Cloning, Resource Exchange and Relation Adaptation: A Self-organising Multi-Agent Framework

(Extended Abstract)

Dayong Ye
University of Wollongong
NSW 2522 AU
dy721@uowmail.edu.au

Minjie Zhang
University of Wollongong
NSW 2522 AU
minjie@uow.edu.au

Danny Sutanto
University of Wollongong
NSW 2522 AU
danny@elec.uow.edu.au

ABSTRACT

In this paper, a self-organising multi-agent framework is proposed. Different from current related approaches which concerned only a single principle of self-organisation, this framework synthesises the three principles of self-organisation, i.e., agent cloning/spawning, resource exchange and relation adaptation. In this framework, an agent can autonomously generate new agents when it is overloaded, exchange resources with other agents if necessary, and adapt relations with other agents to achieve a better network structure. In this way, agents in this framework can adapt to dynamic environments.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Self-organisation, Adapation, Learning

1. INTRODUCTION

Self-organisation is defined as "the mechanism or the process enabling the system to change its organisation without explicit external command during its execution time [2]". For self-organising sytems design, Mathieu et al. [1] pointed out that a self-organising system should include three principles. The first one is that agents within the system will generate new agents to take part of their load once they are overloaded. The second one is that agents can exchange skills or resources, if necessary, between each other to increase autonomy. The last one is that agents should be able to create new specific relations between agents in order to remove the middle-agents. Currently, each of the three principles has attracted many research efforts. However, to the best of our knowledge, there lacks an attempt which combines the three principles in a single framework in order to achieve better performance compared with those self-organisation approaches which considers only one of the three principles. Towards this end, in this paper, we present a self-organising multi-agent framework which combines the three principles together, i.e., agent cloning/spawning, resource exchange and relation adaptation. Our framework is illustrated within a general platform, i.e., distributed task allocation. By employing a general platform, instead of a particular existing system, our framework can be potentially applied to a wide variety of applications.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. MODEL DESCRIPTION

The task allocation network is modeled as a tuple $\langle A, N, T, \Gamma \rangle$. Each element is described as follows.

- A is a set of collaborative agents in the network, i.e., $A = \{a_1, ..., a_n\}$.
- $N = \{N_1, ..., N_n\}$, where each N_i demonstrates the neighbours of agent a_i.
- $T = \{t_1, ..., t_m\}$ is a set of task types which will arrive at the network.
- $\Gamma = \{\gamma_1, ..., \gamma_l\}$ is a set of resource types which exist in the network.

The neighbour set of each agent N_i consists of three different neighbours, i.e., *peer*, *subordinate* and *superior*, which are formed by two relations, i.e., *peer-to-peer* relation and *subordinate-superior* relation.

A task type defines which resource types are needed and the quantity of each type of resources is required. In addition, a task type also dictates the task benefit, the task service time and the task maximum waiting time before being executed. If a task cannot be completed before the deadline of its service time, the benefit of this task decreases gradually with time elapse till 0.

There is a continuous dynamic stream of tasks which arrive at the network. Each task Φ randomly corresponds to a task type in T. Φ can be divided into several subtasks and each subtask, $\varphi_i \in \Phi$, requires a particular resource type and a specific amount of this type of resources which are indicated by the corresponding task type. In addition, each subtask has a relevant benefit paid to the agent which successfully completes the subtask. In this paper, a subtask φ_i is modeled as a token Δ_i which can be passed in the network to find a suitable agent to complete. Each token consists of not only the information about resource requirement of the corresponding subtask, but also the token traveling path which is composed of those agents that the token has passed.

3. THE SELF-ORGANISING MULTI-AGENT FRAMEWORK

3.1 Agent Cloning/Spawning

When an agent is overloaded (i.e., it cannot complete the subtasks in its subtask waiting list before their respective deadlines or it has too many neighbours to keep with), the agent will create a new agent to handle part of its load. The agent has two options, namely cloning an agent or spawning an agent.

Specifically, for a single agent, spawning is triggered when the task load exceeds the agent's ability to finish it on time, given the agent's current status and resource level. In this condition, the agent spawns some new agents and assigns the most benefit tasks and corresponding resources to them. These spawned agents are as *subordinate*s of the original agent, but they cannot establish relations with other agents. When the spawned agents finish the assigned tasks, they are in an idle status. When the spawned agents keep in an idle status for a pre-defined period, namely that no more such subtasks need to be completed, they will be destroyed by the original agent to save relation management load.

On the other hand, cloning happens when an agent has too many neighbours, which means that the agent has a heavy overhead for managing relations with other agents. In this situation, to avoid possible communication congestion, the agent clones an agent which has the same resources as itself, and assigns some neighbours to the cloned agent. The original agent keeps a *peer* relation with the cloned one. Different from the spawning agents, the cloned agent cannot be destroyed by the original agent. Instead, the original and cloned agents will compose together, once the total number of neighbours of them is less than a pre-defined threshold.

3.2 Resource Exchange

For a single agent, when a resource has not been used for a long time, the agent will transfer the resource to a neighbouring agent which really needs this resource. Here, we devise a Q-learning algorithm to handle it (**Algorithm 1**). An action in this Q-learning algorithm represents transferring a resource to a neighbour.

