

BI-OBJECTIVE SCHEDULING IN PRINTED CIRCUIT BOARD ASSEMBLY

Patcharaporn Neammanee
School of Applied Statistics, National Institute of Development Administration,
Bangkok Thailand, patchara@as.nida.ac.th

Sabah U. Randhawa
Office of Academic Affairs, Oregon State University, Corvallis OR 97331, USA,
Sabah.Randhawa@orst.edu

Manop Reodecha
Faculty of Engineering, Chulalongkorn University, Bangkok Thailand,
manop@eng.chula.ac.th

ABSTRACT

This paper presents an approach to solve the problem of scheduling printed circuit board assemblies on single production line to minimize total weighted tardiness and total feeder setups, which are sequence dependent. Preemptive integer goal programming model is constructed to represent the bi-objective problem. Although an optimal branch-and-bound method is available, its computational requirements are impractical for solving large problems. This paper describes a genetic-algorithm based heuristic for this model and examines its performance.

KEYWORDS: bi-objective scheduling, genetic algorithm, printed circuit board

INTRODUCTION

High competition in the last decade has forced electronics assembly to move from a high volume, low product mix to a low volume, high product mix environment, which requires setups that are more frequent. In a low volume high mix product environment, feeder setups in printed circuit board (PCB) assembly account for a significant portion of the total production time with potential to be reduced through effective process planning. Feeder setup times are sequence dependent. Therefore, total production time can be reduced if PCBs that require common components are processed sequentially. On the other hand, scheduling must also meet delivery due dates. This paper addresses an approach to schedule printed circuit board assemblies on single production line with objectives to meet due dates and to minimize feeder setup time.

Multi-objective optimization receives a lot of attention during the last two decades. By nature, most real-world problems are multi-objective. However, an approach to solve for optimality does not exist. Several studies have been working on multi-objective scheduling. Luzzatto and Perona (1993) applied Group Technology (GT) to schedule PCBs with the objectives of minimizing feeder setups and maximize workload balance. Ammons et al. (1985) considered a bi-objective model, which

attempted to accomplish workload balance and to minimize the number of board visitings to the machines. Koksalan and Keha (2003) described two bi-objective scheduling problems on a single machine: namely minimizing flow time and number of tardy jobs, and minimizing flow time and maximizing earliness (minimizing slack time). Baker and Smitj (2003) focused on scheduling jobs ordered by three criteria to a single processor. A job is assigned with one objective. There are three scheduling criteria: minimizing makespan, minimizing maximum lateness and minimizing total weighted completion time. Johtela et al. (1998) scheduled PCBs to a production line by minimizing the number of PCB families as a major objective and treating other objectives as minor.

MATHEMATICAL MODEL

Two approaches of multi-objective optimization in the prior articulation of preferences are weighted approach and goal programming. Weighted approach applies when all objectives have the same priority but may have different degrees of penalty weights. With this approach, determining weighting coefficients, which are relative to importance levels of the objectives, effects on trade-off solution. Goal programming is efficient when desired goals are known and their targets are in the feasible region. Additionally, the decision maker can classify preference levels for all objectives and identify appropriate weighting coefficients for objectives in the same level. This will discard the non-commensurable characteristics of the problem.

From studying some PCB assemblers in Thailand, these two major objectives, previously discussed, are common in scheduling PCB assemblies with different priorities. One is minimizing the total weighted tardiness, which is more important than the other, which is minimizing the total feeder setup time. Consequently, goal programming is appropriate to model an optimal scheduling with minimizing weighted tardiness as the first objective and minimizing number of feeder setups as the second one.

In order to make problems manageable, the following assumptions have to be made: (1) All PCBs are assembled on one production line; (2) The feeder of a component type cannot be duplicated on different machines at the same time; (3) All boards of each PCB type are processed in the same batch.

