A MANPOWER SCHEDULING SYSTEM FOR A HELICOPTER REASSEMBLY SHOP

Ki-Young Jeong
Industrial Engineering Tech. Program, South Carolina State Univ.
300 College Street, NE Orangeburg, SC 29117, Tel (803) 516-4127, Email: kjeong@scsu.edu

Olga Bagatourova
Optimization and Capacity Modeling Team, Bank of America
1401 ELM st., Dallas, TX 75202, Tel (214) 508-5333 Email: Olga.Bagatourova@bankofamerica.com

ABSTRACT

This paper presents the design concept, functionalities and application of the simulated annealing-based manpower scheduling systems (SAMSS) at a labor-intensive helicopter reassembly shop in the Southern US. A helicopter maintenance process typically follows a disassembly, a repair or an overhaul, and a reassembly process. Among these, the reassembly process is considered as a bottleneck, and significantly relies on human labor. Hence, the shop management requested a user-friendly manpower scheduling tool providing a feasible and an optimized schedule under diverse what-if conditions. The design of SAMSS started from the capture of the reassembly process information with the IDEF3 (Integrated DEFinition) method, which was translated into a task network with shop status data, then heuristics and simulated annealing-based scheduling algorithms were applied to provide a feasible solution and/or a near optimal solution. To increase the compatibility with existing software, the MS-Excel® and MS-Project® interfaces were developed. A manpower scheduling system needs to promptly provide a short term schedule and diverse what-if scenario analyses to shop managers for better decision making. The Gantt chart, workload distribution chart, and flowtime sensitivity analysis for each crew supported in SAMSS were considered useful in a real life case study presented.

INTRODUCTION

A military helicopter has several types of maintenance programs such as phase maintenance and overhaul maintenance for safety and for asset readiness. The phase maintenance is a schedule maintenance program where a helicopter is sent to a maintenance shop for inspection after specific flight time hours, which depends on the helicopter type. In the overhaul maintenance, the helicopter is completely overhauled, repaired, and reassembled after specific calendar time based use – i.e. 3 years. A depot facility provides both an overhaul maintenance program and a heavy maintenance program for the helicopter requiring significant repair service. This paper is based on the overhaul case study performed in an army depot located at the Southern part of the United States.

Once a helicopter is inducted for the overhaul service, the helicopter frame is placed at one of the bays, and typically moves through 1) a pre-shop analysis and disassembly, 2) a cleaning process, 3) a paint and strip process, 4) a structure and electrical service process, 5) an exterior and interior prime process, 6) a reassembly process, 7) a final paint process, and 8) a final flight test process area. Through the pre-shop analysis, the helicopter is disassembled into several modules and components for inspection. Based on the inspection result, the modules and components are routed to appropriate shops not shown in the above processes - People call these shops back-shops since they are located in the back side of the depot. For example, an engine is routed to the engine shop, and the blades are routed to the blade shop. The disassembled main frame is moved in succession to be cleaned and stripped. Then the structure and

electrical check and the repair of the helicopter are performed, followed by an exterior and interior prime process necessary for repainting the helicopter. After that, the frame is reassembled at a bay with required modules and components through manual operations. The helicopter is then cleaned before going through the final paint and flight test process. It should be noted that the disassembly and reassembly processes use different bays on the same floor to accelerate both processes. The total flowtime - time between the job starting and ending - through all these processes currently varies from 60 to 300 eight-hour days depending on work contents.

Since the previous shop capacity analysis showed that the reassembly process was a bottleneck, the shop management wanted to improve the bottleneck performance with an optimized reassembly schedule. The management also wanted to obtain the answers to the following questions: 1) "What is the expected reassembly flowtime with current manpower and workload?", 2) "What percentage of the flowtime can be reduced if additional manpower is added?", 3) "What is the bottleneck resource in the reassembly process?" There were five bays for reassembly, and each one was dedicated to a single helicopter. There were seventeen workers with five different skill types, and the members with the same skill type is called a crew. Currently, the five crews were required to reassemble a helicopter: six mechanical engineers (AMG), four electirical engineers (AEE), four avionics engineers (AV), two sheet metal engineers (AS), and one inspector (INSP). However, there was no written document on "who does what" information. Hence, we first observed the environment and then interviewed the domain experts to define basic tasks (operations) with appropriate level of abstraction to avoid the excessive complexities and efforts associated with the data collection. The basic task is a unit process for which the process time is observed to decide the schedule. The PROSIM® [3] software was used to define basic tasks since this software supports the IDEF3 process description method and provides a convenient user-interface to capture resource requirements.

