A NEW DECISION SUPPORT SYSTEM FOR PAPER MANUFACTURING

Rama Akkiraju, Pinar Keskinocak, Sesh Murthy, Frederick Wu

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
tel: +1 914 945 3566, fax: +1 914 945 4527
e-mail: {akkiraju, pinar, murthy, fwu}@watson.ibm.com

1. Introduction

In this paper, we describe a cooperative multi-objective decision-support system for scheduling operations in a multi-mill multi-machine environment in the paper industry. Scheduling production and distribution in real-world paper manufacturing environments is an extremely complex task requiring the consideration of numerous constraints and objectives. The dependencies among the various stages of the production and distribution process compound the problem complexity. To the best of our knowledge our system is the first on the market or in the literature, which addresses scheduling problems at different stages of the paper manufacturing process in an integrated fashion. Our system follows a multi-criteria optimization approach and generates *multiple* alternative solutions, using an Asynchronous Team (A-Team) of cooperating software agents. In the A-Team framework, multiple heuristics and optimization based solution methods are encapsulated as agents. The system at any time presents a filtered subset (the non-dominated set) of these alternatives to the scheduler. The scheduler can modify and improve the schedules created by the system and *vice versa*. Thus, our system supports cooperation between the computer and the schedulers to work towards improved scheduling decisions.

In the remaining part of the paper, we present an overview of our system. In Section 2, we discuss the algorithms employed by the system, which are encapsulated as agents in the A-Team framework. In Section 3, we give a brief description of the A-Team framework. We discuss the positive business impacts of our system and present our conclusions in Section 4.

2. Algorithm Overview

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 (i.e. there are order-machine restrictions). When a machine switches from one product to another, there is a sequence dependent setup. Paper machines produce large *reels* of paper, with a fixed width (called *deckle*). The reels are later cut into rolls of paper (sized to customer specifications); this is called *trimming*. Finally, the rolls are packed onto trucks and rail cars for shipment to customers, warehouses and ports. Further details of the paper manufacturing process can be found in (Biermann 1993).

Each customer order defines the quantity, product type, dimensions (roll width and diameter), due date and shipping destination. Given the status of the manufacturing enterprise, future orders and forecasts, the schedulers must decide how to:

- *allocate orders* to mills and machines.
- *sequence* the orders on each machine and form production runs (batches).
- *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 or distribution centers.

in order to

- maximize profit by minimizing trim waste and transportation costs.
- maximize on-time delivery by minimizing earliness and tardiness.
- maximize production efficiency by minimizing manufacturing disruptions.
- maximize product quality by minimizing customer preference violations.

Each of these decisions requires the solution of difficult optimization problems such as the scheduling of jobs to multiple non-identical machines with setups and the cutting stock problem. Problem complexity is compounded by process interactions wherein the optimization of one stage of the production process may negatively impact downstream production or shipping.

During the allocation of orders to mills and machines, order-machine restrictions and downstream consequences on run formation, sequencing, trimming and shipping have to be considered. Since most paper companies are responsible for the transportation of the orders to the customers, it is important for them to produce the orders in the mills close to their final destinations. But an allocation based on transportation cost alone may result in very poor schedules in terms of on-time delivery and trim efficiency. We use two approaches for order allocation: *linear programming* and *dispatch algorithms*. In the linear programming approach we divide the planning horizon into time buckets and use the following decision variables in the formulation:

X(jkb) : quantity (tons) of order *j* produced on machine *k* in bucket *b*

The objective is to minimize a weighted sum of transportation cost with earliness and tardiness penalties, subject to order-machine restrictions and machine capacities in each time bucket. The solution of the linear program may result in an allocation in which orders are split across multiple mills and machines. In most cases order splitting is not desirable. We apply several heuristics for "fixing" such allocations, in which we merge pieces of an order and allocate the whole order to one machine based on some priority rules. The second approach we use for order allocation and initial sequencing is to allocate orders to the machines one at a time, i.e. *dispatch*ing. We use several dispatch algorithms following different rules and preferences, based on a (weighted) combination of order and machine properties, such as due date, processing time, tardiness penalty, setup time and transportation cost from machine location to order destination. The order allocation and initial sequencing problem we consider at this stage generalizes some other machine scheduling problems studied earlier in the literature (Ho and Chang, 1991; Lee and Pinedo, 1997). It can be characterized as scheduling jobs on parallel non-identical machines, subject to job-machine restrictions, sequence dependent setups, batch size preferences, job-machine assignment costs and 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).

