Minimizing the Total Completion Time in a Two-Machine Group Scheduling Problem with Carryover Sequence Dependency

Rasaratnam Logendran
Department of Industrial and Manufacturing Engineering
Oregon State University
USA

Chelliah Sriskandarajah
School of Management
The University of Texas at Dallas
USA

Abstract

The two-machine group-scheduling problem for minimizing the total completion time with carryover sequence dependency is addressed in this paper. These problems are typically encountered in electronics manufacturing. A procedure called *Minsetup* is developed to determine the lower bound for the total completion time, and a search algorithm based on tabu search is proposed to solve the problem. Unlike in conventional tabu search, the search algorithm is implemented on a tabu tree, and the steps associated with the algorithm are further demonstrated by solving a representative example from industry.

1 Introduction

The impact of cellular manufacturing systems in industry practice has triggered a significant number of research investigations in group scheduling in recent years. Group scheduling problems are characterized as those requiring scheduling at two levels. At the first level, a sequence for jobs (parts) in a group that would either maximize or minimize some measure of effectiveness must be determined. At the second level, a sequence for groups of jobs in order to optimize the same measure of performance must be determined. As the jobs that belong to a group are similar, the setup time required to change from one job to another within a group can be considered negligible or included in the processing time. Also, the operational sequence on machines for jobs within a group is the same, i.e., flow-line. However, a changeover from one group to another would require such a substantial setup time that it cannot be disregarded.

Recognizing these properties, several heuristic algorithms for determining the job sequence (i.e., level 1) have been developed by incorporating some of the basic ideas of both optimal and heuristic solution algorithms for solving static flowshop scheduling problems [7,16,4,15]. For minimizing the total completion time (makespan), these include the work by Radharamanan [18] Al-Qattan [2], and Allison [1]. For the same performance measure, a new heuristic proposed by Logendran and Nudtasomboon [11] showed a superior performance to those by Radharamanan [18] and Al-Qattan [2] on a series of test problems.

At level 2, each group consisting of jobs has a setup time and a run time on the machine it is processed. The run time is equal to the sum of the processing time of jobs used in the level 1 analysis. Also, when the setup time can be separated from the run time, it is typical to perform the setup on M_2, in anticipation of the arriving job, a feature commonly referred to as anticipatory setup. The processing of groups of jobs also follows a flow-line arrangement. Yoshida and Hitomi [21] were the first to investigate the two-machine group-scheduling problem for minimizing the makespan. Their algorithm focused on an extension of Johnson's [7] algorithm originally developed for a two-machine flowshop problem with setup times included. Sule [19] extended this to a case where the setup, run, and removal times are separated. Even with removal times separated, an extension of Johnson's algorithm provided an optimal group sequence for minimizing the makespan. Proust et al. [17] presented heuristic algorithms for minimizing the makespan of an m-machine flowshop-scheduling problem with setup, run, and removal times separated. Logendran et al. [10] developed combined heuristics for minimizing the makespan on m-machine, bi-level group scheduling problems when setup and run times are separated.

The research reported in this paper pertains to group scheduling in electronics manufacturing. McGinnis et al. [14] provide a description of three related problems in printed circuit board assembly: grouping, allocation, and arrangement and sequencing. First, the machine groups and board families should be selected, and families should be assigned to groups. Second, when multiple machines are present in a machine group, the various components should be assigned to machines. Finally, the feeders carrying components should be staged and the sequence in which to populate each PCB with components should be determined. Setup strategies employed in PCB assembly can be broadly classified into two groups – single setup strategy and multi-setup strategy [3].

The single setup strategy, when used with one board type, is referred to as the unique setup strategy, and when used with several board types that belong to a family, it is referred to as the family setup strategy. The multi-setup strategy is founded on the idea of insufficient staging capacity on the placement machine to perform a single setup. Tang and Denardo [20] proposed the Keep Tool Needed Soonest (KTNS) rule in order to minimize the number of tools changed in a tool magazine of a machine in flexible manufacturing systems (FMS). With respect to electronics manufacturing, Lofgren and McGinnis [9] use the multi-setup decompose and sequence (DAS) strategy for solving the multi-board single-machine component allocation problem. Leon and Peters [8] evaluate different setup strategies for a single PCB machine.