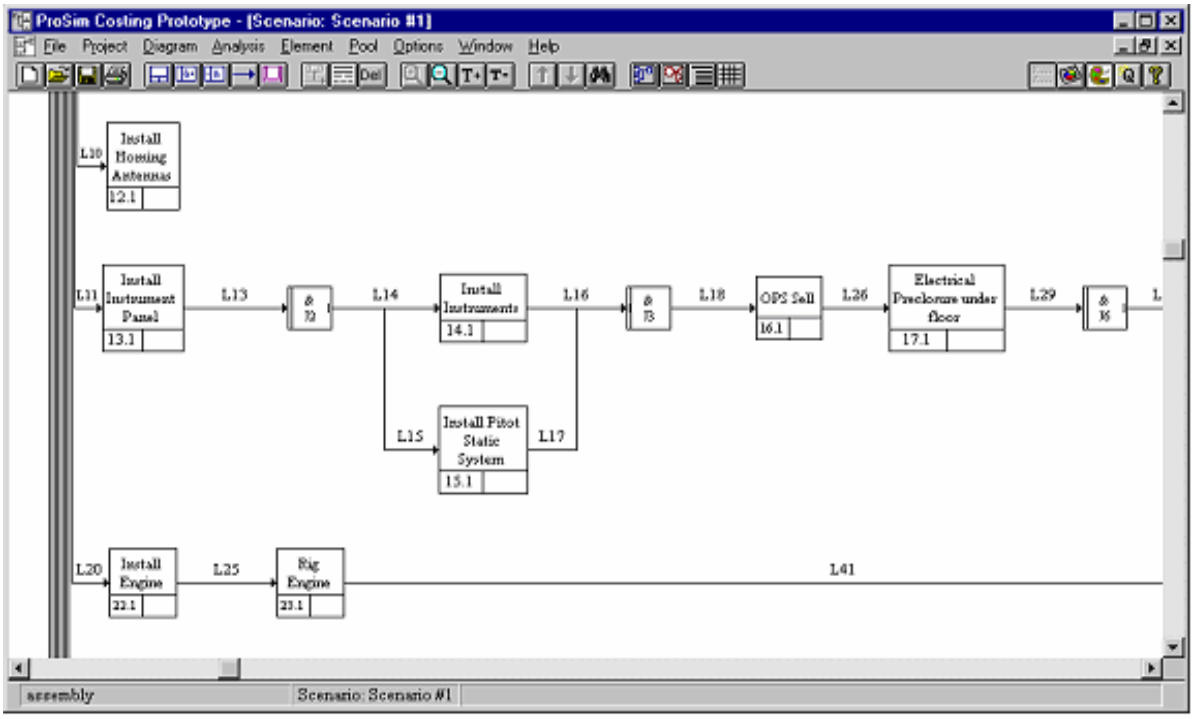

Figure 1. An Example of Helicopter Reassembly Process Map

In the IDEF3 schematics, a UOB (unit of behavior) shown as a rectangular box represents a process or a task. The temporal, logical, and timing relations between the processes are captured by three types of junctions: “AND”, denoted by “&”, inclusive OR, denoted by “O”, and “exclusive OR”, denoted by “X”. These junctions can be classified into a Fan-in type or a Fan-out type based on their role. In a Fan-in type junction, there are one or multiple incoming processes and one outgoing process to and from the junction, respectively, and in a Fan-out type junction, one or multiple outgoing processes are linked to one incoming process. The IDEF3 method graphically represents the process information using these UOBs and Junction. For detailed information on IDEF3, readers are encouraged to refer to Mayer [4].

A part of the process diagram in the reassembly shop is represented in Figure 1. According to the analysis of the IDEF3 diagram, a helicopter required 186 reassembly tasks (operations) under precedence relation constraints. As seen in Figure 1, some tasks such as “Install Instruments” and “Install Pitot Static System” could be performed in parallel by a single electircal engineer (AEE). Hence they are capsulated by a Fan-out and a Fan-in “AND” junctions. These tasks may compete to seize the required resource. Hence, because of this competition, several alternative scheduling scenarios will be generated. Among these alternative, the shop management preferred the schedule with the shorter flowtime to reduce the total helicopter maintenance lead time.

ANALYSIS OF REASSEMBLY SHOP

The manpower scheduling problem for a single helicopter reassembly within a bay can be modeled as a resource-constrained project-scheduling problem (RCPSP), where a complete task network consists of multiple nodes (tasks) and arcs (precedence relation) among tasks, and each task is competing for limited resources. Within a bay, workers are the constrained resource. However, the bay availability also affects the flowtime and the worker’s workload since the workers are shared across the bays. Hence, each helicopter will seize a bay, and get services provided by the workers, and eventually it will release the bay, and move to the next process.

The specific requirements for the manpower scheduling system requested by the shop management were as follows: 1) It needs to provide a feasible schedule within a short period of time since the manager wants to frequently deploy it for a short term schedule – a shift and/or a day, 2) It needs to provide a near optimal schedule in terms of the flowtime, 3) It needs to provide manpower workload distribution and “what if” analysis, and 4) It is supposed to provide the user-friendly graphic interface.

