Agent-Based Cooperative Scheduling

Seshahayee Murthy
John Rachlin

Rama Akkiraju
Frederick Wu

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

Abstract

Cooperative scheduling is concerned with integrating multiple problem-solving perspectives or objectives when generating solutions to a scheduling problem. Cooperation may involve interactions between humans, between scheduling agents, or between humans and agents. The purpose of this paper is to focus on the later two types of cooperation in which a team of software-based asynchronous agents play a major role in identifying candidate scheduling alternatives. This so-called A-Team architecture is at the heart of a set of decision-support tools that employ cooperative scheduling techniques for scheduling the operations of real-world paper manufacturing facilities. The IBM Paper Mill Scheduling System was developed at the IBM T.J. Watson Research Center over the past couple of years and is currently being sold world-wide.

Introduction

The IBM Paper Mill Scheduling System is an integrated suite of programs for scheduling the operations of a paper mill from manufacturing to product delivery. Scheduling is at the heart of manufacturing process. It impacts profitability, customer satisfaction, and whether or not the manufacturing line runs smoothly. Because scheduling a paper mill is an enormously complex task, involving thousands of constraints, it requires substantial effort to define solutions that are both feasible and efficient. The IBM Paper Mill Scheduling System consists of a patent pending decision-support framework that uses multiple asynchronous agents cooperating to produce an evolving population of candidate scheduling solutions. By evaluating many scheduling alternatives, the user gains important insights into the tradeoffs between multiple competing objectives.

The human scheduler cooperates with other agents by making manual modifications to existing solutions. By allowing for manual intervention, the experienced human scheduler is able to interject special considerations or contingencies not necessarily captured by the objectives of the scheduling system. The human scheduler can try to improve upon schedules created by agents and *vice versa.*

These forms of cooperation provide a very powerful approach to multiobjective decision-support and optimization in complex manufacturing environments. One alternative approach to multiobjective optimization involves assigning weights to different objectives. The resulting objective function or "fitness" function is then minimized or maximized depending upon how the objectives are defined. This approach, although common, has serious limitations: First, these weights are difficult to define particularly for inherently qualitative objectives such as product quality or customer satisfaction. Secondly, constant coefficients imply a linear relationship between objectives that does not necessarily exist. Finally, as economic conditions change, the relative importance of different objectives may change, requiring that users periodically change the values of these weights. However, in practice this is inconvenient and may be untenable. In short the special relationships between the mill, its customers, and its distributors may dictate policies that cannot be encompassed by a single weighting factor.

Our agent-based scheduling system addresses many of the limitations of existing job-shop scheduling systems identified by (McKay Safayeni & Buzacott, 1988) by providing a decision-support framework for scheduling that naturally accounts for conflicting and changing goals. By providing mechanisms for manual intervention and *evaluating the impacts of such changes,* the scheduler can react sensibly to near-term uncertainties and contribute his or her expert knowledge to the system. In this sense, the human scheduler becomes like one of the agents of our system, participating on an equal basis to the formulation of candidate alternatives.

Paper Manufacturing and Scheduling

The manufacture of papers of all types is a multi-billion dollar business worldwide. There are hundreds of different kinds of paper used for everything from printing to packaging and different manufacturing processes tend to be favored for a specific type of paper product. However, almost all paper manufacturing processes share several major elements, or subprocesses. In this section we will briefly describe a generic paper manufacturing process as background information to motivate the AI solution that is the subject of this paper. Further details can be found in (Biermann 1993).

A paper company may have hundreds of different customers each with their own specific requirements. These customers place orders with specific characteristics, and it is the job of the mill to fill these orders in a way that minimizes the cost of production, but satisfies the requirements of each order. There are many characteristics that define an order. The customer may order whole rolls wound onto a core, or stacks of cut size sheets of a specific dimension (e.g., 8.5" X 11"). Other characteristics include desired grade (quality, composition, tensile strength, etc.) basis weight, dimensions (roll width and diameter, or cut sheet width and length), thickness, total tonnage, possible coating or embossing requirements, packaging, labeling, and delivery requirements (due date and transport mode options).

