Cooperative Multi-Objective Decision-Support for the Paper Industry

Sesh Murthy[1], Rama Akkiraju[1], Richard Goodwin[1], Pinar Keskinocak[1], John Rachlin[1], Frederick Wu[1],
Santhosh Kumaran[2], James Yeh[1], Robert Fuhrer[1], Alok Aggarwal[1],
Martin Sturzenbecker[3], Ranga Jayaraman[4], Bob Daigle[5]

In this paper, we discuss a new decision-support system for scheduling paper manufacturing and distribution. Scheduling production and distribution of paper is an extremely complex task requiring the consideration of numerous constraints and objectives. Problem complexity is compounded by process interactions wherein the scheduling of one stage of the production process may negatively impact downstream processes. In contrast to earlier approaches, our system considers multiple scheduling objectives and multiple stages of paper manufacturing and distribution simultaneously in a global multi-criteria optimization framework. It generates multiple scheduling alternatives by using several algorithms based on approaches such as linear/integer programming, network flow methods and heuristics. These alternatives reveal tradeoffs with respect to competing objectives, such as maximizing profitability, on-time delivery, product quality, and minimizing manufacturing disruptions. Each scheduling alternative is a complete enterprise schedule that includes sub-schedules for each stage of the manufacturing process and loading. The scheduler can work cooperatively with the system to explore alternatives and to improve solution quality. As the final decision maker, the scheduler can choose the best alternative for the enterprise. By functioning as an intelligent assistant, our system relieves the schedulers of mundane computational tasks and allows them to focus on the objectives of the enterprise and decision making. For generating multiple solutions and facilitating cooperation with the scheduler, our system is implemented using the agent based Asynchronous Team (A-Team) architecture in which multiple solution methods cooperate by evolving a shared population of solutions**.** The successful deployment of our system at several paper mills in North America has resulted in significant savings and improved customer satisfaction. These positive results arose from improved schedule quality and improvements in the business process that our decision-support approach has fostered.

Key words: Paper manufacturing; scheduling; multi-criteria optimization; decision support; A-Teams; agent-based scheduling; cooperative scheduling.

[1] IBM T.J. Watson Research Center, Route 134, Yorktown Heights, NY 10598.
[2] IBM Supply Chain Optimization Development, 3200 Windy Hill Road, Atlanta, GA 30339-5610.
[3] IBM Internet Division, 30 Saw Mill River Road (Route 9A), Hawthorne, NY 10532.
[4] IBM Logistics and SAP Development, 5600 Cottle Road, San Jose, CA 95193.
[5] IBM, 3 Canal Plaza, Portland, ME 04112.

1 Introduction

This paper describes the successful application of a new multi-objective decision-support approach to the problem of scheduling production and distribution for paper manufacturing enterprises. The scheduling system based on this approach produces global schedules that prescribe the operations for all the major steps in the manufacture and distribution of paper. It considers the objectives of the key constituencies in the enterprise, such as customer service, manufacturing, and finance. The system serves as an intelligent assistant (Reddy 1996), presenting multiple good alternative schedules to the decision maker. The scheduler cooperates with the decision support system by injecting expertise and refining the alternatives to create improved solutions, and finally selects the best one.

The manufacture of paper products is a major worldwide industry. The total sales volume of paper and allied products in the United States alone was over $180 billion in 1996 (Stanley 1996). Paper manufacturing is an extremely capital-intensive business, with equipment and facilities for a new production line costing on the order of half a billion US dollars. To stay competitive, paper companies need to utilize their capacity efficiently, be highly responsive to customer demand, ensure on-time delivery and provide short lead times. This requires scheduling production so as to minimize production and distribution costs, inventory levels and manufacturing disruptions (Shaw 1998). The relative importance of these competing objectives varies with the state of the production environment and market conditions. The complexity of the scheduling problem is compounded by process interactions wherein the scheduling of each stage of the production process affects downstream production or shipping.

The traditional approach to paper mill scheduling is to schedule each stage in the process independently. Typically, paper manufacturers allocate orders to paper machines and sequence them manually. Then they use one software package for trim scheduling and use another one for outbound logistics scheduling. Each of these packages focuses on a single process step and attempts to create an optimized schedule based on local objectives. Since there is no interaction between applications, the complete schedule obtained by combining the sub-schedules is usually of very low quality. For example, a trim schedule that minimizes trim loss may cause vehicles to be loaded inefficiently, unacceptably increasing shipping costs. To improve the loading schedule, the scheduler must make extensive changes in the trim schedule. This process is time- and labor-intensive since posing "what-if" scenarios requires moving between applications. To make the process more manageable, companies adopt business rules, such as only taking orders in full vehicle loads and requiring trim schedules to produce exactly the amount ordered. This simplifies scheduling but reduces the ability to respond to customer requests, lowering customer satisfaction and can reduce efficiency. Additionally, most scheduling packages combine multiple objectives into a single objective function and produce a single schedule that minimizes this composite objective function (Pickard 1997). Since the importance of the underlying objectives may not be precisely known, this approach rarely generates a satisfactory solution (Goodwin et al. 1998; Steuer 1989). Furthermore, presenting schedulers with a single take-it-or-leave-it choice does not illustrate the tradeoffs between competing objectives needed to make an informed decision. To explore the solution space, schedulers repeatedly modify the objective function and re-run the scheduling application to see other possible schedules (Yu 1985).

In constrast, our new scheduling system considers all stages of paper production and distribution simultaneously and generates multiple enterprise-wide schedues. Each enterprise schedule contains an allocation of orders to machines, a sequencing of orders on the machine, a trim schedule for each machine and a loading schedule. The schedules are created by algorithms that take into account the interactions between the process stages and focus on enterprise-wide objectives. The algorithms that we have developed use approaches such as linear programming, integer programming with and without randomized rounding, network flow and heuristic methods. By combining multiple approaches and considering interactions between processes, our system significantly improves solution quality compared to earlier approaches. Each generated schedule is evaluated in terms of multiple objectives and the best schedules are presented to the scheduler. By examining these schedules and comparing the alternatives, schedulers gain an understanding of the tradeoffs that must be made and select a solution that strikes a suitable balance among the sometimes conflicting objectives. Our system also allows schedulers to work cooperatively with the software to improve schedule quality and pose "what-if" questions. In addition, our system treats business rules as objectives rather than constraints. Some of the schedules presented by our system may violate the rules in order to substantially improve customer satisfaction or manufacturing efficiency. These schedules can be shared with schedulers responsible for different stages of production and with customer service representatives and serve as a basis for negotiating an exception to a business rule.

Our scheduling software has been deployed at several paper mills and is saving paper manufacturers millions of dollars per year (Hoffman 1996). In addition to improving profitability through reductions in trim waste and shipping costs, it has improved customer service and manufacturing operations (Shaw 1998).

