Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

SIMULATION SOFTWARE FOR SURFACE MOUNT ASSEMBLY

Thomas M. Tirpak
Flexible Manufacturing Systems Laboratory
Corporate Manufacturing Research Center
Motorola
Schaumburg, Illinois 60196, U.S.A.

ABSTRACT

SMT Sim is a simulation program specially developed for surface mount assembly. Detailed models have been constructed for studying feeder setup and chip placement operations. This software tool can be used for estimating the production cycle time for a particular set of boards and for evaluating the impact of various production policies on a surface mount factory's throughput. Following some introductory comments regarding the need for developing a tool such as SMT Sim, this paper presents an overview of the functional elements of the software, i.e., the routines for managing input data, outputs, and process models. The object-based code libraries of the SMT Sim software are also discussed. Use of the software is explained by means of an example simulation run, from build plan to simulated production report. Two types of applications are addressed: benchmarking setup procedures and analyzing assembly cost.

1 INTRODUCTION

This paper addresses the development of a simulation tool for surface mount (SMT) assembly operations. SMT has been widely adopted in the electronics industry and typically involves the following process steps: screen printing (solder paste), glue dot application, automated placement of small and large chips, robotic and/or manual placement of odd-shaped parts, reflow, inspection, and test. In some cases, SMT is combined with existing through-hole technology. Driels and Klegka (1992) and McGinnis, et al. (1992) provide good descriptions of typical surface mount production environments. Time studies and simulation analyses of SMT factories have consistently indicated that the automated small-part placement process is most often the bottleneck, as depicted in Figure 1. Thus, the high-speed, small-part placement machines or "chip-shooters" should be the focus of attention for minimizing the overall cycle time and increasing the throughput of an SMT line.

An important step in addressing the optimization of chip-shooter throughput is to develop a thorough understanding of the factors impacting cycle time. These include the sequence of component placements on a printed circuit board, the allocation of components to feeder slots (on one or more available placement machines), and the order in which pending lots are released into production. For a

