On Collaborative Planning for Time-constrained Multi-agent System

Jian Li, Junwu Zhu*, Ling Teng

College of Information Engineering, Yangzhou University, Yangzhou 225127, China

*Corresponding Author: jwzhu@yzu.edu.cn

Abstract

Existing Multi-Agent System (MAS) pursue a high degree of concurrency, to improve the efficiency of the task on the study while neglecting the communication and time restriction from bidders. In this paper, we design a time-constrained MAS with inflexible schedule. Tenderer propose tenders to bidders and encourage bidders propose more choices and real ability. For all plan sequences, we check them and finally find out the best solution with shortest deadline or max total revenue.

Keywords: MAS, time-constrained, inflexible schedule

1. Introduction

A Multi-Agent System (MAS) has multi agents, determining the execution sequence according to the constraints among tasks and executing the tasks those can be concurrently executed. The purpose of the system is taking full advantage of the time and resource related, lifting efficiency and finally maximizing the benefits.

In MAS, relationships among tasks are very complicated. There may be existing different connections between one single task and other tasks. Plenty of the possible results multiply the difficulty of assigning tasks. In real life and social practice, without time factor, making a plan for a large number of tasks existing complicated constraints among them is impossible. Tenderer and bidders have different demands. They both want to maximize their benefits. Assignment of tasks and deadline directly influences the benefit of tenderer and bidders. Without time factor, concurrent execution of tasks is not obvious, hard to express the plan and lifting efficiency seem to be not easy.

To solve the above problems, this paper adds time factor and ability into such MAS, making it much more fit the actual. There are time constraints in tenders proposed by tenderer. Also, bidders propose their production time for each task. Time factor makes it easier to quickly check the plan sequence, and finally make assignment easier. In addition to this, time-constrained MAS can fully reflect tenderer’s demand, bidders’ ability and deadline is as important as the total benefit can achieve for tenderer.

This paper is developed from Mixed Multi-unit Combinatorial Auction (MMUCA) and crowdsourcing. We enforce that tender of tasks are clear and bids form bidders can fully represent his ability, time needed and reward. In a time-constrained MAS, we design a mechanism to encourage bidders propose more bids and represent their real abilities. Also, we propose several factors that influence the valuation. All these are for more choice bidder can propose, and tenderer benefits from it.

The paper is organized as follows. In section 2 we review the related work in MAS. In section 3 we describe the framework of the mechanism and the data flow that occurs. In section 4 we define the notions of bids and some sequence. In section 5 we propose some schemes for system. At the end, we remark time-constrained MAS.

2. Related Work

Up till now, there have some study in MAS, such studies mainly focused on organization, communication, negotiation, dependability and fault-tolerance.

In literature [1], it defined the notion of transformation and mentioned multisets. Using multisets can easily describe the goods whatever it is Input or Output and duplicate elements in such multisets can represent variety of Input (or Output). With bidding languages, bids represent what the bidder want to achieve. In literature [2], it introduces the relevant knowledge of the bidding language, refers to several kinds of auctions and figures out the most suitable bidding language in these auctions. With the help of combination of bidding languages, the bids can be shorter and more precise.

In literature[3][4], it reviews the development of
crowdsourcing and introduces the three aspects of incentive design, task assignment and quality control. In the tasks assignment part, emphasis assigning tasks depending on the ability, time and other factors. It emphasizes that true ability of bidders could be report in represent clearly is most important.

3. Framework

Tenderer decompose a big task \( T \) into several little tasks. \( T = \{ \text{task}_1, \text{task}_2, \ldots, \text{task}_n \} \). There are some certain constraints among these tasks like starting one task may have to finish other tasks first. The tenderer issues tender consists of restriction of time for each task, maximum budget cost and a set of tasks priorTasks before we start such a task.

Bidders bid for the tasks from Tenderer. Bidders propose bids for some tasks which they are confident on. A bidder may propose several choices full of plans. Once the bid was chosen to be executed in sequence, the content in the bid would be realized in the time restricted. In one bid, aim task’s name taskName, bidder’s ability transformation, production time needed productionTime and the valuation \( p \) are given. A schedule consists of three parts: taskName, transformation and productionTime. valuation: \( v((\text{schedule})) = p \). Here, \( p \) is a comprehensive offer for transformation, productionTime in the bid and corresponding cost in the tender.

