N_{H_i} consists of I subsets, $N_{H_1}, N_{H_2}, \ldots, N_{H_I}$, which are the demand nodes representing the I product types.

Step 2) (Network expansion) Copy each node in N_M as its dummy node in $N_{M'}$. The subset $N_{M'} = \{m'_k | k = 1, 2, \ldots, K\}$ is the set of dummy depot nodes duplicated from N_M. Also, add a subset $N_S = \{m_0\}$ which contains a super depot node m_0.

Step 3) (Node collection time assignment) For the super depot node in N_S, the depot nodes in N_M, and the

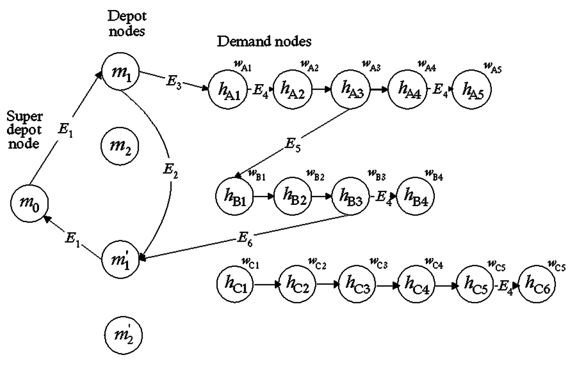
Depot nodes
Demand nodes
Super depot node
m_0
m_1
m_2
m'_1
m'_2
E_1
E_2
E_3
E_4
E_5
E_6
h_{A1} h_{A2} h_{A3} h_{A4} h_{A5}
h_{B1} h_{B2} h_{B3} h_{B4}
h_{C1} h_{C2} h_{C3} h_{C4} h_{C5} h_{C6}
w_{A1} w_{A2} w_{A3} w_{A4} w_{A5}
w_{B1} w_{B2} w_{B3} w_{B4}
w_{C1} w_{C2} w_{C3} w_{C4} w_{C5} w_{C6}

Fig. 4. Transformed network for the PISP with six subsets, E's, of the edge set E.

dummy depot nodes in $N_{M'}$, the node collection time is set to 0. For the nodes in subset N_{H_i}, the node collection time is set to $n_{ij}p_i$, which is equal to the processing time of job h_{ij} in the PISP.

Step 4) (Node reward assignment) For the super depot node in N_S, the depot nodes in N_M, and the dummy depot nodes in $N_{M'}$, the node reward is set to 0. For the nodes in subset N_{H_i}, the node reward is set to w_{ij}, which equals to the job weight of job h_{ij} in the PISP.

Step 5) (Edge distance assignment) Edge set E can be expressed as $E = \cup_{n=1}^{6} E_n$ and depicted as Fig. 4. The definitions of each subset of edge set E are described in detail later in this section. It should be noted here that the value of the edge distance is the setup time incurred with different product types in the PISP.

Edge set E can be expressed as follows, where

$E_1 = (N_S \times N_M) \cup (N_{M'} \times N_S)$ represents a set of directed edges from the super depot node m_0 to the depot nodes N_M, and a set of directed edges from the dummy depot nodes $N_{M'}$ to the super depot node m_0. Each edge (ℓ, v) in E_1 represents distance $d_{\ell v} = 0$. The edges in E_1 ensure that exactly K tours (the total number of machines) will be obtained in any feasible solution. A tour selecting an edge in E_1 indicates that the tour is initiated or terminated.

$E_2 = \{(m_k, m'_k) | m_k \in N_M, m'_k \in N_{M'}; k = 1, 2, \ldots, K\}$ represents a set of directed edges from the depot nodes N_M to the corresponding dummy depot nodes $N_{M'}$. Each edge (ℓ, v) in E_2 represents distance $d_{\ell v} = 0$. The edges in E_2 will be routed to form a tour (machine schedule), if no demand nodes are serviced by that tour (the machine will be idle in this case).

$E_3 = (N_M \times N_{H_i})$ represents a set of directed edges from the depot nodes N_M to the demand nodes N_{H_i}. Each edge (ℓ, v) in E_3 represents the distance equal to the setup time $d_{\ell v} = s_{Ui}$. A tour selecting an edge in E_3 shows that a demand node in N_{H_i} is to be serviced. That is, a job of product type i is scheduled on a machine as the first job to be processed.

$E_4 = \{(h_{ij}, h_{ij'}) | h_{ij}, h_{ij'} \in N_{H_i}; i = 1, 2, \ldots, I; j, j' = 1, 2, \ldots, J_i, j \neq j'\}$ represents a set of directed edges from the same node of demand nodes N_{H_i} in the same product type i. Each edge (ℓ, v) in E_4 represents distance $d_{\ell v} = 0$. A tour selecting an edge in E_4 indicates that the tour services a demand node associated with product type i in N_{H_i} immediately after completing the service of a demand node in the same product type in N_{H_i}.

$E_5 = \{(h_{ij}, h_{i'j'}) | h_{ij} \in N_{H_i}, h_{i'j'} \in N_{H_{i'}}; i, i' = 1, 2, \ldots, I; i \neq i'\}$ represents a set of directed edges in the demand nodes N_{H_i}. Each edge (ℓ, v) in E_5 represents the distance equal to the setup time $d_{\ell v} = s_{ii'}$. A tour selecting an edge in E_5 indicates that the tour services a demand node in N_{H_i} immediately after completing the service of a demand node in $N_{H_{i'}}$.

$E_6 = (N_{H_i} \times N_{M'})$ represents a set of directed edges from the dummy demand nodes N_{H_i} to the dummy depot nodes $N_{M'}$. Each edge (ℓ, v) in E_6 represents the distance equal to the setup time $d_{\ell v} = s_{iU}$. A tour selecting an edge in E_6 indicates that the tour completes the service of its last dummy demand node in N_{H_i} and returns to the depot. That is, a job in N_{H_i} is scheduled as the last job on the machine.