Variables:

- i index of component types; $i = 1, 2, 3, \dots, N$
- j index of PCB type j ; $j = 1, 2, 3, \dots, N$
- k sequence index for the k^{th} PCB type produced
- m index for machines; $m = 1, 2, 3, \dots, R$
- w_j weighting coefficient of penalty cost when PCB type j being tardy for one unit of time; $0 \leq w_j \leq 1$
- f_i number of slots required by component type i
- q_k units of the k^{th} PCB type produced
- d_{ik} number of component types i required by one board of the k^{th} PCB type

- N_k set of components required by the k^{th} PCB type
 p_{ikm} processing time to pick and place one component of type i on one board of the k^{th} board type using machine m
 s_{im} estimated feeder setup time for component of type i on machine m .
 F_m number of feeder slots available on machine m
 Υ_k component setup time for the k^{th} PCB type
 λ_k processing time for processing all boards of the k^{th} PCB type
 C_k completion time of the k^{th} PCB type; $C_0 = 0$
 T_k total production time including the processing time and setup time of the k^{th} PCB type
 D_j due date of PCB type j
 t_j tardiness of PCB type j
 TD target of total weighted tardiness
 FS target of total number of feeder setups
 $Y_{ijm} = 1$ if the component of type i is assigned to machine m for producing PCB j
 0 otherwise.

Decision variables

- $X_{jk} = 1$ if PCB j is produced in the k^{th} sequence;
 $= 0$ otherwise.
 $W_{ikm} = 1$ if component type i needs to be loaded on machine m for the k^{th} PCB in a sequence on machine m ;
 $= 0$ otherwise.

Objective functions:

- First objective : $\text{Min } p_1$
 Second objective : $\text{Min } p_2 + n_2$

$$\sum_j w_j t_j - p_1 = \text{TD} \quad (1)$$

$$\sum_k \Upsilon_k - p_2 + n_2 = \text{FS} \quad (2)$$

Constraints:

Setup time for each PCB type

$$\Upsilon_k \geq \sum_i s_{im} W_{ikm} \quad \forall i \in N_k, \forall m, k \quad (3)$$

Processing time for each PCB lot

$$\lambda_k \geq \sum_i q_k t_{ikm} d_{ik} Q_{ikm} \quad \forall i \in N_k, \forall k, m \quad (4)$$

Feeder capacity

$$\sum_i f_i Y_{ikm} \leq F_m \quad \forall i \in N_k, \forall k, m \quad (5)$$

The number of setups

$$Q_{ikm} \geq Y_{ijm} X_{jk} \quad \forall i \in N_k, \forall k, m \quad (6)$$

$$W_{ikm} \geq Q_{ikm} \quad k = 1, \forall i \in N_k, \forall m \quad (7)$$

$$W_{ikm} \geq Q_{ikm} - Q_{i(k-1)m} \quad k \geq 2, \forall i \in N_k, \forall m \quad (8)$$

Each PCB lot produced at one time

$$\sum_k X_{jk} = 1 \quad \forall i \in N_k, \forall k, m \quad (9)$$

Only one PCB type being produced

$$\sum_j X_{jk} = 1 \quad \forall i \in N_k, \forall k, m \quad (10)$$

Production time

$$T_k = Y_k + \lambda_k \quad \forall k \quad (11)$$

Completion time

$$C_0 = 0 \quad (12)$$

$$C_k \geq C_{k-1} + T_k \quad k \geq 2, \forall i \in N_k, \forall m \quad (13)$$

Tardiness

$$C_k - D_j X_{jk} \leq t_j \quad \forall j, k \quad (14)$$

Binary variables

$$X_{jk}, W_{ikm}, Q_{ikm} \in \{0, 1\} \quad \forall i, k, m \quad (15)$$

Non-negativity constraint

$$t_j \geq 0 \quad (16)$$

The two objectives of the model are to minimize deviation of total weighted tardiness and minimize sum of deviations of setup time from the targets, which are determined in equation (1) and (2). Constraints (3) and (4) determine component feeder setup time and processing time to produce each PCB type. Constraint (5) corresponds to the feeder capacity limitation for each machine. Constraints (6), (7), and (8) are used to compute the number of feeder setups. Constraint (7) determines component setups for the first PCB type on each machine whereas constraint (8) identifies the additional component setups required by the next PCB type on each machine. Constraint (9) ensures that each PCB type will be processed at one and only one time. Constraint (10) ensures that all boards of each PCB type will be produced in the same batch. Equation (11) presents the calculation of the production time, including setup time and processing time. Constraint (12) and (13) define the completion time. Constraint (14) measures the tardiness of each PCB type. Constraints (15) and (16) present binary and non-negative set of constraints for decision variables.