The manufacture of paper begins with the production of pulp from a variety of sources, including logs, wood chips, recycled fiber, etc. The pulp is fed into a paper machine along with a number of other ingredients that define the "recipe" for producing a particular grade and basis weight. Thus, at any given time, the mill produces a single product on each of its machines. Because it takes time to change production from one grade to the next, the mill will try to transition between similar products so as to minimize setup time and waste. A paper machine outputs large reels of paper. The width of the sheet (called the deckle) is limited by the size of the paper machine. The scheduler must define a sequence of runs for each machine, as well as the orders to be fulfilled in each run.

Reels are then processed through a *winder* that unrolls the reel, and uses slitters (knives) to cut the deckle into narrower lengths of paper. These narrower strips are rewound to form *rolls.* This subprocess is called *trimming* and exemplifies the classic cutting-stock problem (Gilmore & Gomory 1961), where the goal here is to maximize utilization of the deckle while satisfying the requirements of each order to within specified tolerances.

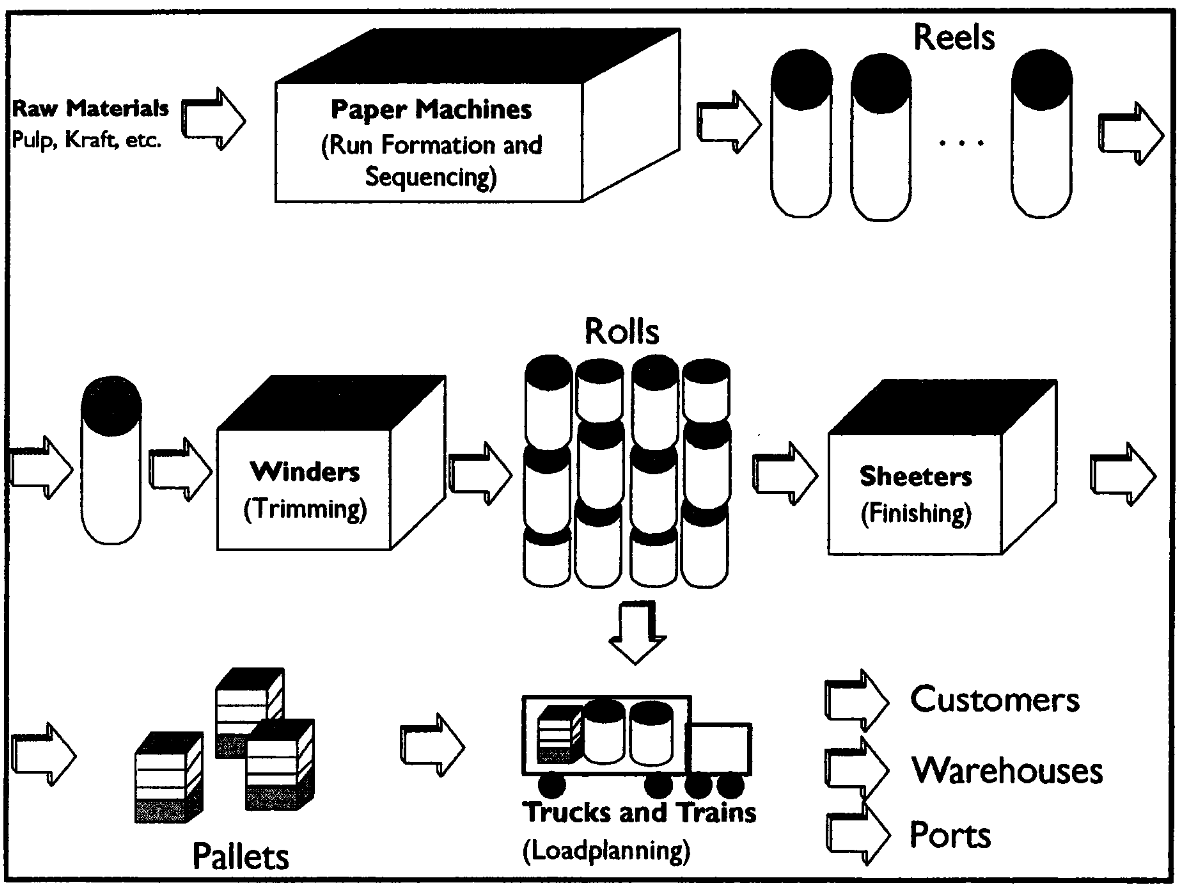

Figure 1. An overview of the paper manufacturing process.