Let \( G \) be the finite set of all the types of goods under consideration. \( U_{in} \& U_{out} \) are both multisets over \( G \). \( U_{in} \) is what tenderer has at the beginning. \( U_{out} \) consists of what the tenderer expect to have. A plan: \( ((\text{taskName}, \text{transformation}, \text{productionTime}), p) \). taskName is the task’s name, transformation means the ability and productionTime is time for production. \( p \) is valuation. productionTime consists two elements: timeBegin and timeEnd. The former represents time to start a production while latter represents the deadline.

Definition1 (transformation): A transformation is a pair of multisets over \( G \): \((I, O) \in \mathbb{N}^G \times \mathbb{N}^G \). A agent declares that it can deliver \( O \) after having received \( I \).

Definition2 (schedule): A schedule is a triple consists of taskName, transformation and productionTime. A plan is one plan proposed by a bidder.

Definition3 (valuation): A valuation is a mapping from schedules to the real number. Intuitively, valuation: \( v((\text{schedule})=p \). If \( p \leq 0 \), bidder will be paid \([p]\). If \( p > 0 \), tenderer will get \( p \).

Definition4 (plan): A plan is a pair consists of schedule and its valuation. It is an atomic element in choice.

Definition5 (choice): A choice consists of several different plans. choice: \{ plan_1, plan_2, \ldots, plan_n \}. A bidder can propose several choices while only one of them can be selected.

Definition6 (\( \gg \)): Transformation: \((\text{input}, \text{output})\) and transformation: \((\text{input}', \text{output}')\) transformation \( \gg \) Transformation iff \( \text{output} \subseteq \text{output}' \).

Definition7 (valid solution sequence): A sequence that meets the tenderer’s expectation and each task has only one plan.

Definition8 (best solution sequence): A valid solution sequence with shortest deadline or max revenue among valid solution sequences is one of best solution sequence.

Proposition1: valuation is not \( \bot \) iff:

1. \( I, O \) are defined over \( G \).
2. transformation\( \gg \)Transformation.
3. productionTime \( \subseteq \) timeRestriction.
5. Algorithm Design

5.1 Rule sequence

Algorithm 1: generation of Rule sequence

**Input:** \( \{task_1, task_2, ..., task_n\} \) \( priorTasks = \{priorTasks_1, priorTasks_2, ..., priorTasks_m\} \), \( j = 0 \)

**Output:** Rule \( [m] \) \( [n] = \{rule_1, rule_2, ..., rule_m\} \)

1. For \( i: 1 \rightarrow n \)
2. If (\( priorTasks_i == \emptyset \))  
3. Continue
4. Else \( rule_{j_1} = task_{i}; \) \( rule_{j_0} = priorTasks_i; j + + \)
5. End for

To put it simply, we put task and its priorTasks into a dimensional array, \( rule_{j_0} \) will be executed before \( rule_{j_1} \) in rule_j, a complete rule consists of all rules like rule_j.

5.2 Rule sequence check

Algorithm 2: Rule sequence check

**Input:** Rule = \( \{rule_1, rule_2, ..., rule_m\} \)

**Output:** true or false

1. for \( i: 1 \rightarrow m \)
2. for \( j: 1 \rightarrow m \)
3. if (\( rule_{i_0} = rule_{j_0} \) \( \land \) \( rule_{i_1} \subseteq rule_{j_0} \))
4. if (\( rule_{i_0} \subseteq rule_{j_1} \) \( \land \) \( rule_{i_0} \neq \emptyset \) \( \land \) \( rule_{i_1} \neq \emptyset \)) \( rule_{j_0} \subseteq rule_{j_1} \) \( \land \) \( rule_{i_0} \neq \emptyset \))
5. Return false; Break
6. End for
7. End for
8. Return true

For any task in one Rule sequence, there is no such a task \( ^{'} \) before the task in one rule while in another rule it is opposite.

5.3 Generation of Plan sequence

Each bidder proposes several choices consist of plans while only one choice will be selected in the end. We select one choice from each bidder and all plans in such choices make up a Plan sequence.