The rest of this paper describes our decision support system and the decision support approach that it embodies in more detail. We begin with a brief description of the paper industry and the production and distribution scheduling problems in paper manufacturing (Section 2). We then describe our approach to decision support (Section 3). In developing our system, we used an innovative software architecture, called Asynchronous Teams (A-Teams) that allowed us to combine multiple algorithms to generate a non-dominated set of solutions that illustrate tradeoffs between competing objectives (Talukdar et al. 1993). We describe the essential components of the architecture and how the architecture supports our approach to decision support (Section 3.1). We then give an overview of the multiple solution approaches that we use for generating schedules (Section 4). Following this, we briefly explain how cooperation is achieved between the scheduler and the system (Section 5). Finally, we discuss some of the improvements in business process that our system has fostered and show how our approach to decision support has contributed (Section 6).

2 Paper Manufacturing

The paper industry manufactures several different kinds of products to satisfy the diverse needs of printing and packaging. Production of paper goes through several stages, which vary depending on the kind of paper being produced, but the overall process is generally the same. In this section, we describe a generic paper manufacturing process and use it to introduce the operations that need to be scheduled in a paper mill. Further details on paper production can be found in (Biermann 1993).

Figure 1 presents an overview of the paper manufacturing process. The first step in paper manufacturing is the production of pulp from logs, wood chips, recycled paper and other sources of fiber. The pulp is fed into a *paper machine* along with the other ingredients that define the "recipe" for producing a particular *grade* and *basis weight* of paper, i.e. a *product*. The grade of paper is determined by physical and optical characteristics, such as smoothness, oil absorbency, gloss and shade. The basis weight is the weight of a *ream*[6] of paper. A paper machine produces large *reels* of paper. The width of the reel, called

[6] A ream is a specified area of paper or paperboard; its size depends on the type of paper. For example, a ream of newsprint is 500 sheets of 24” by 36” (or 3000 sq. ft.).

the *deckle*, is fixed for each machine. Another machine called a *winder* unwinds the reel while slicing it into narrower strips that it then rewinds to form *rolls*. The process of cutting a reel to make rolls is called *trimming* and the portion of the deckle that is not consumed by the rolls is the *trim loss*. Typically, several sets of rolls are made from each reel. The widths and diameters of these rolls must match the customer requirements. As each set of rolls is produced, the rolls are wrapped for shipping or temporary storage. In the case of cut sheet paper products, the rolls are loaded onto a *sheeter*, which unwinds the roll and slices the paper into sheets of the desired size. The sheets are then wrapped and packaged for shipping. (We do not consider cut-sheet production in this paper.) Finally, the end products are loaded onto trucks and rail cars for shipment to customers, warehouses and ports.

Most paper manufacturers produce to order because the large number of combinations of product type and roll size makes it impractical to stock inventory. Each customer order specifies a quantity (in tons or number of rolls), a product type, roll dimensions (width and diameter), due date and shipping destination. While customers prefer to have their orders filled exactly, there is typically a standard tolerance (e.g. +/-3%) on the quantity that can be produced to satisfy an order. To improve efficiency, manufacturers sometimes take advantage of this tolerance and produce more or less than the ordered amount. The amount produced in excess of the order quantity is called *overrun*. Similarly, any production shortfall is called *underrun*. The paper manufacturer is usually responsible for the freight cost from its mill to the customer's location, which can constitute as much as 15% of the selling price. In ordet to reduce transportation costs, manufactures prefer to produce an order in a mill close to the order's final destination.

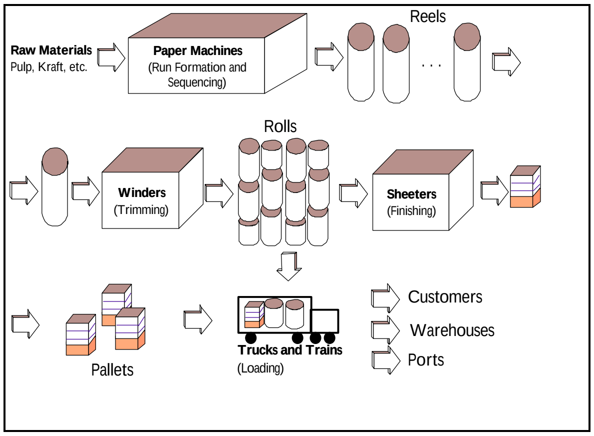


Figure 1. An overview of the paper manufacturing process

A typical large paper manufacturing enterprise has several mills in different locations, each mill having one or more paper machines. Each paper machine is capable of producing a subset of the company's products at different production rates. Paper production is a continuous process in which a machine can make only one product at a time because each product has its own unique recipe. When the product being made on a machine is changed, the machine continues to operate, but the paper it produces is of poor quality for some time after the change is initiated. The length of this transition time or *setup time* depends on the products being produced before and after the transition; transitions between similar products are shorter than transitions between very different products (i.e. the setup times are sequence-dependent). The setup times between products can also be machine dependent. The production between two consecutive setups is called a *run*. Often there is a restriction on the minimum length of a run since very short runs are likely to produce paper with quality problems.

Given a set of firm orders and forecasted demand, the schedulers must decide how to:

- *allocate* orders to mills and machines

- *sequence* the orders on each machine and form a sequence of runs
- *trim* the reels of paper to create rolls of the right size for each order
- *load* the rolls into vehicles for shipping to the customers and distribution centers

in order to

- maximize profit by minimizing trim loss, transportation costs and inventory carrying costs.
- maximize on-time delivery by minimizing both early and late delivery.
- minimize manufacturing disruptions.
- maximize product quality by minimizing customer preference violations.

Most problems in schedule generation are variations of well known NP-complete problems. For example, run formation and sequencing is an extension of multi-machine job sequencing problem with sequence dependent setup times, trimming is an extension of the cutting stock problem and loading falls into the domain of the multiple knapsack problem.

3 Decision Support Approach

Our approach to decision support is to augment the skills of the schedulers by offering an intelligent assistant capable of providing the information needed to make a good decision. In our decision support approach, the scheduler plays two roles: participating in solution generation and deciding on the final schedule. Before presenting the algorithms that we use for scheduling, we briefly explain our approach to decision support in order to motivate our choice of software architecture.

A good decision support system should relieve the schedulers of mundane computational tasks and provide them with high quality solutions to allow them to focus on decision making. The system should apply advanced solution methods to systematically search a larger space of possible solutions than a scheduler could consider. In addition, it should track and identify all constraint violations and perform extensive bookkeeping operations.

The schedulers that use our system are experts in their field and have an intimate and extensive knowledge of the production environment, operating procedures and company policies. Through years of experience, they have developed heuristics which quickly identify promising schedules. However, these heuristics search only a small part of the solution space and overlook opportunities for alternative and possibly better solutions.

