

Job-Shop Scheduling Can Assist in Improving Manufacturing Budget Control

Timothy Deane Mather, PMP
Chief Technology Officer & Managing Principal, PMA Consultants LLC

Introduction or is this the Abstract?

Best route scheduling is an intractable problem, in both the intangible world of mathematics and the tangible world of job shop manufacturing. The contention between multiple jobs and limited resources quickly overwhelms even the most powerful computers' ability to solve the problem. This paper looks at the origins of the problem and suggests approaches to decrease overall production time for a group of jobs.

What is the Job Shop Scheduling Problem?

The real question being asked here is: what is the best way to do the work that needs to be done in order to produce a specified group of jobs in the shortest amount of time. The more formal definition reads more like: Given n jobs j_1, j_2, \dots, j_n , each unique in the path they must or could follow through the shop, but sharing resources within the shop, and minimizing the production time for the job list as a whole, what is the best ordering of jobs and steps within jobs.

There are really two separate groups working on job shop scheduling. One party, the mathematicians, see the problem from ivory towers. The other is management, trying to meet production schedules. Mathematicians have been working on this problem for half a century. They gather to argue with each other about a certain type of mathematical problem and whether or not it can be solved within polynomial time on a determinative device (a Turing machine). Can a computer solve the job shop scheduling problem when there are three or more machines involved in the computation? The jury is still out on the solvability of the scheduling problem within polynomial time on a determinative device. If it is resolved that the job shop scheduling problem cannot be solved in this way, then the problem would then be termed NP-complete. This term of art describes a class of decision problems within computational complexity theory. The defining feature of an NP-complete problem is that using any known algorithm, the time required to solve the problem increases disproportionately with the complexity and number of variables. Thus, when you have three or more workstations in the job shop scheduling problem the time to calculate the answer becomes absurdly long.

The other group attacking the job shop scheduling problem are people who work at job shops. They are interested in the practical application of computing power towards the goal of completing work more efficiently. In the author's experience, the creation of complex custom machines is a typical example of a job shop in need of assistance in the calculation of the best path for multiple custom machines being built simultaneously and contending for specific resources. If the job shop scheduling problem is truly an NP-Complete problem then finding "the solution" is impossible. Typically, applications focus on the use of known algorithms, such as the critical path method, resource leveling, and Monte Carlo simulations, in an attempt to bring coherence to a dynamic system with many dependent variables. Most of these tools fall under the category of "better than nothing."

Planning Visualization using Logical Diagram Method (LDM)

The use of LDM simplifies the task of visualizing alternate paths of individual jobs within the shop, and highlights resource contentions. An annotated LDM diagram spells out the specific elements of [what?] in figure 1.