Based on the shop description and customer requirements, we initially considered a simulation-based optimized scheduling system as in Paul and Chanev [5] where they combined a simulation engine with a genetic algorithm to search an optimal schedule. The simulation is a generic approach, and the generic algorithm works as an optimizer to improve the schedule generated by the simulation. However, in our case, we are specifically interested in the manpower schedule. Hence the time consuming simulation approach is not needed since a simulation based approach takes longer execution time due to the event scheduling methods as described in Jeong [2]. Therefore, we decided to use the simulated annealing-based- approach. The main reason why we adopted the simulated annealing as our optimizer is its simplicity of implementation compared to the genetic algorithm while the performance of the simulated annealing is still r as good as the genetic algorithm for many scheduling problems.

THE ARCHITECTURE OF SAMSS

This section describes the concept and architecture of the simulated-annealing-based-manpower-scheduling system (SAMSS) to implement the requirements discussed in the previous section. The SAMSS consists of three main components: “Shop Status Database” (SSD), “Task Generator” (TG), and

"Schedule Generator" (SG). The simplified architecture and information flow among these three components is described in Figure 2. The "Task Generator" (TG) provides the input to "Schedule Generator" (SG) by reading "Shop Status Database" (SSD). The precedence relations, the remaining task time and the tasks' resource requirement are used to construct the input. The "Schedule Generator" builds a schedule based on this input.

Users can select either a feasible schedule generation option or an optimized schedule generation option. In the feasible schedule generation option, it uses popular heuristic rules, and in the optimized schedule generation option, it activates the simulated-annealing engine. The "Schedule Generator" presents a Gantt chart, manpower workload distribution and several performance metrics. The shop managers can use this information as a decision support aid to estimate the actual workload and work progress. For better results, it is important to keep consistency between the "Shop Status Data" and the reality. Hence, once a daily or shift work is finished, the progress of each task must be recorded into the "Shop Status Database". The selection of the MS-Project® and the MS-EXCEL® interface makes this data collection and update easier since many domain experts are already very familiar with these two products.

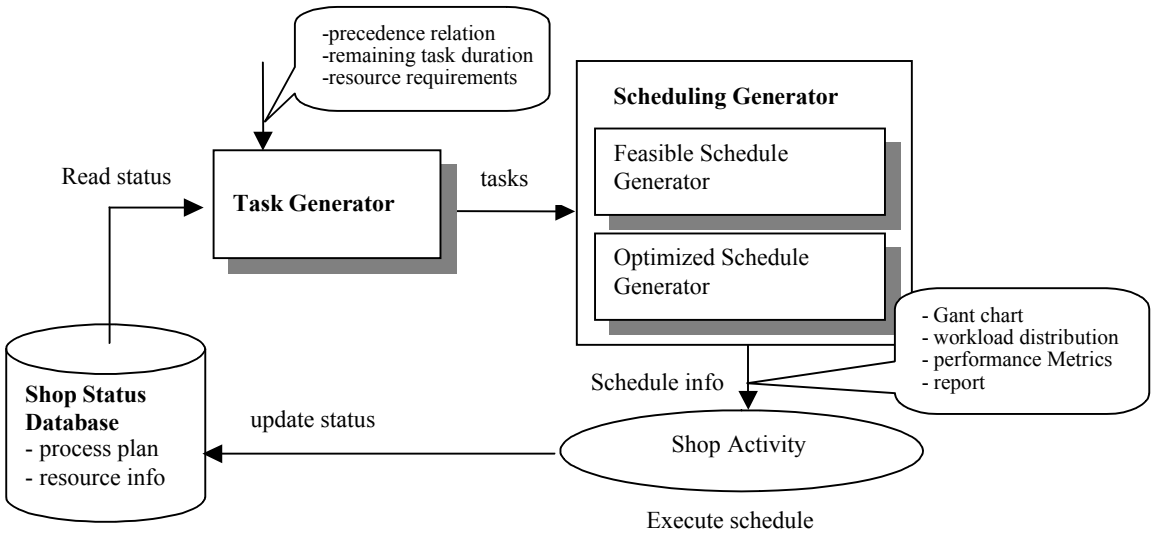

Figure 2. Concept of SAMSS

Shop Status Database

The SAMSS was designed to use an MS-ACCESS® database. A simplified data model with IDEF1X is presented in Figure 3. The IDEF1X is a graphical language to capture the information to build a data model. It captured the entities (objects) and their relations. In this case, there are six major entities: "Resource," "Task," "TaskResource," "Assignment", "Aircraft", and "TaskPredecessor". Each entity can be classified as either independent (rectangular box) or dependent (rounded rectangular box). A dependent entity depends on the migration of the primary key from independent entities for the definition of its own primary key. In our IDEF1X model, "TaskResource" "TaskPredecessor", and "Assignment" are the dependent entities and the remainders are independent entities. The primary keys of each entity are represented above the horizontal grid line while the other attributes are denoted below the line. A foreign key is labeled by "FK" and the one-to-many (at least one) relation is denoted by "P". The solid line without any label indicates a one-to-many (zero, one or more) relation. For instance, the "TaskResource" entity has two foreign keys as a composite primary key: one migrated from the "Task"

entity and the other from the "Resource" entity. The "Resource" object has at least one or more "TaskResource" objects and the "Task" object has zero, one, or more "TaskPredecessor" objects.

