Multi Machine Scheduling: An Agent-based Approach

Rama Akkiraju   **Pinar Keskinocak**
Sesh Murthy   **Frederick Wu**

IBM T.J. Watson Research Center
Route 134 and Taconic
Yorktown Heights, NY 10598
{ *akkiraju|pinar|murthy|fwu*}@watson.ibm.com

Abstract

Scheduling of multiple parallel machines in the face of sequence dependent setups and downstream considerations is a hard problem. No single efficient algorithm is guaranteed to produce optimal results. We describe a solution for an instance of this problem, in the domain of paper manufacturing. The problem has additional job machine restrictions and fixed costs of assigning jobs to machines. We consider multiple objectives such as minimizing (weighted) tardiness, minimizing job-machine assignment costs. We solve the problem using a simple agent architecture called the Asynchronous team (A-team), in which agents cooperate by exchanging results. We have built agents each of which encapsulates a different problem solving strategy for solving the multi-machine scheduling problem. The A-team framework enables the agents to cooperate to produce better results than those of any individual agent. In this paper we define the problem, describe the individual agents, and show with experimental results that the A-team produces very good results compared to schedulers alone.

Introduction

Effective production scheduling is at the heart of an efficient manufacturing process, and can result in improved on-time delivery of products, reduced inventory costs, increased productivity, and fewer setups. However, scheduling is a very complex task due to the need to optimize multiple competing objectives such as customer satisfaction, production efficiency, and profitability. In this paper, we present an effective way of addressing the multi-machine scheduling problem for non-identical, parallel machines subject to sequence dependent setups, job-machine restrictions, batch size preferences, and downstream scheduling consequences, considering multiple objectives simultaneously. To the best of our knowledge, this problem (especially with multiple objectives) has not been considered earlier. We developed a novel application of an agent-based architecture, called *A-Teams,* to solve this complex multi-criteria optimization problem.

We consider a set of jobs that must be run and there is a set of machines available to execute the jobs. Jobs may be classified into groups of similar job types. Each machine can run only one job type at a time, and each machine may have some unique characteristics, such as production rate. Each job has a due date, on which production should have been completed, and certain jobs may be restricted to a subset of the available machines. When a machine changes from one job type to another, there may be some setup time which depends on the similarity of the job types. To avoid frequent setups, schedulers[1] form batches of jobs of the same type to be run on a machine in sequence. Machine characteristics may also dictate batch size preferences if, for example, very small batches result in higher defect rates. When the machines are not located in the same facility, the cost of transporting the product to its destination is machine-dependent, which implies a job-machine assignment cost. In many applications, there are downstream consequences of scheduling that strongly influence the goodness of a schedule. In such complex scheduling problems, performance measures to be minimized include: tardiness, setup cost and time, inventory cost, job-machine assignment cost, batch size violations, and downstream bottlenecks or inefficiencies.

Armed with an understanding of the realities of manufacturing scheduling, one can see that a solution that deals only with a subset of the objectives and constraints is of limited usefulness. On the other hand, a monolithic solution that considers all the aspects of scheduling for a given domain would be impractical and unwieldy. What is needed is a modular architecture in which each component can be concerned with a subset of the objectives and constraints, and a framework that allows the components to work together to solve the entire problem. In our work, we use an agent architecture called Asynchronous Teams (A-Teams) to address·this multi-criteria optimization problem of multi-machine scheduling. A-Teams is a simple framework wherein multiple problem solving methods

---

[1] In this paper, we use the term 'scheduler' to refer to the human scheduler.

(agents) cooperate with each other by exchanging results. In the next section, we briefly discuss this agent architecture and our solution approach in detail.