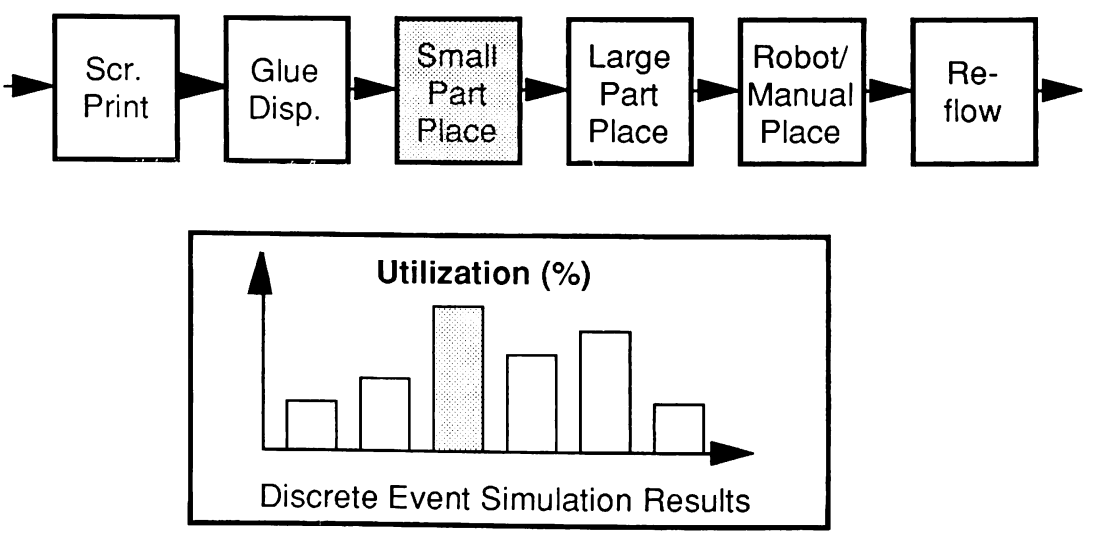


Figure 1: Layout and Utilization of a Typical Surface Mount Line

typical SMT production environment, cycle time reduction involves a complex set of interrelated optimization problems. The recent literature has dealt with each of these subproblems. An extensive list of references has been compiled by McGinnis, et al. (1992). Heuristics for solving the placement sequence optimization problem are discussed by Ball and Magazine (1988) and Mehra (1993). The feeder setup problem, which may include constructing both standard setups and individual or custom setups, is addressed by McGinnis, et al. (1992) and Sadiq (1991). Several techniques have been developed for sequencing the production of SMT lots. The algorithm developed by Piramuthu, et al. (1992) simultaneously addresses the feasibility of a schedule with respect to assigned due dates, sequence-dependent feeder setup reduction, and work-in-progress minimization. Learning-based scheduling for SMT has been integrated with Monte Carlo simulation (Sikora, et al. 1992). Tirpak (1993) provides an overview of current work in scheduling for high-mix surface mount assembly. Clearly, a wide variety of optimization techniques exists, but how can one test a methodology and validate the results for a particular production environment without interrupting production in that factory? The answer lies in constructing a detailed simulation model of the chip assembly process.

The remaining sections of this paper address the design and application of SMT Sim, a software tool developed by the Motorola Corporate Manufacturing Research Center (CMRC). Detailed simulation models have been constructed for studying feeder setup and chip-shooter placement operations. This software can be used for estimating the production cycle time for a particular set of boards and for evaluating the impact of various production policies on an SMT factory's throughput. Section 2 presents an overview of the functional elements of the software, the routines for managing input data, outputs, and process models. The object-based architecture of the SMT Sim software is discussed in Section 3. Use of the software is explained in Section 4 by means of an example simulation run, from "job script" (or build plan) to simulated production report. In Section 5, two types of applications are presented: benchmarking setup procedures and analyzing assembly cost. The paper concludes with a brief discussion of possible enhancements as well as further applications for this software.

2 FUNCTIONAL DESIGN OF THE SMT SIM SOFTWARE

SMT Sim is a simulation tool for studying the effects of feeder setups, part assignments, job scheduling, and other related factors on SMT production operations. The main feature of the software is that one can accurately predict the cycle time required to produce a given assortment of boards. Though the initial implementation of the software was for a common chip-shooter operating in a relatively high-mix, moderate-volume environment, design considerations for a variety of equipment types and production environments have been taken into account.

In addition to being a tool for evaluating the cycle times and feeder changes for various production scenarios, SMT Sim can perform some straightforward but important tasks. First, the software incorporates a set of heuristics for optimizing machine placement programs. The software can be run as a batch-mode optimizer for a collection of placement files. Second, if a new board design arrives in the factory, SMT Sim can construct the necessary setup instructions and an optimized placement file (updated for this new setup). A *delta-setup* can be created based on a standard setup or the current setup. The inputs to and outputs from the program are depicted in Figure 2.

Inputs to the software include several sources of information describing the build-plan and the part feeder reels which must be setup on a placement machine. The "job script" contains a list of the file names of the placement programs for each pending lot, in the order in which they are to be produced. Changes in feeder setups are also noted in this file. A sample "job script" is explained in detail in Section 4. The second type of input, the placement programs, consists of the machine-readable numerical code used to specify the sequence of chip placements for a given board design, i.e., the X location, Y location, part rotation, etc. for each placement. The current version of SMT Sim reads placement information in two standard machine-readable data formats. One also needs to specify the feeder

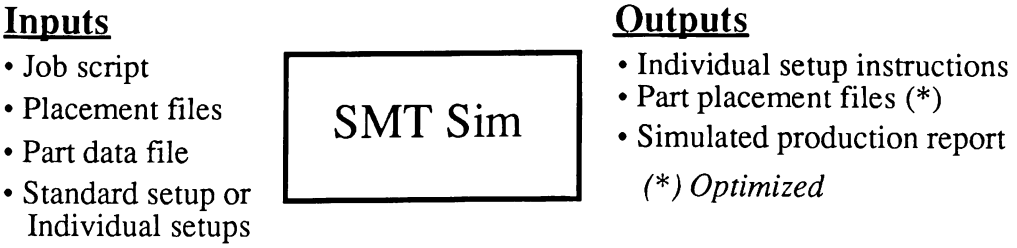


Figure 2: A Diagram of Inputs to and Outputs from SMT Sim

type to be used for each part number. This defines the feeder tape width, e.g., 8 mm, and thus the number of slots required on the feeder carriage for each part type. The fourth input is used in constructing feeder setups for the different board designs. A feeder setup is essentially a list of part numbers and their assigned feeder slots. One can either specify feeder setups for each board individually or provide a "standard" setup to which additional "custom" (non-standard) parts can be added, if necessary.