Algorithm 1: Resource transfer according to a_i

for each $a_i \in A$ **do**
  $Res \leftarrow GetResource()$;
  for each resource $\gamma_i \in Res$ **do**
    a_i initialises Q-values and $\boldsymbol{\pi}$;
    a_i informs its neighbours;
    neighbours respond to a_i;
    $\mathbf{r} \leftarrow$ the reward vector for neighbours;
    for each neighbour of a_i, i.e., $n_j \in N_i$ **do**
      $Q(n_j) \leftarrow (1-\alpha_1)Q(n_j) + \alpha_1 \cdot r_j$;
    end for
    $\bar{r} \leftarrow \frac{1}{|N_i|}\sum_{n_j \in N_i} r_j$;
    for each neighbour of a_i, i.e., $n_j \in N_i$ **do**
      $\pi(n_j) \leftarrow \pi(n_j) + \zeta(Q(n_j) - \bar{r})$;
    end for
    $\boldsymbol{\pi} \leftarrow Normalise(\boldsymbol{\pi})$;
    a_i selects a neighbour based on $\boldsymbol{\pi}$;
  end for
end for

3.3 Relation Adaptation

Our relation adaptation algorithm is based on the past information of the individual agents. Specifically, agents use the information about the past task allocation processes to evaluate their relations with other agents. We develop a multi-agent Q-learning algorithm to tackle the relation adaptation problem. **Algorithm 2** demonstrates our relation adaptation algorithm in pseudocode form.

Algorithm 2: Relation adaptation according to a_i

1 $Candidates_i \leftarrow a_i$ selects agents in the network;
2 **for each** $a_j \in Candidates_i$ **do**
3 $Act_i \leftarrow available\_actions(a_i, a_j)$;
4 $Act_j \leftarrow available\_actions(a_i, a_j)$;
5 **for each** $x \in Act_i$, $y \in Act_j$ **do**
6 Initialise Q_{ix} and Q_{jy} arbitrarily;
7 **for** $k = 0$ to a predefined integer **do**;
8 calculate $\pi_{ix}(k)$ and $\pi_{jy}(k)$;
9 $Q_{ix}(k+1) = Q_{ix}(k) + \pi_{ix}(k)\alpha_2(\sum_y r_i^{x,y}\pi_{jy}(k) - Q_{ix}(k))$;
10 $Q_{jy}(k+1) = Q_{jy}(k) + \pi_{jy}(k)\alpha_2(\sum_x r_j^{x,y}\pi_{ix}(k) - Q_{ix}(k))$;
11 **end for**
12 **end for**
13 $\langle x_{opti}, y_{opti}\rangle \leftarrow argMax_{match(x,y)}(Q_{ix} + Q_{jy})$;
14 a_i, a_j take actions x_{opti} and y_{opti}, respectively;
15 $\mu_{ij} \leftarrow \mu_{ij} + (L_j^i/\rho_1 - 1)$;
16 **if** $\mu_{ij} > 1$ **then** $\mu_{ij} \leftarrow 1$;
17 **if** $\mu_{ij} < 0$ **then** $\mu_{ij} \leftarrow 0$;
18 $\mu_{ji} \leftarrow \mu_{ij}$;
19 **end for**

When finishing learning Q-values, a_i and a_j (Line 13) cooperate to find the optimal actions for both of them.

Algorithm 3 illustrates the reasoning aspect of each agent for selecting a group of agents to initilise the relation adaptation process.

Algorithm 3: Candidates selection of each agent

for each $a_i \in A$ **do**
  $Candidates_i \leftarrow \emptyset$;
  for each $\Delta_k \in tokens_i$ **do**
    statistics of $\Delta_k.owner$;
  end for
  if $\exists$ # of same $\Delta_k.owner > \rho_2$ **and**
    $\Delta_k.owner \notin Neig_i^{\sim} \vee Neig_i^{\succ} \vee Neig_i^{\prec}$ **then**
    $Candidates_i \leftarrow Candidates_i \cup \{\Delta_k.owner\}$;
  end if
  if $\exists$ # of same $\Delta_k.owner < \rho_3$ **and**
    $\Delta_k.owner \in Neig_i^{\sim} \vee Neig_i^{\succ} \vee Neig_i^{\prec}$ **then**
    $Candidates_i \leftarrow Candidates_i \cup \{\Delta_k.owner\}$;
  end if
end for

4. CONCLUSION

This paper introduced a self-organising multi-agent framework which considers the three principles of self-organisation, i.e., agent cloning/spawning, resource exchange and relation adaptation. Through combining the benefits of the three principles, our framework outperforms state of the art approaches which focus on a single principle only. Since our framework is decentralised and continuous over time without external control, it meets the definition of self-organisation given by Serugendo et al. [2].

5. REFERENCES

[1] P. Mathieu, J.-C. Routier, and Y. Secq. Principles for dynamic multi-agent organizations. In *PRIMA'02*, pages 109–122, Tokyo, Japan, Aug. 2002.

[2] G. D. M. Serugendo, M.-P. Gleizes, and A. Karageorgos. Self-organization in multi-agent systems. *The Knowledge Engineering Review*, 20(2):165–189, 2005.