By combining the advanced solution techniques used by our system with the domain expertise of the schedulers, our system facilitates cooperation to produce a set of high quality alternative solutions for consideration. A premise of our decision support approach is that the scheduler's expertise can not be fully captured in software. As stated by (Wagner 1969) " ... an operations research model is never sufficient unto itself: it cannot become entirely independent of judgment supplied by knowledgeable managers. ... The question is not when to apply science and when to rely on intuition, but rather how to combine the two effectively." Our system allows the scheduler to create new schedules or modify existing schedules at any level of detail in order to provide expert guidance. Unlike some earlier approaches (IBM 1998; Yu 1985) the iterations in the solution generation process are not only based on changing the objective weights, but also based on more detailed input from the scheduler about the solutions. The schedules generated by the scheduler, like those generated by the system, serve as a basis for further search and refinement by our system. This contrasts with some other systems, such as PRM (IBM 1998), where once a scheduler edits a solution, the system cannot further improve it. There are cooperative expert systems, such as Scheiker (Masayuki and Morishita, 1991), where the cooperation between the scheduler and the system is similar to our approach; however, expert system rules are typically far less capable of making useful improvements to scheduler generated solutions than the dircet improvement methods we employ.

In the end, the final decision is in the hands of the scheduler who is responsible for the quality of the resulting schedule. We understand that a software system that offered a single take-it-or-leave-it solution for a complex multi-objective problem would not fulfill the information needs of the schedulers. It is difficult, if not impossible, for a software system to generate the "best" solution for a multi-objective problem, since the relative importance of the objectives is usually not known a priori and may change depending on the changing conditions (Bennett 1998; Das 1998; Steuer 1989; Yu 1985). Furthermore, a single solution does not provide any insight to the scheduler about the tradeoffs between the objectives and about how the solution compares to the other possible "good" solutions. On the other hand, a system that overwhelmed a scheduler with thousands of solutions would also be of little use. In order to present a reasonably sized set of alternatives that illustrate real tradeoffs, we evaluate each alternative in business terms with respect to key objectives and present the set of non-dominated solutions to the scheduler. A solution is called *non-dominated* with respect to a given set of solutions, if none of the other solutions in the set is better in every objective. The objectives used in the evaluation are profitability (includes trim efficiency and transportation cost), customer satisfaction (measured by tardiness and other customer satisfaction measures, such as deviations from ordered quantities) and manufacturing disruptions (includes setup time and number of short runs). These objectives represent the goals of the key constituencies of the enterprise. As the ultimate decision maker, the scheduler examines the tradeoffs offered by the system, and selects the "best" schedule.

To implement our approach to decision support, we employ the Asynchronous Team (A-Team) architecture. This architecture supports our aim of providing an intelligent assistant that can work cooperatively with schedulers to produce multiple solutions that illustrate tradeoffs (Murthy 1997). In addition, the A-Team architecture allows us to combine multiple problem-solving approaches which increases the likelihood that good solutions can be found in a reasonable amount of time.

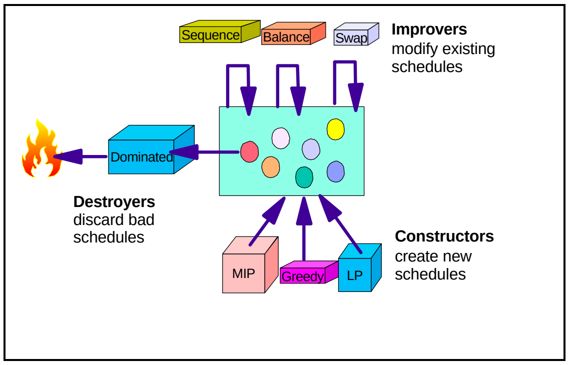
3.1 Asynchronous Team Architecture

An asynchronous team is a team of software agents that cooperate to solve a problem by dynamically evolving a shared population of solutions (Talukdar et al. 1983, 1993; Murthy 1992). The agents work asynchronously and embody their own methods for deciding when to work, which solutions to work on, and how to generate new or improved solutions. Agents do not explicitly coordinate their activities with one another. Cooperation is achieved by allowing an agent to modify a solution in the population created by other agents. The solution alternatives are usually the result of contributions made by many agents (Salman et al. 1997).

As shown in Figure 2, there are three types of agents in an A-Team: constructors, improvers, and destroyers. *Constructors* take only the problem definition and create new solutions. Deterministic constructors attempt to run only once, while constructors that use randomized algorithms may run many times to add a variety of solutions to the population. *Improvers* attempt to improve upon the current set of solutions in the population by modifying or combining existing solutions. *Destroyers* attempt to keep the population size in check and focus the efforts of the improvers by removing bad solutions from the population. A scheduler can interact with the A-Team, taking on the role of an agent by adding new solutions or by removing or improving existing solutions. Within the population, each solution is evaluated with respect to multiple objectives. Typically, solutions to scheduling problems are evaluated in terms of cost, timeliness, product quality, and manufacturing disruptions. As a group, the set of agents in an A-Team evolve the population of solutions to provide a set of good alternatives that illustrates important tradeoffs among the competing objectives. Because the agents share access to the populations, an A-Team is like a blackboard system but without a central controller (Erman et al. 1980). A-Teams also have certain characteristics of a genetic algorithm in that a population of solutions evolve over time (Holland 1975). However, unlike genetic algorithms, the mechanisms for altering individual solutions may be highly directed by taking into account domain specific knowledge, rather than depending upon random mutation or crossover.

Figure 2: A-Team architecture.

For difficult optimization problems, any single algorithm is not likely to produce high quality solutions in a reasonable time for all problem instances. The combination of agents in an A-Team constitute a portfolio of algorithms which is more robust than any single algorithm (Huberman 1997; Murthy 1992; Lee et al.1996). Beyond the benefits of using multiple algorithms independently, the A-Team architecture allows the agents to cooperate synergistically, and leads to higher quality and more diverse solutions. Agents may focus on optimizing different objectives and explore different parts of the search space for feasible solutions. The A-Team architecture further enhances this synergistic cooperation by enabling the scheduler to function as an agent.



Recently it has been shown that agent-based architectures hold promise for solving complex multi-objective optimization problems. (Liu and Sycara, 1996) have shown the benefits of a tightly-coupled agent architecture in solving the job shop scheduling problem. (Beck and Fox, 1994) presented a mediated approach to multi-agent coordination in the context of a supply chain management system. A model of the supply chain in which each of the key players in the chain is encapsulated in an autonomous agent is described in (Swaminathan et al. 1996). A blackboard-style architecture is used in (Smith et al. 1990) for generating and revising factory schedules, wherein a set of distinct methods are selectively employed to generate, revise or analyze specific components of the overall schedule.