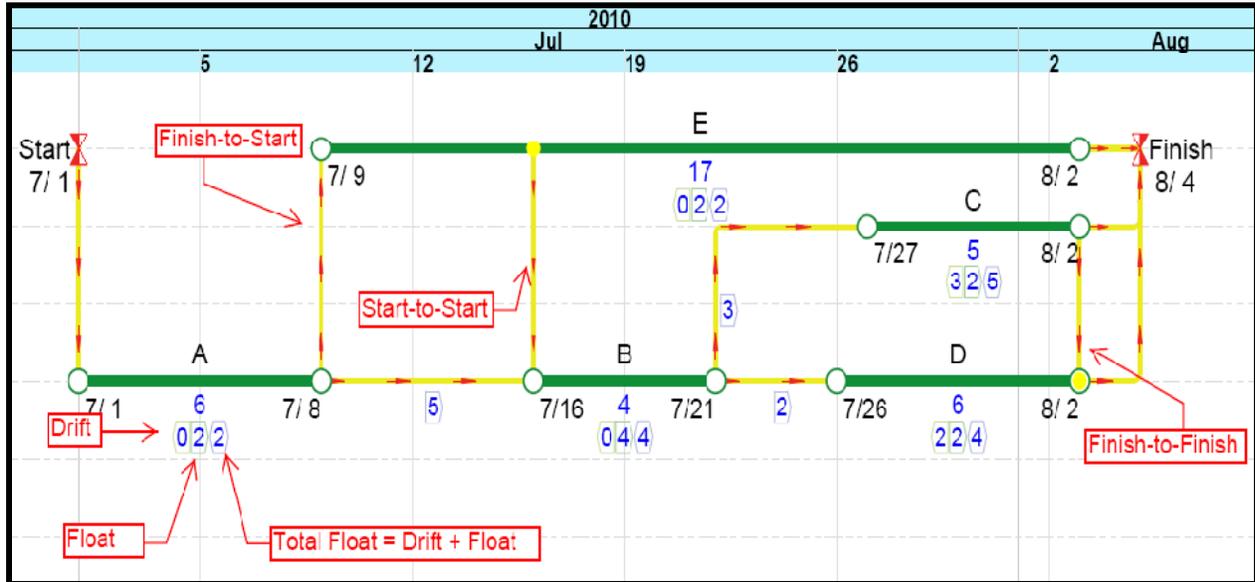


Figure 1. Annotated LDM Diagram

CPM v GPM in the Job Shop

Application of Critical Path Scheduling (CPM) scheduling to a real world resource-loaded manufacturing scenario yields suboptimal results. Figure 2 is a visual representation of a critical path schedule created to plan the production timing on two jobs (A and B) with resource constraints on software design and QC through a production facility. In this super-simplified example, the resource limit for each of our two constrained resources is set to 1 resource applicable on any given day for both constraints. As a result, neither software design nor QC can overlap during production of the two subject jobs. As you can see from the diagram, the unconstrained CPM schedule yields a substantial resource over allocation in software design and somewhat less in QC. We have produced a schedule which cannot be accomplished given our resource constraints. As you can also see from Figure 2, the CPM schedule yields an expected completion date of 2/29/2012 for Job A and 3/27/2012 for Job B. However, as noted above this is not an obtainable schedule due to resource constraints.