Task Generator

The "Task Generator" generates the input data for the "Schedule Generator" by reading the "Shop Status Database". The input data consists of two types: task information and resource information. The task information includes the task arrival time, task duration, immediate predecessors, required resources, and required quantity of resources.

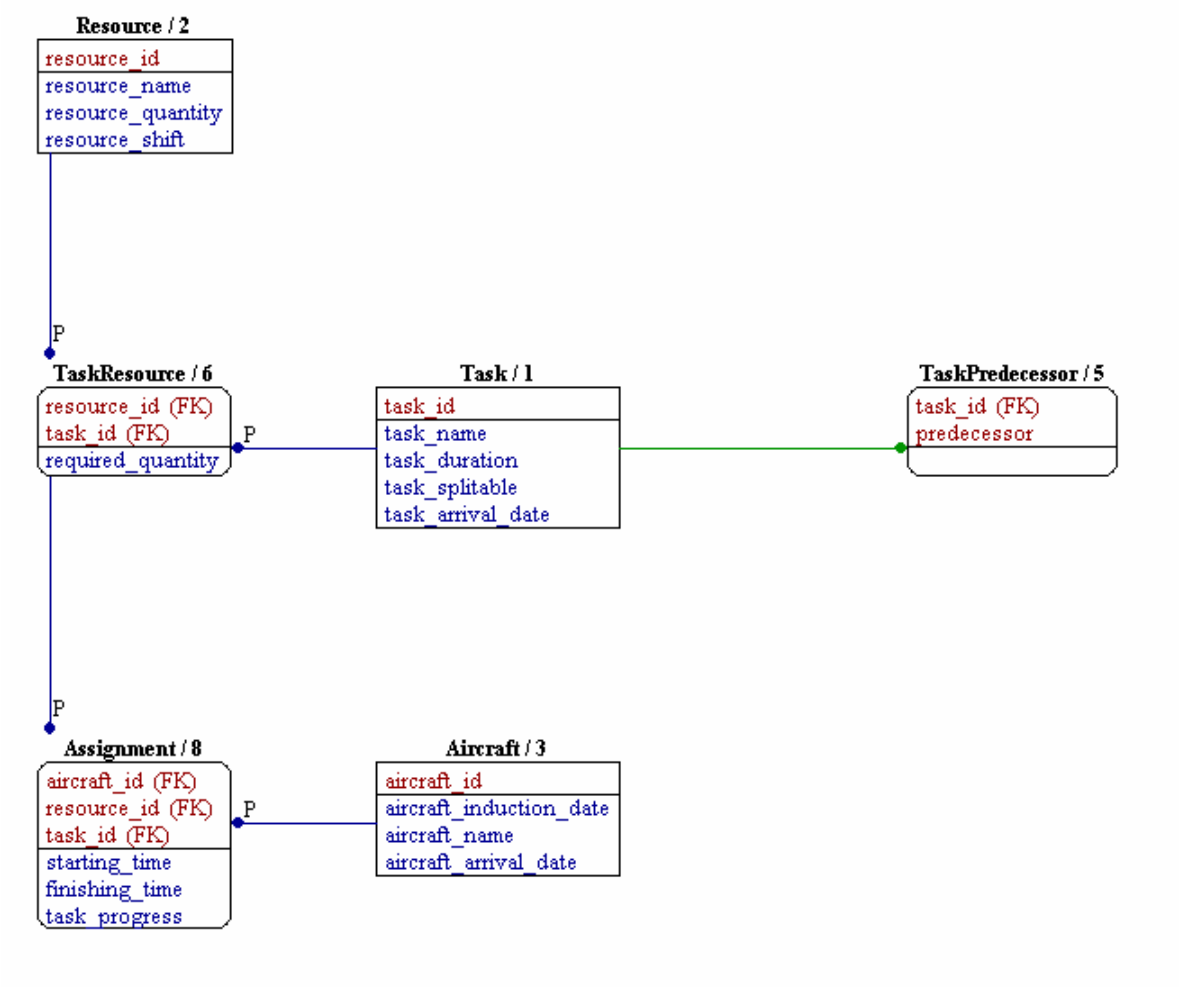

Figure 3. IDEFIX Data Model for Shop Status

The resource information includes available resources and their quantity. Note that the shop status database is periodically updated based on the task completion or progress. Hence, a task network constructed by the "Task Generator" at time $\mathbf{t + \Delta}$, $\mathbf{TN(t+\Delta)}$, is a sub-network of the task network constructed at time t, $\mathbf{TN(t)}$ since $\mathbf{TN(t+\Delta)}$ is generated from $\mathbf{TN(t)}$ by eliminating the completed tasks during the Δ period. By the same reasoning, all task networks are a sub-network of $\mathbf{TN(0)}$, which describes an entire task network before any reassemble work starts at a bay.