Several related problems have been extensively studied in the literature (Pinedo 1995). Scheduling jobs on a single machine to minimize (weighted) tardiness is studied in (Abdul-Razaq et. al. 1990), (Koulamas 1994), (Panwalkar and Iskander 1977), (Panwalkar, Smith and Koulmas 1993), (Potts and Van Vassenhove 1991), (Russell and Holsenback 1997). The extension to multiple machines is considered in (Arkin and Roundy 1989), (Barnes and Brennan 1977), (Ho and Chang 1991). Scheduling subject to sequence dependent setup times is considered in (Lee, Bhaskaran and Pinedo 1997) for single machine; in (Lee and Pinedo 1997) and (Clements et.al. 1997) for parallel identical machines. Du and Leung (1990) proved that minimizing the total tardiness on one machine is NP-hard. Further complexity results on machine scheduling problems can be found in (Lenstra and Rinnoy Kan 1997). Most of these papers consider only one scheduling objective, e.g. minimizing total tardiness. To the best of our knowledge, there are no models that address the complex problem of job scheduling on parallel non-identical machines with sequence dependent setup times and job-machine restrictions, considering multiple objectives (such as the total weighted tardiness and job-machine assignment cost). In many manufacturing environments these types of restrictions and different objectives are important and are part of the scheduling problem the planners and schedulers face every day.

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, Smith and Sadeh 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.

As an example of the problem studied in this paper, we consider the enterprise wide scheduling problem in paper manufacturing, which was one of the prime motivations for this study. A typical large paper manufacturing enterprise has a number of paper mills in various locations and one or more paper machines in each mill. Each paper machine is capable of producing some subset of the company's products (job-machine restrictions) at different production rates. Paper production is a continuous process in which a machine can make only one type, or "grade" of paper at a time (Bierman 1993). When the grade 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" depends on the composition of the grades before and after the transition; transitions between similar grades are shorter than transitions between very different grades (sequence-dependent setup times). The transition times between grades can also be machine dependent. Often there is a manufacturing policy that bounds the length of a production batch, or "run." A lower bound on run length is typically set because very short runs are likely to produce paper with quality problems.

Each customer of the paper company orders a number of rolls of a grade of paper of specified dimensions to be delivered on the due date to a specified location. The paper manufacturer is responsible for the freight cost from its mill to the customer's location, which can constitute as much as 15% of its total costs. Minimizing tardiness and transportation costs (job-machine assignment cost) are important objectives in scheduling paper production.

Scheduling of production on paper machines has major downstream implications that must be taken into account. One of the key downstream processes is known as "trimming." The output of a paper machine is large rolls of paper of fixed width (on modern machines this may be as large as 10 meters). Trimming is the process of cutting the large rolls into smaller rolls having widths ordered by the customers. A production run usually consists of many orders for different widths. Since it is important to utilize the full width of the reel efficiently, this is an example of the cutting stock problem. The best achievable trim efficiency is strongly dependent on the mix of order widths and quantities in the production run. Maximizing trim efficiency is an important scheduling objective that is determined by a downstream process.

Solution Approach

A-Team Agent Architecture

An Asynchronous Team or A-Team (Talukdar, Souza, Murthy 1993), (Murthy 1992) is an agent based architecture that is well suited for multi-objective optimization problems (Souza 1993). In this architecture, multiple problem solving methods (agents) cooperate with each other in evolving a population of solutions towards what is called a Pareto-optimal frontier. The outcome is the non-dominated set of solutions from the population.

Agents communicate through a population of candidate solutions. There are three types of agents which create and modify this population. They are:

1. *Constructors* that create initial solutions.
2. *Improvers* that take existing solutions, and modify them to produce a new solution. Some of the solutions they create may not be better than existing solutions, but are valuable because they serve to explore the solution space and may discover a path to a good solution.
3. *Destroyers* that keep the size of the population of solutions in check by deleting bad or redundant schedules.

Figure 1 shows the essential features of the A-Team architecture. 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 make its own decisions of *when* to work, *how* often to work, and *what* to work on.