SOLUTION APPROACHES

The mathematical model presented in the previous section is an Integer Goal Programming (IGP). Initially, the branch-and-bound method was used, on the SAS software, to find optimal solutions of the IGP model. As expected, computation time for solving an integer programming problem increased exponentially when the problem size increased. The problem size is characterized by two dimensions - the

number of PCBs and the number of components. Problems with the number of PCBs ranging from 3 to 6, and the number of components ranging from 10 to 20 components, were executed. The computation time of the problem with 6 PCBs and 20 components was infeasible. However, in actual production, it is common to have PCB types ranging from 5 to 40 per week on a production line with the number of component types ranging from 50 to 200. Therefore, there is a need to seek for a more efficient approach to solve this problem.

Numerous studies have proposed Genetic Algorithm (GA) to accomplish bi-objective or multi-objective scheduling problems with sequence dependent setups. Murata et al. (1996) developed multi-objective genetic algorithm for scheduling problem with objectives of minimizing the makespan, total flowtime and total tardiness. To search for pareto optimal solution, weights of individual objectives in fitness function are considered as variables. The performance of multi-objective genetic algorithm is better than single-objective genetic algorithm with constant weights of each objective. Rubin and Ragatz (1995) employed GA to scheduling problem with the objective of minimizing the total tardiness in a single-stage process with sequence dependent setup environment and also examined the capability of GA to obtain near optimal schedules. Taboun et al.(1995) scheduled sequence dependent jobs on identical parallel machines to minimize total makespan and number of tardy jobs. With problem sizes ranging from 25 to 100 jobs and number of parallel machines ranging from 3 to 10, the results show that the bi-criteria GA can perform better than single criterion case. France et al. (2001) proposed a new memetic algorithm to schedule jobs on single machine with the purpose of minimizing total tardiness in sequence dependent setup environment. This algorithm is an extension from generic GA by adding hybrid population approach and local search algorithm with neighborhood reduction schemes to improve effectiveness of the algorithm.

GENETIC ALGORITHM BASED HEURISTIC

Genetic is a biological term. Biologically, offspring inherits characteristics of parents through genes. Therefore, selection of genes, termed genetic selection, can produce better offspring. In general, the rule ‘survival of the fittest in the environment’ applies in natural selection. In addition, genes can evolve through combination and mutation. The same concept is the basis for development of Genetic Algorithm, GA. Genetic Algorithm is an optimization method that utilizes the theories of evolution and natural selection to solve a problem within a complex solution space. GA searches a population of structures, which evolves according to rules of selection and genetic operators such as recombination and mutation. Each individual in the population receives a measure of its fitness in the environment. Reproduction focuses attention on high fitness individuals. Recombination and mutation provide general heuristics for exploration by adding variability.

The following describes the elements and the heuristic of the GA for solving the PCB assembly scheduling problem.

Structure Representation

A PCB assembly schedule is presented in a permutation vector of the index set $\{1, 2, \dots, N\}$, where N is the total number of PCB types. Therefore, permutation representation is suited for the problem. A chromosome, representing as a scheduling solution, contains the job numbers (PCB types) listed in the same order that they will be produced on the production line.

Initial Population and Population Size

Selecting a good initial population and population size will increase efficiency of GA. To improve the convergence rate, the Earliest Due Date, EDD, rule is applied to generate chromosomes in the initial population. The probability that a PCB type be selected in the first gene is higher than another PCB type if it has earlier due date. Therefore, the PCB type that has the earliest due date has the highest probability to be selected in the first gene and the one with the latest due date has the smallest.

Population size refers to the number of chromosomes in each generation. Generally, population size has an impact on the performance and the efficiency of GA. With small population size, the opportunity to find better solution will be small. However, with large population size, GA will consume higher computation time to solve the problem.