The simplicity and modularity of the A-Team architecture makes it well suited for creating systems that are used daily in production environments. Modularity allows new algorithms, in the form of agents, to be added to an A-Team without major modifications to the existing system. Algorithms can be incomplete since partial solutions can be added to the population and serve as seeds for the improver agents. As a practical matter, modularity allows a customer support engineer to easily configure and tune the system at installation time by turning agents on or off.

4 Algorithm Overview

We now give an outline of the algorithms that we use for paper mill scheduling. Our construction algorithms, embodied in A-Team agents, can be grouped based on the sub-problem(s) they focus on:

- Allocating orders to machines and creating initial sequences.
- Sequencing orders and forming runs on a given machine.
- Trimming reels to form rolls.
- Loading rolls into vehicles for shipping.

Orders are allocated to paper machines based primarily on transportation cost considerations. Machines are geographically distributed and producing an order on a machine close to the order's final destination saves in transportation costs. Order allocation should also consider the due dates of the orders and the load balance on each machine to ensure on-time delivery. We use the following approaches for order allocation: *linear programming*, *integer programming*, *network flow models* and *dispatch algorithms*. The order allocation methods create partial solutions in which orders are allocated and given an initial sequence on each machine. The initial sequence of orders defines a sequence of runs on each machine. The problem we consider at this stage generalizes other machine scheduling problems studied earlier in the literature (Ho and Chang, 1991; Lee and Pinedo, 1997). It can be characterized as scheduling orders on parallel non-identical machines, subject to job-machine restrictions, machine- and sequence-dependent setups, batch size preferences, job-machine assignment costs and earliness/tardiness penalties, with additional implications on downstream processes. Further details of our approach for solving this complex scheduling problem are discussed in (Akkiraju et al. 1998).

The linear programming and network flow based models are fast, but they create allocations in which some orders may be split between mills and machines. In some cases, splitting orders across multiple machines may be undesirable, especially if the machines are geographically dispersed (which complicates order tracking) or produce paper with different properties. In these cases, we "fix" the resulting allocation by merging or partially merging some of the split orders. We use several heuristic approaches for fixing allocations with order splitting. The idea is to merge pieces of an order, while considering transportation cost, due dates and the utilization of the machines.

In dispatch algorithms, the method is to first sort the orders according to some criteria and then allocate each order in the list to a machine based on a "priority index" of the order-machine pair. The dispatch approach does not split orders across mills and machines. Sort criteria are based on combinations of order properties, such as due date, processing time and tardiness penalty. For example, we sort orders in increasing order of their due dates and then we sort orders with the same due date by grade. In our implementation, we use several combinations of primary and secondary sort criteria for the initial sorting of the orders. The priority index is calculated by combining several measures of an order-machine pair, such as order completion time, earliness, tardiness, transportation cost and processing times.

For each order-machine allocation, we create new sequences for the orders on each machine, as alternatives to the initial sequence generated by the order allocation agents. In order sequencing, the goal is to produce the orders on time, to ensure smooth transitions between grades and to obtain good trim efficiency from the resulting runs. The sequencing problem, at this stage, falls into the domain of single machine sequencing with sequence-dependent setups with the objective of minimizing weighted tardiness (Abdul-Razaq et al. 1990; Du and Leung, 1990; Koulamas 1994; Lee et al. 1997). We use dispatch algorithms for sequencing orders on a given machine; the idea is to select one job at a time, from the set of remaining jobs, based on some "priority index" and schedule the job with the highest priority as the next job on that machine. The priority index is calculated by combining different properties such as earliness, tardiness, and processing and setup times. We have several algorithms which follow this framework, but use different priority indices. We also use some priority rules which are known in the literature, such as the PSK (Panwalkar et al. 1993) and NBR (Holsenback and Russell, 1992) rules. When we sequence the orders to form runs, we also consider downstream implications on trim efficiency and shipping. We compute estimates of the trim efficiency and shipping cost, and modify the runs as necessary.

To improve order allocations and sequencing, our improvers change the sequence of orders on a machine or swap orders between mills and machines. Some improvers move only one order at a time, while others may move a group of orders, e.g. a whole run, to a new position. Order allocation improvers focus mainly on reducing the transportation cost and balancing the load on the machines, while considering the impact of any move on other measures, such as tardiness. Sequencing improvers focus on reducing tardiness and setups, while considering the impact of any move on other measures such as transportation cost and trim efficiency. We also have some improvers, which try to reduce the number of very short runs and setups. Having very short runs is undesirable because they may violate the minimum run size requirements and they are more likely to have poor trim efficiency. These improvers may merge two different runs of the same grade, or move orders from a long run into a short run. Again, they consider the impact of these moves on other measures, such as the transportation cost and tardiness.

During each production run, a paper machine produces a set of reels. Cutting the reels into smaller rolls of ordered sizes is called *trimming*. For example, a reel of width 200" can be cut into 4 rolls of width 43" and one roll of width 27" resulting in 1 inch of trim loss (200 - (4)(43) - (27) = 1). The main objective in trimming is to minimize the trim loss while minimizing deviations from customer ordered quantities. Our agents first generate a set of "efficient" patterns, which utilize the deckle maximally. They then select a subset of these patterns to produce the rolls in that run. Pattern generation is done using a randomized procedure to ensure that the generated patterns exploit the diversity of roll widths in the run. To select the patterns to use and the repetitions of each pattern, some of the agents use integer programming, while others use linear programming with randomized rounding. Having selected the patterns and their repetitions, the agents sequence the patterns with a user-selectable combination of objectives such as making high priority orders first, finishing orders in the sequence of their due dates, minimizing pattern changes (winder setups), minimizing diameter changes and minimizing the number of orders in process. Once a collection of patterns is selected and sequenced to produce and cut the reels in a run, this is called a *trim sheet* (for that run).

Some improver agents increase trim efficiency by modifying runs that have relatively low trim efficiency. Such agents may move certain orders or rolls from one run into a different run, possibly on a different machine. The two modified runs are then trimmed again. If the resulting trim is improved, the change is kept. We have several methods for selecting orders or rolls to be moved. In one method we examine the trim patterns having the lowest efficiency in the trim sheet, and select an order that is made in those patterns. In another method we look for a rollsize that makes up a large fraction of all the rolls to be made in the run. If this rollsize does not trim well, then some of its orders are moved to another run. This operation is repeated using different selections of runs and orders to improve the trim efficiency of the entire schedule.

In many paper mills, rolls are loaded directly onto vehicles as they emerge from the wrapping machine. *Loading* is the process of deciding which vehicles to use and how to load them. The choice involves:

- selecting the mode of transportation,
- selecting the sizes of vehicles to use
- deciding how to load the rolls into the vehicles.