The software outputs the feeder setup and (optimized) chip placement instructions for each board in the "job script" as well as a simulated production report, a sample of which is explained in Section 4 of this paper. The original release of the software receives all its inputs from and writes all its outputs to ASCII text files. In a subsequent implementation, however, the software has been integrated within an existing manufacturing data base, and all placement and feeder data are retrieved from and stored to prespecified tables in the relational data base.

The construction of a simulation model for the SMT Sim software is logically divided into two parts, one addressing chip placements and the other covering feeder setups. Cycle time estimates computed by the software are based on a characterization of the placement operations of a common chip shooter as well as the time-study data for feeder change operations in a typical Motorola SMT factory. The chip placement model includes all the main factors impacting the total assembly time for a particular surface mount board, i.e., board load time, fiducial checks, chip placements, and board unload time. Each of these cycle time components can be identified separately and incorporated into the model. Figure 3 shows the directions of motion for a typical chip shooter.

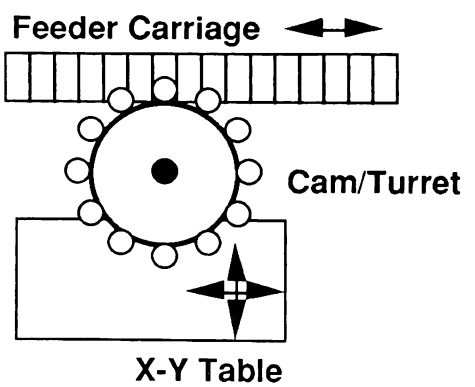
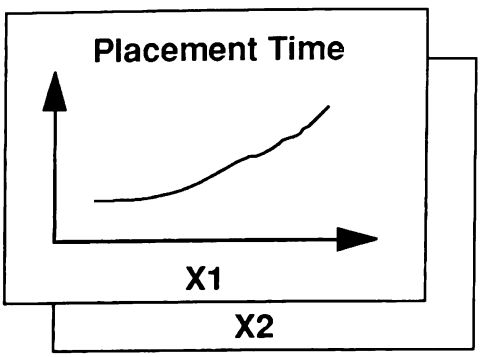


Figure 3: Machine Layout for a Typical Chip Shooter

Using designed experiments, mathematical models have been developed to predict the cycle time for a placement program on a several chip-shooters. Figure 4 shows the generic form of such a model.



Characterization Experiments

PLACEMENT TIME = f(X1, X2, ...)

Figure 4: Cycle Time Characterization for Chip Shooters

The simulation model addresses feeder setups with respect to three issues: the cycle time required for changing setups, the number of feeder changes, as well as the mechanics for constructing a feasible feeder setup for a particular board, given its bill-of-materials (BOM) and the setup currently on the machine. The detailed model of the feeder setup process enables the user to specify parameters such as the time required for an "ADD" (mounting a new feeder reel in a previously empty slot), a "BREAK" (removing a feeder from a given slot), or a "BREAK/ADD" (replacing a feeder with a new one). An assumption has been made that feeders can be prepared off-line and that the only down-time is the time actually spent changing feeders on the chip-shooter. One additional factor identified in the time study for a Motorola SMT factory has been incorporated into the SMT Sim model. Namely, the time study data showed that the duration of feeder change operations depends on the sizes and types of the feeders which are moved. Inspection times for a new setup have likewise been added to the model. These estimates are computed using a per-feeder inspection time, assuming that the machine operators verify each ADD or BREAK whenever the feeder setup is changed. Although the actual times for feeder setup and inspection may vary from factory to factory, the form of the model remains the same.