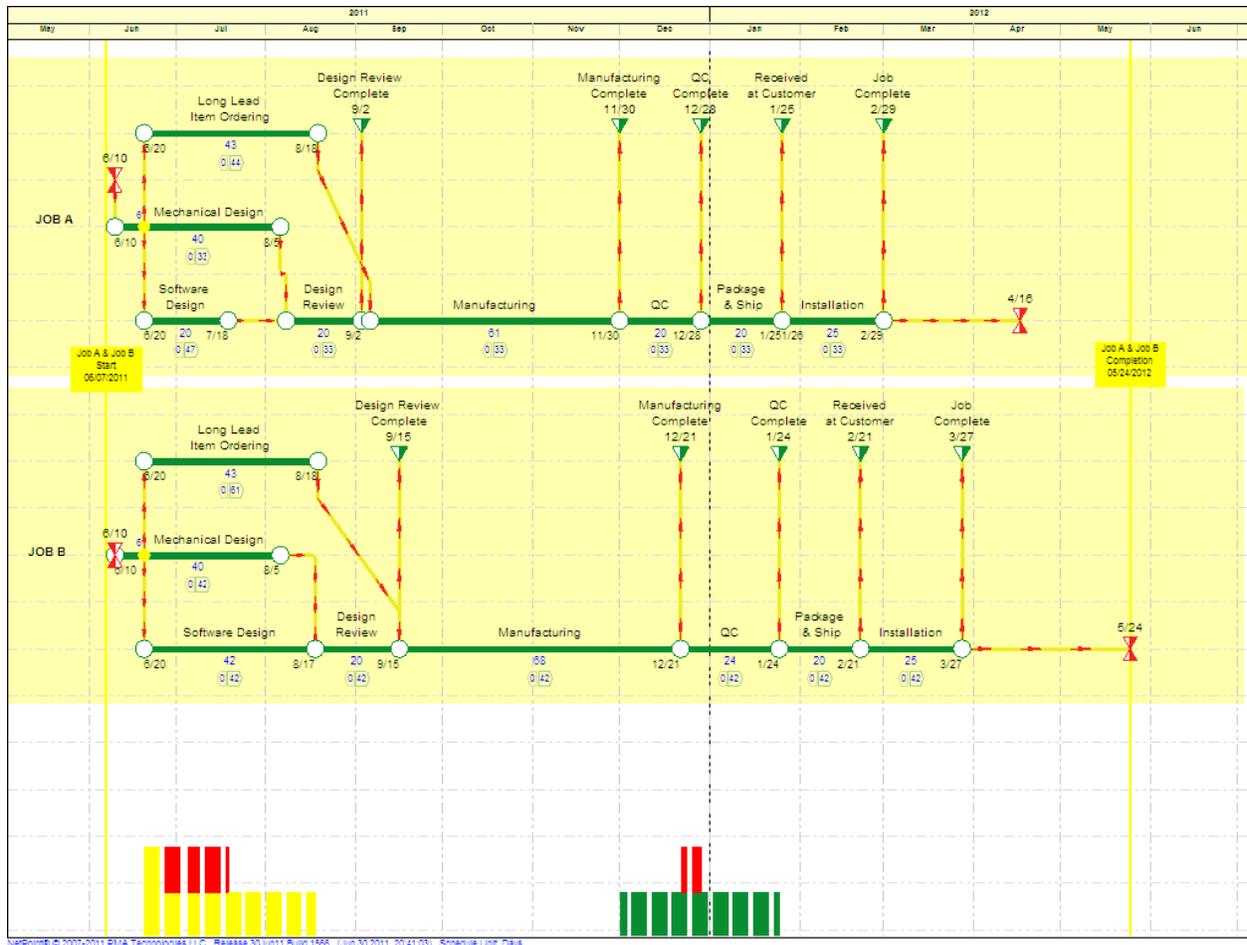


Figure 2. Annotated LDM Diagram with a Resource Key

As demonstrated in Figure 3, by setting start constraints on Job B Software design and QC (start no earlier) we can see that we now have a schedule we can accomplish. This “resource leveled” schedule reflects the same end date for Job A, but a new finish date for project B of 4/24/2012, nearly a month of added time to the job shop schedule.

