Reaching Agreement in Multiagent System: Optimization in Allocation Resources

Veena
Student, M.Tech
Dept. of Computer Science and Engineering
Graphic Era University,
Dehradun, India

Mr. Manish Sharma
Assistant. Professor
Dept. of Computer Science and Engineering
Graphic Era University,
Dehradun, India

Abstract— One aspect of Multiagent systems (MAS) that has been only partially studied is their role in software engineering and in particular their merit as a software architecture style.We studies a particular multiagent resource allocation problem with indivisible and sharable resources. The utility of an agent for using a bundle of resources is the difference between the valuations of that bundle of resources. The valuation and the delay can be agent-dependent. Currently, the great majority of agent-based systems consist of a single agent. However, as the technology matures and addresses increasingly complex applications, the need for systems that consist of multiple agents that communicate in a peer-to-peer fashion is becoming apparent. Central to the design and effective operation of such multiagent systems (MAS) are a core set of issues and research questions that have been studied over the years by the distributed AI community.

Keywords- agent, multiagent, allocation, resources, payment scheme, communication protocol.

I. INTRODUCTION

Multiagent system when a number of autonomous artificial entities (agents) interact more or less loosely, more or less cooperatively, with the aim of achieving the objective they have been designed for Often these agents need to reach agreements: among a set of possible alternatives they have to jointly agree upon one of these available choices. Reaching agreements is challenging because individual agents may have conflicting views about the issue at stake. The issue, as we shall see in this document, may be of different nature: allocating a number of resources among a set of agents.

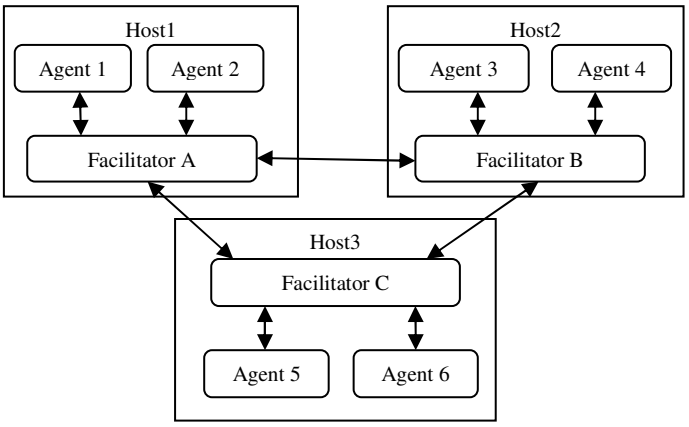

Figure1: Multiagent System Organization

MAS are relevant to a wide range of applications. These include, Background of distributed Resource allocation framework Reference number[1,2], Issues Allocation Resources References number[3,4], Challenges and

Characteristic Multiagent System Reference number[5,6], Side payments Scheme and Function Reference number[7,8], Interaction Protocol in Multiagent System Reference number[9,10].
This paper is a survey of some of the most salient optimization in MAS. In the remainder of this introduction, we first give a tentative definition of MAS and introduce its main Background (Section II). To illustrate the interdisciplinary character of the field, we then list some of the Issues and Challenges that we consider particularly interesting and challenging (Section 2).MAS Resource allocation and Framework discuss in (Section 3,4).An important feature Resource Allocation framework is side payment define in(Section 5).

II. HISTORICAL BACKGROUND

Multi-agent systems are an emerging sub-field of artificial intelligence that is concerned with a society of agents interacting in order to solve a common problem. Multi-agent systems are a relatively new field of research. They have only been studied since about 1980, and the field has only gained recognition since about the 1990s.In 1980 a group of researchers held the first Distributed Artificial Intelligence (DAI) workshop at MIT to discuss issues concerning intelligent problem solving with systems consisting of multiple problem solvers, termed multi-agent systems. Multi-agent systems have become more and more important in many aspects of computer science by introducing the issues of distributed intelligence and interaction. They represent a new way of analyzing, designing, and implementing complex software systems.

III. MOTIVATIONS

With the growth in the Internet and electronic commerce, use of computer technology in MAS is changing in major ways:

- Complex systems are beyond direct control. They operate through the cooperation of many interacting subsystems, which may have their independent interest, and modes of operation.
- Real-world problems are heterogeneous. Heterogeneous environments may use different data and models, and operate in different modes.
- Scalability and flexibility. Because multi-agent systems are open and dynamic structures, the system can be adapted to an increased problem size by adding new agents, and without affecting the functionality of the other agents.