Thus, the transformation converts a PISP with K machines and I product types of jobs containing $N'_I = \sum_{i=1}^{I} J_i$ jobs into the MTMCP. The network that transformed from the PISP to the MTMCP contains $1 + 2K + N'_I$ nodes and $3K + 2N'_I K + \sum_{i=1}^{I} J_i(J_i - 1) + \sum_{i=1}^{I} J_i(N'_I - J_i)$ edges.

In the transformed network, we know MTMCP feasible tours are the following four situations: 1) E_1-E_2-E_1; 2) E_1-E_3-E_6-E_1; 3) E_1-E_3-E_4-E_6-E_1; 4) E_1-E_3-E_4-E_5-E_6-E_1. In the first situation, it represents that the tour services no demand node. It states the recruiter dose not visit any member in surrounding high school football team. In the second situation, it represents that the tour services a demand node from one single cluster. It states that the recruiter only meets with one high school student in just one school. If the recruiter meets with two students in one school, then the edge sequence would be E_1-E_3-E_4-E_6-E_1 in the third situation. In the fourth situation, it represents that the tour services two demand nodes from different clusters (with changing the processing of job clusters only once). It states the recruiter visit two high schools in his limited class time. If three demand nodes with changing the processing of job cluster twice, then the traveling time (including initial setup time (loading time), process time, setup time for two consecutive jobs of different product types on the same machine, and unloading time) would be the same as the total time required for the recruiter (machine) leave his college to meet (process) with football team members (jobs) in different high schools (job clusters). The total scores associated with each high school visited are the same as the weighted throughput of the serviced jobs. Consequently, any feasible MTMCP tour can be converted into a PISP schedule with equal solution value.

On the other hand, any feasible PISP schedule on a machine can be converted into a MTCMP on the transformed network. The job schedules in any PISP solution must also contain the following cases: 1) no jobs are scheduled on the machine; 2) only one job scheduled on a machine; 3) multiple jobs of the same product type are scheduled on the machine; 4) multiple jobs of different product types are scheduled on one machine. If no jobs are scheduled on a machine, then it represents that the machine is in idle status. The tour corresponding to the schedule must be the situation E_1-E_2-E_1, with traveling time and total score are both 0. If only one job scheduled on a machine, then the tour corresponding to the schedule must be the situation E_1-E_3-E_6-E_1. If multiple jobs of the same product type are scheduled on the machine, then the setup time for two consecutive jobs of different product types on the same machine does not occur at that machine. The tour corresponding to the schedule must be the situation E_1-E_3-E_4-E_6-E_1. Clearly, no edges in E_5 will be included in the tour. Finally, if multiple jobs of different product types are scheduled on one machine, then the tour corresponding to the schedule must be the situation E_1-E_3-E_4-E_5-E_6-E_1 for three jobs changing the processing of product type once, and E_1-E_3-E_5-E_4-E_5-E_6-E_1 for four jobs changing the processing of product type twice. The total required for the machines to process those jobs would be the same as the traveling time for the recruiter to visit those high school football team members. The total weighted throughputs of the scheduled jobs are the same as the total scores associated with each high school visited. Consequently, any feasible PISP schedule can be converted into a MTMCP tour with equal solution value.

V. Solutions of PISP

In this section, we present a heuristic procedures proposed by Butt and Cavalier [4] on the transformed network to solve the PISP efficiently. We then present an algorithm to identify an upper bound in order to enable accessing the quality of the solution determined by the heuristic algorithm applied in the PISP. Finally, an illustrative example is provided.

A. Heuristic Algorithm

The maximizing importance of the MTMCP procedure, named MAXIMP (MAXimize IMPortance), developed by Butt and Cavalier [4] on the transformed network is presented. There are two main stages in the MAXIMP algorithm. The first stage constructs the joined weight table (JWGT) list and the reward (RWD) list. The RWD is a list of the job indices in descending order of their corresponding weights. The joined weight table (JWGT) is also a list of the job pairs in descending order of their corresponding joined weights. The joined weight is used to calculate the total weight of a job pair in the time unit. The equations used for this calculation are defined as follows: for all pairs of jobs h_{ij} and $h_{i'j'}$, where $i = i'$ and $j = j'$ do not hold simultaneously, $t_{h_{ij}h_{i'j'}}$ is equal to $s_{Oi} + n_{ij}p_i + s_{ii'} + n_{i'j'}p_{i'} + s_{i'O}$, and the term O denotes an idle machine status and *Cap* is the capacity of one machine:

$$W_{1h_{ij}h_{i'j'}} = [(w_{ij} + w_{i'j'})/Cap] \times t_{h_{ij}h_{i'j'}} \quad (18)$$

$$W_{2h_{ij}h_{i'j'}} = [(w_{ij} + w_{i'j'})/t_{h_{ij}h_{i'j'}}] \times Cap. \quad (19)$$

The weight for each job pair $(h_{ij}, h_{i'j'})$ is defined as

$$W_{h_{ij}h_{i'j'}}(\beta) = \beta W_{1h_{ij}h_{i'j'}} + (1-\beta)W_{2h_{ij}h_{i'j'}} \beta \in [0,1]. \quad (20)$$

The two weight types, $W_{1h_{ij}h_{i'j'}}$ and $W_{2h_{ij}h_{i'j'}}$, apply different approach to define the best two-job combinations [4]. $W_{1h_{ij}h_{i'j'}}$ is used to find those job pairs associated with greater weights for the total time needed to setup and process. Thus, the job pairs associated with great weights and setup times would be favored using this weighted type. On the contrary, $W_{2h_{ij}h_{i'j'}}$ generates the great values when $t_{h_{ij}h_{i'j'}}$ is small. Job pairs have small setup times would be favored. Therefore, Butt and Cavalier [4] created the joined weight, which calculated using (20), to obtain the alternative solutions to MTMCP by varying the value of β. After the JWGT list is constructed, some job pairs should be removed from the JWGT list when $t_{h_{ij}h_{i'j'}} > Cap$ because the summation of setup times and processing times exceeds the machine capacity.

In the second stage of the MAXIMP procedure, each route on the parallel PI machines is constructed using the same procedure. First of all, the top job pair in JWGT list is chosen and assigned to one machine m_k. Second, a candidate job is determined by moving down the JWGT list from the current position until a job pair is found, which contains one job assigned to machine m_k and one job that is unassigned. The selected job is then inserted using a TSP calculation. At the same time, the total traveling time must not violate the capacity limitation.