The work reported here specifically relates to scheduling the assembly processes required of printed circuit boards (PCBs) in a two-machine environment and is motivated by a real industry application. Unlike in FMS, in PCB assembly it is impractical to keep changing the component feeders on a frequent basis. The board types that exhibit similar features are typically included in one group. The setup is thus performed to enable placement of *all* of the components required of several board types that belong to a single board group, which is referred to as the *surrogate board group*. However, the setup on the second machine can be performed in anticipation of the arriving surrogate board group.

The characteristics of group scheduling in electronics manufacturing is contrastingly different from that of hardware manufacturing. The setup time required of the current group of PCBs on a machine is not only sequence dependent, but is carried over from the first group to the group preceding the current group. In other words, if N board groups are scheduled in the order 1,2,…,N, then the setup time required to stage the components required of the Nth board group is dependent *not just only on the* preceding group (i.e., (N-1) th group), but all of the preceding groups (i.e., 1,2,3…,(N-1)). We characterize this as a two-machine, carry-over sequence-dependent group-scheduling problem with anticipatory setups. In addition, the challenges encountered here are far greater and distinctly different from those reported previously, including that on FMS. These include: 1. The setup change required of a board group is dependent upon the component and the *feeder location to which it is allocated*, and 2. Multiple sequential-machining environment. In the next section we provide a representative example obtained from the industry.

2 A Representative Example

Figure 1 presents the two-stage PCB assembly system, comprised of two machines. Machines 1 and 2 (M_1 and M_2) are referred to as the high-speed placement machine (HSPM) and multi-function placement machine (MFPM), respectively. Typically, these machines consist of 20 feeders (slots) on the magazine rack to which components can be assigned. The representative example, obtained from industry data, is presented in Tables 1-4.

The example consists of five different board types (G_{11}, G_{21}, G_{22}, G_{31}, and G_{32}) that belong to three different board groups, G_1, G_2, and G_3. It demonstrates the use of all 20 feeders on the HSPM and 10 feeders on the MFPM, although a few are never used in this example. The entire process of scheduling tasks required of PCBs is typically broken down into several different planning horizons, each consisting of a few weeks. Thus the initial/reference configuration for the current planning horizon is given by the

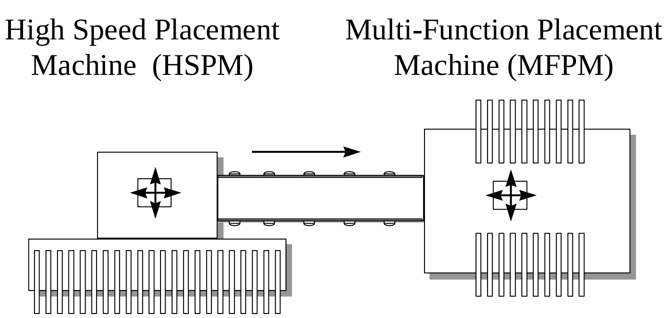

Figure 1. Two-Stage PCB Assembly System

Table 1. An Example with Three Board Groups

Groups	G_1	G_2		G_3	
Boards in a Group	G_{11}	G_{21}	G_{22}	G_{31}	G_{32}
Number of Boards to Build	5	7	3	14	9

Table 2. Feeder Setup Times on HSPM and MFPM

Average Setup Time per Feeder (sec)	
HSPM	MFPM
180	220

Table 3. Feeder Configuration for the High Speed Placement Machine (HSPM)

Reference Configuration on the M/C	Machine Feeders	Feeder Carrier Configuration Required per Group			Configuration Required for Each Board				
		G_1	G_2	G_3	G_{11}	G_{21}	G_{22}	G_{31}	G_{32}
	1			101-0002				101-0002	101-0002
101-0006	2		101-0004			101-0004	101-0004		
101-0010	3	101-0010	101-0009		101-0010	101-0009			
101-0001	4		101-0001				101-0001		
	5	101-0008	101-0005		101-0008	101-0005	101-0005		
101-0030	6	101-0006	101-0013	101-0013	101-0006		101-0013	101-0013	101-0013
101-0007	7	101-0007	101-0007	101-0006	101-0007	101-0007		101-0006	
101-0013	8	101-0013		101-0003	101-0013			101-0003	101-0003
101-0033	9		101-0008	101-0008		010-0008	101-0008		101-0008
	10	101-0004			101-0004				
101-0035	11								
	12	101-0009			101-0009				
	13		101-0006			101-0006	101-0006		
101-0038	14	101-0003	101-0003	101-0012	101-0003	101-0003	101-0003		101-0012
	15								
101-0040	16	101-0011	101-0012		101-0011		101-0012		
	17								
101-0011	18		101-0011	101-0009		101-0011		101-0009	101-0009
	19		101-0014	101-0010		101-0014		101-0010	101-0010
	20								