Mode choices typically include rail and truck. Rail is generally less expensive, but has longer transit times. For each mode, the available vehicles come in fixed sizes with weight and volume limits. In addition to the physical limitations, there are also insurance regulations that restrict loading alternatives. In order to be insured for shipment by rail, rolls must be placed in the vehicle following a prescribed *loading pattern*. It is an NP-complete problem to determine if a set of rolls can be loaded into a given vehicle. In order to improve vehicle utilization, rolls going to different locations can be mixed in the same vehicle. However, mixing rolls going to different locations incurs drop-off charges and extra distance charges and may increase tardiness. In addition to limits on individual vehicles, there are also limits on the number of vehicles that can be loaded in a shift and the number and types of vehicles that are available per day.

The loading problem is similar to a multiple knapsack problem (Martello and Toth, 1990) with multiple sizes of knapsacks, knapsack availability constraints and complex rules for determining legal combinations of items in a knapsack. To address this problem, we use two basic approaches: traditional knapsack packing algorithms and algorithms that work with a set of legal loading patterns. Our greedy knapsack packing algorithms select vehicles based on cost or on-time delivery and load the vehicles until the weight or volume limits are reached. The results are then checked to ensure that the rolls can be loaded following prescribed loading patterns. These algorithms work best for vehicles like trucks where this check almost always succeeds. The second class of algorithms generate and share a set of legal loading patterns. A pattern generator enumerates subsets of rolls that meet weight and volume constraints for each type of vehicle and then filters the set to remove the subsets of rolls that cannot be loaded into a vehicle using a prescribed loaded pattern. The loading problem is then reduced to selecting how many of each pattern to choose in order to ship all the items. We use a combination of mixed integer programming algorithms and greedy algorithms to do the pattern selection. This second approach is most effective for rail cars. In addition to these two basic approaches, we also make use of local improvement routines. These algorithms look for opportunities to reduce costs, reduce drop shipments and improve vehicle utilization by swapping items between vehicles and changing vehicle sizes. These algorithms perform a stochastic hill climbing and serve to explore the area around the solutions generated by the first two types of algorithms, looking for improvements.

Some of the algorithms we discussed focus only on certain components of the global scheduling problem, but also consider the objectives of downstream processes. In addition, the system uses several algorithms that specifically address problems created by process interactions, and modify schedules to resolve them. The details of these algorithms are discussed in (Keskinocak et al. 1998) Assembled in the A-Team framework, these algorithms cooperate synergistically to generate high quality, complete enterprise schedules.

5 Facilitating Cooperation Between the Scheduler and the Software

In our decision support approach, we promote cooperation between the scheduler and the system to improve solution quality and the decision making process. Through interacting with the system, the scheduler can suggest alternatives, understand and modify solutions (Figure 3). The system can take guidance from the scheduler, offer solution alternatives and point out constraint violations. Schedulers can submit manually created or modified solutions back to the scheduling system for evaluation and comparison with other solutions, as well as for further improvement by the system. By working with the system to explore the solutions space, the schedulers gain insight that allows them to be more confident of their final decision. To achieve the level of cooperation that is required, the scheduler and the system need to communicate effectively. In this section, we discuss the type of communication that is needed and how our graphical user interface supports the required communication.

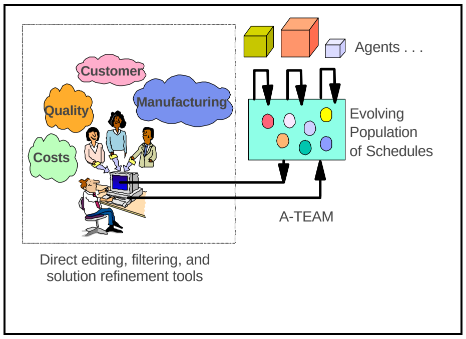
Our graphical user interface, developed specifically for paper mill scheduling allows the schedulers to explore the particulars of a scheduling problem and its solutions. The schedulers have deep knowledge about the domain, while the system has the details of the current set of orders, the state of production and the currently released schedule. Using this information, the system can identify changes in the order book and discrepancies between the scheduled and actual production. One of the responsibilities of the scheduler is to modify the existing schedule to account for these differences. The interface must convey the required information to the scheduler in an easily understood form. In addition, both the scheduler and the system need to be able to express their own solutions and suggest modifications to each other's solutions.

Figure 3. A-Teams provide the basis for building enhanced decision-support systems.

The interface facilitates communication by providing multiple views of the data at different levels of granularity and by using the scheduler's own graphical language as a basis for direct manipulation of a schedule. The views provided by the interface include the following:

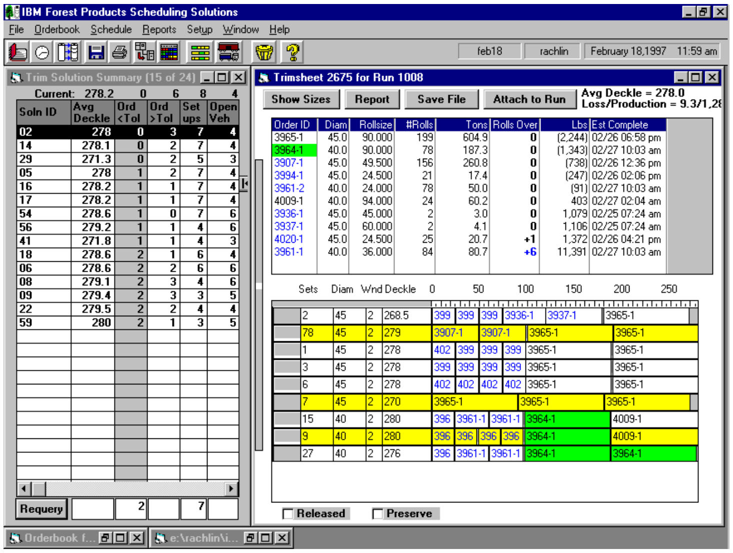
- *Summary of results*: presents the objective function evaluations for each enterprise schedule in the non-dominated set.
- *Detailed view of each enterprise schedule*: for all the machines in the schedule, shows the duration of each run and the run sequence.
- *Detailed view of each machine schedule in a given enterprise schedule*: shows the duration and product type of each run and the run sequence on the machine.
- *Detailed view of a run in a given machine schedule* (Figure 4): shows the orders in the run and the trim sheet.
- *Detailed view of a loading schedule*: shows how orders are loaded onto vehicles.