In an A-Team, agents can cooperate since one agent can work on the output of another. In general, agents may take as input any number of existing solutions and produce as output any number of solutions. For example, in the job allocation and sequencing problem, a solution may consist of a set of machine schedules, each schedule containing a particular sequence of jobs. Imagine two agents A and B. Agent A moves a job from one machine to another in order to balance the load on the machines but might have delayed the production of the job, while agent B moves a job ahead of the time on to another machine to reduce it tardiness. It is thus possible to achieve cooperation between agents in a

way that reveals potential tradeoffs between cost and on-time delivery.

In order for an A-Team to achieve a balanced population of solutions that approach the Pareto-optimal frontier, it is important that the agents be well balanced as a collection. No single agent need have the ability to optimize or even improve a solution with regard to *all* the objectives. However, the collection should have at least one agent which can address each of the objectives. This provides assurance that any damage done to a solution by an agent with regard to one objective can be repaired or more than overcome by another, possibly on the path toward a solution that is improved in all respects.

In addition to enhanced optimization performance, A-Teams also provide an intrinsically modular, scalable and fault-tolerant computing environment. If a new algorithm is discovered for solving a particular problem, its capability can be readily assimilated without requiring fundamental changes to the underlying software.

Decision Support Using A-Teams

In any complex scheduling problem, there may be considerations and constraints that are not known to the scheduling system. The system may not know of them because they are highly dynamic, or because they exist only in the minds of the schedulers. For example, a customer may have expressed some willingness to take part of an order late as long as one truckload is ready on time. Because such contingencies arise very often, a scheduling system should be designed to allow the schedulers to interact with the system and to modify schedules easily. If the system meets these criteria, then it is truly a decision-support system, and not a decision-making system.
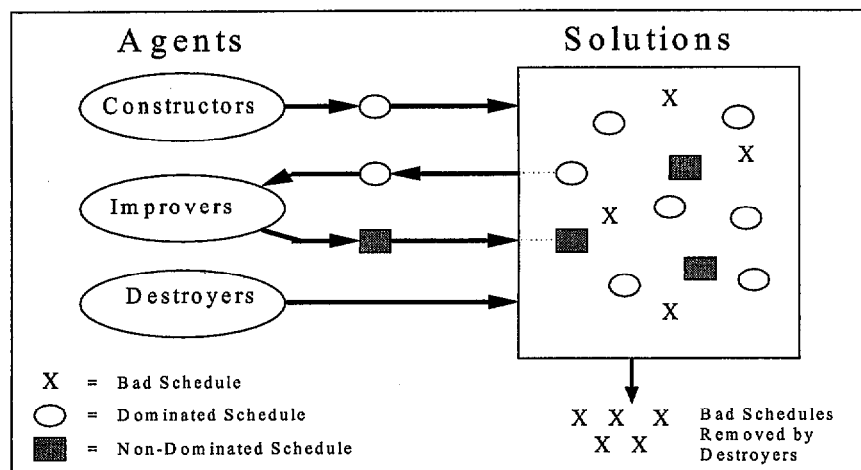

Figure 1. The Essential Features of the A-Team architecture employed by our scheduling system

completion time of job j; tardiness of job j; the absolute difference between the due date of job j and its completion time on machine k; slack of job j; the weighted combination of any of these items with the transition time.

Improvement Heuristics: Improvement agents take the existing schedules from the population and improve them in several different dimensions. For example they move
- a single job in order to improve tardiness,
- batches in order to decrease the number of small batches (if we move a batch next to a batch of the same type, i.e. if no setup is required in between, those two batches can be combined into a single larger batch),
- a subset of jobs in a batch to improve the solution of a downstream problem (e.g. moving a set of jobs to a different run to improve trim efficiency),
- jobs/batches to improve transportation cost or machine load balance,
- jobs/batches to improve any combination of the measures above.

The jobs can be moved to a new position on the same machine, or on a different machine. The goal of any exchange may be to improve a single objective only, such as tardiness, or a combination of objectives such as minimizing transportation cost and load balancing on the machines. Each agent selectively picks a solution to work on based on certain criteria. For instance, a tardiness improvement agent picks a schedule from the population that needs improvement in tardiness. A load balancing agent picks a schedule whose load is unevenly balanced and attempts to fix it and so on.