Table 4. Feeder Configuration for the Multi-Functional Placement Machine (MFPM)

Reference Configuration on the M/C	Machine Feeders	Feeder Carrier Configuration Required per Group			Configuration Required for Each Board				
		G_1	G_2	G_3	G_{11}	G_{21}	G_{22}	G_{31}	G_{32}
101-0063	1	101-0015		101-0015	101-0015			101-0015	
101-0084	2								
	3								
	4		101-0015	101-0018		101-0015	101-0015	101-0018	101-0018
	5	101-0018	101-0017		101-0018	101-0017			
101-0018	6								
101-0123	7			101-0019				101-0019	101-0019
	8								
101-0020	9		101-0016				101-0016		
	10			101-0020				101-0020	101-0020

allocation of components to the different feeders at the end of the previous planning horizon. Notice that G_{21} and G_{22} are two board types that belong to the same group, yet require the same as well as different components. Thus, the setup needed should not focus on either G_{21} or G_{22}, but rather the *surrogate* representative of both G_{21} and G_{22}, which, for simplicity, is referred to as board group G_2. The underlying issue now is to address the sequencing of the various surrogate board groups as well as the individual board types that make up each surrogate board group. Clearly, this is a two-stage (HSPM and MFPM) group-scheduling problem, requiring that sequencing decisions be made at two levels. The first level deals with identifying the sequence of board types within each group, while the second level deals with the sequencing of the groups themselves.

3 Procedure for Evaluating the Lower Bound

For minimization of makespan, the two-machine flowshop (job)-scheduling problem with no carry-over and sequence-dependent setups, has been shown strongly NP-hard [6]. The problem investigated in this paper, and described as the two-machine group scheduling problem with carry-over and sequence-dependent setup times, is easily reducible to the problem investigated by Gupta and Darrow [6]. Thus, it is also strongly NP-hard. Therefore, a procedure, called Minsetup, is developed for identifying an effective lower bound for the makespan of the research problem.

3.1 Procedure Minsetup

The feeder configuration for M_1 shown in Table 3 for the example problem with 3 board groups (G_1, G_2, and G_3) can be advantageously used to demonstrate how an effective lower bound can be evaluated for the makespan. Notice that with three groups and R being the reference group, there are six possible group sequences: R-G_1-G_2-G_3, R-G_2-G_1-G_3, R-G_3-G_1-G_2, R-G_1-G_3-G_2, R-G_2-G_3-G_1, and R-G_3-G_2-G_1. If the focus is on G_1 and feeder 3, the best possible scenario would require no setup change (as component 101-0010 is in feeder 3 of R), and if it is feeder 5, the best possible scenario would require one setup change (as 101-0008 is not in feeder 3 of R, G_2, or G_3), irrespective of the position held by G_1. Thus the total *minimum* number of feeder changes/setups required for G_1 is 5 (0+1+1+0+0+1+1+0+1), irrespective of the position held by G_1. Using the same analogy, the total minimum number of feeder changes required for G_2 and G_3 is 6 (1+1+0+1+0+0+0+1+0+1+0+1) and 6 (1+0+1+1+0+1+1+1), respectively. Similarly, for M_2, the total minimum number of feeder changes required for G_1, G_2 and G_3 are 1, 3, and 3, respectively.