To illustrate our approach to user interaction, Figure 4 shows the screen used for detailed trim scheduling, the process where reels are cut into rolls. The grid on the upper right of Figure 4 shows the specifications of the orders that need to be trimmed in this run. It includes the roll size, the diameter and the number of rolls for each order. These attributes specify the details of the trim scheduling problem. The grid on the lower right portion of Figure 4 is the *trim sheet* which gives a graphical representation of a solution. Each line in the trim sheet is a trim pattern which shows the placement of the slitting knives and the assignment of the resulting rolls to orders. Each pattern is repeated one or more times to create a pattern set. Within the trim sheet, the background color changes for each pattern that requires a new knife setup. Also, selecting an order in the top grid highlights its position in the trim sheet, as shown with order 3964-1. Finally, the grid on the left gives a summary view of the set of alternative trim solutions and their evaluations. For a trim sheet, there are 35 evaluations that summarize both the characteristics of a trim sheet, such as the number of setups, and downstream consequences, such as the number of open vehicles. For a loading dock, having too many partial loads waiting on the dock can interfere with the loading of other vehicles. The number of open vehicles is a measure of the maximum number of partial loads waiting to be completed at any one time. The columns in the solution summary grid can be rearranged to show the subset of evaluations that the scheduler is interested in or can be scrolled to show the complete set.


Customer
Quality
Manufacturing
Costs
Agents . . .
Evolving Population of Schedules
A-TEAM
Direct editing, filtering, and solution refinement tools

To understand how the interface is used, consider a typical scenario where a scheduler is trying to create a trim sheet for a production run and initially has the system generate a set of candidate solutions. Such is the case in Figure 4 where the system has produced multiple solutions and presented a set of 24 non-dominated solutions. The scheduler has then narrowed the set, by using the criteria boxes at the bottom of the grid, to view only solutions with no more than seven setups and no more than two orders under tolerance. Clicking on a column, the scheduler can sort the solutions by a particular evaluation. In this case, the scheduler has sorted by "orders under tolerance", which is the number of orders for which the quantity produced is under the minimum order quantity. Solutions that don't underrun orders are of particular interest if the mill has more capacity than it has orders for. Double clicking on a solution shows its details on the right hand side and allows the scheduler to determine which orders are being overrun and by how much. In this case, orders 4020-1 and 3961-1 are being overrun by 1 and 6 rolls respectively. To improve the solutions and explore alternatives, the scheduler can directly manipulate the trim sheet by dragging rolls between patterns, reordering, adding or deleting patterns. Once the scheduler finalizes a trim sheet, it can be released for production, using the check box at the bottom of the screen, and attached to the production run as its current trim sheet using the button at the top of the screen.

Each screen in our graphical user interface is organized to provide the information needed to accomplish a specific task or a set of related tasks, as is the case for the trim sheet editor we described above. The information is presented in an easy to understand graphical format. By allowing direct manipulation of the graphical representation of a schedule and by providing multiple levels of detail, our interface enables the communication that is required for effective cooperation between the scheduler and the system.


IBM Forest Products Scheduling Solutions
File Orderbook Schedule Reports Setup Window Help
feb18
rachlin
February 18,1997 11:59 am
Trim Solution Summary (15 of 24)
Trimsheet 2675 for Run 1008
Show Sizes
Report
Save File
Attach to Run
Requery
Orderbook f...
e:\rachlin\i...

Current: 278.2 0 6 8 4

Soln ID	Avg Deckle	Ord <Tol	Ord >Tol	Set ups	Open Veh
02	278	0	3	7	4
14	278.1	0	2	7	4
29	271.3	0	2	5	3
05	278	1	2	7	4
16	278.2	1	1	7	4
17	278.2	1	1	7	4
54	278.6	1	0	7	6
56	279.2	1	1	4	6
41	271.8	1	1	4	3
18	278.6	2	1	6	4
06	278.6	2	2	6	6
08	279.1	2	3	4	6
09	279.4	2	3	3	5
22	279.5	2	2	4	4
59	280	2	1	3	5
Requery		2		7	

Avg Deckle = 278.0
Loss/Production = 9.3/1,28

Order ID	Diam	Rollsize	#Rolls	Tons	Rolls Over	Lbs	Est Complete
3965-1	45.0	90.000	199	604.9	0	(2,244)	02/26 06:58 pm
3964-1	40.0	90.000	78	187.3	0	(1,343)	02/27 10:03 am
3907-1	45.0	49.500	156	260.8	0	(738)	02/26 12:36 pm
3994-1	45.0	24.500	21	17.4	0	(247)	02/26 02:06 pm
3961-2	40.0	24.000	78	50.0	0	(91)	02/27 10:03 am
4009-1	40.0	94.000	24	60.2	0	403	02/27 02:04 am
3936-1	45.0	45.000	2	3.0	0	1,079	02/25 07:24 am
3937-1	45.0	60.000	2	4.1	0	1,106	02/25 07:24 am
4020-1	45.0	24.500	25	20.7	+1	1,372	02/26 04:21 pm
3961-1	40.0	36.000	84	80.7	+6	11,391	02/27 10:03 am

Sets Diam Wnd Deckle 0 50 100 150 200 250

Sets	Diam	Wnd	Deckle						
2	45	2	268.5	399	399	399	3936-1	3937-1	3965-1
78	45	2	279	3907-1	3907-1	3965-1	3965-1		
1	45	2	278	402	399	399	399	3965-1	3965-1
3	45	2	278	399	399	399	399	3965-1	3965-1
6	45	2	278	402	402	402	402	3965-1	3965-1
7	45	2	270	3965-1	3965-1	3965-1			
15	40	2	280	396	3961-1	3961-1	3964-1	4009-1	
9	40	2	280	396	396	396	396	3964-1	4009-1
27	40	2	276	396	3961-1	3961-1	3964-1	3964-1	

☐ Released ☐ Preserve

Figure 4. Screens from the graphical user-interface. The GUI enables effective cooperation between the scheduler and the agents in the A-Team.

6 Business impact

Our paper mill scheduling system has significantly improved the scheduling and decision making process for our customers, giving them substantial monetary returns. Improvements come both from the higher quality of the solutions that our system generates and from positive changes in business processes that our approach to decision support fosters. In terms of solution quality, one of our customers, Madison Paper Industries, reports a reduction in trim loss by 6 tons per day and a 10% reduction in freight costs (Shaw 1998). Each of these savings amounts to millions of dollars per year. Our system has allowed other customers to change their business practices to achieve “perfect trim” consistently and to improve their ability to negotiate with customers when accepting orders. In this section, we highlight some of the ways in which our system improves the quality of schedules and give examples of how it has allowed companies to improve their business processes.

Customarily, the value of a scheduling system is determined only by the quality of the schedules that it produces. The schedules that our system generates for each stage of the process improve upon the state-of-the-art in scheduling. Adam Stearns of Madison Paper Industries reports “We would use our old trim package, throw orders into it and let it trim them the best it could. Then we would let the IBM module take the same orders and manipulate them to come up with a trim. We saw that the IBM package was consistently saving over two inches [of trim loss] -- just an incredible amount.” (Shaw 1998, pg. 74) “Testing shows that we are getting about 10% savings annually on distribution costs from the load planning piece alone, which amounts to millions of dollars. ... We expected the system’s GUI [graphical user interface] to make load planning easier, but we didn’t expect to gain these efficiencies.” (Shaw 1998).