Finally, it needs to perform the final tour check to see whether the weight associated with the job at the top of RWD list is greater than the summation of the weight collected from the jobs assigned to machine m_k. When the occurrence happens, the jobs currently assigned to machine m_k should be removed and assign only the top of job in RWD list to machine m_k. The tour construction and insertion procedures are repeated until all machines are arranged.

B. Upper Bound

An upper bound is also proposed to be used as a convenient reference point for accessing the accuracy of the efficient algorithm applied in the PISP. The PISP tackles the problem characterized by sequence-dependent setup times, for which the setup matrix is an asymmetric matrix. We modify the concept of Yalaoui and Chu [24] to obtain the upper bound. For the estimated setup time, the summation of all minimal setup times among different product types are considered as being the proportional to the load ratio, which is the result of the division of total machine capacity by the total processing time of all jobs. The procedure used to calculate the upper bound is as follows.

Step 1) Calculate the estimated setup capacity C_S which is proportional to the load ratio:

$$C_S = \left\{ \frac{K}{I}\left(\sum_{i=1}^{I} s_{Oi}\right) + \frac{I-K}{I}\left(\sum_{i'=1}^{I} s_{\bullet i'}\right)\right\} \times \frac{K \times Cap}{\sum_{i=1}^{I}\sum_{j=1}^{J_i} n_{ij}p_i}$$

where $s_{\bullet i'} = \min_{1 \le i \le I} s_{ii'}, i = 1, 2, \ldots, I$.

Step 2) Calculate the remaining capacity C_R.

$$C_R = K \times Cap - C_S.$$

Step 3) Define an index called *weight ratio* respecting the division of weight by corresponding processing time.

TABLE II
SETUP TIMES REQUIRED FOR SWITCHING ONE PRODUCT TYPE TO ANOTHER FOR H_A, H_B, AND H_C. (UNIT: MINUTES)

From \ To	O	H_A	H_B	H_C
O	-	120	120	120
H_A	0	0	240	180
H_B	0	300	0	120
H_C	0	120	60	0

Sort the weight ratios in descending order of magnitude. Choose the jobs from the top of the list until the remaining capacity (C_R) is fully utilized.

Step 4) Calculate the upper bound of total weighted throughput from the selected jobs.

C. Illustrative Example

The PISP example contains a set job of three independent product types: H_A, H_B, and H_C. Job cluster 1 (product type A) contains four jobs, h_{A1}, h_{A2}, h_{A3}, and h_{A4}. Job cluster 2 (product type B) contains four jobs, h_{B1}, h_{B2}, h_{B3}, and h_{B4}. Job cluster 3 (product type C) contains three jobs, h_{C1}, h_{C2}, and h_{C3}. One job is stacked in a cassette containing 20 pieces of a particular product; thus, $n_{ij} = 20$ for $i = A, B$, and $C, j = 1, 2, \ldots, J_i$. These jobs are nonpreemptive and can be processed on either one of the two identical PI machines (m_1 and m_2). The unit processing time of each job is unaffected by the machine processing it.

The limited capacity of each machine is 1440 min. The individual plate processing time of the three product types, A, B, and C is 12, 15, and 9 min, respectively; thus, the job processing time of the three product types is 240, 300, and 180 min, respectively. It is assumed that the unit profits for product types A, B, and C equal 95, 110, and 60 dollars, respectively. It is also assumed that the job profits are the initial weights of jobs in the example and the enhanced weight value of the target (α) is 1.2 for all product types. While the target production volume of product type A is 40 pieces and each job size is 20 pieces, two jobs (h_{A1} and h_{A2}) should be weighted by the value of α. Consequently, the weight of h_{A1} (as for h_{A2}) is set at 2280 weighted dollars (wd), ($95 \times 20 \times 1.2 = 2280$: unit profit $\times$ job size $\times$ α, respectively), which is higher than for h_{A3} and h_{A4} ($95 \times 20 \times 1 = 1900$ wd each). Their higher weights ensure that h_{A1} and h_{A2} can easily be assigned to a schedule. The target production volumes of product types B and C are 40 and 20 pieces, respectively. Therefore, the weights of h_{B1}, h_{B2}, and h_{C1} are 2640, 2640, and 1440 wd, respectively. The weights of h_{B3}, h_{B4}, h_{C2}, and h_{C3} are 2200, 2200, 1200, and 1200 wd, respectively.

The setup time required for switching one product type to another type is shown in Table II (the term O denotes an idle machine status).

In the illustrative example where the PISP is transformed into the MTMCP network formation, the five steps are described as follows.

Step 1) (Network construction) Construct a network with 13 nodes (comprising two depot nodes and 11 demand nodes for the 11 jobs to be scheduled on the two machines) and 134 directed edges (including two edges in E_2, 22 edges in E_3, 30 edges in E_4, and 80 edges in E_5) initially.

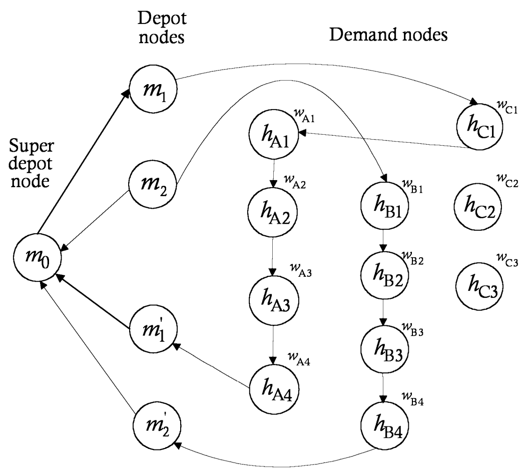

Fig. 5. Routes of the two machines on the transformed network for the PISP.