3 THE OBJECT-BASED SOFTWARE ARCHITECTURE

The SMT Sim software has been developed as a set of C language libraries according to an object-based architecture. Each library contains routines for handling a specific part of the simulation, e.g., the file input, the feeder setup model, the chip placement model, the simulation of a "job script," etc. SMT Sim is a numerical simulator which

computes the cycle time for each lot in a "job script" and then sums these times together to form the overall cycle time for the build plan. The cycle time estimates are computed via specialized software models incorporating the features of SMT assembly outlined in Section 2. A simulation run essentially consists of a sequence of functional evaluations, each yielding a cycle time estimate based on the particular placement program and feeder setup. This is in contrast to discrete-event simulations, which incorporate an event calendar and a set of possible state transitions. As is the case with SMT Sim, specialized simulators typically utilize a standard programming language, such as C or FORTRAN, to achieve the desired level of modeling detail. Dudewicz and Karian (1985) and Law and Kelton (1991) provide some useful guidelines for developing such software.

Figure 5 depicts the organization and interrelation of the SMT Sim software libraries and provides a brief description of the routines in each library. A library fully contained within another library is one such that its functions and data are used exclusively by the library which contains it. The SMT setup and placement model has been organized according to the diagram in Figure 5 so that the software can be programmed and debugged in a modular fashion. To adopt this software for another factory model or different machine types, the code in only a few libraries would need to be modified.

The main program module, top-most in the hierarchy, is found in the file *smtsim3.c*. The remaining portion of program execution is controlled by the function evaluate() in library *driver.c*. A "driver object" represents the methods and data used for one particular piece of machinery in a surface mount line. For the current version of SMT Sim, this is a common chip-shooter used in many Motorola factories. One will note in Figure 5 that the "driver object" accesses libraries related to both feeder setups *(feeders.c and setup.c)* and placement sequences *(fcpmdr.c)*. The evaluate() routine also calls routines in the *parts.c* library, which manages a data base of discrete components to be placed on each type of board. The functions in *fcp3opt.c* handle placement sequence optimization. Finally, each of the above-mentioned libraries accesses the *utility.c* library which contains a number of functions for error-handing, I/O and expression parsing. Figure 6 lists the functions in the *feeders.c* library.

```
int copy_setup();
int read_setup();
int zero_setup();
int construct_setup();
int assign_feeders();

int mark_assigned_slots();
int fill_not_assigned_slots();
int write_delta_setup_file();
int is_feasible();
int check_spacing();
int count_num_slots();
```