Notice that the procedure for evaluating the minimum number of setup changes required for each board group is really a *comparison* between the components used by each board group with that of the other groups, including R, and is tied to the number of components used by that group. In this example, let x_1, x_2, and x_3 be the number of components used by board groups G_1, G_2, and G_3, respectively, and $x\text{max} = \text{Max}(x_1, x_2, x_3)$. The polynomial time complexity for evaluating the minimum number of setup changes in this example can thus be expressed as 3 * 3 * $x$max, as there are three comparisons for each group and there are a total of 3 groups. In general, for a problem with N board groups, this procedure has polynomial time complexity of order $O[N^2 (x\text{max})]$. For clarity, we provide the pseudo code for the procedure *Minsetup* next in Steps 0 through 4.

Define the following notations:

i = 0,1,…,N groups (0 refers to the reference group R)

s = 1,2,…,T feeders

r = counter for feeder

u = counter for group

Feedercomp [i,s]: Component number in feeder F_s for group G_i

Countsetup [i]: Counter that counts the minimum number of setups required for group G_i,i = 1,…,N

Step 0: *Initialization*. Initialize Feedercomp [i,s] by storing components in each feeder F_s for each group G_i, where i = 0,1,…,N and s = 1,2,…,T. Set Counsetup [i] = 0 for each group G_i, i = 1,…,N. Set r = 1 and u = 1.

Step 1: For each i = 0,1,…,N, compare the following components when $i \neq u$ and Feedercomp [u,r] and Feedercomp [i,].

Compare: Feedercomp [u,r] and Feedercomp [i,r].

If they are not the same component in all N comparisons, then set Countsetup [u] = Countsetup [u] + 1.

Step 2: If $r < T$, set $r = r + 1$ and go to Step 1.

Step 3: If $u < N$, set $r = 1$, $u = u + 1$, and go to Step 1.

Step 4: Output. Countsetup [i], $i = 1,\ldots,N$. Stop.

Determining the minimum number of setup changes required by each group on M_1 and M_2 has allowed us to *convert* a notoriously hard, two-machine sequence dependent group-scheduling problem with carry over setups into a two-machine sequence independent group-scheduling problem with no carry-overs. Yet, the provision for performing *anticipatory* setups on M_2 is maintained in both problem structures. The polynomial time algorithm (of order $O(N\log N)$, where N is the number of board groups scheduled to be processed) by Yoshida and Hitomi [21] can be applied to identify the optimum minimum makespan for the converted two-machine group scheduling problem with sequence-independent anticipatory setups. This optimum makespan is also the lower bound for the minimum makespan of the original problem with carry over sequence dependency. We state this property as Theorem 1 below, and omit providing a formal proof, as it is easy to see. In the next section, we present the data pertinent to the representative example and apply the procedure *Minsetup* to evaluate the lower bound.

Theorem 1: The algorithm by Yoshida and Hitomi [21], when applied with minimum setup times obtained as described in the procedure Minsetup above, provides a lower bound on the makespan for the original problem with carry-over sequence dependency.

4 Application of the Algorithm to Evaluate the Lower Bound

Table 5 below presents the setup time for each board group and run time for each board type of the *converted* sequence independent group-scheduling problem. The sequence independent setup times are evaluated based on the data presented in Table 2, and the run times are evaluated based on the number of boards to be built as presented in Table 1. For example, the sequence-independent setup time for G_1 on M_1 is 900 sec. as the number of setups is 5 and the average setup time per feeder is 180 sec.

Table 5. Setup and Run Times for the Sequence-Independent Problem

Board Groups	G_1	G_2		G_3	
Board Types (Jobs)	G_{11}	G_{21}	G_{22}	G_{31}	G_{32}
Setup Time on M_1 (sec.)	900	1080		1080	
Setup Time on M_2 (sec.)	220	660		660	
Run Time on M_1 (sec.)	318	319	141	1096	1083
Run Time on M_2 (sec.)	20	41	3	488	355

Let, n_i = Number of jobs in group G_i.

S_{ij} = Setup time of group G_i on machine j ($i = 1,2,...,N$; $j = 1, 2$).

$t_{i(k)j}$ = Run time of the *rank ordered* job k in group G_i on machine j ($i = 1,2,\ldots,N$; $k = 1,2,....,n_i$; $j = 1, 2$).

The following algorithm determines a group schedule, which minimizes the makespan of a two-machine sequence-independent group-scheduling problem.

Step 1:
Use Johnson's rule to determine an optimal job sequence.
Step 2:
Use the following approach to determine an optimal group sequence.