Schedule Generator

The "Schedule Generator" is a core engine of SAMSS. It consists of two main modules: a feasible schedule generator (FSG), and an optimized schedule generator (OSG). The FSG generates a feasible schedule within reasonable time using a specific dispatching rule, and the OSG generates an optimal schedule using a simulated annealing approach. In fact, the solution generated by FSG serves as an initial solution at OSG since OSG automatically calls the FSG. We define the following notations to explain the pseudo-algorithm in Figure 4.

SEQ: a set representing a feasible task sequence.
SEQ[i]: i^{th} task in a feasible sequence of tasks.
Pred(**SEQ**[i]): a set of immediate predecessors of SEQ[i].
S: a set of schedulable tasks.
C: a set of scheduled tasks.
T_i: a task in **S**.

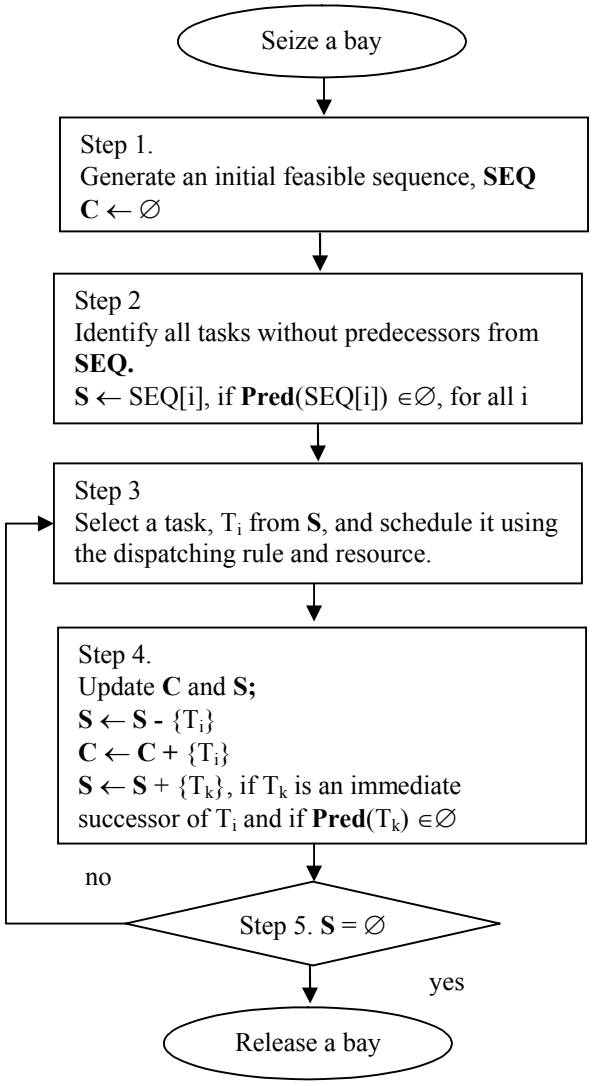

Figure 4. A Feasible Schedule Generating Algorithm

The FSG first generates a feasible sequence of tasks in terms of the precedence relations, and classifies all tasks into two sets: *Schedulable* and *Scheduled*. The scheduled task set is initially empty (Step 1). A task is considered as schedulable if either 1) there is no predecessor or 2) all of the immediate predecessors are scheduled. Initially, the only tasks without any predecessor are added to the *Schedulable* set, **S** (Step 2). Currently, the earliest possible starting time (EPST) and the largest number of successors (LNS) rules are available as a heuristic dispatching rule. The EPST computes the starting times of all schedulable tasks based on the required resource availability, and then it selects the combination of a task and a resource with the earliest starting time. In the LNS rule, a task with the largest number of successors is first selected, then, an available resource is assigned to it. When multiple available resources exist, the EPST is used to choose a resource (Step 3). Once a selected task, T_i, is assigned to a resource, the status of this task changes from the *Schedulable* (**S**) to the *Scheduled* (**C**). Note that this status change of T_i may change the status of its immediate successor, T_k, to the *Schedulable* (Step 4). The procedure is repeated until all tasks are scheduled (Step 5).

The OSG initially activates the FSG with the EPST rule to generate an initial solution - a task sequence - for the simulated annealing. It then randomly chooses two tasks within the precedence relation constraints, and exchanges their positions. Then the resource scheduling is performed in the order of the tasks in the task sequence. In other words, the solution quality is sensitive to the order of tasks in the task sequence. To generate a feasible sequence of tasks in terms of the precedence relations, we used the method described by Boctor [1]. For a task, SEQ[i], in any feasible sequence, the possible movements of the task are constrained between $L_i + 1$ and $H_i - 1$, where L_i is the maximum of the positions of the immediate predecessors of SEQ[i] and H_i is the minimum of the positions of the immediate successors in the sequence. According to our experiments, an optimized schedule using this approach showed 5% to 10 % performance improvement against a feasible schedule in terms of makespan - longest completion time of all helicopters, and flowtime - average time spent for all helicopters for the problem with five helicopters with three bays. With current working time - 8 hours a day, this corresponds to one to three calendar day saving.

