

## Multi-agent Based Integration of Scheduling Algorithms

Dr. Bo Zhao, Prof. Dr. Yushun Fan  
Department of Automation, Tsinghua University  
Tsinghua Yuan, Beijing 100084  
P. R. China Name

### ABSTRACT

Up to date more and more research of scheduling use Multi-agent System (MAS) technique. In this paper MAS is used to realize integration of scheduling algorithms. Firstly, Multi-agent Scheduling System (MASS) is divided into two types: Entity-type MASS and Process-type MASS. Some of the researches are introduced. Secondly, the concept of integration of scheduling algorithms is put forward. Thirdly, the models of agents, computing agent and manager, are proposed. Then a Process-type MASS of multi-agent based integration of scheduling algorithms, which compose above two sorts of agents, is built. Finally, we conclude by describing the significance of our research and highlighting future extensions.

**KEY WORDS:** Scheduling, Multi-agent System, Integration of Scheduling Algorithms, Multi-agent Scheduling System

### 1. INTRODUCTION

Algorithm is the key for the scheduling theory. The principle of scheduling is to find an optimal scenario of job allocation or resource distribution with scheduling algorithms. In the past decades many algorithms originated from other fields have been used in scheduling research and therefore formed the so-called scheduling algorithms. These works have enriched scheduling theory. However, any single algorithm that people have used so far are only applicable to a few special environments and can not adapt to dynamic production environment. This makes scheduling algorithms have not exerted all of their power in practical production. It is often the case that the plan formulated by scheduling algorithms is disabled because of some disturbance such as machine failure, unexpected jobs coming into workshop, material shortage. The inconsistency of scheduling theory with the scheduling practice has remained a big issue in manufacturing.

To solve the problems one may think of two solutions:

- 1) Finding an all-purpose scheduling algorithm that is applicable to almost all sorts of scheduling cases.
- 2) Finding a mechanism by which appropriate algorithms of scheduling algorithms library can be called dynamically and integrated rapidly to respond to the change of production environment.

Unfortunately there is no one-fits-all algorithm that meets the requirement of solution 1. The promising and recommendable approach would be the latter. The purpose of this paper is to find such a mechanism named Integration of Scheduling Algorithms (ISA) and use MAS technique to realize it.

MAS (Multi-agent System) technique, which is a branch of distribute artificial intelligence, has been regarded as one of the most promising approaches to solve scheduling problems under dynamic environments and has attracted a lot of attention recently. In this paper, the MAS technique is used to realize dynamic integration of scheduling algorithm. And the solution will ensure either theoretical efficiency or operation robustness.

We may call a scheduling system that uses multi-agent technique as Multi-agent Scheduling System (MASS). By reviewing some important literatures, we find that MASS can be divided into two types:

#### 1) Entity-type MASS

Agents in such MASS map physical entities in real-life systems as jobs and resources (machine, conveyance, storage, etc.). The major feature of such MASS is the reciprocity between resource agents and job agents. Every agent has intention of itself, goal and benefit. They are capable of self-advancement and self-control. They can also be distinguished from environmental information and then take action. Resource agents and job agents, as supplier and customer in market, achieve their maximal benefits and system goals through negotiation or transaction.

Research of Entity-type MASS is very plentiful. Lin et al.[1] used agents to response functions and entities (machine, job, database, etc.) of manufacturing system in their framework. And they used mark-like model to

realize negotiation among agents. Ramos[2] also put forward a scenario that compose of resource agents and job agents. Gomes et al.[3] view a MASS as an three level organization. Agents are signed different roles and functions depending on their position within the structure of the system. Agents of the low level are classified resource agents and job agents. Ouelhadj et al.[4] defined an “actor” architecture where agents is associated with particular functions which are distributed over resource agents and use contact net protocol for dynamic scheduling. Rabelo et al.[5] studied multi-agent based scheduling in virtual enterprise environments on the base of HOLOS scheduling system, which is a framework devoted to derive “instance” of agile scheduling system.

## 2) Process-type MASS

Predominant agents in such MASS are called process agents. They map processes that realize a function [6], a computation [7], an activity [8], etc. Each process agent can only solve part of a problem. Different agents work together by collaboration to achieve system’s goal, as people coming from different fields to a team will do.