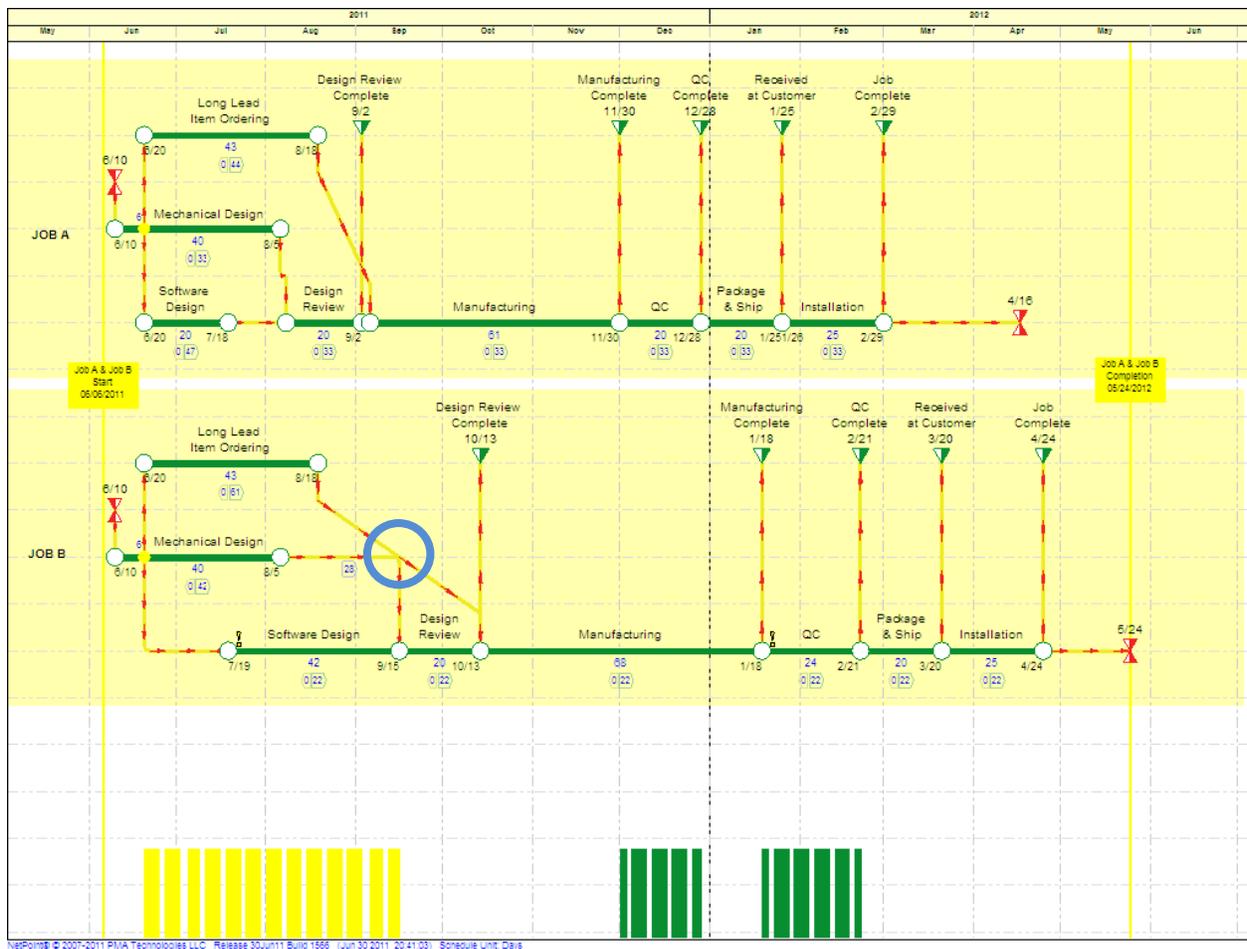


Figure 3

Release from the earliest date constraints of CPM and the ability to visually evaluate the schedule as a whole may lead to a non-polynomial solution, which is better than the solution generated by a CPM evaluation alone.

Graphical Path Method (GPM), an alternative to CPM, allows a graphical model of the planned production schedule. Non-polynomial evaluation is aided by real-time graphical feedback through the logic diagramming method (LDM). GPM uses planned dates selected by the planner, which can reside anywhere on or between the early and late dates of CPM. Planned dates are not calculated. Floats are calculated by working through the network path and adding the activity level floats and gaps between activities. This is not intuitive to experienced CPM schedulers because CPM would calculate early dates. The key difference between CPM and GPM is the availability of back float, or "drift" as GPM describes it. Drift is the number of days an activity can move to earlier dates without forcing an earlier start day for the project. Drift is a calculated activity attribute that measures the number of days an activity may backslide or extend to an earlier position without forcing an earlier project start or earlier interim release date. Drift plus float equals total float (Ponce de Leon, G., Jentzen, G., Fredlund, D., Field, D. & Spittler, P., 2010).