1. For each group under the job sequence determined by step 1, calculate the following values:

$$A_i = S_{i1} - S_{i2} + \max_{1 \le r \le n_i} \left(\sum_{k=1}^{r} t_{i(k)1} - \sum_{k=1}^{r-1} t_{i(k)2} \right)$$

$$B_i = \max_{1 \le r \le n_i} \left(\sum_{k=r}^{n_i} t_{i(k)2} - \sum_{k=r+1}^{n_i} t_{i(k)1} \right)$$

2. Find the minimum value among the A_i's and the B_i's. Break ties arbitrarily.
3. If it is A_x, place G_x first, and if it is B_x, place G_x last.
4. Remove the assigned group from further consideration and go back to step 2.

The application of Step 1 to the example problem results in the following job sequence within each sequence. G_1: G_{11}; G_2: G_{21} – G_{22}; and G_3: G_{31} - G_{32}. The application of Step 2 to the example problem results in the evaluation of the following A_i and B_i values for each of the three groups as shown in Table 6. Thus the group sequence is: G_3 – G_1 – G_2, and the complete sequence comprised of both the groups and jobs within each group is: G_3 (G_{31} - G_{32}) – G_1 (G_{11}) – G_2 (G_{21} - G_{22}). The Gantt chart for this sequence is shown in Figure 2, thus resulting in a makespan of 6020 sec. The value 6020 is also a lower bound for the original problem with carryover sequence dependency.

Table 6. Values of A_i and B_i for the Example Problem

Group (G_i)	A_i	B_i
G_1	998	20
G_2	839	3
G_3	2111	335

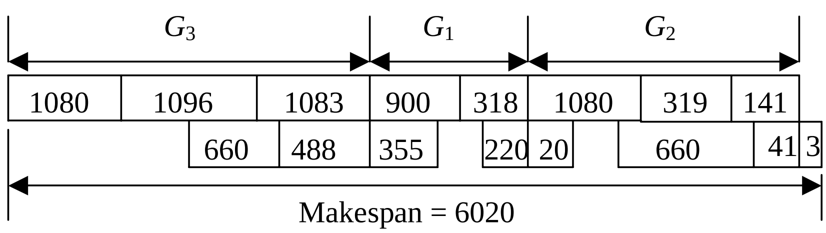

Figure 2. Gantt Chart for the Example

5 A Search Algorithm

With a lower bound for the makespan of the original problem with carryover sequence dependency known, we present a higher-level search algorithm, based on tabu-search for solving large problems in electronics manufacturing encountered in industry practice. Tabu search [5] has been proven to be an effective approach for solving complex scheduling problems, including that of a flexible manufacturing system's scheduling problem [12] and real-time scheduling on unrelated-parallel machines with job splitting [13]. To trigger the search based on the concept of tabu search, an initial solution is required. For ease of implementation, the solution identified above is considered to be the initial solution to the original problem. Yet, to evaluate the true makespan for the original problem, certain modifications have to be made to compensate for carryover sequence-dependency. From Tables 3 and 4, the number of setup changes on M_1 and M_2 required to execute the group sequence $R – G_3 – G_1 – G_2$ are evaluated as follows: R to G_3 – 8 setups on M_1 and 4 setups on M_2; $(R – G_3)$ to G_1 – 8 setups on M_1 and 1 setup on M_2; $(R – G_3 – G_1)$ to G_2 – 8 setups on M_1 and 3 setups on M_2. Given a group sequence, the optimal job sequence within a group must follow the Johnson's ordering, as the objective is to minimize the makespan. Thus, the job sequence for each group is the same as that determined in Step 1 in Section 4. Recognizing that a setup on M_1 and M_2 takes 180 and 220 sec., respectively, the makespan for the initial solution (i.e., G_3 (G_{31} - G_{32}) – G_1 (G_{11}) – G_2 (G_{21}- G_{22})) can be evaluated as 7280.

