Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 112 (2017) 91–98

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France

Adaptive Task Allocation Based on Social Utility and Individual Preference in Distributed Environments

Naoki Iijima[a], Ayumi Sugiyama[a], Masashi Hayano[a], Toshiharu Sugawara[a,*]

[a]*Department of Computer Science and Communications Engineering*
Waseda University, Tokyo 1698555, Japan

Abstract

Recent advances in computer and network technologies enable the provision of many services combining multiple types of information and different computational capabilities. The tasks for these services are executed by allocating them to appropriate collaborative agents, which are computational entities with specific functionality. However, the number of these tasks is huge, and these tasks appear simultaneously, and appropriate allocation strongly depends on the agent's capability and the resource patterns required to complete tasks. Thus, we first propose a task allocation method in which, although the social utility for the shared and required performance is attempted to be maximized, agents also give weight to individual preferences based on their own specifications and capabilities. We also propose a learning method in which collaborative agents autonomously decide the preference adaptively in the dynamic environment. We experimentally demonstrate that the appropriate strategy to decide the preference depends on the type of task and the features of the task reward. We then show that agents using the proposed learning method adaptively decided their preference and could maintain excellent performance in a changing environment.

Keywords: Task allocation; Preference; Reinforcement learning; Cooperative agent

1. Introduction

Recent advances in computer and Internet technologies have enabled IoT (Internet-of-Things) applications in which a huge number of networked devices and machines such as computers, sensors, cameras, and smartphones are interconnected, and the services are provided by combining their different functions/capabilities and executing them in a cooperative manner. Therefore, the allocation of tasks to appropriate collaborative agents is a central issue because (1) a huge number of agents with different capabilities exist on the Internet, (2) tasks have a variety of requirements regarding to deadlines and solution quality, and (3) the number of service requests from users are also massive and

* Corresponding author. Tel.: +81-3-5286-2596 ; fax: +81-3-5286-2596.
E-mail address: sugawara@waseda.jp (Toshiharu Sugawara).

E-mail address: {n.iijima,a.sugiyama,m.hayano}@isl.cs.waseda.ac.jp, sugawara@waseda.jp

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of KES International
10.1016/j.procs.2017.08.177

they are geographically distributed over a wide area. Nevertheless, efficient task allocation to appropriate agents is required.

Thus, a number of autonomous task (re)allocation/negotiation methods are proposed. These studies are usually based on auction- or market-like algorithms and have been proposed in multi-agent system contexts[1,2,3,4,5,6]. For example, Asadpour and Sabei[1] proposed a semi-optimal algorithm for max-min indivisible goods under a condition where agents are assumed to have the known linear utility function. Aziz et al.[2] proposed an algorithm that finds the Pareto optimal allocation using only an agent's ordinal preference that is consistent with the utility function. However, how to define the utility in a large-scale real-world application in which tasks with a variety of characteristics and requirements must be performed by heterogeneous agents is not obvious. This is because the performance of distributed systems is affected by the capabilities and the desire of agents and vary in accordance with the changes in service requests.

For this issue, we have developed a strategic and flexible task allocation method that takes into account both social utility and local (i.e., the agent's individual) preference[7], even when the social utility and the local preference may be incompatible. This method is based on a resource allocation framework called *single-object resource allocation with preferential order* (SORA/PO), where each agent is allocated only one resource at a time on the basis of social cardinal utility but is also permitted to declare a preference for the resources that are likely to be allocated[8]. However, Iijima et al.[7] only indicated the findings where the appropriate preference of each agent is strongly affected by the social utility (such as the processing time) and the specifications of tasks (such as the deadlines, required resource patterns, and associated rewards).

The contribution of this paper is to propose a method with which collaborative agents learn the appropriate preference strategy to improve the entire performance with fixed social utility. We assume that the social utility is shared with all agents, so it is costly to change it, but the local preference can be changed by individual decisions. In the proposed methods, agents learn the best (or a better) strategy to decide the local preference from a number of possible strategies. Then, even if the structure and the distribution of required tasks vary, agents can adaptively select the strategy so that the local and entire performance are maintained and improved. The local preference may be inconsistent with the social utility; thus, in this sense, agents may not be socially rational. However, we believe that if they can consider their own specifications and capabilities, the entire high performance would be maintained even under the changes of environments by combining the proposed learning and allocation methods.

This paper is organized as follows. Section 2 describes the basic concept to discuss our issue and our proposed method. Then, we introduce a number of strategy functions to decide agent preferences and present the proposed method in Section 3. In Section 4, we experimentally show that the entire system's performance is affected by the preference strategies that decide the agents' preferences. Then, we evaluate our proposed method to learn the appropriate preference individually in the various environment, and show that it can exhibit good performance in all situations. Finally, Section 5 presents a brief summary of our result.

2. Preliminary

This section describes the preliminary concepts, task allocation problem and single-object resource allocation with preferential order, to explain our issue addressed here and our proposed method.

2.1. Task allocation problem

Let $A = \{1, \cdots, n\}$ be the set of agents, and introduce a discrete time whose unit is known as a *tick*. Agent $i \in A$ has the associated resources denoted by $R^i = \{r^i_1, \dots, r^i_h\}$, where $r^i_k \geq 0$ and h is a positive number expressing the number of resource types. The sum of resources in i is denoted by $\|R^i\| = \sum_{k=1}^{h} r^i_k$.

The task t is characterized by (S^t, w_t, o_t), where $S^t = \{s^t_1, \dots, s^t_h\}$ and $s^t_k \geq 0$, and it represents the resources required to complete t. Then, the agent receives the associated reward, $w_t > 0$. In this paper, w_t is the sum of resources $\|S^t\| = \sum_{k=1}^{h} s^t_k$, but, of course, we can define the reward differently. Thus, we also consider the tasks whose rewards change over time. The deadline of t is denoted by d_t, and agents must complete t before it; otherwise t is dropped. The solution of our problem is to increase the total rewards of tasks and/or reduce the dropped tasks due to the time-out.