Fitness Function and Parent Selection

The fitness value of each chromosome is key measure to guide direction of search in GA. In this research, the first objective is to minimize the total weighted tardiness. The total weighted tardiness of the i^{th} schedule, TW_i , is calculated by

$$TW_i = \sum_{j=1}^N w_j * \max \{0, c_j - d_j\}$$

where c_j represents completion time of PCB type j , d_j is due date of PCB type j , and w_j reflects the penalty cost per unit of time for late delivery PCB the j to the customers. Thus, the fitting function of chromosome i (f_i) is

$$f_i = \frac{1}{TW_i + 1}$$

Chromosome with large fitness value has small total weighted tardiness. Adding one to the total weighted tardiness is to avoid a division by zero when it is equal zero. This function is used in France et al. (2001).

Reproduction, Mutation, Migration and Elitism

After evaluating the solution from the current population, a better solution may be found through reproducing another population. Reproduction procedure will randomly select parents (chromosomes) from the current population, and then create new offspring using crossover operators. A two-point crossover is used to produce offspring by exchange genetic materials between pairs of parents. Figure 1 illustrates an example of two-point crossover with sequencing of eight PCB types. Parents A and B are randomly selected and point 2 and point 5 are also randomly selected. The first child copies genes at the positions 1-2 and 6-8 of parent A. Positions 3-5 of the child are then filled according to the order they appear in the chromosome of parent B. The second child receives genes at the positions 3-5 from parent A. The rest of the positions on the second child are then filled by genes of parent B in the order they present on parent B.

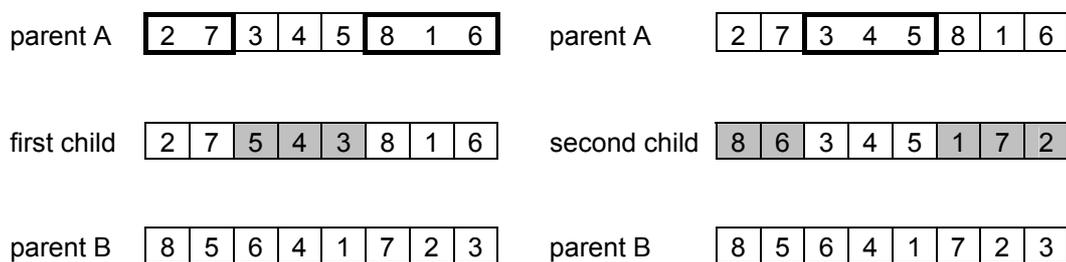


Figure 1. An example of two-point crossover

The algorithm applies elitism strategy to increase the quality of the population and force convergence. With this strategy, the next population copies some of the best chromosomes, measured by fitness value, from the current population.

In order to prevent premature convergence of the population, mutation and migration procedures is also applied to develop offspring. Each child generated from crossover procedure will get an opportunity to mutate some genes from their parents. Migration procedure will allow new solutions, which are not generated by any parents in current population, to be included the next population.

Local Search Procedure

In order to satisfy the second objective, the local search procedure is employed to reduce the number of feeder setups, as long as the first objective value does not increase. The procedure tries to move a PCB type to be processed next to the PCB types of the same family. Since applying local search for all chromosomes may consume high computation time, it might be a good idea to apply this search procedure for some of the best chromosomes in each population.

Termination Criteria

The maximum number of populations and the degree of solution improvement are used as criteria to determine whether the GA procedure should be stopped. When the best solutions of populations do not converge to a value, the maximum number of populations will terminate the algorithm. On the other hand, when the quality of

the best solutions from one population to the next does not improve significantly, it would not be valuable to continue searching solution from the next population. In this case, the degree of solution improvement will be applied to terminate the procedure.

Heuristic

Having described the elements of the Genetic Algorithm, the following describes the heuristic, which is summarized in Figure 2.