TABLE III
NODE REWARDS AND TIMES OF COLLECTION ON THE TRANSFORMED NETWORK FOR THE PISP EXAMPLE

node	reward	collection time	node	reward	collection time
h_{A1}	2280	240	h_{B3}	2200	300
h_{A2}	2280	240	h_{B4}	2200	300
h_{A3}	1900	240	h_{C1}	1440	180
h_{A4}	1900	240	h_{C2}	1200	180
h_{B1}	2640	300	h_{C3}	1200	180
h_{B2}	2640	300			

Step 2) (Network expansion) Copy each node in depot nodes N_M as dummy depot nodes $N_{M'}$ (including m_1' and m_2'). Also, add a super depot node $N_S = \{m_0\}$.

Step 3) (Node collection time assignment) For the super depot node (m_0), the depot nodes (m_1 and m_2), and the dummy depot nodes (m_1' and m_2'), the node collection time is set to 0. For the 11 demand nodes, the node collection times are equivalent to their processing times in the PISP and as shown in Table III.

Step 4) (Node reward assignment) For the super depot node (m_0), the depot nodes (m_1 and m_2), and the dummy depot nodes (m_1' and m_2'), the node reward is set to 0. For the 11 demand nodes, the node rewards are equal to the job weight of job h_{ij} in the PISP and are also presented in Table III.

Step 5) (Edge distance assignment) After the constructed network is expanded in Step 2, the transformed network shown in Fig. 5 contains 16 nodes (comprising one super depot node, two depot nodes, two dummy depot nodes, and 11 demand nodes) and 160 directed edges (including four edges in E_1, two edges in E_2, 22 edges in E_3, 30 edges in E_4, and 80 edges in E_5, and 22 edges in E_6). The values of the edge distances between every two nodes correspond to the setup times incurred between the two jobs in the PISP example and are presented in Table IV.

TABLE IV
EDGE DISTANCE MATRIX OF THE TRANSFORMED NETWORK FOR THE PISP EXAMPLE

$u \backslash v$	m_0	m_1	m_2	m'_1	m'_2	h_{A1}	h_{A2}	h_{A3}	h_{A4}	h_{B1}	h_{B2}	h_{B3}	h_{B4}	h_{C1}	h_{C2}	h_{C3}
m_0	-	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-
m_1	-	-	-	0	-	120	120	120	120	120	120	120	120	120	120	120
m_2	-	-	-	-	0	120	120	120	120	120	120	120	120	120	120	120
m'_1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
m'_2	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
h_{A1}	-	-	-	0	0	-	0	0	0	240	240	240	240	180	180	180
h_{A2}	-	-	-	0	0	0	-	0	0	240	240	240	240	180	180	180
h_{A3}	-	-	-	0	0	0	0	-	0	240	240	240	240	180	180	180
h_{A4}	-	-	-	0	0	0	0	0	-	240	240	240	240	180	180	180
h_{B1}	-	-	-	0	0	300	300	300	300	-	0	0	0	120	120	120
h_{B2}	-	-	-	0	0	300	300	300	300	0	-	0	0	120	120	120
h_{B3}	-	-	-	0	0	300	300	300	300	0	0	-	0	120	120	120
h_{B4}	-	-	-	0	0	300	300	300	300	0	0	0	-	120	120	120
h_{C1}	-	-	-	0	0	120	120	120	120	60	60	60	60	-	0	0
h_{C2}	-	-	-	0	0	120	120	120	120	60	60	60	60	0	-	0
h_{C3}	-	-	-	0	0	120	120	120	120	60	60	60	60	0	0	-

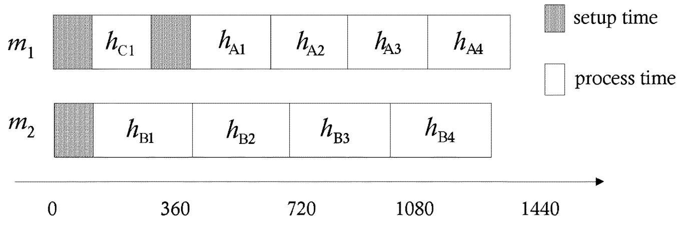

Fig. 6. Heuristic solution for the PISP.

TABLE V
COMPARISON BETWEEN THE OPTIMAL SOLUTIONS AND THE HEURISTIC ALGORITHM

Prob. No.	*J*	*K*	*I*	*Cap*	MILP		Upper Bound		MAXIMP		
					Optimal value	CPU (sec)	*Value*	*Ratio **	*Value*	*Ratio ***	CPU (sec)
1	10	2	3	18	15840	50.78	15840	1.000	15840	1.000	0.03
2	10	2	4	17	14320	159.23	14320	1.000	14320	1.000	0.06
3	10	2	4	19	15760	321.36	15760	1.000	15760	1.000	0.05
4	11	2	3	21	18480	809.09	18480	1.000	18480	1.000	0.03
5	11	2	3	24	19680	20599.61	**20680**	1.051	19480	0.990	0.03
6	11	3	4	17	19540	79.5	19540	1.000	19540	1.000	0.03
7	12	2	3	24	18940	20440	**19940**	1.053	18740	0.989	0.03
8	12	2	4	23	19040	2250.88	19040	1.000	19040	1.000	0.06
9	12	3	4	16	17600	13148.92	**19040**	1.082	17600	1.000	0.06
10	13	3	4	20	23080	36253.81	**24180**	1.048	22980	0.996	0.05

Underlines indicate the heuristic algorithm get the optimal solution.
Boldfaces represent that the upper bound greater than the optimal value.
Symbol * represents the division of the value of upper bound by the value of optimal.
Symbol ** represents the division of the value of weighted throughput by the value of optimal.

TABLE VI
PROBLEM CHARACTERISTICS

Problem characteristics	Values
Variation of processing times	S, L
Variation of setup times	S, L
Number of jobs	75, 100, 150
Number of machines	2, 3, 4
Number of product types	12, 24
Values of α	1.5, 2
Total problem instances	144

It is noted that each tour made by the coach which is to be constructed on the transformed network must begin with the super depot node m_0, pass through the depot node m_k, and pass through the dummy depot node m'_k before ending at the super depot node m_0. Therefore, an initially constructed route appears to be in the form $\{(m_0, m_k, \ldots, m'_k, m_0)\}$. After the first node pair $(h_{ij}, h_{i'j'})$ is selected and inserted in the conducted route using the MAXIMP algorithm, the route can be expressed as $\{(m_0, m_k, \ldots, h_{ij}, h_{i'j'}, \ldots, m'_k, m_0)\}$. By processing the MTMCP algorithm, some unselected nodes can be found and inserted into the current route without violating the available time constraint of each recruiting visit.