Since multiple agents work on each others' results, these agents together attain results that may not be achieved by any single agent on its own. These iterative refinements to the set of schedules in the population drive them towards the optimal frontier in the search space.

Experimental Results

We have applied our model to a real-world scheduling problem in paper industry, with multiple mills and machines. The problem consists of more than 5000 orders (jobs) to be scheduled on the 9 machines which are part of 4 different mills. The given set of orders constitute about 8 weeks of production. Each order is for one of the 15 grades of paper that the company produces. Each of these machines has a different production capacity and can only produce certain grades of paper. The geographical location of a particular machine plays a significant role in allocating a job to a machine as it has significant impact on the transportation cost of the schedule. There is a setup time to switch from one grade of paper to another on each paper machine and smooth cycles of grades is desired to reduce the setup times. The goal is to allocate the orders to the machines and sequence them for production such that:
1) The overall transportation cost is minimized.
2) The tardiness and earliness of the orders is minimized.
3) The total setup time is minimized.
4) The number of batches (grade runs) that violate the minimum batch size requirements is minimized.
5) The composition of the orders within each batch does not negatively affect the downstream processes. (Each grade run should be able to trim well with minimum possible waste of paper.)
6) The load is evenly balanced on all the machines.

In this implementation, the measures we used for evaluating the solutions are the following: tardiness and earliness (in ton-days), number of orders late and early, transportation cost (in dollars), number of run size violations (runs longer or shorter than the desired bounds) and the trim efficiency (paper lost due to trimming as a percentage of total production). (It is to be noted that these evaluation metrics are customer dependent and can change from one customer to another. We obtain this information during the design study and incorporate them into the system during the benchmarking process.) These measures are presented in the first 6 columns of Table 1 for each of the 10 non-dominated solutions generated by the system.

We used the A-Team framework to enable the cooperation among various problem solving methods to address this multi-machine scheduling problem. Our A-Team consisted of 15 constructors and 5 improvers. Each agent (constructor or an improver) is an embodiment of the algorithms described in the earlier sections (run with various parameter settings). For an A-Team invocation of 1 1/2 hours of CPU time on a single processor IBM RS/6000 Model 43P (64M of memory), the scheduling system generated 10 non-dominated solutions. (The termination criterion is set by the user by specifying the amount of time to run the A-Team.) As the A-Team approach aims toward monotonic improvement of the solutions, the longer we run the system, the better the results would be. However, within the system we provide several configuration settings for the users to choose to from. Table 1, given below, is a summary of the solutions.

Schedule	Tardiness (ton-days)	Earliness (ton-days)	# of Orders Late	# of Orders Early	Transp. Cost ($)	# of Run Size Violations	Trim Efficiency %	Total Production Tons	Trim Loss Tons
01	464,516	216,810	621	537	$6,503,788	11	97.682	239,892	5,693
02	464,131	159,592	644	436	$6,503,890	9	97.751	240,002	5,522
03	659,086	1,231,163	882	1863	$6,500,465	2	98.523	240,094	3,600
04	460,285	223,339	616	571	$6,500,465	13	97.682	239,911	5,693
05	465,734	903,938	793	1487	$6,507,095	5	98.26	243,433	4,310
06	674,495	1,079,403	909	1737	$6,505,312	2	98.511	239,975	3,628
07	465,780	200,199	587	548	$6,505,511	10	97.741	243,260	5,623
08	649,871	1,162,401	851	1859	$6,505,511	2	98.535	243,220	3,617
09	656,482	1,094,900	856	1770	$6,510,074	2	98.556	243,277	3,565
10	560,422	251,897	717	650	$6,511,978	9	98.862	243,204	2,801
100	613,015	176,012	Not given	Not given	$6,565,750	Not given	98.84	243,405	2,967

Table 1. A summary of the experimental results obtained by the application of A-Teams to an instance of multi-machine scheduling problem in paper domain.