As each set of rolls is completed, the rolls are wrapped for shipping or temporary storage.

For cut-sheet paper products, the next step is to cut rolls into sheets of the desired size. This is called *sheeting* or *finishing*. During this process, rolls are loaded onto a sheeter, which unwinds the roll and slices the paper in two dimensions to the finished size. Standard-sized sheets are then typically wrapped into bundles to form a ream. These reams are then placed into cardboard cartons which, in turn, are stacked onto skids in preparation for shipping. The scheduler allocates orders to each sheeter and then must define a production sequence that satisfies due dates while minimizing expensive transition setups.

Finally, the packaged products (either rolls or sheets) are loaded onto vehicles for shipment to the customer or to other distribution centers. Load planning involves selecting the right mode of transportation or mix of modes, allocating loads to carriers, and in some cases pooling loads together into single vehicles in an effort to pack vehicles efficiently (bin packing) while satisfying loading and delivery constraints

In our discussions with dozens of manufacturing companies, we find that certain objectives reappear time and time again like themes in a musical canon. These four basic objectives are:

1. Maximize profitability
2. Maximize on-time delivery
3. Maximize quality and customer satisfaction
4. Minimize manufacturing disruptions

Obviously, these four objectives are interrelated. A late delivery impacts customer satisfaction and possibly long-term profitability while *avoiding* a late delivery can create severe manufacturing problems. Each of these four objectives may have many contributing factors. The decision-support/optimization framework provided by the IBM paper mill scheduling system allows the user to modify the definition of these objectives. Our discussion is independent of the way in which particular objectives are defined.

Better solutions through agent-agent cooperation

There has been a great deal of work on applying AI to the problem of scheduling. (Mortan & Pentico 1993) provides a broad overview of scheduling issues and heuristic techniques. It also includes extensive references to a variety of scheduling methods including computer simulation, mathematical approaches, simulated annealing, genetic algorithms, beam search, expert systems, neural networks, as well as methods that combine AI with operations research (OR) techniques.

An Asynchronous team or *A-Team* (Murthy 1992, Souza 1993, Talukdar, Souza, & Murthy 1993, Talukdar, Baerentzen, & Souza 1996), is an AI architecture that consists of multiple problem-solving methods (called agents) working together on a common problem. Communication (and cooperation) takes place through a shared population of candidate solutions. Figure 2 shows the essential features of the A-Team architecture. The A-Team architecture employed by the IBM Scheduling System does not represent any single method or heuristic, but is rather an attempt to combine multiple techniques including some of those referenced above by encapsulating individual algorithms as agents. While this paper focuses on the application of A-Teams to paper manufacturing scheduling, A-Team based scheduling systems have also been developed for the steel industry (Lee *et al.* 1996).

An A-Team architecture consists of a population of solutions and three types of agents which create and modify this population. The different agent types are:

1. *Constructors* that create initial solutions.
2. *Improvers* that take existing solutions, and modify them to produce a new solution which is then added to the current population. The original solution is maintained. Technically, improvers are not required to make measurable improvements. They may make random modifications that lead to worse solutions that are nevertheless valuable because they serve to explore the solution space and in so doing may discover a path to a good solution.
3. *Destroyers* keep the size of the population of solutions in check. Their main function is to delete clearly bad or redundant schedules, thus separating the wheat from the chaff. When a destroyer finds a bad schedule, it tosses a weighted coin to determine whether the schedule is actually deleted. Again, we avoid always deleting what we think are bad solutions to account for the possibility that a bad solution may be on the path to large improvements.

Because the agents all have access to the population, an A-Team, in this sense is like a blackboard system. A-Teams also have certain characteristics of a genetic algorithm in that a population of solutions evolves over time. However 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. (Horn, Nafpliotis, & Goldberg 1994) presents a method for finding the Pareto frontier using a genetic algorithm, and also describes related work on multi-objective optimization using GAs.