Figure 6: Functions in the SMT Sim Feeders Library

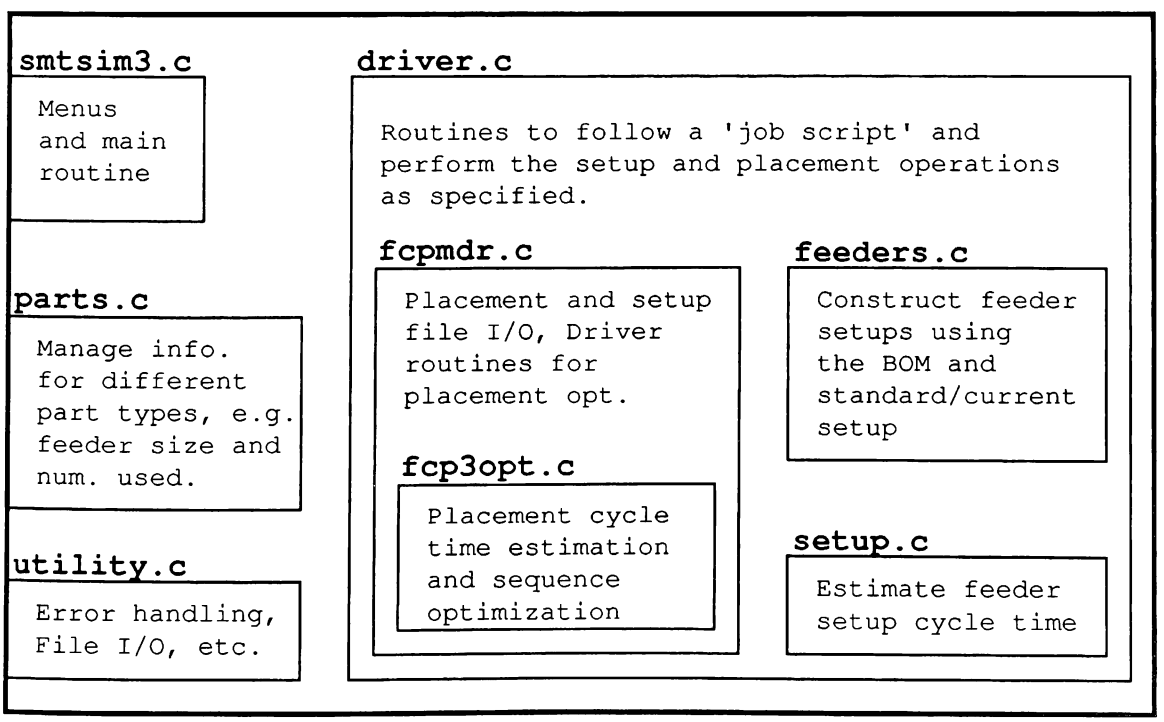


Figure 5: Object-Based Design of the SMT Sim Software

4 EXAMPLE SIMULATION RUN

The best way to explain the operation of the SMT Sim software is by showing an example simulation run. This example will focus on the "job script" as the main input and the estimated production report as the main output. The "job script" basically defines the "degrees of freedom" or variables in simulating high-mix SMT production. In addition to listing the types and quantities of boards to produce, the "job script" includes information regarding standard feeder setups, setup changes, rules for ordering standard and custom parts, as well as settings for the level of optimization to perform. The format of a "job script" is explained in Figure 7.

Figure 8 shows the information displayed to the computer screen when one runs the SMT Sim software. The "job script" defined in Figure 7 has been used for this simulation run. It should be noted that all placement programs for boards listed in the "job script" must be in the local file directory. The part data base file *parts1.dat* provides the mapping between part number and feeder type. A list of component shape codes and their associated cam speeds is found in file *pdgfile*. Information displayed to the screen, as in Figure 8, is likewise printed to the estimated production report file.

The first step in the simulation is to initialize the standard feeder setup. The placement programs listed in the "job script" subsequently use this standard setup as the

```
Production forecast data. File: plan.in                    (Line 1)
Date: 6/22/93                                              (Line 2)
-1  plan.std            -1  31                             (Line 3)
^   ^                   ^   ^
a   b                   c   d

1   26  board1.csc       11  board1.outc       1          (Line 4)
^   ^       ^           ^       ^             ^
e   f       g           h       i             j          ...
Continued
```

Key:

- **a** ***Feeder setup data line:*** The first field of the third line is always set to **-1**. When using the "job script" for simulating a production scenario with SMT Sim, there may be more than one feeder setup data line, i.e., one for each change in standard (current) setup and/or parameters governing the addition of non-standard parts.
- **b** The name of the file containing the new feeder setup: **plan.std**.
- **c** A **-1** signifies that the entire setup in file **plan.std** is considered a "standard" setup. Otherwise, this field contains the number of standard slots.
- **d** A code representing the set of rules used to generate and apply new feeder setups for a particular board, given the current feeder setup.

- **e** ***Board assembly data line:*** This first data field can be 0 or 1. A **1** flag denotes that the feeder setup must be modified in order for this board to be produced. A zero (0) value is used when the feeder carriage is to be left unchanged.
- **f** There are **26** boards of this type to produce.
- **g** The name of the input file containing the placement data: **board1.csc**.
- **h** The type of placement sequence optimization to be performed. Option **11** is one of the available settings and will invoke a fast, first-cut optimization.
- **i** The name of the output file for the optimized placement sequence: **board1.outc**.
- **j** A flag specifying whether or not the feeder setup is to be changed back to the standard (or initial) setup. For a fixed standard setup, this flag would be set to **1**. For a "rolling setup" or "delta-setup" constructed for each new board based on the parts in the current feeder setup and the BOM for the board, this flag would be set to 0.

Figure 7: Explanation of the SMT Sim Job Script Format

base setup and add parts as necessary. For each entry in the "job script," cycle time estimates are computed for the placement operations (after sequence optimization) as well as for the setup and inspection operations. The totals are printed as the last step before SMT Sim finishes executing. In this example, the production totals indicate that roughly 17% (178/1061) of the total time is spent for setup and inspection.