THE IMPLEMENTATION OF SAMSS

In this section, we show the performance and features of SAMSS. We executed the SAMSS with an optimal scheduling generation (OSG) option with five helicopters for a demonstration purpose. The initial task progress information is read from the "Shop Status Database" and displayed in the Excel spreadsheet to indicate which tasks are partly or completely finished.

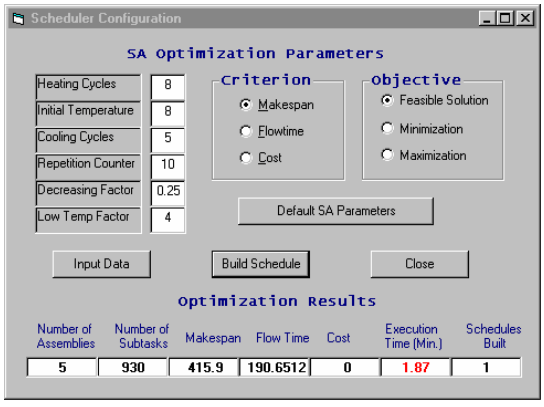

Figure 5. Parameters for Simulated Annealing

The SAMSS also generates the individual manpower schedule and workload for a crew. Users can choose the Task Generator's configuration options to define the simulated annealing parameters and objective function type as seen in Figure 5. The detailed schedule can be displayed in both the MS-Excel® and the MS-Project®. For example, the expected starting time and ending time of each task with consideration of resource are displayed in the Excel spreadsheet, and this information can be transformed into the MS-Project® using the MPX format feature in the MS-Project. This transformation feature allows users to use the diverse features of the MS-Project® including the Gantt chart and the task progress tracking etc. The Gantt chart of this case is displayed in Figure 6 as an example.

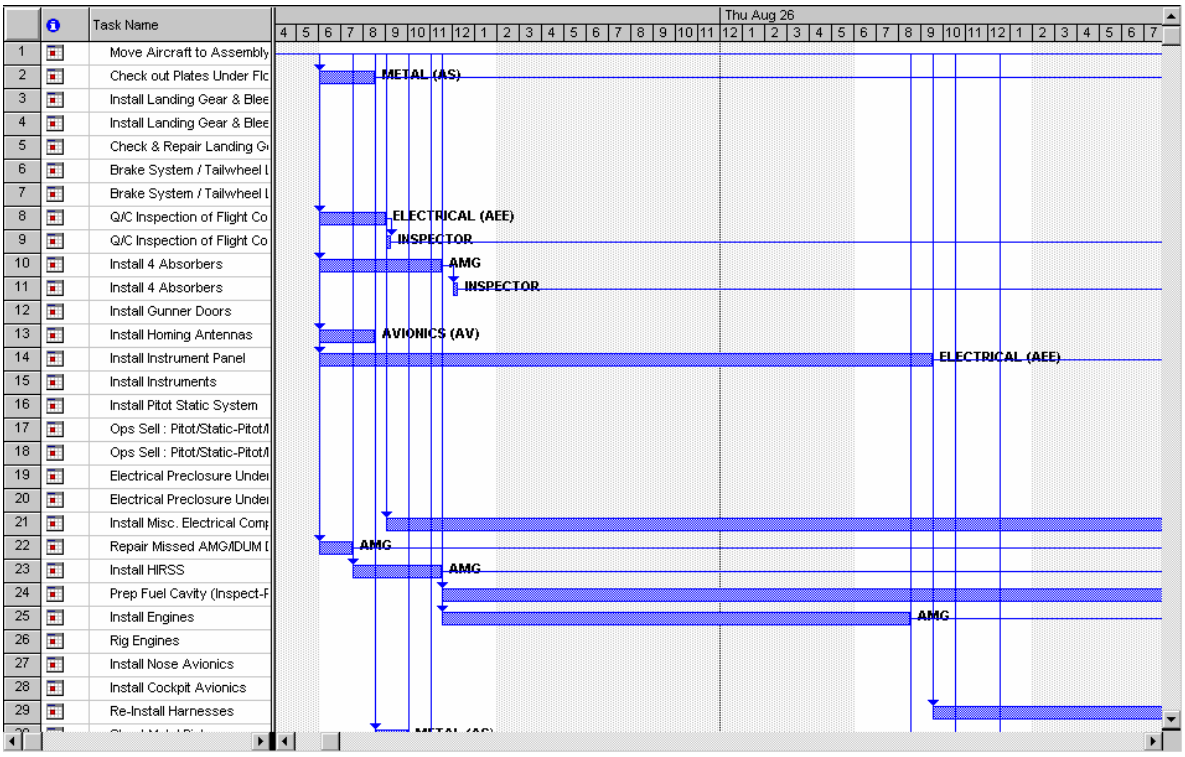

Figure 6. A Gantt Chart

The workload distribution per crew is plotted in Figure 7. The number of business days in terms of eight-hour day (x-axis), skill types (y-axis), and the number of busy workers (z-axis) are plotted in each axis. Through this visualization, the shop managers can predict who will be a bottleneck. Hence, they may attempt to resolve this bottleneck with workload redistribution. For example, we recognized that the mechanical engineers (AMG), sheet metal engineers (AS), and the inspector (INSP) were fully utilized for more than half of the reassembly period, while the AV was under utilized for all periods. This trend is confirmed in Table 1, which shows the effect of the crew size - number of workers in each team classified by skill type.

