Using Graphics to Display Messages in an Intelligent Decision Support System

Peter. G. Higgins
School of Mechanical and Manufacturing Engineering, Swinburne University of Technology,
Hawthorn 3122, Australia, phiggins@swin.edu.au

Abstract. The paper addresses the form of the graphical interface for an IDSS used for scheduling small-batch production. The IDSS has been developed to aid inferential processing by bringing “visibility” to those attributes experienced schedulers use to schedule jobs on the shop floor. From the patterns in the values of the attributes, schedulers can see interactions and dependencies between jobs.

Introduction

Higgins (1992) has developed an intelligent decision support system (IDSS) for hybrid human-computer decision-making in production scheduling for a job shop. It combines a knowledge base with heuristics (Hejamadi, 1986). The scheduling objective is complex (Higgins, 1995; Higgins and Wirth, 1995). The decision-making process for producing a suitable schedule is in itself ill-defined. Being only partially explicit it therefore fits the semi-structured requirements of DSS (Keen and Scott Morton, 1978; Speranza and Woerlee, 1991).

Theoretical discussion

As scheduling per se is not the concern of this workshop, I will limit my comments to a few basic aspects of scheduling problems and their formulation to show that IDS (Intelligent Decision Support) has a very significant role. The traditional approach in theoretical scheduling has a simple link between an Operations-Research (OR) based system and the human decision maker (Speranza and Woerlee, 1991). The decision maker provides the data for the OR model. After processing, the system presents the solution back to the decision maker. This approach:
1. ignores the difference between a plant and its model
2. disregards the role of the human decision maker in solving real problems
3. does overlooks the issue of decision-making responsibility

There is a vast gap between the scientific conception of OR and the non-scientific bases of real decision making. The traditional OR approach reduces scheduling problems to greatly simplified forms, to allow the scientific appraisal of nominated heuristics. There are two aspects of this reduction: the description of the problem and the performance objective. When describing jobs, only a few attributes are considered important and all others are ignored. Often due dates and expected processing times are the only characteristics used. The most difficult problem that heuristic methods tackle is the inclusion of time for one major and one minor set-up between jobs. The job description has to include the job attributes that trigger
these set-ups. This method also simplifies the production shop. A shop with many machines is commonly reduced to a single bottleneck machine (Figure 1). All machines except the one perceived to be problematic are ignored. Parallel machines that may technically differ are considered equal, with at most the processing speed being the sole permissible variant. Single uncomplicated mathematical relations are used to measure performance. For example, the meeting of due dates is a substitute for customer satisfaction.

The OR approach is so abstracted from reality that it does not address many needs and concerns of practising human schedulers. Schedulers have to satisfy many stated and unstated conflicting goals, using hard and soft information that is possibly incomplete, ambiguous, biased, outdated, and erroneous. Using intuition, they fill in the blanks about what is happening, and, what can and what will happen, on the floor. This includes sensory data and a mental model of the situation. What they need to ponder upon varies: ‘When scheduling, some issues will be paramount while others will be ignored. These subsets will change with time, date, mood, climate, and so forth. Any issue can at sometime affect the scheduling decision’ (McKay, Safayeni and Buzacott, 1988).

Thus scheduling is not a closed problem with clearly stated boundaries and strict rule for manoeuvring within these boundaries (Bright and Johnston, 1991). There is not a finite, specifiable set of options. The scheduling objective generally is neither singular nor clear. By applying a limited set of heuristics and rules an open problem is treated as a closed one. Such an artificial constraint on the solution space leads to inappropriate solutions.

Incorporating a knowledge base, with domain-specific rules, into the model helps to address weaknesses in the standard OR approach. Even if the rule base is large enough, they are prone to brittleness. What happens when the environment changes? Machines may change. Products may change. Unforeseen methods or materials may be introduced. These are but some possible changes. What then happens to the rule base? Is it upgraded? Are the changes ignored and its advice accepted? Is the schedule used only after whosoever holds responsibility has vigilantly made the necessary modifications? Factors that may result in quite divergent scheduling outcomes have often to be reconciled when a schedule is developed. How can this be done?

Many aspects of scheduling jobs within a plant are understood best by humans working in the particular environment. Within the context of local knowledge, they know how to handle information that may be diverse, inexact, or conflicting. Instead of focussing on mathematical techniques, the problem’s locus should move to the needs of persons who have to take responsibility for the planning of production.
For the planning and control of power systems, Venikov, Jouravlev and Soukhanov (1971) found that simpler algorithms could be used when a human decision maker is in the decision loop. Even the less probable events could then be taken into account. Haider (1976) used this approach in interactive scheduling of a four-machine shop. He found that his interactive scheduler, using a simple performance measure (average tardiness), was more effective than a batch simulation, using a sophisticated performance objective.