In the illustrative example, the subset of the selected nodes involves $\{m_0, m_1, m_2, m'_1, m'_2, h_{A1}, h_{A2}, h_{A3}, h_{A4}, h_{B1}, h_{B2}, h_{B3}, h_{B4},\}$ h_{C1} which have a collective reward score of 19480 using the MAXIMP algorithm. The two visiting routes are $\{(m_0, m_1, h_{C1}, h_{A1}, h_{A2}, h_{A3}, h_{A4}, m'_1, m_0)\}$, and $\{(m_0, m_2, h_{B1}, h_{B2}, h_{B3}, h_{B4}, m'_2, m_0)\}$. Thus, they can be viewed as the production sequences h_{C1}-h_{A1}-h_{A2}-h_{A3}-h_{A4} and h_{B1}-h_{B2}-h_{B3}-h_{B4} on Machine 1 and Machine 2, respectively. The combined weighted throughputs of the two machines are 19480 wd. The Gantt chart for the illustrative example is presented in Fig. 6 and another equivalent representation of the transformation network is depicted in Fig. 5.

The optimal solution in terms of weighted throughput for the PISP example used in this paper is achieved by using the MILP model formulated in Section III is 19680 wd. Simultaneously, this example also has an upper bound of 20680 wd when the first ten jobs are selected. There is no guarantee that the selected ten jobs can form a complete scheduling sequence with limited machine capacity, but the solution generated certainly provides

TABLE VII
COMPUTATIONAL COMPARISONS INVOLVING DIFFERENT β VALUES

Problem instance no.	α	J	K	I	*Upper bound*	*solution ratio* (MAXIMP/Upper bound) $\beta = 0$	$\beta = 0.5$	$\beta = 1$	c.p.u. (sec)
1	2	75	2	12	153080	$\underline{0.967}$	$\underline{0.967}$	0.703	0.91
2	2	75	3	12	196980	$\underline{0.949}$	$\underline{0.949}$	0.825	1.02
3	2	75	4	12	210940	$\underline{1.000}$	$\underline{1.000}$	0.911	1.05
4	2	75	2	24	151120	$\underline{0.982}$	$\underline{0.982}$	0.799	0.91
5	2	75	3	24	188620	$\underline{0.954}$	$\underline{0.954}$	0.860	1.04
6	2	75	4	24	204020	$\underline{1.000}$	$\underline{1.000}$	0.920	1.03
7	2	100	2	12	170520	$\underline{0.958}$	$\underline{0.958}$	0.729	1.66
8	2	100	3	12	224560	$\underline{0.961}$	$\underline{0.961}$	0.815	1.95
9	2	100	4	12	268560	$\underline{0.952}$	0.949	0.840	2.09
10	2	100	2	24	173920	$\underline{0.951}$	$\underline{0.951}$	0.852	1.72
11	2	100	3	24	227340	$\underline{0.958}$	$\underline{0.958}$	0.897	2.02
12	2	100	4	24	263840	$\underline{0.975}$	$\underline{0.975}$	0.893	2.15
13	2	150	2	12	183000	$\underline{0.977}$	$\underline{0.977}$	0.743	3.73
14	2	150	3	12	257800	$\underline{0.946}$	0.926	0.790	4.80
15	2	150	4	12	322920	$\underline{0.966}$	$\underline{0.966}$	0.800	5.53
16	2	150	2	24	187280	$\underline{0.971}$	$\underline{0.971}$	0.828	3.91
17	2	150	3	24	262280	$\underline{0.945}$	$\underline{0.945}$	0.822	4.98
18	2	150	4	24	322280	$\underline{0.967}$	0.954	0.856	5.66
19	1.5	75	2	12	121610	$\underline{0.933}$	$\underline{0.933}$	0.793	0.89
20	1.5	75	3	12	162910	$\underline{0.932}$	$\underline{0.932}$	0.864	1.01
21	1.5	75	4	12	176870	$\underline{1.000}$	$\underline{1.000}$	0.968	1.02
22	1.5	75	2	24	116820	$\underline{0.951}$	$\underline{0.951}$	0.817	0.90
23	1.5	75	3	24	154320	$\underline{0.938}$	0.929	0.880	0.99
24	1.5	75	4	24	169720	$\underline{1.000}$	$\underline{1.000}$	0.975	1.02
25	1.5	100	2	12	130830	$\underline{0.963}$	$\underline{0.963}$	0.770	1.66
26	1.5	100	3	12	178930	$\underline{0.937}$	$\underline{0.937}$	0.826	1.94
27	1.5	100	4	12	221730	0.920	$\underline{0.940}$	0.814	2.07
28	1.5	100	2	24	133860	$\underline{0.965}$	$\underline{0.965}$	0.833	1.74
29	1.5	100	3	24	178960	0.967	$\underline{0.978}$	0.864	2.04
30	1.5	100	4	24	215460	0.977	$\underline{0.979}$	0.883	2.16
31	1.5	150	2	12	138750	$\underline{0.968}$	$\underline{0.968}$	0.757	3.86
32	1.5	150	3	12	198610	$\underline{0.971}$	$\underline{0.971}$	0.782	4.97
33	1.5	150	4	12	255370	$\underline{0.971}$	$\underline{0.971}$	0.755	5.63
34	1.5	150	2	24	142260	$\underline{0.972}$	$\underline{0.972}$	0.818	4.00
35	1.5	150	3	24	199560	$\underline{0.947}$	$\underline{0.947}$	0.811	4.98
36	1.5	150	4	24	246810	$\underline{0.957}$	$\underline{0.957}$	0.839	5.66

Underlines indicate the best solutions among all parameters setting.

an upper bound. In the later section, computational evidence shows that this bound is indeed strong.