The first row shows the current crew size and the two objective function values - MSPAN for makespan and FLTIME for flowtime - using those crew size combinations. The other rows represent unit increment in the crew size and the corresponding objective function values. The columns labeled MS IMP and FL IMP show the percentage improvement of the makespan and the flowtime, respectively, compared to the current configurations in the first row. For example, if the crew size of AMG increases by one, the makespan is improved by 2.21 %, and the flowtime by 4.98 % as denoted in the second row – Note that AMG was fully utilized as seen in Figure 7. The addition of AV does not improve the objective functions

since the AV was underutilized as represented in Figure 7. That is, the workload distribution in Figure 7 provides useful insight on the bottleneck, and the resource addition problem.

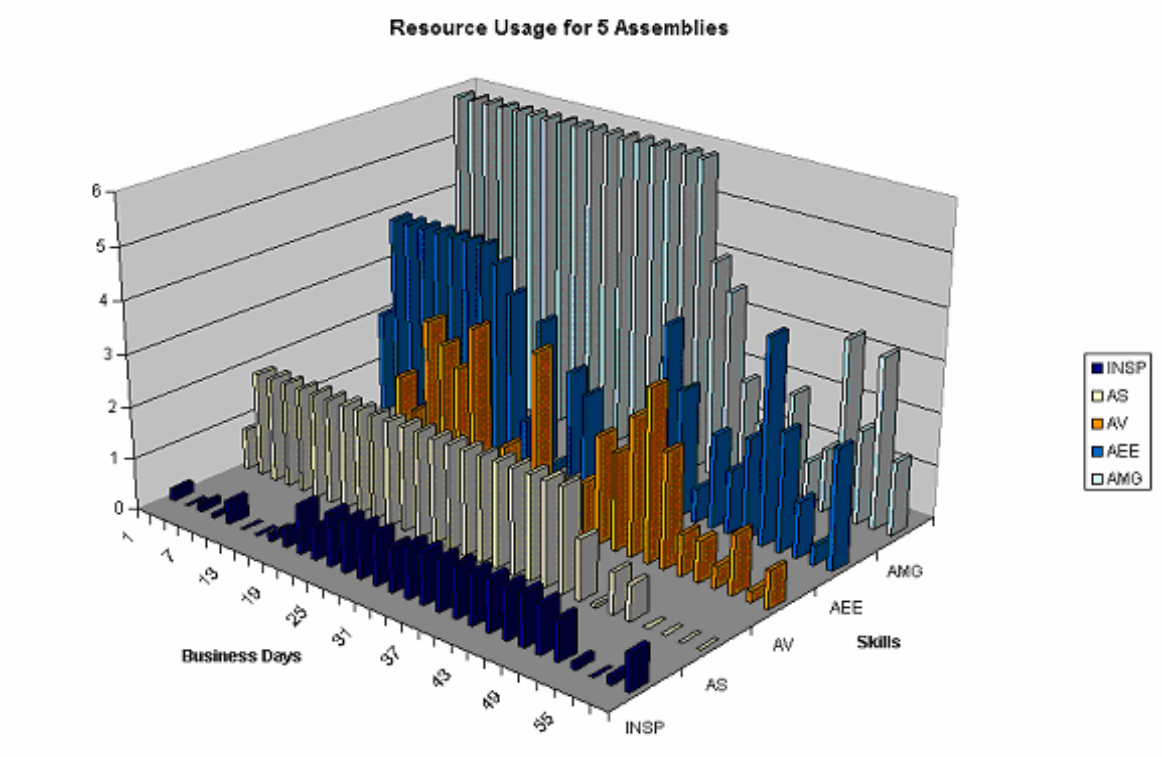

Figure 7. Workload Distribution per Crew

	AMG	AEE	AV	AS	INSP	MSPAN (hours)	MS IMP	FLTIME (hours)	FL IMP
Current	6	4	4	2	1	415.9	0 %	190.6	0 %
AMG INC	+1	0	0	0	0	406.7	2.21 %	181.1	4.98 %
AEE INC	0	+1	0	0	0	410.2	1.37 %	185.7	2.57 %
AV INC	0	0	+1	0	0	418.4	-0.60 %	190.6	0.0 %
AS INC	0	0	0	+1	0	428.7	-3.08 %	183.4	3.78 %
INSP INC	0	0	0	0	+1	392.9	5.53 %	182.5	4.25 %

Table 1. Performance Summary

More extensive tests were performed and their results were displayed in Figure 8 where the x-axis shows the crew size increments, and y axis shows the percentage improvement in the objective function values. The "current" in x-axis represents the current crew size combination. In many cases, both flowtime and makespan were improved when we increased the crew size. However, there was a threshold value in the increment. For example, the flowtime was improved up to three AMG additions (AMG FLTIME) while the makespan was improved up to two AMG additions (AMG MSPAN). Beyond those threshold values, none of the objective functions changed or it was slightly reduced. It was also interesting to see that the threshold values of both flowtimes were higher than those of makespans - note that INSP MSPAN had two additions until it reached the threshold value while INSP FLTIME steadily increased within the selected range of these tests. That is, when we increased either AMG or INSP, both flowtime and the makespan were improved until the number of addition reached the first threshold value (for makespan), and after that, only flowtime was improved until the addition reached the second threshold value.