For a given problem instance, it is conceivable that the initial solution is of acceptable quality that the search can be curtailed. As the lower bound on the makespan for the original problem with carryover sequence dependency is known, the quality of the solution (MS) can be determined as a percentage deviation, given by $[(MS – MS_{LB})/ MS_{LB}]*100$. For the lower bound of 6020, the percentage deviation for the initial solution is 20.93%. Typically, a solution with a percentage deviation of 3% or lower is considered acceptable quality. As computation time is a significant issue in solving industry size problems, even a percentage deviation of 5% or lower may be deemed acceptable. As the percentage deviation for the initial solution is above 20%, it is considered as the seed/parent solution to trigger the application of the tabu search-based solution algorithm.

To keep the application simple, at this time only pair-wise exchanges or perturbations of groups including the cyclical perturbation is considered to generate a new set of group sequences. Within the context of group scheduling, the applications of tabu search typically exhibit two levels (layers) of search. The outer level is dedicated to the perturbations on the group sequence, and the inner level is dedicated to the perturbations on the job sequence within each group. Because the intent is to minimize the makespan and the optimal job sequence within each group must follow the Johnson's ordering, the inner level search is not required.

To enhance the computational efficiency of the search, we deviate from the conventional tabu search to propose a search that is truly implemented on a 'tabu tree.' The initial solution when perturbed would produce the following group sequences: $\underline{G}_1 – \underline{G}_3 – G_2$; $G_3 – \underline{G}_2 – \underline{G}_1$; and $\underline{G}_2 – G_1 – \underline{G}_3$. As the tabu list size is assumed equal to one, G_1 and G_3 are underlined in the first perturbed solution to indicate that they are tabu in the next iteration. That is, if a pairwise exchange on G_1 and G_3 is considered, it would result in $G_3 – G_1 – G_2$, which is the parent (initial) solution and revisiting the parent solution would not result in newer solutions (children), thus must be prohibited. In the tabu tree, the root node is reserved for the initial solution at level 0. The three 'off springs' generated from the initial solution are then be represented by three separate nodes at level 1 of the tree.

The makespan for each of these three new group sequences is: $\underline{G}_1 – \underline{G}_3 – G_2 = 7100$; $G_3 – \underline{G}_2 – \underline{G}_1 = 7297$; and $\underline{G}_2 – G_1 – \underline{G}_3 = 7272$. As $\underline{G}_1 – \underline{G}_3 – G_2$ results in the minimum makespan, it is selected as the solution for the next perturbation. As G_1 and G_3 are considered tabu, two new solutions are obtained. Thus the two nodes contained at level 2 of the search tree are $G_1 - \underline{G}_2 - \underline{G}_3$ and $\underline{G}_2 - G_3 - \underline{G}_1$ with a makespan of 6912 and 6937, respectively. Observe that further perturbation on $G_1 - \underline{G}_2 - \underline{G}_3$ would result in $\underline{G}_2 - \underline{G}_1 - G_3$ and $\underline{G}_3 - G_2 - \underline{G}_1$ as two new solutions that ought to reside at level 3 of the search tree. As the same group sequences have already been generated at level 1 of the search tree, the two new solutions evaluated at level 3 are disregarded. Strictly speaking, at this point the search should back track to level 1 of the tree and consider perturbations on $\underline{G}_2 - G_1 - \underline{G}_3$ = 7272, as this is the next best unperturbed solution. With three groups in this example, however, the maximum number of permutation group sequences that can be generated is 6 (= 3!). As all six of them are already identified on the search tree, the search is terminated without considering any further perturbations on $\underline{G}_2 - G_1 - \underline{G}_3$.

The search mechanism developed in this research can be described as follows. There are three candidate solutions (i.e., ($\underline{G}_1 - \underline{G}_3 - G_2$; $G_3 - \underline{G}_2 - \underline{G}_1$: and $\underline{G}_2 - G_1 - \underline{G}_3$) generated from the initial solution $G_3 - G_1 - G_2$ with a makespan of 7280. The aspiration level (AL) is also set equal to 7280. As the search mechanism is built based on the concept of tabu tree (and not on a conventional tabu search), all of the candidate solutions are inserted into a temporary candidate list (TCL). The best among the three (i.e., $\underline{G}_1 - \underline{G}_3 - G_2^{*}$) with a makespan of 7100 is chosen and inserted into the candidate list (CL) as the next solution for perturbation. The AL is also revised to 7100. Observe that as the new solution inserted into the CL has a better makespan than the initial solution it receives a star (*) to indicate that it has the potential of becoming the next local optimum. It will become the next local optimum provided the makespan evaluated for the candidate solutions in its neighborhood is inferior, i.e., greater or equal to 7100. If it did, it will receive two stars (**), and be inserted into the index list (IL) as the first local optimum. As this being the case for the next candidate solution (i.e., $G_1 - \underline{G}_2 - \underline{G}_3^{**}$) with a makespan of 6912, it is first inserted into the CL with a star and then into the IL with two stars. The AL is now set to 6912. When the search terminates, $G_1 - \underline{G}_2 - \underline{G}_3^{**}$ is the only entry in IL, and is identified as the best solution for the example problem.