IV. PROBLEM STATEMENT

Often these agents need to optimization: among a set of possible alternative they have to jointly agree upon one of these available choices. Optimization is challenging because individual agents may have conflicting views about the issue at stake. The issue, as we shall see in this document, may be of different nature: allocating a number of resources among a set of agents, specific payments deals and functions. Further the procedure that leads to such an agreement being made can be of various types.

V. ISSUSES AND CHALLENGES

Although MASs provide many potential advantages, they also present many difficult challenges. Here, I present problems inherent in the design and implementation of MAS. The list of challenges includes problems first posed in Bond and Gasser (1988), but I have added some:

First, how do we formulate, describe, decompose, and allocate problems and synthesize results among a group of intelligent agents?

Second, how do we enable agents to communicate and interact? What communication languages and protocols do we use? How can heterogeneous agents interoperate? What and when can they communicate? How can we find useful agents in an open environment?

Third, how do we ensure that agents act coherently in making decisions or taking action, accommodating the nonlocal effects of local decisions and avoiding harmful interactions? How do we ensure the MAS do not become resource bounded? How do we avoid unstable system behavior?

Fourth, how do we enable individual agents to represent and reason about the actions, plans, and knowledge of other agents to coordinate with them; how do we reason about the state of their coordinated process.

VI. MULTIAGENT RESOURCE ALLOCATION(MARA)

A tentative definition would be the following:

"Multiagent Resource Allocation is the process of distributing a number of items amongst a number of agents".

Properties of systems where agents can reallocate resources among them by means of individually acceptable deals. The resources considered are non shareable and indivisible.

The interest of computer scientists in resource allocation problems has been reinforced in recent years by the deployment of large-scale distributed applications involving a number of autonomous entities, possible synergies between resources, and where central computation is in practice infeasible. Canonical examples can be given, the coordination of task-allocation within a team of autonomous Agent and the allocation of computational resources on grid-like systems. All these systems typically build on the Contract-Net, the famous protocol designed to assign a set of tasks to agents, from an initial allocation, agents reassign the tasks among themselves 'in some way', until a 'satisfying' allocation is found. As noticed 'in some way' can be instantiated by a number of reallocation mechanisms.

6.1 RESOURCES

We refer to the items that are being distributed as *resources*, while *agents* are the entities receiving them. We should stress that this terminology is not universally shared. In the context of applications of MARA in manufacturing, for instance, we usually speak of *tasks* that are being allocated to *resources*. That is, in this context, the term "resource", refers to what we would call an "agent" here. We can distinguish different types of resources. For instance, resources may or may not be divisible. For divisible resources, different agents may receive different fractions of a resource.

6.2 ALLOCATION

A particular distribution of resources amongst agents is called an *allocation*. For instance, in the case of non-sharable indivisible resources, an allocation is a partition of the set of resources amongst the agents. The set of resources assigned to a particular agent is also called the *bundle* allocated to that agent.

6.3 OBJECTIVES

The objective of a resource allocation procedure is either to find an allocation that is *feasible, e.g*. to find any allocation of tasks to production units such that all tasks will get completed in time; or to find an allocation that is *optimal*. In the latter case, the allocation in question could be optimal either for the central entity choosing the allocation or with respect to a suitable aggregation of the preferences of the individual agents in the system. (e.g. an allocation of resources that maximizes the average utility enjoyed by the agents).

VII. RESOURCE ALLOCATION FRAMEWORK

7.1 ALLOCATION

Let G be a finite set of indivisible goods. An allocation A: N = 2G is a partitioning of the items in G amongst the agents in N (i.e. each good must be owned by exactly one agent). As an example, allocation A, defined via A (i) = {g1} and A(j) = {g2, g3}, would allocate g1 to agent i, and g2 and g3 to agent j. There are NG possible allocations, and the system is initially in one of these allocations that are, the goods are initially distributed. Observe that we do not put any 1constraints on the number of goods that agents may obtain, so that we have to deal with bundles of resources.