Once an order-machine allocation is given, the orders on each machine have to be sequenced. The goal is to produce the orders on time, to have smooth transitions from one product to another and to obtain good trim efficiency from the resulting runs. The sequencing problem at this stage falls into the category of single machine sequencing with sequence dependent setups, and is known to be NP hard (Lee et.al. 1997, Pinedo 1995). Given a machine and a set of orders allocated to it, we iteratively select the job with the highest PRIORITY, from the set of remaining jobs, and schedule it as the next job on that machine. To synthesize the PRIORITY criterion we use several different combinations of due dates, setup times, diameter change penalties and so on, thereby generating multiple alternative sequences.

The main objective in trimming is to minimize the *trim loss,* while producing the rolls according to customer order specifications and considering other manufacturing objectives such as on-time delivery. The reels are cut into rolls according to *pattern*s. A pattern is simply a combination of different roll widths (for a given diameter). 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). To generate a trim solution for a run, we first generate (randomly) a large number of feasible cutting patterns. Then we select a subset of these patterns and decide how many of each pattern to use to create a "good" trim solution. We use several selection policies based on linear and integer programming formulations. Finally, we allocate rolls to orders so as to minimize percentage deviations from ordered amounts.

So far we have discussed methods which generate (partial) solutions from scratch. These methods are encapsulated as construction agents in the A-Team. We also have several improvement methods, which take the existing schedules and improve them in several different dimensions. For example they move

- a single order to improve tardiness,
- a run next to another run of the same product and merge the two to decrease the number of short runs,
- a subset of orders to a different run to improve trim efficiency,
- orders/runs to improve transportation cost or machine load balance,
- orders/runs to improve any combination of the measures above.

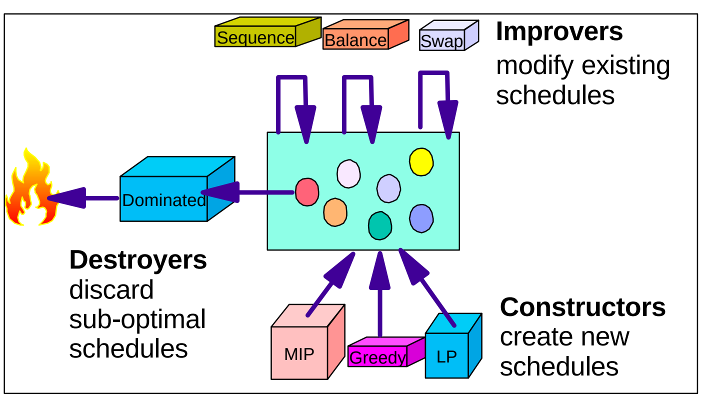
The orders can be moved to a different run on the same machine or on a different machine. The goal of any move may be to improve just a single objective, such as tardiness, or a combination of objectives such as minimizing transportation cost and trim loss.

To facilitate the cooperation between these multiple problem solving methods, we use the Asynchronous Team (A-Team) framework. We discuss this framework next.

3. A-Team Framework

In our decision support system, we use an agent architecture called *A-Team* (Talukdar et.al. 1993). In this architecture, multiple problem solving methods (agents) cooperate through a population of candidate solutions (Figure 1). The outcome is the non-dominated set of solutions from the population. Since the agents share access to the population, an A-Team is like a blackboard system but without a central controller. A-Teams also have certain characteristics of genetic algorithms in that a population of solutions evolve over time. However, unlike genetic algorithms, the mechanisms for altering individual solutions may be highly directed by using domain specific knowledge, rather than depending upon random mutation or crossover.

Figure 1. The Essential Features of an A-Team



In an A-Team, there are three types of agents which create and modify solutions:

1. *Constructors* that create initial solutions.

2. *Improvers* that take existing solutions and modify them to produce new (improved) solutions.
3. *Destroyers* that keep the size of the population of solutions in check by deleting bad or redundant schedules.

The A-Team architecture does not define the content of the agents, but only their possible roles. This gives us complete freedom to use a broad range of algorithms encapsulated as agents. Each agent is independent and can decide for *when* to work and *what* to work on.

4. Conclusion

We have developed a system for enterprise-wide scheduling of paper manufacturing and distribution. Our system has been successfully deployed at 14 paper mills in North America. Computer World reports that within one year after starting to use our system, Madison Paper Industries saved $2 to $3 million in trim and transportation costs at its one 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. Our decision support system is thus empowering schedulers with new tools for managing the complexities associated with scheduling the operations of an entire enterprise while providing meaningful insights into the inherent tradeoffs associated with competing objectives.

References

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.

Biermann C. 1993. Essentials of Pulping and Papermaking, San Diego Academic Press.

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.

Lee Y.H., Bhaskaran K., 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.

Pinedo M. 1995. Scheduling : Theory, Algorithms and Systems, Prentice Hall.

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

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