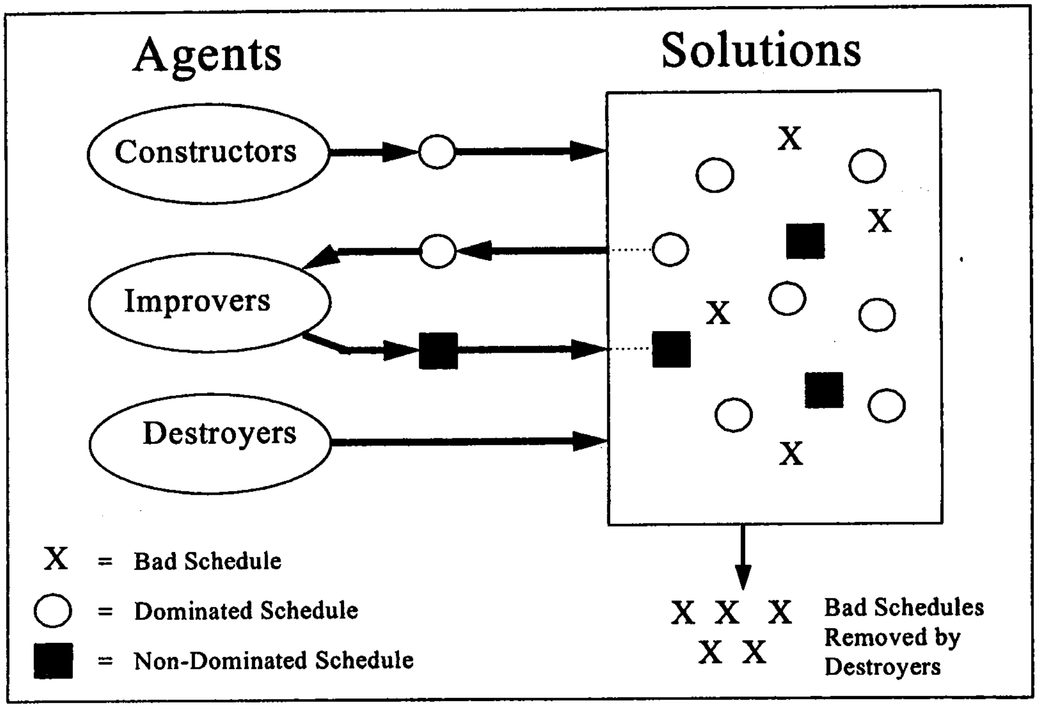

Figure 2. The A-Team Architecture employed by the IBM Paper Mill scheduling system

From a software development point of view, the system is more robust because if a bug appears in one agent, the system will still function. No single agent is critical. (Huberman, Lukose, & Hogg 1997) demonstrates how combining multiple algorithms together produces solutions which are preferable to the solutions generated by individual algorithms operating alone, and offers a measure for the degree of cooperation involved based on statistical correlations of performance. A performance analysis of parallel implementations of asynchronous team algorithms can be found in (Barán, Kaszkurewicz, & Bhaya, 1996).

In an A-Team, cooperation between agents is possible because one agent can work on the output of another. In general, agents may take as input any number of existing solutions as input and produce as output any number of solutions. For example, in transportation planning, a solution may consist of multiple vehicle loading plans, each defining the particular items loaded inside. Imagine two agents A and B. Agent A converts some randomly chosen set of trucks to rail shipments while Agent B does the opposite - it converts railcars to trucks. By themselves, these two agents will produce a very limited set of possibilities. Together, they produce a broader range of choices which combine two modes of transportation. This is useful because shipments by truck tend to be more expansive than rail, though require less time for delivery. It is thus possible to achieve cooperation between just two agents in a way that reveals potential tradeoffs between cost and on-time delivery. In practice, the IBM paper mill scheduling system employs a broad range of mathematical programming techniques as well as some very simple heuristics depending upon the particular problem.

Better decisions through human-agent cooperation

The IBM paper mill scheduling system is an *interactive* decision-support system. Human schedulers knowledgeable about the objectives of manufacturing, customers, and management play an important role in developing a final scheduling solution. Over many years, customer service and sales personnel have come to know and understand the special requirements of their customers, their suppliers, and their distributors. Indeed, it is this attention to the individual needs of each customer that is a driving force behind many changes taking place not only in the paper industry, but in other industry sectors as well. We believe that it is impractical and undesirable to try to capture too many of these individual constraints within the scheduling system itself. Such systems tend to be less efficient, and more brittle.