Unlike Entity-type MASS that mainly composes of resource agents and job agents, Process-type MASS has no typical architecture. There is much difference among researches of such system by now. Lau[6] defined a MASS for FMS scheduling, which is capable of individual learning and group learning. Agents in the system are scheduling models that have ability of predictive scheduling and making reaction toward environment or other agents. Morikawa et al.[7] use agent maps genetic algorithm in his research of scheduling in process of CIM. The whole process of solving problem is divided into several stages. Each agent responses one stage. They work one by one. One agent gets input from upriver agents and output result to downriver agents. Gary Knotts[8] present a multi-agent scheduling method to solve multimode, resource-constrained project scheduling problem. Agents map activities of project.

Baker[9] reviewed most scheduling algorithms and proved that they can be used into multi-agent heterarchy. So we consider to integrating more scheduling algorithms into one framework to adapt requirement of complicated production environment by defining a process-type MASS. The agents in our architecture map scheduling algorithms.

The rest of the paper is organized as follow. In section 2, we introduce concept of integration of scheduling algorithms. In section 3, we detail the solution of multi-agent based integration of scheduling algorithms. In the last section, we conclude by describing the significance of our research and highlighting future extensions.

## 2. INTEGRATION OF SCHEDULING ALGORITHMS

Different scheduling algorithms have their own feature and using condition or area. As units, their capabilities are limited. But they can be used to solve complex scheduling problem if only they are combined together. Integration of Scheduling Algorithms (ISA) is such a process or mechanism of integrating various scheduling algorithms to generate a single architecture, which process more applicability.

To some extent, some mixed algorithms, which are compose of other algorithms, are particular instances of ISA. But these combination are simple and immovable, and the result, which are also single algorithm, is still limited to some special production environments and we are not interested in it here.

ISA that be defined here is a flexible and intelligent scenario of collaboration among scheduling algorithms. It can dynamically call or joint different algorithms together for different environment under various conditions and constraints. Each algorithm (we call it atomic algorithm) joining the scenario is independent.

There are two cases of ISA. One is that atomic algorithms store in a system are dynamically called to response the change of environments. The other is that scheduling problem is divided into several subproblems. Each subproblem is sloved by one atomic algorithm. The whole problem is solved by dynamic collaboration of several atomic algorithms.

We may design an intelligent system that consists of a rule base, an algorithm base, a reason machine and other components. The rule base stores knowledge of using algorithm. The algorithm base stores all sorts of scheduling algorithms. The system can make estimation in reason machine and select a good algorithm from algorithm base according to rules from rule base. Theoretically it can solve any complex scheduling problem. But its efficiency is hard to ensure because we have too many algorithms and rules of how to use these algorithms. To retrieve rules and algorithms need very long time.

Then, of course, we think of distributed structure. MAS is surely a good architecture to help realizing ISA.

## 3. MULTI-AGENT BASED ISA

A scheduling algorithm is a process of solving scheduling problems. The process needs to keep contact with the environment. Assembled with a rule base, an analysis module, an interface (to contact with outside), and a

reason machine, etc., a scheduling algorithm can be characterized as an intelligent agent. The agent can make decisions based on the response from the environment and take action (computation). We name this agent computing agent. The dynamic integration of scheduling algorithms is the integration of different computing agents under the scheduling of a manager.

### 3.1 DEFINITION OF AGENTS

A whole computation consists of several steps: environment analysis, goal setting, evaluation of computation capability, decision making, computing, output conclusion, etc. We integrated these steps into a general model of computing agent as Fig.1

---

Fig.1

---

Elements of a computing agent are detailed as follow:

#### 1) Algorithm Base

Stores algorithms that belong to the same type, e.g. scheduling rules. Each algorithm can be used inside the agent according to condition of their being used. Also, new algorithms belong to the type can be added in.

In fact, the contents in the base may be information as: ID of an algorithm, Input, etc. It points to a program of an algorithm.

#### 2) Rule Base

Stores knowledge of using algorithms: applicable conditions, capability, efficiency, etc. New rule can be added in at any time.