Our system achieves a global perspective that results in further schedule improvements by considering options for changing a schedule for one stage of production to improve the schedule for subsequent stages. Schedulers have long recognized that such changes could result in dramatic improvements. However, the effort needed to explore such options was prohibitive when incompatible software was used to schedule each stage and each scheduler was only familiar with the software used to schedule their stage of production. We cite two examples to illustrate how our scheduling system takes advantage of this global perspective.

The first example involves moving orders between mills, machines and production runs to reduce trim loss. It is often possible to improve trim efficiency by adding orders to a run to get better trim patterns. The difficulty is in identifying potentially good orders to move and in determining the impact of the move on the rest of the schedule. Moving an order to an earlier run can increase inventory costs, reduce paper quality if the length of the future run becomes too short and increase tardiness by delaying the start of subsequent runs. Moving an order between mills can increase shipping costs and increase shipping times. Without an integrated system, determining the extent of such impacts can be difficult and time consuming. To reduce complexity, schedulers often use simple business rules to limit their search and the impacts that they need to consider. Such rules might include only looking at orders already assigned to the same machine and scheduled for production in the next month. With such a limitation, impacts on inventory cost, shipping cost and load balance can be ignored and the scheduler only needs to consider whether the move increases tardiness for the other orders assigned to the same machine in the next month. Our integrated software allows these business rules to be relaxed. Both the scheduler and the software can explore a larger set of alternatives to improve the quality of the schedules.

Our second example involves sequencing trim patterns in a trim sheet to reduce shipping cost. In a stand-alone trim system, trim patterns are sequenced by their due dates or by the ship date needed to meet their due dates to reduce tardiness. This ordering also tends to reduce transit costs by increasing the time allowed for transit, permitting slower and less expensive modes of transportation to be used. However, this approach misses several opportunities to reduce transportation costs that a trim scheduler may not recognize. Specifically, producing an order early may allow it to be shipped with another order going to the same or nearby location. However, without looking at how the orders could be combined in vehicles, it is impossible to determine if there are any cost savings. Furthermore, simply ordering trim patterns to increase the time available for transit does not reduce transit costs unless the increase is large enough to allow the mode of transportation to be changed. The amount of the time increase needed depends on the destination, the modes of transportation available for the order and even the time of day that a carrier makes pickups at the mill. Conversely, decreasing the time available for transit increases costs only if it requires switching to a more expensive mode of transportation and increases tardiness only if a sufficiently fast mode of transportation is not available. Our integrated system allows a scheduler to quickly determine how changes in trim pattern sequence affects transit costs and tardiness.

Improving schedule quality has obvious impacts on profitability. However, just as important are the opportunities to change business processes that our system affords by showing multiple good alternatives, some of which may violate some constraints. These solutions can suggest good opportunities to look for alternative means of production and can suggest when it would be profitable to negotiate to change customer requirements or business policies.

One of our customers regularly achieves perfect trim (i.e. no trim loss) by using solutions with underruns that violate the minimum order quantity for one or more orders. At this mill, orders can be filled from production, from inventory or from a combination of inventory and production. The mill keeps a small inventory of rolls that it can retrim to meet the specifications of a particular order. The problem for the schedulers is to determine when to use the inventory. Using our software, the schedulers look for solutions that give perfect trim and try to complete underrun orders from inventory. Overruns become new inventory.

Order specifications and manufacturing policies are generally viewed by scheduling software as hard constraints that a schedule must meet. However, changes to these constraints are often negotiable if there is significant financial or customer service motivation. For example, some mills have a policy that limits the number of setups on a winder to at most one per hour. However, the production manager may be willing to violate the policy if it substantially increases trim efficiency or reduces order tardiness. Similarly, customer service representatives can negotiate with a customer to change the specifications of an order, but would only do so if it would substantially benefit another order or significantly reduce expenses. "...when our sales staff calls with an order for a specific tonnage, roll size and destination, we can enter this into the scheduling system as 'pending' and it will tell us when we can do it and what it is going to cost if we do it. That is powerful, powerful information for sales to provide a customer." (Shaw 1998 pg. 81). If the solution suggested by the system underruns a pending order, then the sales staff may offer to split the order, filling the rest of the order at a later date. If a solution overruns an order, the sales staff may offer a volume discount if the customer takes the extra production.

Finally, the global perspective of our scheduling software has sped up the scheduling process and improved communications among schedulers since they can share schedules and see the impact of their changes on other stages of production. This reduces the time needed to generate a new schedule and allows production to react more quickly. "For our group, we thought this system was going to make our job much easier, and it has for the most part. ... But, we have also found that rather than having a one or two-day window to react to customer or production concerns, we have just closed that window, since they know we need less time to make changes. But it has certainly given both production and customers a lot more flexibility in how they manage their processes" (Shaw 1998 pg. 81).

Our scheduling system has been well received in the market place because it helps paper manufacturers to satisfy the needs of their customers while reducing costs. Improvements in schedule quality result in measurable monetary returns that give our system a payback period of less than one year (Shaw 1998). Other improvements, like reduced scheduling time and changes in business processes are harder to quantify but are appreciated by paper manufacturers as they operate in a highly competitive market.

7 Conclusions

We have developed a system for enterprise wide scheduling of paper manufacturing and distribution. Our system has been successfully deployed at multiple paper mills in North America. Computer World reports that within one year after starting to use our system, Madison Paper saved $2 to $3 million in trim and transportation costs at its Madison mill (Hoffman 1996). Positive effects of our system on scheduling flexibility, efficiency and greater responsiveness to order changes are reported in Pulp and Paper (Shaw 1998), in addition to the improvements in cost and customer satisfaction.

Three factors have contributed to the success of our scheduling system. First, our system takes a global perspective and accounts for upstream and downstream interactions in the manufacturing and distribution process. The schedules generated by the system are *global*; they take into account enterprise-wide objectives and include detailed schedules for each stage of production and distribution. Second, the system employs a new multi-objective decision support approach. It offers schedulers a set of high quality solution alternatives which reveals tradeoffs with respect to competing objectives, such as increasing profitability, on-time delivery, product quality, and reducing manufacturing disruptions. Each alternative is evaluated in business terms reflecting these objectives. To produce these scheduling alternatives, we developed multiple solution methods based on linear/integer programming, network flows and heuristics. In doing so, we have extended the set of problems considered in the scheduling literature and have advanced the state-of-the-art in solving problems in this area. Our methods are encapsulated as agents in an A-Team. The A-Team architecture facilitates cooperation between different solution methods and the scheduler, resulting in solutions that no single method would likely achieve working alone. The scheduler can modify and improve a solution generated by the system, and unlike many other systems, a solution generated by the scheduler can also be fed back to the system for further refinements and improvement. Our graphical user interface enhances cooperation between the scheduler and the system. The scheduler examines these alternatives and their business evaluations, understands the tradeoffs and using business judgment selects the “best” schedule. Third, our system enables schedulers to discover new alternatives and thereby effectively negotiate with customers and manufacturing personnel. Therefore, in addition to improved solution quality, our system has played an important role in fundamentally changing the planning and scheduling process.