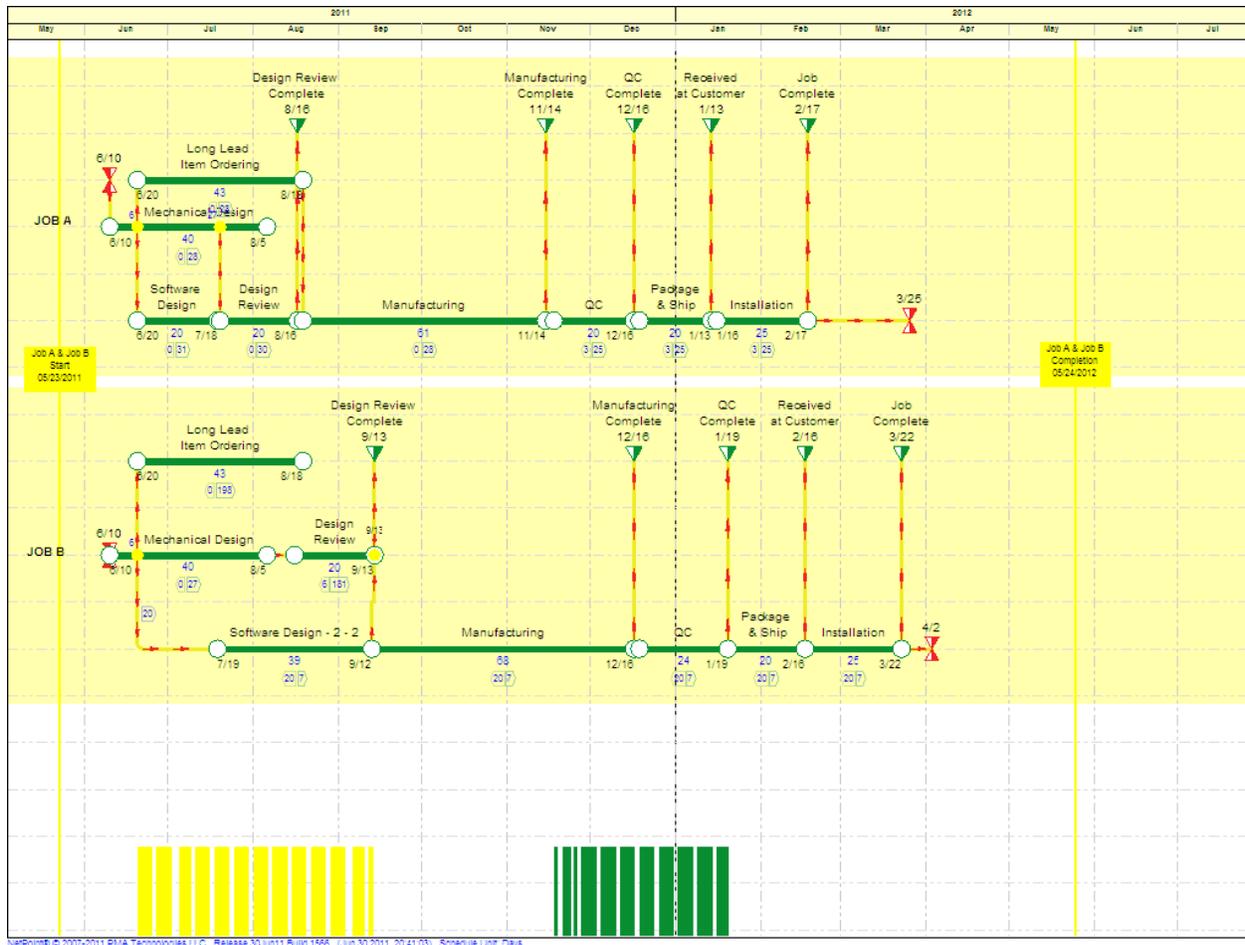


Figure 4.

Figure 4 is the result of a planner reviewing various possible scenarios for completing both Job A and Job B on the earliest possible date. By interactively receiving instant feedback from the plan as various scenarios are attempted, the planner can quickly and easily assess the impact of various logic and timing changes. In the final version shown above the planner has brought Design Review into a finish-to-finish relationship with Software Design with a one-day offset. In essence, when the software design is complete the design review will be nearly complete, and will take one additional day to complete the review of the final software design (check out offset). The days gained through this parallel task performance can then be used to offset software design in Job B from software design in Job A, resolving the resource conflict while reducing days required to complete both plans.

This example illustrates the importance of drift and the impact of accurate total float calculation. Planners need accurate total float information such as resources and cost to plan projects realistically. Some argue that experienced planners can spot such opportunities to shift activities despite the inaccurate total floats calculated by CPM algorithms. However, as the scope of activities increases, most planners cannot find such information readily. The illustrative examples above were created using NetPoint™, a GPM® based planning system, which accurately calculates total floats inclusive of drifts, and provides real-time interactive graphics for planners to experiment with what-if scenarios on resource leveling and optimization.

Recognizing the importance of back floats in project planning, GPM uses the concept of “drift,” as defined in the previous section, to compensate for the error of CPM’s total float calculation. In fact, the sum of drift and float is the real total float of an activity, drift is the amount that an activity can move to the left (towards the project beginning) and float is the amount that an activity can move to the right (towards the project end) without impacting the beginning and ending of the project. The calculated drift, float and total float (shown in the three boxes below each activity in Figure 2), provide planners with the accurate information needed to optimize resource utilization and leveling.

Manufacturing has often been at the forefront of adopting new technologies that can produce increased efficiencies and productivity. In the area of Man Machine Interface (MMI), Computer Numeric Control systems adopted touch screens early in their adoption cycle. In large production facilities, widespread adoption of Radio Frequency Identification (RFID) to track Work in Process (WIP) brought a level of simplicity and automation previously unknown to the tracking of actual progress on the manufacturing floor. Finally, the widespread adoption of automated storage and retrieval techniques for both production and warehousing at manufacturing locations enhanced productivity while lowering headcount. With the widespread adoption of tablet computing, it may be that a touch-based, interactive planning and scenario tool will offer advantages to those faced with the job shop scheduling problem.

By using NetPoint to develop a GPM plan the planner gains the additional benefit of real time network updates. The NetPoint system is developed as an event/object-driven tool. All current CPM software features an architecture that has three separate components: a database engine, a CPM engine, and a graphical display engine. Changes in the database are processed through the CPM engine and then rendered through the graphics engine to update the network graphic. NetPoint does not use a database; it invests control of the network in the actual graphical objects on the screen. Events, such as changes in logic or duration, cause each of the objects on the screen to react, in real time, to any changes which impact them, according to the rules of GPM. With GPM the planner is in complete control of a mathematically and algorithmically grounded graphical representation of their network model.