VI. REAL-WORLD APPLICATIONS

In this section, ten small-size problem instances and 144 large-scale PISP problem instances are considered. These problem instances are solved using the MAXIMP (for MAXimize IMPortance), proposed by Butt and Cavalier [4] and the upper bound provided by Yalaoui and Chu [24], and described in Section V.

For the purpose of comparing the exact solutions solved by the MILP model formulated in Section III and the MAXIMP algorithm, ten small-size problem instances which are an extension of the example stated in Section V are first experimented. They are generated by randomly linking number of machines and number of product types with various numbers of jobs. For these problem instances, a C++ programming code has been written to generate the constraints and variables of the MILP model. In addition, to obtain optimal solutions, the IP software CPLEX 7.5 on Pentium IV 3.2 GHz PC is used. The results are presented in Table V.

Table V indicates that the solutions solved by MAXIMP reach the optimality in seven problem instances (out of ten) in terms of weighted throughput. Each gap between the optimality and the MAXIMP algorithm is less than 1.1%. The average run times (in CPU seconds) for the MAXIMP algorithm is indeed significantly faster than that of the MILP model. As Table V also shows, the upper bound values are greater then the optimality

TABLE VIII
RESULTS IN MEANS WITH DIFFERENT PROBLEM CHARACTERISTIC GROUPS

Problem characteristics						*Upper bound*	*MAXIMP*	*e* (%)
α	Variation of setup time		Variation of processing time					
	Small	Large	Small	Large	n	Mean	Mean	
2	✓		✓		18	214942.2	206941.5	3.72
2	✓			✓	18	218323.3	210335.6	3.66
2		✓	✓		18	217077.8	208678.3	3.87
2		✓		✓	18	220503.3	212736.7	3.52
1.5	✓		✓		18	168868.9	162783.3	3.60
1.5	✓			✓	18	172204.4	165567.2	3.85
1.5		✓	✓		18	170390.0	163559.4	4.01
1.5		✓		✓	18	174632.2	167985.6	3.81
Average						197048.9	189241.6	3.76

in four problem instances (out of ten) in terms of weighted throughput. The largest gap between the upper bound and optimality is 7.56%, which makes it a tight upper bound.

For the large-scale problem instances, they are randomly generated which cover most of the processing characteristics of the cell assembly process of a TFT-LCD factory in Hsinchu Science-Based Industrial Park, Taiwan. The processing characteristics of these PISP problem instances include 1) variation of the job processing time, 2) variation of the setup times for switch product types, 3) number of jobs, 4) number of machines, 5) number of product types, and 6) enhanced weight value of the target (α), as displayed in Table VI. The job processing times for each product type were generated using uniform [46, 70] for large variation of job processing times and uniform [54, 62] for small variation of job processing times. Furthermore, setup time is one of the essential factors for increasing the impact of results. In the data set investigated, two levels of setup time variations are included. For instance, the large variation of setup times is 9209.7 and the small variation is 2436.3 in the problem instances which included 24 product types. Taking a problem instance which has the characteristics of 100 jobs with $\alpha = 2$, a large variation of job processing times, and a large variation of setup times as an example, the related jobs, product types, processing time, weight, and daily target production jobs are shown in Table A1. The weight settings for the problem instances are determined by the profit of each job which is a good indicator in TFT-LCD factories. The machine capacity is set to 1440 min (one day). The "minute" is used as the unit for the processing time, setup time, and machine capacity. The setup times required for switching one product type to another for the 24 product types are presented in Table A2.

Solving the large-scale PISP problem instances on the transformation network using the MAXIMP algorithm (the program code is written in Visual Basic 6.0), the set of tours on the transformation network is generated. Initially, we experiment with the algorithm on 36 problem instances which have the characteristics of large variations of job processing times and setup times. The β parameters are set to 0, 0.5, and 1 in the MAXIMP which represent the minimum, midpoint and maximum, respectively, of the range used by Butt and Cavalier [4], the originators of this algorithm. The computational results based on different α and β values are shown in Table VII, where the solution ratios are computed as the division of the heuristic solutions by the value of upper bound.

These results indicate that the best solutions to these problem instances can be achieved when $\beta = 0$ and $\beta = 0.5$. Therefore, we limit the choice of β values for this algorithm to $\beta = 0$ and 0.5 and then we choose the best solution (with the maximal weighted throughput) of each problem instance between the two values as the final MAXIMP value. In Table VII, it also shows that the heuristic algorithm can solve the PISP efficiently.

To evaluate the accuracy and effectiveness of the heuristic algorithm, the upper bound is used here as a convenient reference point. The percentage gap e is defined as the percentage of deviation of the best weighted throughput solution from the upper bound (where $e = [(\text{the average value of upper bound} - \text{the average value of the heuristic solutions})/\text{the average value of upper bound}] \times 100$). As Table VIII shows, the average percentage gap in the upper bound is 3.76%. These gaps are partially due to the fact that the upper bounds are obtained from the strictly small setup time choices. The jobs, selected by the upper bound procedure, may not form a complete scheduling sequence, but the total weighted throughput generated certainly provides an upper bound for the solution. Therefore, the MAXIMP algorithm solves the PISP on a transformed network and can obtain a near-optimal solution within a few CPU seconds.

VII. CONCLUSION

This paper considers the PISP involving constraints based on sequence dependent setup time, job-cluster processing time, machine capacity, and daily production target volume, which is difficult to solve and is a great challenge to those involved in the industrial production. In this investigation, the PISP is formulated as a mixed integer linear programming problem model in order to maximize weighted throughput. The transformation which converts the PISP into a multiple tour maximum collection problem (MTMCP), a well-known network problem which has been investigated extensively [4]–[8], has also been presented. An example has been provided to illustrate how we may apply the proposed transformation to convert the PISP into the MTMCP. To demonstrate the applicability of the network transformation, an efficient MTMCP algorithm is applied to solve the real-world PISP in a cell assembly factory. The computational results indicate that the algorithm can provide excellent solutions for real-world PISP within a few CPU seconds. The effective algorithm investigated in this paper may assist those involved in cell assembly factories to make judicious scheduling decisions.