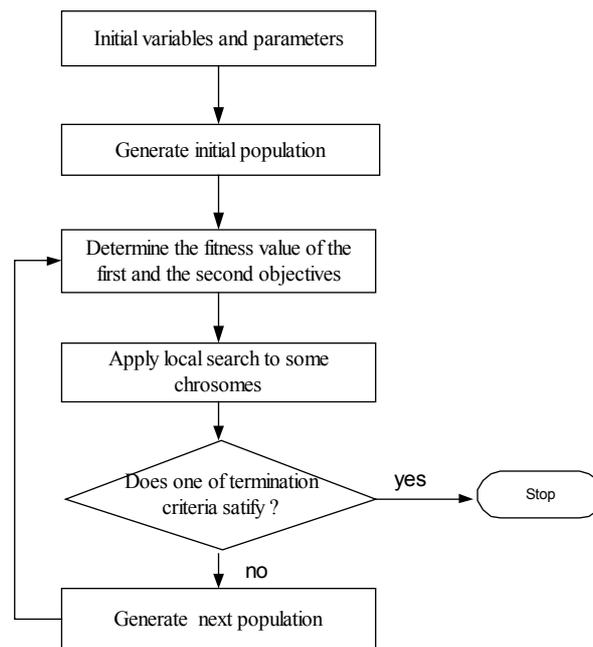


Figure 2. Genetic algorithm based heuristic

- Step 1. Determine production information and initial parameters. Production information that must be available are: the number of PCB types, the component types required by individual PCB types, the quantity of a component type per board, the PCB lot sizes, and due dates. Production characteristics including setup times, placing time per component, feeder capacity, and the number of machines are also required. Initial parameters to be specified are: the maximum number of populations, the population size, the degree of fitness function improvement, the proportion of members using the elitism strategy and the probabilities of mutation and migration.
- Step 2. Calculate the fitness value of each chromosome. In this step, the total weighted tardiness, the total number of setups, and the fitness value of each chromosome will be determined.

Step 3. Apply local search to some of the best members. Select some of the best solutions and try to improve the quality of the solution based on the second objective, minimizing the feeder setups.

Step 4. Check termination criteria. Stop if either of the criteria is satisfied; otherwise, go to Step 5.

Step 5. Generate the next population. Using reproduction, mutation, migration and elitism procedures to create offspring for the next generation. Then go to Step 2.

EVALUATION OF GENETIC ALGORITHM BASED HEURISTIC

The effectiveness of GA based heuristic was tested on four previously published problems by comparing both the total weighted tardiness and total feeder setups obtained from this heuristic with their published optimal values. Table 1 describes the test problems drawn from Ben-Arieh and Dror (1990); Gronalt et al. (1997); Hashiba and Chang (1991) and Shtub (1992), respectively. Each problem is represented by a PCB-to-component incident matrix with zero/one entries. Presence of “one” entry in row i and column j indicates that a component in column j is required by a PCB in row i .

In Table 1, the density is the total number of 1s in PCB-to-component incident matrix divided by the number of PCB types multiplied by the number of component types. The component usage is the total number of PCB types that need a particular component type, and is represented as the sum of the matrix columns. The PCB requirement is the total number of component types demanded for a specific PCB assembly; that is the sum of the matrix rows. The average component quantity is the average amount of all components on all PCB types, and equals the sum of all elements in production data matrix ($m \times n$ elements) divided by $m \times n$. The average PCB volume is an average size of all PCB lots.

Table 1. Characteristics of test Problems

	Test Problems			
	1	2	3	4
Number of PCBs (m)	7	10	8	10
Number of Components (n)	14	11	53	25
Density (%)	0.418	0.445	0.189	0.532
Minimum Component Usage	1	1	1	2
Maximum Component Usage	4	7	8	8
Average Component Usage	2.93	4.45	1.51	5.32
Minimum PCB Requirement	1	4	6	7
Maximum PCB Requirement	8	7	14	20
Average PCB Requirement	5.86	4.90	10.00	13.30
Average Component Quantity per board	15.57	12.70	74.88	69.30
Average PCB Volume	1,957	1,670	1,213	473

In order to run the GA heuristic, its parameters have to be specified. In this paper, the maximum number of populations is 50, the population size is 100, the degree of

fitness function improvement is 2, the proportion of members using elitism strategy is 0.10, probability of mutation (P_{mu}) is 0.60, and the probability of migration (P_{mi}) is 0.40.

The total weighted tardiness (TWT) and the total number of feeder setups (TFS) obtained from the GA based heuristic are compared with the optimal solutions in Table 2. Computation time (CT) of each problem on a Pentium III 930 MHz computer with 256 Mb memory is also presented.