Thus, agents cooperate to define candidate solutions that address *most* constraints and which are *nearly* satisfactory. It is still necessary for a human scheduler to intervene by modifying existing solutions, and in so doing, interject into the solutions knowledge not necessarily captured by individual agents or evaluation criteria. This approach is not entirely new. For example, (Bertolotti 1992) describes a scheduling system, AERPLAN, which, like the IBM paper mill scheduling system, relies on the interaction of a user to address special case scenarios.

To support effective cooperation between the human and agents, the scheduling system must have an intuitive user-interface allowing the user to manipulate schedules down to the smallest detail. The interface can assist by giving a warning if the user tries to do something potentially invalid (for example, loading a truck beyond its allowed weight limit, or creating trim patterns that exceed the deckle) but it is fundamental to our design philosophy that the *users know best and can do what they want.*

The IBM paper mill scheduling system is a client-server application. Each user has his or her own PC-based graphical user-interface connected either by token-ring or by modem to an RS/6000 workstation running AIX for running the computationally intensive scheduling engine. (Stand-alone trim optimization is also supported under Windows NT.)

Figure 3 shows a portion of the PC interface by which users manipulate particular winder schedules if necessary. On the left is a summary of multiple trim alternatives, each evaluated according to serveral important criteria. The user can view one of the solutions (on the right) and make manual changes if desired. In the above example, our system generated 24 trimming alternatives in about 3 seconds when powered by an IBM RISC System/6000 model 591 with 512 Meg of RAM.