For Sanderson (1989) the applicability of a DSS depends on the type of problem. A human would expect to perform better than simple priority rules where rules must change to meet new demands or adapt to certain plant configurations. The human’s ability would be further enhanced when ‘equipped with a well-engineered graphical display with good predictive and bookkeeping capabilities’. If the best decisions depend on keeping many factors in memory simultaneously and choosing between many fairly acceptable options under time pressure, then the computer should fare better. This is particularly so, if the human has to use a basic alphanumeric interface. The unique feature of human performance, which may be most useful, is the ability to solve ill-defined problems on an intermittent basis, instead of solving well-defined problems regularly. This requires a sound mental model of system properties and constraints, and a clear mental picture of the current system state.

In summary, a decision support approach is pertinent to scheduling (Speranza and Woerlee, 1991) as:

- No model can capture all the characteristics of a problem;
- The decision maker always has the best understanding of the problem;
- The knowledge the user has about the problem can be very important for the recognition of solutions (This is especially true when using an approximation procedure, because the complexity of the model causes computational problems);
- The use of a model is only one step in the decision making process;
- The decision situation and conditions are dynamic;
- The user can control the whole decision process, whereas “black box” systems require the user to “trust” the system without understanding it;
- Models are often difficult to use (The data preparation and interpretation of results can be complicated).

A case study may be help to explain how what seems to be on the surface a highly structured activity is in practice ill-structured.

**Scheduling Case Study**

The job shop has single-stage parallel processors that have similar, but not identical, features. Because of the difference in features, the number of machines on which a job can be processed varies. For most jobs, this depends on the value of one job attribute. In planning beyond the immediate shift, the scheduler has some indication of expected arrivals, which is not at all comprehensive. New jobs arrive intermittently, machines break down and management often demands specific jobs to be expedited. In planning work for a shift, the scheduler includes anticipated arrivals. The actual times at which jobs arrive may differ from those predicted.
Other arrivals are unanticipated. Setting up a processor for an operation is sequence dependent. A major delay occurs where the value of a particular attribute for a job differs from its predecessor. Other attributes trigger minor set ups. One attribute only triggers a minor set up if its value increases from one job to the next. There is a quick turnaround service for premium jobs. On the arrival of a “hot” job the scheduler has to decide whether to pre-empt the current job. These, and other, constraints make the scheduling process beyond the capabilities of heuristics developed for single-stage parallel processors.

The scheduling objective (Higgins, 1995; Higgins and Wirth, 1995) is only partially explicit. It encompasses qualitative factors tacitly understood by the scheduler but not directly expressed. It includes customer satisfaction. A quantitative component of this goal is meeting a customer’s due date. Operationally this becomes the minimisation of the average tardiness (although the scheduler would not state it in this way).

When the shop is heavily loaded, the scheduler aims to maximise utilisation to meet the tardiness objective. Endeavouring to do so, the scheduler strives to minimise the number of set ups and idle time. To minimise major set ups on a machine, the scheduler seeks to run as a group all jobs that have the same value of the triggering attribute. This group forms an uninterruptible string among other jobs in a queue of waiting jobs. The scheduler arranges the order of jobs in the string to meet some objective criteria. For example, to minimise tardiness a scheduler may order jobs from shortest to longest processing time (SPT). This may violate the constraint imposed by the minor set-up criterion. The scheduler has to decide whether to accept this extra set-up cost. The scheduler may allocate the group to the most versatile machine. If another machine has the highest availability, the scheduler may decide to split the group into those jobs that fit on the machine and those that require a different machine. Spreading the work may increase set-up costs.

These are but a few situations and strategies that can arise, as the possibilities are myriad. The purpose of this brief précis is to provide some insight into the types of activities involved in the scheduling process. It is obvious that it is not possible to schedule this job shop just by using a simple priority rule, such as SPT. Hard (e.g., machine capability) and soft constraints (e.g., grouping jobs requiring the same set-up) dominate the scheduling process. While the hard constraints cannot be violated, a scheduler, who is prepared to pay the penalty, can transgress the soft constraints.

**IDSS Architecture**

Using various classifications, schedulers group jobs. As jobs may fit more than a single classification, the scheduler has to decide between competing choices. On observing the attributes of particular jobs, and, patterns among all the jobs available for processing, a human scheduler decides for each job the most appropriate class. An architecture of an IDSS that facilitates this form of scheduling has pattern-matching at its core (see Figure 2). To find the basis for grouping, the human scheduler searches for patterns among jobs. Central to the interactive decision making is the jobs screen, which displays the attributes of the available jobs. It is made up of several windows: one for each machine and one for unassigned jobs. Using this screen, human schedulers choose classificatory strategies, the formation of groups, and the placement of groups into machine queues. In an opportunistic way, they try various groupings, making amendments and maybe backtracking on previous decisions. In ordering the jobs within a string, they can apply various policies from a suite of operations research...
(OR) scheduling heuristics. A knowledge-based adviser, warns the scheduler when soft constraints are infringed, and disallows violation of hard constraints. The result of this interactive process is a Gantt chart.