7.2 PREFERENCES AND DOMAINS

Preferences of individual agents i € N is modeled using valuation functions vi: 2G -> R, mapping bundles of goods to the reals. We shall assume that agents only care about the bundle they actually hold (no externality), so we can safely use vi (A) as a shorthand for vi (A (i)), the value agent i assigns to the bundle received in allocation A. We also make the assumption that all valuations vi are normalized, in the sense that vi({ A}) = 0. Sometimes the scenario takes place in a specific domain, in the sense that all agents' valuations functions are known to belong to restricted class of functions.

7.3 UTILITY FUNCTION

As an example, take the following utility function, representing in (slightly simplified) k-additive form the preferences of agent a1 over the possible bundles composed from {g1, g2, g3}.
u1 = 3g1 + 5g2 + 2g1g3 + 2g2g3
The interpretation is as follows: for instance, if a1 obtains g3 alone her utility is 0, if she gets {g1, g2} her utility is 8, if she gets {g1, g2, g3} her utility is 12. The utility is 2-additive, not tree-structured (because the terms g1g3 and g2g3 are overlapping), and not 2-separable (but obviously 3-separable, since there are only 3 resources).

7.4 SOCIAL WELFARE

Multiagent systems are sometimes referred to as "societies of agents" and the aggregation of individual preferences in a MARA system can often be modelled using the notion of *social welfare* as studied in Welfare Economics and Social Choice theory. Examples include utilitarian social welfare, where the aim is to maximise the sum of individual utilities, and , where the aim is to maximise the individual welfare of the agent that is currently worst off. If a *distributed* resource allocation procedure is used, then the term "multiagent" indicates that the computational burden of finding an allocation is shared amongst several agents.

VIII. SIDE PAYMENTS

An important distinctive feature of resource allocation frameworks is whether or not they allow using money. Money may be used in deals to compensate the utility loss of some agents. Money also makes more acceptable interpersonal utility comparisons. Most of the work described here is in a framework with money (we shall nevertheless provide some insights on the framework without money), meaning that deals may be accompanied by monetary side payments. This is modeled using so-called payment functions: p: N -> R, which is required to satisfy. A positive value p (i) indicates that agent i pays money, while a negative value means that the agent receives money. It is possible to impose an initial payment on each agent, at the time of awarding them the bundle they receive in the initial allocation. Payment functions and initial payments together are referred to as the payment scheme. In this context, a state of the multiagent system is described as the current allocation of resources, together with a payment balance. Each agent i € N is then equipped with a utility function ui: 2G×R -> R mapping pairs of bundles and previous payments to the reals.

For example, ui (A (i)) is the utility of agent i in state (A, A') Very abstractally, a deal can be described as a pair of (distinct) allocations = (A, A'), fixing the situation before and after the exchange. When no restriction applies we sometimes talk of complex deals. In practice, the range of feasible deals are restricted, by limiting the number of resources or agents involved, or by preventing agents from giving and receiving resources at the same time. We also consider the case of topological restrictions affecting possible exchanges.

IX. THE FOLLOWING TYPE OF DEALS IN SIDE PAYMENT:

- **One-resource-at-a-time** (or 1-deals): a single resource is passed from an agent to another.
- **Swap deals:** one agent gives a resource and receives (simultaneously) a resource in return from another agent.
- **Cluster deals:** a bundle of resources is passed from an agent to another agent.
- **Bilateral deals:** only two agents are concerned all previous cases are special cases of bilateral deals.
- **T -deals:** must involve an entire term (from a set of terms T), from one or more sender(s) to a single receiver.
- **Clique deals:** involves only agents belonging to a common clique3 of the graph G.

X. SPECIFIC PAYMENT FUNCTIONS

A key result of the framework is that a deal (with money) is individually rational iff it increases utilitarian social welfare.
This means in particular that any IR deal _ = (A,A')

generates a social surplus (sw(A')−sw(A)). And this in turn raises the question of choosing a payment function that is, choosing how to distribute this social surplus generated by the deal. There are several options to do that. We start with, arguably, the most natural ones.

- **The locally uniform payment function (LUPF)** divides this amount equally amongst the participating agents N (and does not affect the other agents);

$p(i) = [vi(A') - vi(A)] - [sw(A') - sw(A)]/|N$

• **The globally uniform payment function (GUPF)** divides it equally amongst all agents N:

$$p(i) = [vi(A') - vi(A)] - [sw(A') - sw(A)]/n$$

These payment functions are "uniform" in the sense that they redistribute the surplus, without any consideration for agents' current situation. We may want to design payment functions that perform some compensation, in the sense that agents are that worst-off receive more from the social surplus generated. we introduced the other payments function:

• **The fully locally equitable payment function,** arrange the payment such that each agent (involved in the deal) would enjoy the same utility level after the deal has been achieved. An important aspect of this payment function is that it may enter in conflict with IR.