6 Conclusions

We have addressed the two-machine group-scheduling problem for minimizing the total completion time with carryover sequence dependency, typically encountered in electronics manufacturing. A procedure for determining the lower bound for the makespan is developed, and a search algorithm based on tabu search is proposed to solve the problem. Unlike in the conventional tabu search, the search algorithm is implemented on a tabu tree, and the steps associated with the algorithm are further demonstrated by solving a representative example from industry. At this time, however, a fairly basic tabu search-based algorithm based on short-term memory and characterized by fixed tabu-list size is applied to the example.

This research would, therefore, be extended in the future to include variable tabu-list sizes as well as long-term memory based on both maximal and minimal frequencies to develop a fairly comprehensive approach that would ensure identifying very good near optimal solutions even on large-size industry problems. In particular, as the search is implemented on a tabu tree, parallelization of the search would be exploited to enhance computational efficiency. The percentage deviation of the best solution identified above is 10.83% [((6912-6020)/6020) * 100] above the lower bound, although it is known to be the optimal solution for the example considered. Thus, efforts would also be directed in the future to develop a tighter lower-bounding mechanism for the problem addressed in this paper.

Acknowledgements

Research by the first author is funded in part by the National Science Foundation (USA) Grant No. DMI-0010118, and that by the second author is funded in part by the National Science Foundation (USA) Grant No. DMI-0100327. Their support is gratefully acknowledged.

References

[1] J.D. Allsion, "Combining Petrov's Heuristic and the CDS Heuristic in Group Scheduling Problems," *Proceedings of the 12th Annual Conference on Computers and Industrial Engineering*, Orlando, FL (USA), (1990), pp. 457-461.

[2] I. Al-Qattan, "Designing GT Cells Enhanced by Group Scheduling: Key to Flexible Manufacturing System," *Proceedings of the IIE Integrated Systems Conference*, St. Louis, MO (USA), (1988), pp. 25-30.

[3] J.C. Ammons, M. Carlyle, L. Cranmer, G.W. Depuy, K.P. Ellis, L.F. McGinnis, C.A. Tovey, and H. Xu, "Component Allocation to Balance Workload in Printed Circuit Card Assembly Systems," *IIE Transactions*, 29(4) (1997), pp. 265-276.

[4] H.G. Campbell, R.A. Dudek, and M.L. Smith, "A Heuristic Algorithm for the n Job, m Machine Sequencing Problem," *Management Science*, 16 (1970), pp. 630-637.

[5] F. Glover, "Tabu-Search: A Tutorial," *Interfaces*, 20 (1990), pp. 79-94.

[6] J.N.D. Gupta and W.P. Darrow, "The Two-Machine Sequence Dependent Flowshop Scheduling Problem," *European Journal of Operational Research*, (1986), pp. 439-446.

[7] S.M. Johnson, "Optimal Two- and Three-Stage Production Schedules with Set-up Times Included," *Naval Research Logistics Quarterly*, 1 (1954), pp. 61-68.

[8] V.J. Leon and B.A. Peters, "A Comparison of Setup Strategies for Printed Circuit Board Assembly," *Computers and Industrial Engineering*, 34(1) (1998), pp. 219-234.

[9] C.B. Lofgren and L.F. McGinnis, "Dynamic Scheduling for Flexible Printed Circuit Card Assembly, *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, Atlanta, GA, IEEE, New York (USA), (1986), pp. 1294-1297.

[10] R. Logendran, L. Mai, and D. Talkington, "Combined Heuristics for Bi-Level Group Scheduling Problems," *International Journal of Production Economics*, 38(2-3) (1995), pp. 133-145.