**Human-Computer Interaction**

The primary function of the IDSS is to help human schedulers seek patterns on which to classify jobs and suitably group them on machines. What is the appropriate form for the IDSS to so communicate with users?

The messages that the IDSS sends to the human scheduler have to bring “visibility” to the classificatory attributes. From the patterns in the values of the attributes, schedulers classify and group jobs. Observing interactions and dependencies between jobs they explore and test different arrangements. Higgins (1994) submits that the use of graphical signs to transmit messages supports this inferential process. Users can quickly search for particular graphical signs in a field of graphical signs (Arend, Muthig and Wandmacher, 1987; Treisman et al., 1985). With graphical signs the message conveys more than the value of attributes. During the search process, emergent features arise through the visual interaction of signs. What particular attributes, and the patterns in attributes, signify, and the responses these engender depend upon experiential knowledge. In deciding how to regard attributes, schedulers invoke mental images of them in a field of other images (Higgins, 1995).

A user needs to readily distinguish between an attribute and its value. Therefore, these graphical signs need to have “two-dimensional” features. “Global” features allow identification of all signs denoting the same attribute on a screen cluttered with multifarious signs. Within a sign for an attribute, there must also be “local” features that allow users to identify its value. It is important that these signs are quite distinguishable at both global and local levels. In devising a schedule, the user scans the screen to find jobs that share an attribute value. Figure 3 shows one realisation of such signs. It is important that global and local features are both distinct and distinctive, and that their form befits pattern recognition. The sign for the colour attribute, which uses a substitutive scale, has dashed lines within a rectangle serving as global features. For local features, dark black lines, denote specific colours. A different form supports the scanning of an additive scale (see processing time). A comprehensive discussion of design issues is covered elsewhere (Higgins, 1995).
Human decision makers should be able to attend to constraints at times appropriate to them. The proposition being put is that the IDSS should not needlessly get in the way of decision-making. Users should be able to explore potential decisions in their own way, guided by their mental model of the scheduling process. Normally expert systems advise users through standard message boxes popping up onto the screen. Users have to immediately attend to the message. This can be quite disruptive. Graphical signs enable the IDSS to pass messages unobtrusively. In Figure 3 the black bars in the presses sign denote permissible choices. These constraints being always visible can be considered at any time. Of course, if a user attempts to violate hard constraints the activities of the IDSS will not remain in the background. It then sends an intrusive warning through a pop-up message box.

For soft constraints graphical signs have particularly efficacy. The IDSS passes a message to draw attention to a violation. Consider at the bottom of Figure 3 the bar and box next to the values of width. This denotes a soft constraint that transpires when a job has a lower value of width than its immediate predecessor and the predecessor uses the same cylinder. The message draws attention while not being disruptive.

While Figure 3 helps the human to see patterns among attributes it lacks a key component. In comparing different orderings of jobs, the scheduler has to consider the effectiveness of each. There are both quantitative and qualitative contributors to performance. These have explicit and tacit dimensions. Part of the composite measure will be some overall quantitative indicator, for example, average tardiness. The scheduler needs to know how to move jobs to improve this measure. For each job in Figure 4 there is graphic on the right. For tardy jobs part of this graphic is a dark rectangle. Its size shows its contribution to the performance measure. Note that job 5 is tardy. If the mouse pointer is placed on this job a bar appears between jobs 1 and 4. This bar shows where to move the job so that it would be produced on time. There may be various arrangements that would give the same value of overall tardiness. By observing the distribution of tardiness a scheduler can adjudge qualitatively other aspects of performance. From experience the scheduler may either prefer a
balanced distribution of tardiness or may prefer concentration in only a few jobs. Factors that may be considered are the robustness of the schedule, whether it is preferable to be a little late across a broad group of customers or be very late with only a few customers.

Conclusion

A decision support system that presents information to human decision makers through graphical signs brings “visibility” to patterns in the data, from which inferences can be drawn. By passing messages in a graphical form, users may attend the advice at the most appropriate time in their decision-making process.

References


Venikov, V. A., Jouravlev, V. G., and Soukhanov, O. A., 1971,. On the methods of direct man-computer intercommunication in planning and operation control of power systems. In T. J. Williams (Ed.) Interfaces with the process control computer, the operator, engineer and management. Purdue University.