The rules use the format of 4-vector: (ID, Condition, Capability, Efficiency), thereinto:

- ID: ID of the algorithm;
- Condition: relation between the algorithm with some scheduling models, i.e. if the algorithm can use in one of algorithms.
- Capability: degree of optimization. It is a relative value of an algorithm we select as a standard.
- Efficiency: the time of finishing computation.

#### 3) Analyzer

Analyzes information from the sensor and makes decision of whether or not responding to the information.

Information from sensor is mainly ID of scheduling model. It then be use to retrieve ID of algorithms in Rule Base. Finding a suited ID of an algorithm means agents can response the scheduling model.

#### 4) Reason Machine

If there are several algorithms in the Algorithms Base that are suited with the scheduling model, selects the best one according to capability and efficiency under the support of the rule base.

#### 5) Computing Cell

Finishing computation with selected algorithm.

#### 6) Sensor

Receives information such as jobs, resources and scheduling models from the Manager and responds with bidding or not.

#### 7) Driver

Outputs result.

In order to harmonize computing agents, we need a manager. It's a special agent as shown in Fig.2 and is responsible for following functions:

---

Fig.2

---

- 1) Registers each computing agent with Register Model and stores their properties in Database.
- 2) Searches information from environment through the Sensor and translates it into appropriate scheduling model in the Modeler under the support of the Knowledge Base.
- 3) Communicates with computing agents via Communicator.
- 4) Records and analyzes middle result in Blackboard and then outputs final result via Driver.

We'll present the reciprocity between the manager and the computing agents in 3.2.

### 3.2 SYSTEM ARCHITECTURE

Thousands of scheduling algorithms has been proposed so far. These scheduling algorithms can be classified into several categories. The hierarchy is shown in Fig. 3:

---

Fig.3

---

Of course we can not and need not design agents for each algorithm. But we can do that for each class. Our solution is to joint different classes of computing agents into a MASS to realize dynamic integration of scheduling algorithms. Except for a manager, every node of the system is a computing agent, which provides congener algorithms that store in its algorithm base. A starlike architecture of the system is shown in Fig.4.

---

Fig.4

---

Fig.4 is only one example of multi-agent based integration of scheduling algorithms system. We may choose either two or more computing agents to build a MASS according to the requirements of a real-life production system. And computing agents can be distributed. They work independently and concurrently.

The system works as follow:

- 1) The manager searches information from the environment and translates it into a scheduling model. Then it broadcasts a bid request to all computing agents and send an expression of the scheduling model;
- 2) When receiving bid, computing agents retrieve their own Rule Base. Whether or not there are records concerning with the model means whether or not the agents have capability to solve the problem. According to their capabilities, the computing agents decide to bid or not and then send messages to the manager.
- 3) The manager makes a choice from computing agents who make the bids and send information to the selected agents.
- 4) The selected computing agent accomplishes the task and outputs its result to the manager.
- 5) If the current result does not fulfill system's goal, the manager will invite a new public bid.

Sometimes, one problem can be divided into several subproblems. Then manager will send bids to several computing agents concurrently. And subproblems are solved independently by computing agents.

The computing agent in the above scenario is "fat ". Facing requirement of some cases, We can also design a MASS of "thin" computing agent, i.e. computing agents is relatively simple, e.g. it provide only one algorithm, not a algorithms base. Anyway, the idea of building MASS in this paper is flexible.

#### 4. CONCLUSION

Classical scheduling theory is problem-based. It is difficult to be applied directly to practice since it simplifies the real-life scheduling system, which is usually complex and contains several problems. Whereas, in this paper, we proposed a mechanism named integration of scheduling algorithms and built architecture of process-type MASS to support it. The mechanism is system-based. It can take into account of the complexity of real-life systems and integrate almost all of scheduling algorithms into one system architecture and then use them dynamically to adapt to the change of production environments.