[11] R. Logendran and N. Nudtasomboon, "Minimizing the Makespan of a Group Scheduling Problem: A New Heuristic," *International Journal of Production Economics*, 22 (1991), pp. 217-230.

[12] R. Logendran and A. Sonthinen, "Tabu Search-Based Approach for Scheduling Job-Shop Type Flexible Manufacturing Systems," *Journal of the Operational Research Society*, 48 (1997), 264-277.

[13] R. Logendran and F. Subur, "Real-Time Scheduling with Job-Splitting on Unrelated-Parallel Machines, *Proceedings of the 7th Annual Industrial Engineering Research Conference (CD-ROM)*, Banff, Canada, (1998).
[14] L.F. McGinnis, J.C. Ammons, M. Carlyle, L. Cranmer, G.W. Depuy, K.P. Ellis, C.A. Tovey, and H. Xu, "Automated Process Planning for Printed Circuit Card Assembly," *IIE Transactions*, 24(4) (1992), pp. 18-30.
[15] M. Nawaz, E.E. Enscore, Jr., I. Ham, "A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem," *Omega*, 11 (1983), pp. 91-95.
[16] V.A. Petrov, *Flowline Group Production Planning*, Business Publications, London (1966).
[17] C. Proust, J.N.D. Gupta, and V. Deschamps, "Flowshop Scheduling with Set-up, Processing and Removal Time Separated, *International Journal of Production Research*, 29 (1991), pp. 479-493.
[18] R. Radharamanan, "A Heuristic Algorithm for Group Scheduling," *Proceedings of the International Industrial Engineering Conference*, USA, (1986), pp. 229-236.
[19] D.R. Sule, "Sequencing n Jobs on Two Machines with Setup, Processing and Removal Times Separated," *Naval Research Logistics Quarterly*, 29 (1982), pp. 517-519.
[20] C.S. Tang and E.V. Denardo, "Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches," *Operations Research*, 36(5) (1988), pp. 767-777.
[21] T. Yoshida and K. Hitomi, "Optimal Two-Stage Production Scheduling with Setup Times Separated," *AIIE Transactions*, 11 (1979), pp. 261-263.

Biographical Sketches

Rasaratnam Logendran is an Associate Professor in the Department of Industrial and Manufacturing Engineering at Oregon State University. He received a Bachelor of Science Degree with honors in Mechanical Engineering from the University of Sri Lanka, a Master of Engineering Degree in Industrial Engineering and Management from the Asian Institute of Technology, Thailand, and Ph.D. in Industrial Engineering and Management from Oklahoma State University, USA. Dr. Logendran's research areas of interest include cellular manufacturing, group scheduling, flexible manufacturing systems' scheduling, integration of design and manufacturing, and manufacturing simulation. In 1996, he was awarded the "Most Favorite Teacher of the Year," by the IIE Student Chapter at Oregon State University, USA. Besides teaching, he has held research fellowship positions with various organizations, including McClellan Air Force Base, Sacramento, California, The Boeing Company, Everett, Washington, and NASA – Johnson Space Center, Houston, Texas, USA, and had made significant contributions aimed at solving industry-relevant problems. Most recently, Dr. Logendran was a Fulbright Scholar at the School of Mechanical Engineering, University of Science, Malaysia, for a period of three months (June to September 2000). He is currently serving on the Editorial Board of the IIE Transactions on Design and Manufacturing, a senior member of IIE, and a member of INFORMS, SME, ASEE, and Alpha Pi Mu.

Chelliah Sriskandarajah is a Professor in the School of Management at the University of Texas in Dallas. He has a Ph.D. degree from the Higher National School of

Electrical Engineering of the National Polytechnic Institute of Grenoble (France) in the field of Production Automation and Operations Research, a M.Sc. degree in Industrial Engineering and Management from the Asian Institute of Technology (Bangkok), and a B.Sc. degree in Mechanical Engineering from the University of Sri Lanka. His research interests lie in the general area of production planning and scheduling, machine scheduling theory, maintenance scheduling, supply chain management, group technology and performance evaluation of production systems. He is an associate editor of INFOR and is a member of IIE, INFORMS, IEEE, POM and CORS.