• **The rational locally equitable payment function** is computed so as to make every agent marginally better off (so as to satisfy IR), then allocate the remaining payments induced by the deal so as to reduce inequalities as much as possible.

XI. INTERACTION PROTOCOLS MULTIAGENT

Interaction protocols govern the exchange of a series of messages among agents.

- **Communication** protocols provide rules that structure message assign and produce meaningful dialogues or conversations.

- **Cooperation** protocols provide a framework within which agents can coordinate their actions to achieve a complex task or solve a difficult problem in a cooperative way

- **Negotiation** protocols are used in situations where agents have incompatible goals to enable the parties involved to reach a compromise and resolve.

XII. EXPERIMENTAL RESULTS AND FUTURE WORK

CPLEX consists of software components and options. IBM ILOG CPLEX Optimizer is a tool for solving linear optimization problems, commonly referred to as Linear Programming (LP) problems. A linear programming problem may be defined as the problem of *maximizing or minimizing a linear function subject to linear constraints*. The constraints may be equalities or inequalities. The linear programming problem, of the form:

(1) Maximize
(2) Subject to
(3) With bounds

MARA using the optimization to the utilization of Resource allocation arises in a variety of application areas, which include planning, scheduling, sequencing, resource allocating, design, and configuration.IBM ILOG CPLEX Library is the solution of linear optimization problem. The **CPLEX Interactive Optimizer** is an executable program that can read a problem interactively (like the Multiagent system) or from files in certain standard formats, solve the problem, and deliver the solution interactively or into text files.

XIII. CONCLUSION

Designing and building Multiagent systems is difficult. They have all the problems associated with building traditional distributed, concurrent systems and have the additional difficulties that arise from having flexible and sophisticated interactions between autonomous problem-solving components. The big question then becomes one of how effective MASs can be designed and implemented. At this time, there are two major technical impediments to the widespread adoption of multiagent technology: (1) the lack of a systematic methodology enabling designers to clearly specify the resource allocation as MASs (2) the lack of widely available MAS interaction protocol and the utility of resources. Side payment function and deals are enable designers to specify an agent's problem-solving

behavior, specify how and when agents should interact, and visualize and debug the problem-solving behavior of the agents and the entire system.

XIV. ACKNOWLEDGMENT

I would like to thank Mr. Manish Sharma for his insightful remarks upon which I was able to significantly improve the content of this report. This research could not have been performed without the support of the Multi-agents research group at the Artificial Intelligence.

REFERENCES

[1] S. Airiau and U. Endriss. Multiagent resource allocation with sharable items: Simple protocols and nash equilibria. In Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010), May 2010.

[2] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible-winner determination. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10), 2010. To appear.

[3] S. Aknine, S. Pinson, and M. F. Shakun. An extended multi-agent negotiation protocol. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(1):5–45, 2004.

[4] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Pool. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[5] Durfee, E. 1996. Planning in Distributed Artificial Intelligence. In *Foundations of Distributed Artificial Intelligence,* eds. G. M. P. O'Hare and N. R. Jennings, 231 246. New York: Wiley.

[6] H. Ackermann, H. R¨oglin, and B. V¨ocking. Pure Nash equilibria in player-specific and weighted congestion games. Theoretical Computer Science, 410(17):1552 – 1563, 2009.

[7] Kockesen, Levent, and Efe A. Ok (2004). "Strategic Delegation by Unobservable Incentive Contracts." *Review of Economic Studies*, 71, 397–424.

[8] Wang, H., and Wang, C. 1997. Intelligent Agents in the Nuclear Industry. *IEEE Computer* 30(11): 28–34.

[9] Y. Bachrach and J. S. Rosenschein. Distributed multiagent resource allocation in diminishing marginal return domains. In Proc. 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2008), 2008.

[10] S. Jha, P. Chalasani, O. Shehory, and K. Sycara. A formal treatment of distributed matchmaking. In *Proceeding of Agents98*, pages 457–458, Minneapolis, Minnesota, 1998.