8 References

Abdul-Razaq T.S., Potts C.N., and Van Wassenhove L.N., 1990. A Survey of Algorithms for the Single Machine Total Weighted Tardiness Scheduling Problem. Discrete Applied Mathematics, 26, 235-253.

Akkiraju R., Keskinocak P., Murthy S., Wu F. 1998. An Agent-based Approach for Multi-machine Scheduling. Proceedings of the Tenth Annual Conference on Innovative Applications of Artificial Intelligence, Menlo Park, CA.

Beck J.C. and Fox M.S. 1994. Supply Chain Coordination via Mediated Constraint Relaxation. Proceedings of the First Canadian Workshop on Distributed Artificial Intelligence, Banff, AB, May 15, 1994.

Bennett D.P. 1998. Multicriteria Optimization. http://www.ieor.berkeley.edu/~bennett/multicrit.html.

Das I. 1998, Multi-objective Optimization. http://www.mcs.anl.gov/otc/Guide/OptWeb/multiobj/index.html.

Du J., and Leung J.Y. 1990. Minimizing Total Tardiness on One Machine is NP-hard. Mathematics of Operations Research, 15, 483-494.

Erman L.D., Hayes-Roth F. Lesser V.R. and Reddy R.D; 1980. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. In ACM Computing Serveys 12(2).

Goodwin R.T., Rachlin J., Murthy S., Akkiraju R.; 1998. Interactive Decision Support: Advantages of an Incomplete Utility Model. AAAI Spring Symposium on Interactive and Mixed Initiative Decision-Theoretic Systems.

Ho J.C., Chang Y. 1991. Heuristics for Minimizing Mean Tardiness for m Parallel Machines. Naval Research Logistics, 38, 367-381.

Hoffman T; 1996. A.I. Based Software Models Help Cut Production Costs. Computer World, September 2. http://www.computerworld.com/idx_usea.htm.

Holland J.H; 1975.Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor MI.

Holsenback J.E., Russell R.M., 1992. A heuristic Algorithm for Sequencing on One Machine to Minimize Total Tardiness, Journal of the Operational Research Society, 43, 53-62.

Huberman, B., Lukose, R., and Hogg, T.; 1997. An Economics Approach to Hard Computational Problems. Science 275:51-54.

IBM Production Resource Manager (PRM) 1998. Release 5.0 User’s Guide and Reference.

Keskinocak P., Akkiraju R., Goodwin R.T., Murthy S., Wu F. 1998. Multi-criteria Optimization for Paper Mill Scheduling, in preparation.

Koulamas C. 1994. The Total Tardiness problem: Review and Extensions. Operations Research, 42, 1025-1041.

Lee, H.S., Murthy, S., Haider, S.W., and Morse, D.; 1996. Primary Production Scheduling at Steel-Making Industries. IBM Journal of Research and Development.

Lee Y.H., Bhaskaran K., and Pinedo M. 1997. A Heuristic to Minimize the Total Weighted Tardiness With Sequence Dependent Setups. IIE Transactions, 29, 45-52.

Lee Y.H. and Pinedo M. 1997. Scheduling Jobs on Parallel Machines With Sequence Dependent Setup Times. European Journal of Operational Research, 100, 464-474.

Liu J.S. and Sycara K.P. 1996. Multiagent Coordination in Tightly Coupled Task Scheduling. Second International Conference on Multiagent Systems (ICMAS ’96), Kyoto, Japan, December 1996.

Martello S. and Toth P. 1990. Knapsack Problems: Algorithms and Computer Implementations, Wiley, Chichester, U.K.

Masayuki N. and Morishita S. 1991. Cooperative scheduling and its application to steelmaking processes. IEEE Trans. on Industrial Electronics. 38(2).

Murthy S., Rachlin, J., Akkiraju R., Wu F. 1997. Agent-Based Cooperative Scheduling. In Constraints & Agents. Technical Report WS 1997-97-05. Menlo Park:AAAI Press.

Murthy S. 1992. Synergy in Cooperating Agents: Designing Manipulators from Task Specifications. Ph.D. Dissertation, Carnegie Mellon University.

Panwalkar S.S., Smith M.L., and Koulamas C.P., 1993. A Heuristic for the Single Machine Tardiness Problem. European Journal of Operational Research, 70, 304-310.

Pickard G.C. 1997. Paper Machine Scheduling Yields Gains at James River Naheola Mill. Pulp & Paper 71 (9) 125-129.

Reddy R. 1996. The Challenge of Artificial Intelligence. IEEE Computer, 29 (10), 86-98.

Salman F., Kalagnanam J., Murthy S. 1997. Cooperative Strategies for Solving the Bicriteria Sparse Multiple Knapsack Problem. IBM Research Report RC 21059 (94164).

Shaw M. 1998. Madison Streamlines Business Processes with Integrated Information System. Pulp and Paper 72 (5).

Smith F.S., Ow P.S., Potwin J., Muscettola N., Matthys D.C. 1990. An Integrated Framework for Generating and Revising Factory Schedules. Journal of the Operational Research Society, 41, 539-552.

Stanley, G.L. 1996. Third year of record sales expected for U.S. paper and related industries in 1996. TAPPI Journal 79 (1), http://tappi.org/publicat/journal/1997/jan-stan.pdf.

Steuer, R.E. 1989. Multiple Criteria Optimization: Theory, Computation, and Application. Malabar, FL: Robert E. Krieger Publishing.

Swaminathan J.M., Smith S.F. and Sadeh N.M. 1996. A Multi-Agent Framework for Modeling Supply Chain Dynamics. Proceedings NSF Research Planning Workshop on Artificial Intelligence and Manufacturing, Albuqerque, NM, June 1996.

Talukdar S.N., Pyo S.S., and Mehrotra R. 1983. Distributed Processors for Numerically Intense Problems. Final Report for Electric Power Research Institute (EPRI) Project. RP 1983-1764-3

Talukdar S.N., Souza P. de and Murthy S. 1993. Organizations for Computer-Based Agents. Engineering Intelligent Systems, 1(2).

Wagner H.M. 1969. Principles of Operations Research, With Applications to Managerial Decisions, Prentice-Hall, Englewood Cliffs, N.J.

Yu, P.L. 1985. Multiple-Criteria Decision Making. New York: Plenum Press.