5 APPLICATIONS FOR THE SMT SIM SOFTWARE

The SMT Sim software can be used for a variety of experimental studies which address benchmarking of schedules and throughput optimization procedures as well as cost modeling and design for assembly (DFA). The software routines containing the detailed, low-level model of the chip-shooter placement process can be modified as necessary to reflect changes in the operating parameters of the machine. Many factors, however, may be studied by simply changing a few data fields in the "job script." The

```
*-------------------------------------------------------------*
*  SMT Sim v1.5                                    FCP-III    *
*                                                             *
*  An SMT production cycle time simulation tool               *
*  (Feeder setup and chip placement)                          *
*                                                             *
*  June 22, 1993                              Motorola CMRC   *
*-------------------------------------------------------------*
[MCS-I data format selected]

Name of job script file (file.in): plan.in
Name of report file (plan.rep):
Name of part data file (parts1.dat):
Name of 'pdg' file (pdgfile):
Is this OK? (y/n/q) y

New Setup: plan.std
45 parts - Standard slots: 1 to 48. Setup rules = "31"
Setup and inspection time:    11.14 minutes

Placement file (input): board1.csc
Placement file (output): board1.outc
The optimized cycle time is   41.4790 seconds.

Placement file (input): board1.ssc
Placement file (output): board1.outs
The optimized cycle time is  147.9926 seconds.
      .
      .     Repeat for each placement program.
      .
Total production time:     1061.56 minutes
Total placement time:       883.24 minutes
Total setup/insp. time:     178.32 minutes
Total feeder changes:          743

Program successfully executed.
```

Figure 8: Sample Simulation Run Using SMT Sim

Motorola CMRC has used SMT Sim to study a number of production environments and to investigate the impact of several optimization alternatives. Guidelines for some of these tests are listed in Figure 9. Within Motorola, the software has been most beneficial as a means to test optimization algorithms for cycle time improvement in high-mix SMT factories. Moreover, results from SMT Sim simulations have been used to provide early feedback to production managers so that they can justify the engineering effort required to implement these optimization techniques, based on predicted cycle time and operational benefits.

The SMT Sim software also applies to the calculation of manufacturing costs. Significant improvements over existing methods, such as Activity Based Costing (ABC), can be achieved. It is well known that simply multiplying the number of components in a board design times an average cost per placement (either in dollars or in assembly cycle time) will not yield a consistently reliable picture of what is actually driving the cost. With SMT Sim, however, one can quantify the impact of particular features of a board design on the overall production cycle time. The Motorola CMRC has completed one such cost analysis for a high-mix SMT factory in which the BOMs for the mix of board designs included two sets of equivalent chip resistors. An industrial engineer in the factory brought this fact to the attention of the CMRC noting that the redundant set of resistors was causing longer setup times. The engineer explained that he had mentioned this to the board designers but that they did not see the problem. SMT Sim was used to calculate the component of the overall production time for a typical week-long build plan that can be directly attributed to the presence of redundant parts. This was a fairly straightforward task. About ten lines of code were changed in the *parts.c* library to exchange any "redundant" resistors for their "non-redundant" counterparts. As anticipated, the simulation estimated that on a weekly basis, ninety minutes of production time was lost solely due to the fact that the board designs were not standardized for the two sets of resistors.

6 CONCLUDING REMARKS

The SMT Sim software incorporates a detailed, object-based model of feeder setup and chip placement operations for a chip-shooter. This tool can provide manufacturing engineers, supervisors, and circuit board designers with reliable estimates of the production cycle time for a given collection of boards. Furthermore, a variety of "What if?" analyses can be performed allowing one to quantify the impact of production policies related to scheduling, standard feeder setups, etc. on the overall SMT assembly cycle time.