APPENDIX

TABLE A1
JOB INFORMATION OF THE 100 JOBS IN PISP

Job ID	Product type	Processing time (min)	weight	Target Flag*	Job ID	Product type	Processing time (min)	weight	Target Flag
1	1	49	1600		51	11	66	4800	T
2	1	49	1600		52	11	66	4800	T
3	2	60	1800		53	11	66	4800	T
4	2	60	1800		54	11	66	4800	T
5	2	60	1800		55	12	64	4600	T
6	2	60	1800		56	12	64	4600	T
7	2	60	1800		57	12	64	4600	T
8	2	60	1800		58	12	64	4600	T
9	3	49.6	4800	T	59	12	64	4600	T
10	3	49.6	2400		60	12	64	4600	T
11	3	49.6	2400		61	13	50	1800	
12	3	49.6	2400		62	13	50	1800	
13	4	52	1560		63	14	60	3600	T
14	4	52	1560		64	14	60	3600	T
15	4	52	1560		65	14	60	3600	T
16	4	52	1560		66	15	53	3600	T
17	4	52	1560		67	15	53	3600	T
18	5	57.2	2800	T	68	15	53	1800	
19	5	57.2	2800	T	69	15	53	1800	
20	5	57.2	2800	T	70	16	52	4800	T
21	5	57.2	2800	T	71	16	52	4800	T
22	5	57.2	2800	T	72	16	52	4800	T
23	5	57.2	2800	T	73	16	52	4800	T
24	5	57.2	1400		74	16	52	4800	T
25	6	64	1400		75	17	68	2100	
26	6	64	1400		76	18	64	1800	
27	6	64	1400		77	18	64	1800	
28	7	49	3200	T	78	18	64	1800	
29	7	49	3200	T	79	19	49	2800	T
30	7	49	3200	T	80	19	49	2800	T
31	7	49	3200	T	81	19	49	1400	
32	7	49	3200	T	82	19	49	1400	
33	7	49	3200	T	83	19	49	1400	
34	8	60	1900		84	19	49	1400	
35	8	60	1900		85	20	60	2800	T
36	8	60	1900		86	20	60	2800	T
37	8	60	1900		87	20	60	2800	T
38	9	53	3600	T	88	20	60	2800	T
39	9	53	3600	T	89	20	60	2800	T
40	9	53	3600	T	90	21	53.2	1500	
41	9	53	3600	T	91	21	53.2	1500	
42	9	53	1800		92	21	53.2	1500	
43	9	53	1800		93	21	53.2	1500	
44	10	48	4480	T	94	22	47.2	4000	T
45	10	48	4480	T	95	22	47.2	4000	T
46	10	48	4480	T	96	22	47.2	4000	T
47	10	48	4480	T	97	22	47.2	2000	
48	10	48	2240		98	22	47.2	2000	
49	10	48	2240		99	23	68	1600	
50	10	48	2240		100	24	60	1600	

*: Target flags, ‘T’, indicate that the jobs belong to the daily production target and should be processed.

TABLE A2
SETUP TIMES REQUIRED FOR SWITCHING ONE PRODUCT TYPE TO ANOTHER FOR 24 PRODUCT TYPES (UNIT: MINUTES)

From \ To	*O*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
O	0	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
1	0	0	15	15	15	15	15	280	280	280	280	120	120	120	50	50	50	50	180	180	180	180	180	120	120
2	0	15	0	15	15	15	15	280	280	280	280	120	120	120	50	50	50	50	180	180	180	180	180	120	120
3	0	15	15	0	15	15	15	280	280	280	280	120	120	120	50	50	50	50	180	180	180	180	180	120	120
4	0	15	15	15	0	15	15	280	280	280	280	120	120	120	50	50	50	50	180	180	180	180	180	120	120
5	0	15	15	15	15	0	15	20	20	20	20	250	250	250	250	250	250	180	30	30	30	30	30	180	250
6	0	15	15	15	15	15	0	20	20	20	20	250	250	250	250	250	250	180	30	30	30	30	30	180	250
7	0	20	20	20	20	30	30	0	208	208	208	30	30	30	30	30	30	170	170	250	250	250	250	250	250
8	0	280	280	280	280	280	280	200	0	15	15	130	130	20	150	150	150	50	260	80	80	80	80	80	280
9	0	280	280	280	280	280	280	200	15	0	15	130	130	20	150	150	150	50	260	80	80	80	80	80	280
10	0	280	280	280	280	280	280	200	15	15	0	130	130	20	150	150	150	50	260	80	80	80	80	80	280
11	0	20	20	20	20	150	150	120	50	50	50	0	15	80	80	80	80	150	150	120	120	120	120	120	30
12	0	20	20	20	20	150	150	120	50	50	50	15	0	80	80	80	80	150	150	120	120	120	120	120	30
13	0	50	50	50	50	260	260	260	150	150	150	150	150	0	280	280	280	20	20	50	50	50	50	50	150
14	0	20	20	20	20	30	30	15	208	208	208	30	30	170	0	15	15	170	130	250	250	250	250	250	250
15	0	20	20	20	20	30	30	15	208	208	208	30	30	170	15	0	15	170	130	250	250	250	250	250	250
16	0	20	20	20	20	30	30	15	208	208	208	30	30	170	15	15	0	170	130	250	250	250	250	250	250
17	0	280	280	280	280	280	280	200	50	50	50	50	50	150	150	150	150	0	260	80	80	80	80	80	280
18	0	150	150	150	150	150	150	80	80	80	80	250	250	250	100	100	100	250	0	30	30	30	30	30	20
19	0	280	280	280	280	250	250	100	30	30	30	50	50	150	100	100	100	30	150	0	15	15	15	30	100
20	0	280	280	280	280	250	250	100	30	30	30	50	50	150	100	100	100	30	150	15	0	15	15	30	100
21	0	280	280	280	280	250	250	100	30	30	30	50	50	150	100	100	100	30	150	15	15	0	15	30	100
22	0	280	280	280	280	250	250	100	30	30	30	50	50	150	100	100	100	30	150	15	15	15	0	30	100
23	0	150	150	150	150	150	150	80	80	80	80	250	250	250	100	100	100	250	270	30	30	30	30	0	20
24	0	270	270	270	270	270	270	100	100	100	100	150	150	150	80	80	80	280	30	30	30	30	30	100	0

REFERENCES

[1] K. C. So, "Some heuristics for scheduling jobs on parallel machines with setups," *Manage. Sci.*, vol. 36, pp. 467–475, 1990.

