Engineering Applications of Artificial Intelligence 26 (2013) 1458–1468

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Multi-objective optimization for dynamic task allocation in a multi-robot system

Avraam Th. Tolmidis *, Loukas Petrou

Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece

A R T I C L E I N F O

Article history:
Received 4 July 2012
Received in revised form
17 December 2012
Accepted 5 March 2013
Available online 8 April 2013

Keywords:
Multi-agent
Task allocation
Multi-objective
Optimization
Multi-robot

A B S T R A C T

In this paper, we propose a solution to the Multi-Robot Dynamic Task Allocation problem. We use Multi-Objective optimization in order to estimate, and subsequently, make an offer for its assignment. The motivation is to provide a generic solution, independent of the domain, with an aim to better utilize resources such as time or energy. The algorithm provides a significant degree of flexibility, and can be implemented in a number of diverse domains, provided the modeling of the parameters follows the convention presented. For this, we take into account – besides the distance traveled – the efficiency of a robot in a specific task type. The system has been shown to demonstrate scalability, as the experimental results indicate. It is also capable of responding to changes in the environment.

1. Introduction

Multi-robot task allocation is so far an open problem. As the relevant technology advances, robots can perform more functions. This yields more perspectives, but also introduces more variables to consider. Using multiple robots provides improved efficiency. It minimizes time and other resources required to achieve a task. It also provides increased robustness, and helps respond to possible changes in the functioning environment. This paper studies the online allocation problem. Task information is not known a priori. Characteristics of tasks are available upon detection, and allocation takes place in real time.

The aim of this paper is to provide an allocation mechanism. Therefore, we do not focus on the detection and decomposition process. Tasks are introduced to the system using a standard representation that includes all the parameters used by the algorithm (position, type, importance, etc). They can be detected using sensors, but also introduced by human users. The proposed system can be considered neither strictly centralized, nor decentralized. It consists of clusters of robots that are responsible of covering a geographical area. The structure is shown in Fig. 1. Clusters are autonomous. They have minimal interaction between them. Each cluster has a database system that contains known task types. Clusters synchronize their databases, so that any new task information is available to the entire system. In the case of a failure, a cluster replicates the database of a neighbor. This adds robustness to the system. The process is shown in Fig. 2.

Individual Clusters work in a centralized manner, while the system as a whole is decentralized. After task detection, an auction takes place amongst robots capable of executing it. Robots calculate their bid using Multi-Objective Optimization (MOO). They use a genetic algorithm, and Pareto optimality to solve the problem. Before deciding on a winner, the auctioneer takes into account the previous performance of the bidder. The paper is organized as follows. In the next section a review of other approaches to the task allocation problem is given. Section 3 provides a more detailed analysis of the proposed algorithm. In Section 4 the experimental setup in order to evaluate the method

Abbreviations: $Ag=\{Ag_1,Ag_2,\ldots,Ag_s\}$, set of agents a robot consists of; Au_l, auction for task l; B_l, bid for Task l; $C=\{C_1,C_2,\ldots,C_m\}$, the set of robot clusters; C_i, a cluster; Chr_i, chromosome i; Cr_i, coordinator of cluster i; E_c, current energy level; E_r, energy level after completion of task queue; E_{max}, maximum robot energy level; $F=\{F_1,F_2,\ldots,F_k\}$, set of functionalities of robot j; G_g, gth generation of chromosomes; $Obj=\{Obj_1,Obj_2,\ldots,Obj_u\}$, set of Objectives pursued; Oc_i, occurrences of task i; Pos_l, Position of task l; $Q_i=\{Tk_{i1},Tk_{i2},\ldots,Tk_{il}\}$, task queue of an agent; $Qtemp_i$, temporary task queue of agent i; $R_i=\{R_{i1},R_{i2},\ldots,R_{in}\}$, robots in cluster i; R_j, robot j; Rl_s, task relevancy level; $S=\{S_1,S_2,\ldots,S_k\}$, set of sensors deployed in the environment; T_d, time of task detection; T_l, time until completion of task l; $Taloc_k$, allocated tasks for agent k; $Tfail_k$, failed tasks for agent k; $Tk=\{Tk_1,Tk_2,\ldots,Tk_q\}$, set of tasks allocated to a robot; Tk_{il}, lth task in Q_i; $V_j=\{E_c,F,Obj,Ag,Tk\}$, attributes of robot j; $p_{0,l}$, initial priority of task l; pf_l, factor by which the priority of task l increases; r_l, position (rank) of Task l in an agent's task queue; t_{comp}, estimated time of task completion; t_{exp}, task expiry time

* Corresponding author. Tel.: +30 2310996294.

E-mail addresses: atolmid@gmail.com, atolmid@issel.ee.auth.gr (A.Th. Tolmidis), loukas@eng.auth.gr (L. Petrou).

0952-1976/$ - see front matter
http://dx.doi.org/10.1016/j.engappai.2013.03.001

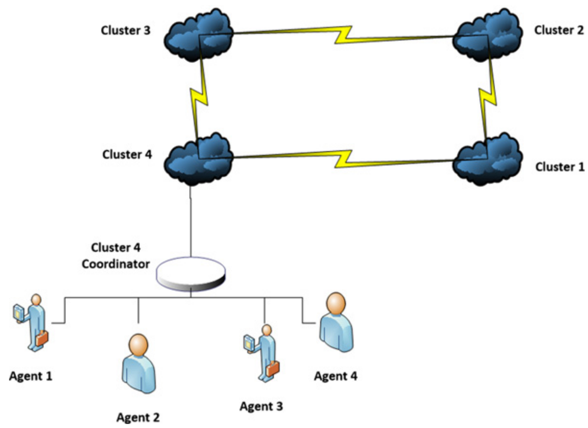

Fig. 1. Structure of the model.