Table 2 Results from GA on four test problems

Problem	Rep.	GA				Optimal		Diff (%)
		TWT	TFS	CT (sec)	Npop	TWT	TFS	
1	1*	475.1	21	20	20	475.1	21	0.00
	2*	475.1	21	25	25			
	3*	475.1	21	21	31			
2	1*	1,631	17	73	46	1,631.0	17	0.00
	2*	1,631	17	74	50			
	3*	1,631	17	72	50			
3	1*	2,146.1	51	75	35	2,146.1	51	0.00
	2*	2,146.1	51	51	24			
	3*	2,146.1	51	105	48			
4	1	1,731.5	33	64	42	1,727.7	33	0.22
	2*	1,727.7	33	78	50			
	3	1,751.5	33	74	48			1.37

* = replication that obtains optimal solution

The results show that GA can achieve optimal solutions in almost all test problems, and near optimal for the rest. The computation time for each run is reasonably short for practical purpose. However, there are some predetermined parameters, which may have impact on the performance of the heuristic, but have not been considered in the evaluation. Further investigation on these parameters may be useful.

CONCLUSIONS

An IGP model is developed to model a bi-objective PCB scheduling problem with sequence dependent setup environment. The problem minimizes the total weighted tardiness as the main objective and minimizes the total feeder setups as the second objective. Attempts to use the branch-and-bound method prove it infeasible to solve practical problems due to high computation time for large problems. A GA based heuristic is developed to solve the problem. It performs effectively and efficiently, producing good results in short computation time.

REFERENCES

- Ammons, J. C., Lofgren, C. B., and McGinnis, L. F., "A Large Scale Machine Loading Problem in Flexible Assembly." *Ann. Ops. Res.*, 3, 1985, pp. 319-332.
- Baker, K. R., and Smitj, C. J., "A Multiple-criterion Model for Machine Scheduling." *Journal of Scheduling*, 6, 2003, pp.7-16.
- Ben-Arieh, D., and Dror, M. D., "Part Assignment to Electronic Insertion

- Machines: Two Machines Case.” *International Journal of Production Research*, 28(7), 1990, pp.1317-1327.
- France, P. M., Mendes, A., and Moscato, P., “A Memetic Algorithm for the Total Tardiness Single Machine Scheduling Problem.” *European Journal of Operational Research*, 132, 2001, pp.224-242.
- Gronalt, M., Grunow, M., Gunther, H. O., and Zeller, R., “A Heuristic for Component Switching on SMT Placement Machines,” *International Journal of Production Economics*, 53, 1997, pp.181-190.
- Hashiba, S., and Chang, T., “PCB Assembly Setup Reduction Using Group Technology,” *Computers and Industrial Engineering*, 21(1-4), 1991, pp.453-457.
- Johtela, T., Smed, J., Johnsson, M., and Nevalainen, O., “Single Machine Scheduling with Fuzzy Multiple Criteria Optimization in Electronic Assembly.” Technical Report 227, Turku Centre for Computer Science, University of Turku, Finland 1998
- Koksalan, M., and Keha, A. B., “Using Genetic Algorithms for Single-machine Bicriteria Scheduling Problems,” *European Journal of Operational Research*, 145(3), 2003, pp.543-556.
- Luzzatto, D., and Perona, M., “Cell Formation in PCB Assembly Based on Production Quantitative Data,” *European Journal of Operational Research*, 69, 1993, pp.312-329.
- Murata, T., Ishibuchi, H., and Tanaka, H., “Multi-objective Genetic Algorithm and Its Applications to Flowshop Scheduling,” *Computers & Industrial Engineering*, 30(4), 1996, pp.957-968.
- Taboun, S.M., Abib, A.H., and Atmani, A., "Generating efficient points of bicriteria scheduling problem by using compromise programming", *Computers & Industrial Engineering* , 29(1-4) , 1995, pp. 227-231.
- Rubin, P. A., and Ragatz, G. L., “Scheduling in a Sequence Dependent Setup Environment with Genetic Search,” *Computers & Operations Research*, 22(1), 1995, pp.85-99.
- Shtub, A., and Maimon O., “Role of Similarity Measures in PCB Grouping Procedures.” *International Journal of Production Research*, 30(5), 1992, pp.973-983.