Algorithm 3: generation of Plan sequence

**Input:** Bidder = \{bidder_1, bidder_2, ..., bidder_m\} \( \) \( Bids, T = \{task_1, task_2, ..., task_n\} \)

**Output:** Plan

1. For \( a: 1 \rightarrow m \)

5.4 Whole-matching sequence

According to the mechanism, a plan aims to a certain task while one choice has several plans. It means that, in a Plan sequence, one bidder bids may bid for several tasks. For each Plan sequence, we try to find a potentialSequence that each task has its own corresponding bidder.

Algorithm 4: generation of whole-matching sequence

**Input:** Plan = \{Plan_1, Plan_2, ..., Plan_m\} \( \) \( \) \( T = \{task_1, task_2, ..., task_n\} \)

**Output:** potentialSequence

1. Put plan for task_i into taskPlan[j]
2. Select *form(taskPlan[i]), i \in (1, n)
3. Put result in potentialSequence[]
4. Put taskName of potentialSequence into T'
5. If (\( T! = T' \))
6. Delete the sequence

Not all potentialSequence are whole-matching sequence for taskPlan may be \( \emptyset \).

5.5 Time sort

Algorithm 5: Time sort

**Input:** T = \{task_1, task_2, ..., task_n\}, potentialSequence: \{plan_1, plan_2, ..., plan_m\}

**Output:** TimeExecution[2*n]

1. For \( i: 1 \rightarrow n \)
2. Temp[i]=timeBegin[i]
3. Temp[n+i]=timeEnd[i]
4. End for
5. Selection sort()
6. For r: 1 \rightarrow 2 * n
7. Forward(timeEnd)
8. End for

At the same time, output should be executed first.

5.6 Expectation check algorithm
Given a whole-matching sequence and its TimeExecution, we check the sequence to find out whether it can meet tenderer’s expectation.

Algorithm6: Expectation check

**Input:** potentialSequence{{Plan_1, Plan_2, ... Plan_n}.
U_in, U_out, goodsOfTendree, TimeExecution[2 * n], m = 1;

**Output:** true /false
1. goodsOfTendree(0) = U_in
2. For i: 1 → 2 * n
   3. If TimeExecution[i] is timeBegin and goodsOfTendree(m-1) >= input.goodsOfTendree(m) = goodsOfTendree(m-1)-input(by timeExecution (m))
   4. Else return False; Break
5. If(TimeExecution[i] is timeEnd)
6. goodsOfTendree(m) = goodsOfTendree(m-1)+output(by timeExecution (m))
7. End for
8. If(goodsOfTendree(2*n) ≥ U_out)
9. Return true
10. Else
11. Return false

If U_out ≤ goodsOfTendree(2*n), the sequence meets the tenderer’s expectation.

U_out ≤ U_in + ∑^2*n_{m=0} execution(m)

6. Conclusion

MAS with time constraint clearly describe the Tenderer’s demand and bidders’ ability. Ability, max budget and production time influence bidders’ valuation. In this paper, we find out all possible sequence and check it step by step. Finally, we find best solutions sequence.

If we describe the system, XOR-ORS language is suitable. In this paper, we select at most one choice from each bidder while several plans in choice. It encourages bidders propose more plans in one choice and more choices. Also, bidders would prefer to propose their real ability.

**Acknowledgment**

Project supported by the National Nature Science Foundation of China (Grant No.61170201, No.61070133, No.61472344); Six talent peaks project in Jiangsu Province (Grant No.2011-DZXX-032). Innovation Foundation for graduate students of Jiangsu Province (Grant No.CXLX12 0916), Jiangsu Science and Technology Project No. BY2015061-06, BY2015061-08, Yangzhou Science and Technology Project No. SXT20140048, SXT20150014, SXT20150013, Natural Science Foundation of the Jiangsu Higher Education Institutions (Grant No.14KJB520041), Innovation Program for graduate students of Jiangsu Province(Grant No.SJZZ16_0261).

**References**

1. Jes ‘us Cerquides, Ulle Endriss, Andrea Giovannucci, and Juan A. Rodríguez-Aguilar: Bidding Languages and winner Determination for Mixed Multi-unit Combinatorial Auctions, International Joint Conference on IJCAI, 2007:1221-1226