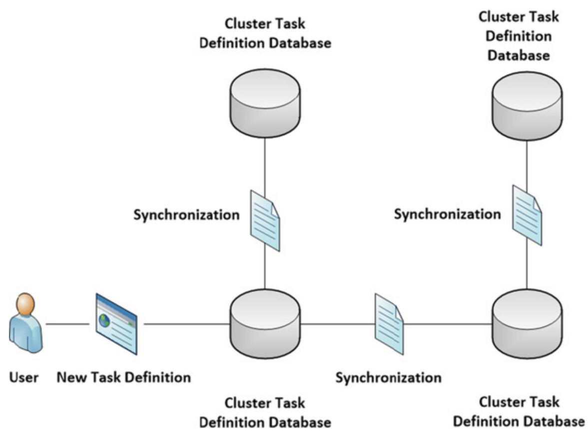

Fig. 2. Insertion of new task type in the system.

is described. Results are presented in Section 5. Section 6 concludes the paper.

2. Related work

Task allocation systems can be classified according to the following criteria:

- The time when computations are performed (online–offline)
- The architecture (centralized–decentralized)
- The types of interactions exhibited in Parker (2008)

In an online allocation system, tasks are identified while the system is functioning. The allocation process takes place after initialization, and new tasks are introduced after that time. Basilico and Amigoni (2011) use Multi-Criteria Decision Making (MCDM), to define exploration strategies in the domain of search and rescue. Another online algorithm is introduced in Jolly et al. (2010), this time in the domain of Robot Soccer Systems. They use a fuzzy neural network in order to plan tasks and select actions.

Korsah et al. (201) used the offline approach. They introduced the xBots system architecture, where tasks are known a priori.

One can also find hybrid systems, that try to get the best from both worlds. An example is Xu et al. (2009), where they propose a Modified Ant Colony System as a solution to the Multi-robot dynamic task allocation problem. Initially, a leader robot allocates predefined tasks, minimizing the traversed distance, as well as balancing workload. Any necessary adjustments after the system has gone online, are performed by the robots without external intervention.

The second major distinction is between centralized and decentralized systems. In a centrally managed system we can make optimal decisions, as the coordinator has global knowledge of the environment. A fault in the coordinator however, makes them unable to operate. Decentralized ones, distribute the processing required, can be more effective in communication, and are more scalable. Most recent proposals use the latter approach.

For instance, Dasgupta and Hoeing (2008) propose a market-based algorithm, along with swarm-based coordination. Agents that encounter new tasks communicate their task lists to nearby robots, and use a dynamic pricing algorithm in order to sell them the task. A slightly different approach, however still distributed, are Distributed In-Network Task Allocation (DINTA) and Multi-Field Distributed In-Network Task Allocation (DINTA-MF) introduced in Batalin and Sukhatme (2005) and Batalin and Sukhatme (2004). They use a static network for communication, sensing and computation. For every detected task the network computes and propagates a utility of the task assignments to the robots. Network nodes make the computations and give nearby robots suggested directions to follow. On the other hand, in Khamis et al. (2011) centralized and hierarchical dynamic and fixed tree task allocation is used.

Considering the types of interactions that are present, a major category are organizational or social approaches. They model group dynamics between agents, forming an organization. Such examples in multi-robot systems are roles, market economies, or teamwork (Parker, 2008). Robots can compete for a task, or cooperate to execute it. A commonly used protocol in market-based architectures is the contract-net protocol discussed in Davis and Smith (1983).

Zlot and Stentz (2005) extend the TraderBots discussed in Dias (2004), a market-based solution. The agents can dynamically act as auctioneers or bidders and facilitate peer-to-peer trades amongst them. In addition, Zlot and Stentz try to address the decomposition problem with the use of task trees. Two different types are used: in the first the goal is satisfied if all subtasks are executed; in the other, one subtask is sufficient. The measure of solution quality is the total distance traveled.

An important parameter in systems that use auctions is the way utility and bids are calculated. Chapman et al. (2010) define individual utilities for each agent, as well as a global utility that they wish to maximize. The global utility is constructed in the same way as in a centralized Markov Decision Problem. Tovey et al. (2005) make an attempt to formalize the formation of bidding rules of auctions, given the objectives. They use a multi-robot exploration task as a case scenario. Jones et al. (2007) use regression in order to calculate bids. Hoogendoorn and Gini (2009) allow agents to express preferences over particular durations, certain time points, or certain types of tasks. In our proposal, agents do not express preferences on how long the task they execute should take to finish, or when in time they want to be more active. They do express a preference however in the types of tasks, represented by the Relevance degree, which – like in the case of Hoogendoorn and Gini (2009) – is expressed by an integer. Finally, Lagoudakis et al. (2005) suggest a generic framework for auction-based multi-robot routing and analyze bidding rules for various objectives.

Lagoudakis et al. are among the first ones to provide upper and lower bounds on the performance, in a market-based system. The allocation of a number of exploration tasks to a team of robots is studied in both papers, Lagoudakis et al. (2005), but also Lagoudakis et al. (2004). They propose an auction-based algorithm, PRIM ALLOCATION, that produces a cost at most twice as large as the one with optimal allocation. Furthermore, they