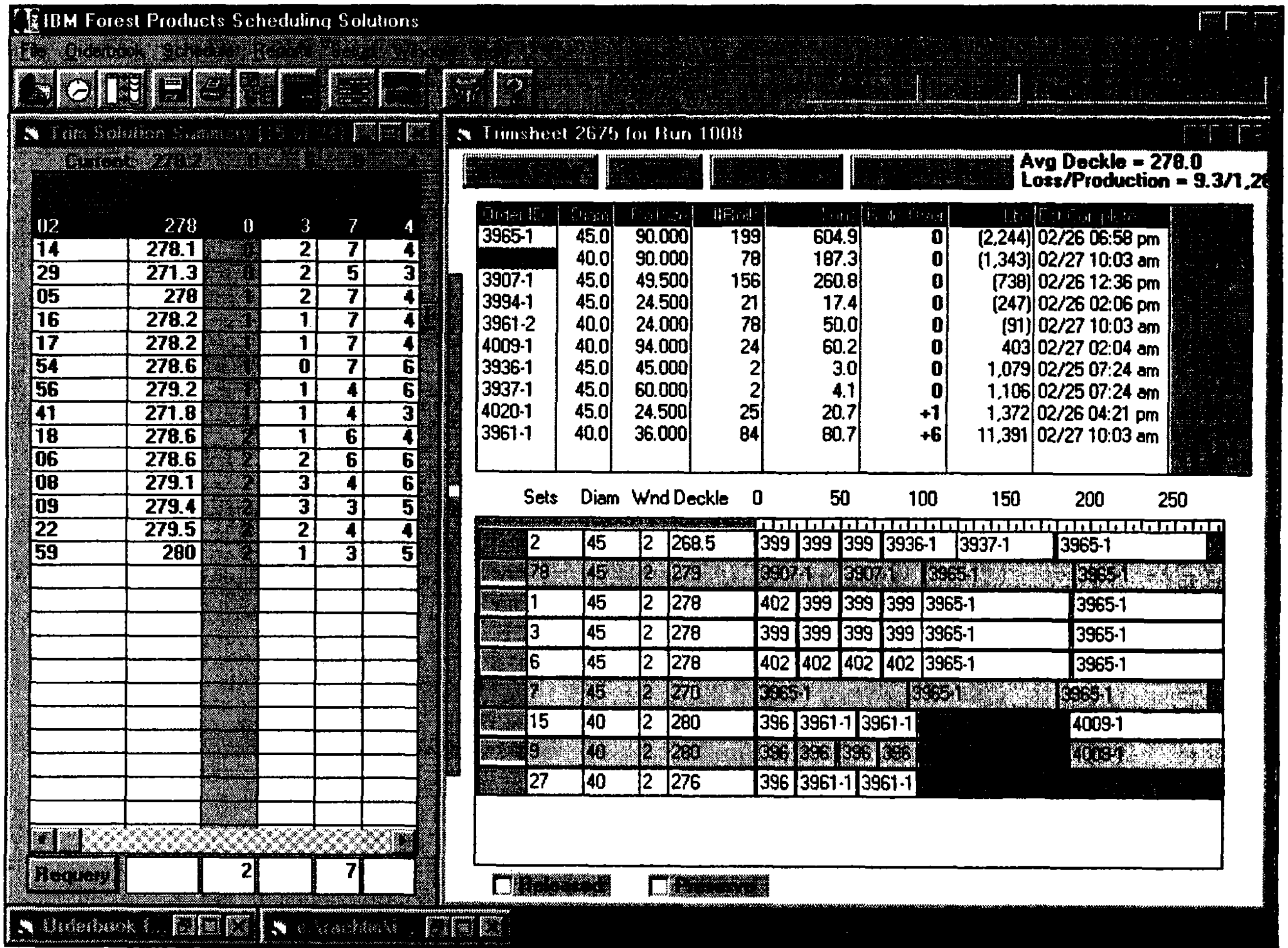

Figure 3. Screens from the graphical user-interface. The GUI enables effective cooperation between the human scheduler and the agents in the A-Team.

By contrast, human trim schedulers may spend hours coming up with a single feasible solution. A-Teams allow humans to consider alternatives they may not have thought of otherwise. This enables a more creative approach to the task of scheduling and empowers schedulers by allowing them to consider objectives of the company as a whole rather than focusing solely upon the goals and objectives of a single subprocess.

The importance of a good interface should not be underestimated as it enables effective cooperation between the human and the agents. Our interaction with the actual users of the system clearly suggests that they view the interface *as the scheduling system itself* without regard for the underlying technology.

Future directions

The IBM paper mill scheduling system demonstrates that A-Teams can be used as a basis for building practical real-world applications that solve complex problems. A-Team technology provides an open architecture by which multiple objectives can be simultaneously addressed, and thus presents a framework for understanding the inherent tradeoffs between competing corporate objectives.

Our group is continuing to explore extensions to the cooperative scheduling theme. We are working to enhance the role of human-human cooperation through workflow mechanisms provided by networking technologies. This is in keeping with the goal of integrated scheduling which seeks to incorporate the knowledge and experience of many diverse points of view. We are also investigating the application of learning techniques that will allow the system to improve its performance by having agents learn to be selective of when they run and what they work on based on the results of past invocations or in response to choices made by the human scheduler.

References

Barán B.; Kaszkurewicz E.; and Bhaya A. 1996. Parallel Asynchronous Team Algorithms: Convergence and Performance Analysis. *IEEE Transactions on Parallel and Distributed Systems* 7(7):677-688.

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

Bertolotti, E., 1992. Interactive Problem Solving for Production Planning. In *Artificial Intelligence Applications in Manufacturing* (Famili, Nau, Kim eds.), Menlo Park:AAAI Press.

Gilmore P.C., and Gomory, G. 1961. A Linear Programming Approach to the Cutting Stock Problem. *Operations Research* 9:849-859.

Horn, J.; Nafpliotis, N.; and Goldberg, D. 1994. A Niched Pareto Genetic Algorithm for Multiobjective Optimization, In *First IEEE Conference on Evolutionary Computation*:82-87.

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

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*.

McKay, K.; Safayeni, F.; and Buzacott, J. 1988. Job-Shop Scheduling Theory: What is Relevant? *Interfaces* **18(4)**: 84-90.

Morton, T., and Pentico D., 1993. *Heuristic Scheduling Systems*. New York:John Wiley and Sons.

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

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

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

Talukdar, S.N.; Baerentzen, L.; Gove, A.; and Souza, P. 1996. Cooperation Schemes for Autonomous Agents. In *Times Assincronos para Problemas Industriais*, Sao Paulo.