It should be noted that the Figure 8 does not provide any optimal combination of crew sizes – note that we attempted to obtain an optimal schedule at a given combination of crew sizes, and the optimal number of the crew size should be independently solved by adding more complexity to this problem. However, this information can still provide many useful insights to the shop management. For example, they can decide who is the bottleneck and mostly required at any given time, and who is underutilized. They can also identify the optimal manpower schedule to minimize the flowtime and/or makespan. If appropriate data is provided, this can also provide a long term capacity planning function too.

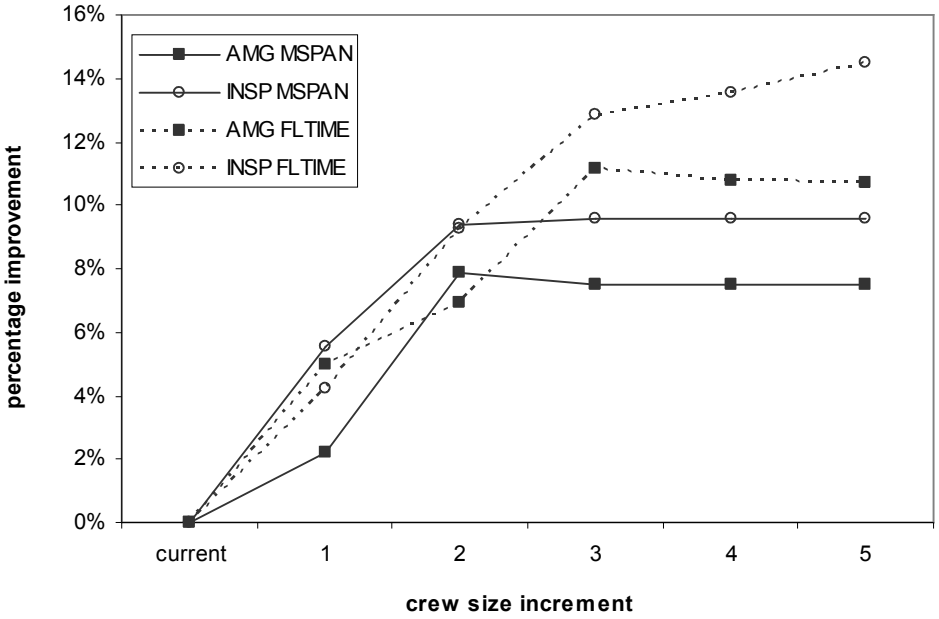

Figure 8. Effect of Crew Size

SUMMARY AND CONCLUSIONS

We have described the brief concept, architecture and application of the SAMSS for the short term manpower schedule at the labor-intensive helicopter reassembly shop in Southern US. SAMSS consists of "Shop Status Database", "Task Generator", and "Schedule Generator". The operational feasibility of the SAMSS was tested during implementation in the helicopter reassembly shop. The shop managers could get "who does what at which time" information to minimize flowtime and/or makespan, which can significantly improve the visibility and quality of managerial decisions. For example, before starting actual tasks, they can set reasonable target progress by executing the SAMSS, and adjust the actual status in the shop by filling out the forms in SAMSS. The managers could also compare the heuristic dispatching rules and see the effect of crew size on the objective functions. The convenient graphical reports are another advantage that helps users to identify the bottleneck resources without time - consuming simulation study. The use of MS-Excel® and MS-Project® interfaces increases the flexibility and user-friendliness of this decision support system. The access to SAMSS via a world-wide-web is under development, which can significantly increase the information sharing between shop crews and manager.

REFERENCES

[1] Boctor, F. Resource-constrained project scheduling by simulated annealing, *International Journal of Production Research*, 1996, 34(8), 2335-2351.

[2] Jeong, K.Y. Conceptual frame for development of optimized simulation-based scheduling system, *Expert Systems with Applications*, 2000, 18(4), 299-306.

[3] KBSI, ProSim™. *Automatic process modeling for Windows, user's manual and reference guide*, Knowledge Based Systems, College Station, TX, 1995.

[4] Mayer R.J. *IDEF3 process description capture method report*, Knowledge Based Systems, College Station, TX, 1994.

[5] Paul R.J. and Chanev, T.S. Simulation optimization using a genetic algorithm, *Simulation: practice and theory*, 1998, 6, 601-611.