Analysis of Results

Table 1 shows the evaluation of each solution according to the above mentioned set of objectives. For comparison, we also show the evaluation of the actual schedule used by our client paper company (solution # 100). The client's schedule was created manually, based on the experience and intelligence of the scheduling team. The schedulers have numerous rules of thumb and guidelines which they apply to all aspects of scheduling, from allocation of orders to mills and machines through run formation and sequencing. However, the process of scheduling by hand is so time consuming that it is not practical for the schedulers to explore a large number of alternate ways to schedule the orders; consequently they may not be considering some significantly better scheduling decisions.

As can be seen from the table, the solutions created by the scheduling system reveal tradeoffs among the multiple objectives such as transportation cost vs. order tardiness and trim efficiency vs. order tardiness. For example, solutions 03 and 04 have the same transportation cost, which suggests that they have almost the same allocation of orders to the mills. However, they differ significantly in their tardiness and trim efficiencies. Solution 03 sacrificed order tardiness for better trim while solution 04 sacrificed trim for better on-time order delivery. Comparison of a solution, such as solution 10, with the client's actual schedule showed that the scheduling system could provide significant reductions in costs (e.g. 6% savings in annual transportation costs) and improvements in customer satisfaction (reduced tardiness). Since our scheduling system is designed as *a decision support* system, the schedulers can modify the solutions

provided by the system and improve them further by using their domain expertise.

Conclusions

In this paper we described an effective solution approach for scheduling jobs on multiple non-identical machines subject to sequence-dependent setups and job-machine restrictions with multiple objectives. The problem is of interest both to the academic community (it generalizes the problems studied earlier) and to scheduling practitioners (it captures additional constraints and objectives that exist in manufacturing environments). To the best of our knowledge, this is the first attempt to solve this general problem.

We used the A-Team solution architecture, encapsulating multiple solution strategies as agents that cooperate with each other and the schedulers. Experimental results on a large sample of data from a paper manufacturer demonstrate the excellent performance of our approach.

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.

Arkin E.M., Roundy R.O. 1989. Weighted-tardiness Scheduling on Parallel Machines with Proportional Weights. *Operations Research*, 39, 64-81.

Barnes J.W., Brennan J.J. 1977. An Improved Algorithm for Scheduling Jobs on Identical Machines. *AIIE Transactions*, 9, 23-31.

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.

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

Clements D.P., Crawford J.M., Joslin D.E., Nemhauser G.L., Puttlitz M.E., Savelsbergh M.W.P. 1997. Heuristic Optimization: A hybrid AI/OR approach. Workshop on Industrial Constraint-Directed Scheduling.

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

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

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

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., 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.

Lenstra J. K., Rinnoy Kan, A. H. G, and Brucker P. 1977. Complexity of Machine Scheduling Problems. *Annals of Discrete Mathematics*, 1, 343-362.

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.

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

Murthy S., Rachlin J., Akkiraju R., and Wu F. 1997. Agent-based Cooperative Scheduling. Submitted to the Fifteenth National Conference on Artificial Intelligence (AAAI-98), Madison WI, July 1998.

Panwalkar S. S., Iskander Wafik 1977. A Survey of Scheduling Rules. *Operations Research*, Vol25, No1, 45-62.

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.

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

Potts C.N., Van Wassenhove L.N. 1991. Single Machine Tardiness Sequencing Heuristics. *IIE Transactions*, 23, 346-354.

Russell R.M., Holsenback J.E. 1997. Evaluation of Leading Heuristics for the Single Machine Tardiness Problem. *European Journal of Operational Research*, 96, 538-545.

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.

Souza P de. 1993. Asynchronous Organizations for Multi-Algorithm Problems, Ph. D. Dissertation, Dept. of Computer Science, Carnegie Mellon University.

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., Souza P.de, and Murthy S. 1993. Organization for Computer-Based Agents, *Engineering Intelligent Systems*, 1:2