Future work with the software will likely include modeling additional types of chip-shooters and studying new production environments. It may be worthwhile to convert the software from the present object-based C code libraries to a full object-oriented implementation in C++. This will simplify the task of adding new placement ma-

What to Test	How to Test with SMT Sim
• Production schedule	• Update the sequence of lots in the "job script."
• Individual feeder setups for different boards	• Add a "feeder setup data line" for each different board with the name of the file containing each setup.
• Standard setup	• Change the name of the standard setup in the first "feeder setup data line" (line 3 in the "job script).
• Method for adding non- standard parts to feeder setup	• Select available options by changing the flag on the first "feeder setup data line" (line 3 in the "job script). New options can be added by modifying the function `fill_not_assigned_slots()` in *feeders.c*.
• Standard setup vs. rolling-setup	• Set the feeder reset flag to 1 or 0, as appropriate, on each "board assembly data line" in the "job script."
• Placement sequence optimization heuristics	• Include the heuristics in the optimizer library *fcp3opt.c*. Measure impact with the placement time characterization.

Figure 9: Some Test Cases and How to Study Them with SMT Sim

chine models, and significantly reduce the effort required for joining several machine models together to construct a detailed simulation of an entire SMT line. Another promising area for work relates to the integration of the software with existing manufacturing and order-entry data bases as well as its ease of use within a factory's computer-aided production planning (CAPP) system and eventually within the circuit board design process.

ACKNOWLEDGMENTS

The author wishes to express his sincerest thanks to the following Motorola engineers who have assisted in developing and implementing the SMT Sim software: Neal Crouse, Kevin Kent, and Ed Winter of the Land Mobile Products Sector, Joe Cawley, Leo Larivee, and Steve Sabetta of Motorola Codex, and Tom Babin of the Corporate Manufacturing Research Center.

REFERENCES

Ball, M.O., and M.J. Magazine. 1988. Sequencing of Insertions in Printed Circuit Board Assembly. *Operations Research*, Vol. 36, No. 2.

Driels, M., and J. Klegka. 1992. An Analysis of Contemporary Printed Wiring Board Manufacturing Environment in the USA. *International Journal of Advanced Manufacturing Technology*, no. 7, pp. 29-37.

Dudewicz, E.J., and Z.A. Karian. 1985. *Modern Design and Analysis of Discrete-Event Computer Simulations*. IEEE Computer Society Tutorial, IEEE Catalog No. EHO227-9.

Law, A.M., and W.D. Kelton. 1991. *Simulation Modeling and Analysis*, second ed. McGraw Hill, Inc., 1991.

McGinnis, L.F., J.C. Ammons, M. Carlyle, L. Cranmer, G.W. DePuy, K.P. Ellis, C.A. Tovey and H. Xu. 1992. Automated Process Planning for Printed Circuit card Assembly. *IIE Transactions*, vol. 24, no. 4, pp. 18-30.

Mehra, V. 1993. Reducing Cycle Times for Surface Mount Production Processes. *Proceedings of the NEPCON West '93 Conference*, Technical Session 18, Anaheim, CA, February 7-11.

Piramuthu, S., N. Raman and M. Shaw. 1992. Learning-based Scheduling in a Printed Circuit Boards Production System. The Beckman Institute, University of Illinois, Urbana, Illinois, Artificial Intelligence Technical Report UIUC-BI-AI-DSS-92-01.

Sadiq, M. 1991. An Automated Methodology for Intelligent Slot-Assignments in Surface Mount Assembly. Master's Thesis, Department of Industrial Engineering, University of Arkansas, Fayetteville, Arkansas.

Sikora, R., D. Chhajed, M. Shaw. 1992. Integrating the Lot-sizing and Sequencing Decisions for Lead-time Reduction in a Printed Circuit Boards Manufacturing Environment. The Beckman Institute, University of Illinois, Urbana, Illinois, Artificial Intelligence Technical Report UIUC-BI-AI-DSS-92-04.

Tirpak, T.M. 1993. A Case Study in Computer-Aided Scheduling for High-Mix Surface Mount Assembly. NEPCON East '93 Conference, Technical Session 6, Boston, Massachusetts, June 14-17.

AUTHOR BIOGRAPHY

THOMAS M. TIRPAK is a Staff Engineer in the Manufacturing Optimization Group of the Flexible Manufacturing Systems Group at Motorola's Corporate Manufacturing Research Center (CMRC) in Schaumburg, Illinois. His current work involves techniques for reducing the cycle time of surface mount assembly processes, with a special emphasis on high-mix operations. He is a member of ORSA, TIMS, and the IEEE Systems, Man, and Cybernetics Society.