Critical Chain Scheduling in the Job Shop Environment

Critical Chain Scheduling was developed by Goldratt, and has gained acceptance in specific fields of endeavor, including job shops. Critical Chain emphasizes the role of resources within a project and embraces the idea that the project manager should control the float. This is combined with generating maximum float by asking subject matter experts (SMEs) to estimate their activity durations only to a 50% probability of on time completion, using bare-bones very short durations. The result is that there is more float than usual in the schedule and the project manager controls all of it.

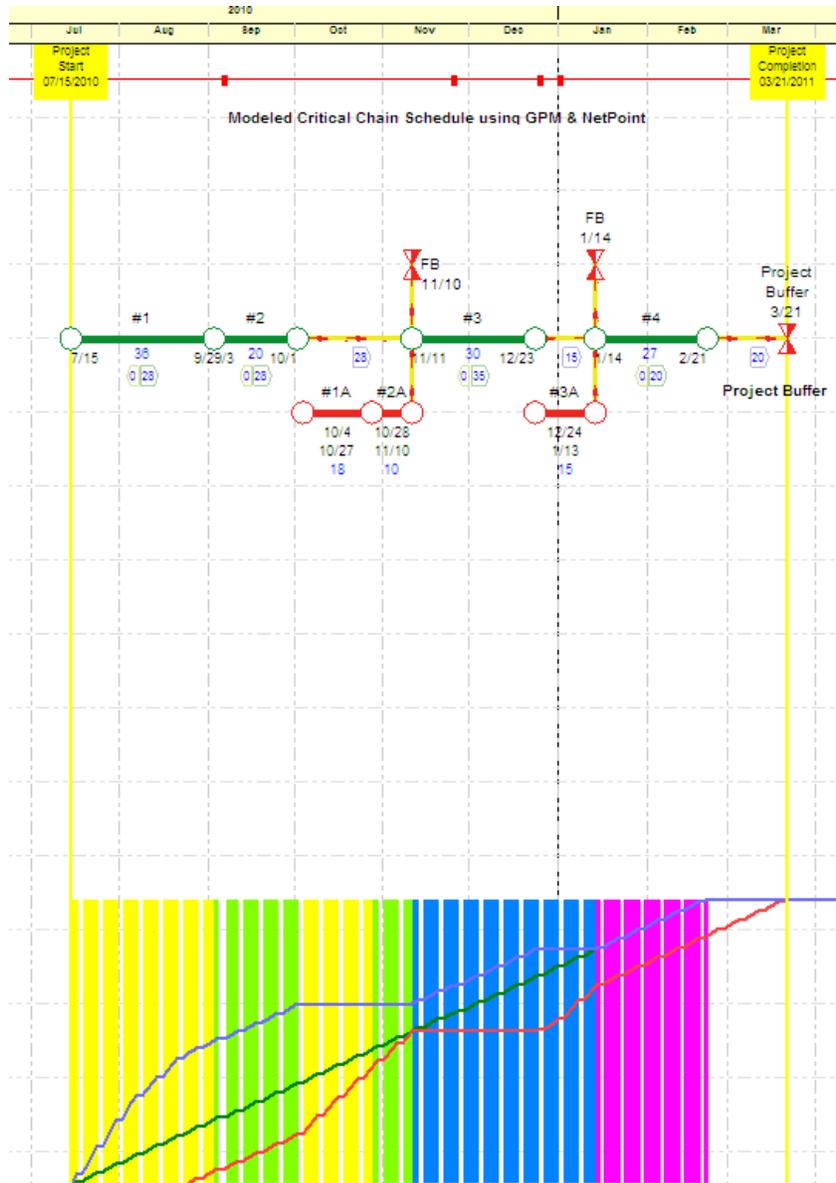


Figure 1.

Future Trends and Convergence

Gartner predicted the death of the mouse as an input device even before the wild adoption of iPad's in the workplace. It seems inevitable that we are headed for a predominantly surface computing paradigm in the coming few years. Mobile, interactive, visual, systems will play an important role in the future of job shop scheduling.

References

Ponce de Leon, G., Jentzen, G., Fredlund, D., Field, D. & Spittler, P. (2010). *Guide to the forensic scheduling body of knowledge Part I*. Ann Arbor: PMA Consultants LLC.