If we look upon the algorithm research of classical scheduling theory as a microcosmic work for simple problems, then the idea described here is the solution to macroscopically applications for complex systems. Further research will be carried out in this aspect as follow:

- 1) Research of MASS need fully utilize results of classical scheduling theory and other production control techniques.
- 2) The functions of both computing agent and manager in this paper are basic and relatively simple. We will expend their functions in next work.
- 3) In general, structure of agent is often complex and need lots of work to design. So we may develop standard model of some types of agent (such as computing agent, resource agent, job agent and etc.) on the base of current research. It will make reusing agent easy.
- 4) As we know, it is unpractical to design directly a universal system for scheduling. But we can dynamically assemble scheduling system for requirements of practical production environment with standard agents which response basic elements of enterprise.

#### REFERENCES

- [1] Grace Yuh-jiun Lin, James J. Solberg. Integrated shop floor control using autonomous Agents. *IIE Transactions, Volume 24*, Number 3, July 1992: 57-71
- [2] Carlos Ramos. An Architecture and a negotiation protocol for the dynamic scheduling of manufacturing systems. *Proceedings - IEEE International Conference on Robotics and Automation* pt 4 May 8-13 1994, Sponsored by IEEE: 3161-3166
- [3] Carla P. Gomes, Austin Tate, Lyn Thomas. Distributed scheduling framework. *Proceedings of the International Conference on Tools with Artificial Intelligence* Nov 6-9 1994: 49~55

Bo Zhao, Yushun Fan, Proceedings of the IASTED International Conference-Intelligent Systems and Control, Florida, USA, 2001.11,pp.55-59

[4] D.Ouelhadj, C.Hanachi, B.Bouzouia. Multi-agent system for dynamic scheduling and control in manufacturing cells. *IEEE International Conference on Robotics and Automation* v 3 May 16-20 1998, Sponsored by IEEE:2128-2133

[5] R.J. Rabelo, L.M.Camarinha-Matos. Multi-Agent-Based agile scheduling. *Robotics and Autonomous Systems*(1999)27,15-28

[6] Rachel Lau, Joel Favrel. Intelligent scheduling agent for distributed decision-making. *Proceedings of the 35<sup>th</sup> Conference on Decision and Control*, Kobe, Japan, December, 1996, Sponsored by IEEE:3849-3850

[7] Koji Morikawa, Takeshi Furuhashi, Yoshiki Uchikawa. Evolution of CIM system with genetic

algorithm. *IEEE Conference on Evolutionary Computation - Proceedings* v/2 Jun 27-Jun 29, 1994, Sponsored by IEEE: 746-749

[8] Gary Knotts, Moshe Dror, Bruce C. Hartman. Agent-based project scheduling. *IIE Transactions* (2000) 32, 387-401

[9] Albert D. Baker. Survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems* v 17 n 4 1998 SME:297-320