[2] F. Tovia, S. J. Mason, and B. Ramasami, "A scheduling heuristic for maximizing wirebonder throughput," *IEEE Trans. Electron. Packag. Manuf.*, vol. 27, no. 2, pp. 145–150, Apr. 2004.

[3] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. RinnooyKan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Ann. Discrete Math.*, vol. 5, pp. 287–326, 1979.

[4] S. E. Butt and T. M. Cavalier, "A heuristic for the multiple tour maximum collection problem," *Comput. Operat. Res.*, vol. 21, pp. 101–111, 1994.

[5] I. M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *Eur. J. Operat. Res.*, vol. 88, pp. 464–474, 1996.

[6] S. E. Butt and D. M. Ryan, "An optimal solution procedure for the multiple tour maximum collection problem using column generation," *Comput. Operat. Res.*, vol. 26, pp. 427–441, 1999.

[7] H. Tang and E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," *Comput. Operat. Res.*, vol. 32, pp. 1379–1407, 2005.

[8] D. Feillet, F. Dejax, and M. Gendreau, "Traveling salesman problems with profits," *Transportation Sci.*, vol. 39, no. 2, pp. 188–205, 2005.

[9] T. C. E. Cheng and C. C. S. Sin, "A state-of-the-art review of parallel-machine scheduling research," *Eur. J. Operat. Res.*, vol. 47, pp. 271–292, 1990.

[10] E. Mokotoff, "Parallel machine scheduling problems: A survey," *Asia-Pacific J. Operat. Res.*, vol. 18, pp. 193–242, 2001.

[11] T. Sen, J. Sulek, and P. Dileepan, "Static scheduling to minimize weighted and unweighted tardiness: A state-of-the-art survey," *Int. J. Production Economics*, vol. 83, pp. 1–12, 2003.

[12] U. Bilge, F. Kirac, M. Kurtulan, and P. Pekgun, "A Tabu search algorithm for parallel machine total tardiness problem," *Comput. Operat. Res.*, vol. 31, pp. 397–414, 2004.

[13] C. Chu, "A branch-and-bound algorithm to minimize total flow time with unequal release dates," *Naval Res. Logist.*, vol. 39, pp. 859–875, 1992.

[14] J. N. D. Gupta and A. J. Ruiz-Torres, "Generating efficient schedules for identical parallel machines involving flow-time and tardy jobs," *Eur. J. Operat. Res.*, vol. 167, pp. 679–695, 2005.

[15] T. K. Hwang and S. C. Chang, "Design of a Lagrangian relaxation-based hierarchical production scheduling environment for semiconductor wafer fabrication," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 566–578, Aug. 2003.

[16] P. Baptiste, C. Marek, C. Dürr, W. Jawor, and N. Vakhania, "Preemptive scheduling of equal-length jobs to maximize weighted throughput," *Operat. Res. Lett.*, vol. 32, pp. 258–264, 2004.

[17] S. P. Y. Fung, F. Y. L. Chin, and H. Shen, "Online scheduling of unit jobs with bounded importance ratio," *Int. J. Foundations Comput. Sci.*, vol. 16, no. 3, pp. 581–598, 2005.

[18] K. Hiraishi, E. Levner, and M. Vlach, "Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs," *Comput. Operat. Res.*, vol. 29, pp. 841–848, 2002.

[19] S. Rojanasoonthon, J. F. Bard, and S. D. Reddy, "Algorithms for parallel machine scheduling: A case study of the tracking and data relay satellite system," *J. Operat. Res. Soc.*, vol. 54, pp. 806–821, 2003.

[20] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *Omega-Int. J. Manage. Sci.*, vol. 27, pp. 219–239, 1999.

[21] B. Jeong, S. W. Kim, and Y. J. Lee, "An assembly scheduler for TFT LCD manufacturing," *Comput. Ind. Eng.*, vol. 41, pp. 37–58, 2001.

[22] H. J. Shin and V. J. Leon, "Scheduling with product family set-up times: An application in TFT LCD manufacturing," *Int. J. Prod. Res.*, vol. 42, pp. 4235–4248, 2004.

[23] W. L. Pearn, S. H. Chung, and M. H. Yang, "A case study on the wafer probing scheduling problem," *Prod. Planning Contr.*, vol. 13, no. 1, pp. 66–75, 2002.

[24] F. Yalaoui and C. Chu, "An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times," *IIE Trans.*, vol. 35, pp. 183–190, 2003.

S. H. Chung received the Ph.D. degree in industrial engineering from Texas A&M University, College Station.

She is a Professor in the Department of Industrial Engineering and Management, National Chiao-Tung University, Hsinchu, Taiwan. Her research interests include production planning, scheduling, cycle time estimation, and performance evaluation. She has published and presented research papers in the areas of production planning and scheduling for IC and TFT-LCD manufacturing.

Y. T. Tai received the Ph.D. degree in industrial engineering and management from National Chiao-Tung University, Hsinchu, Taiwan.

She is an Assistant Professor of the Department of Information Management, Kainan University, Taoyuan, Taiwan. Her research interests include scheduling and semiconductor manufacturing management.

W. L. Pearn received the Ph.D. degree in operations research from the University of Maryland, College Park.

He is a Professor of Operations Research and Quality Assurance at National Chiao-Tung University (NCTU), Hsinchu, Taiwan. He was with Bell Laboratories as a Quality Research Scientist before joining NCTU. His research interests include process capability, network optimization, and production management. His publications have appeared in the *Journal of the Royal Statistical Society, Series C*, *Journal of Quality Technology*, *European Journal of Operational Research, Journal of the Operational Research Society*, *Operations Research Letters*, *Omega, Networks*, *International Journal of Productions Research*, and others.