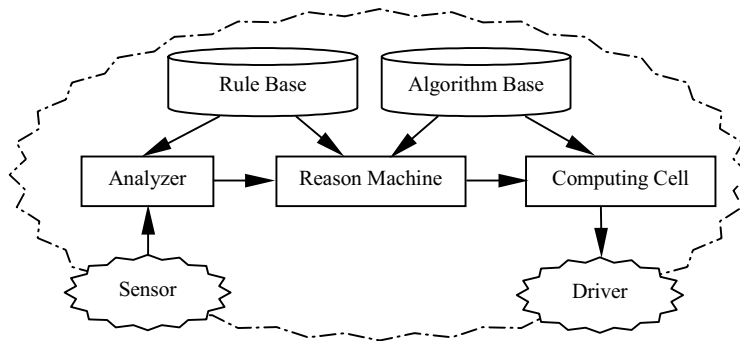


Fig.1 Architecture of computing agent

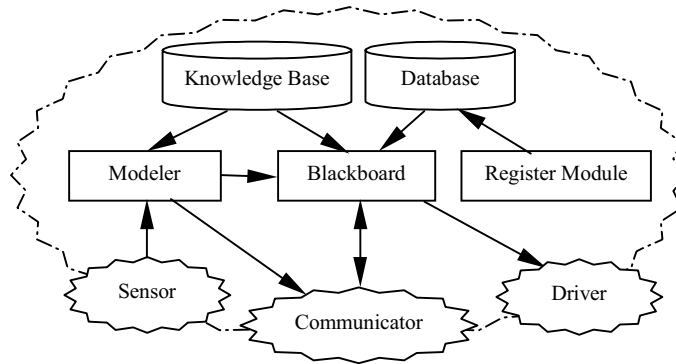


Fig.2 Architecture of manager

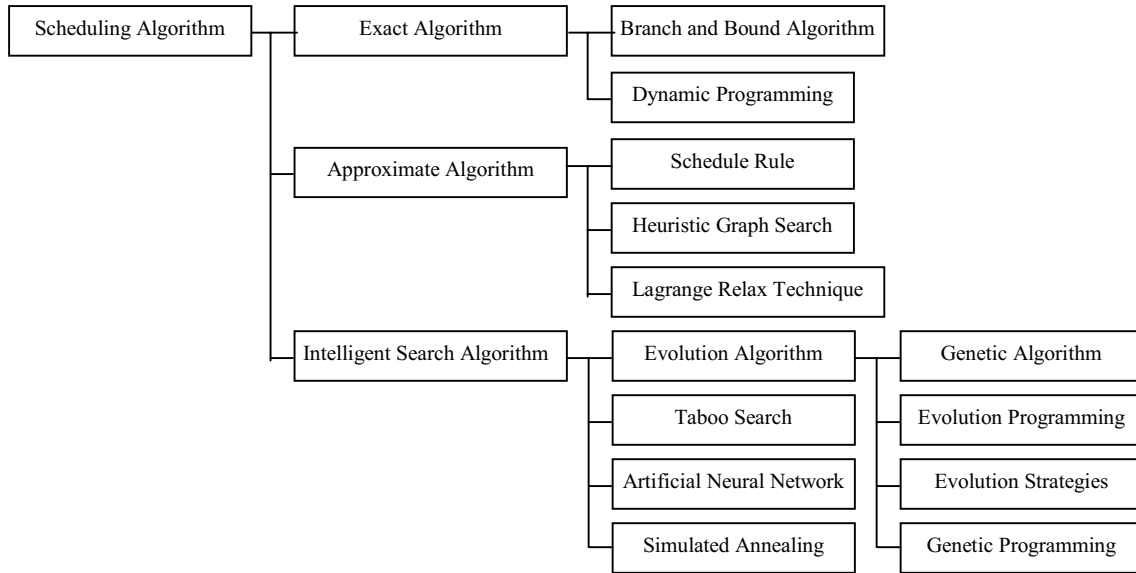


Fig. 3 Partition of scheduling algorithm

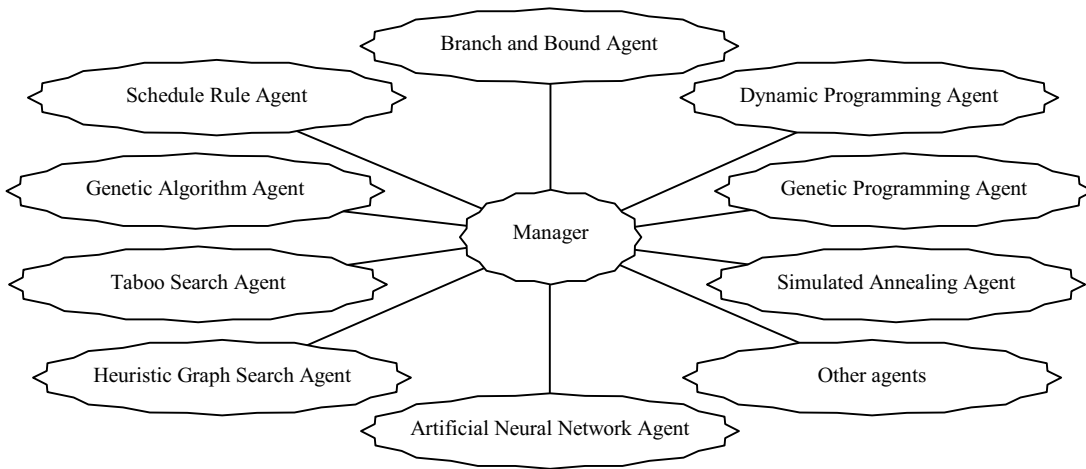


Fig.4 Architecture of Multi-agent-base integration of scheduling algorithms system