Using Distributed W-Learning for Multi-Policy Optimization in Decentralized Autonomic Systems

Ivana Dusparic and Vinny Cahill
Lero - the Irish Software Engineering Research Centre
Distributed Systems Group, School of Computer Science and Statistics
Trinity College Dublin
ivana.dusparic@cs.tcd.ie, vinny.cahill@cs.tcd.ie

ABSTRACT

Distributed W-Learning (DWL) is a reinforcement learning-based algorithm for multi-policy optimization in agent-based systems. In this poster we propose the use of DWL for decentralized multi-policy optimization in autonomic systems. Using DWL, agents learn and exploit the dependencies between the policies that they are implementing, to collaboratively optimize the performance of an autonomic system. Our initial evaluation shows that DWL is a feasible algorithm for multi-policy optimization in decentralized autonomic systems. Our results show that a multi-policy collaborative DWL deployment outperforms individual single policy deployments, as well non-collaborative deployments.

Categories and Subject Descriptors

H.3.4 [Systems and Software]: Distributed systems; I.2.11 [Distributed Artificial Intelligence]:Multiagent systems

General Terms

Algorithms, Design, Experimentation

Keywords

Autonomic Computing, Reinforcement Learning, Decentralized Systems

1. AUTONOMIC SYSTEMS

Autonomic computing systems should be capable of self-management based only on high-level objectives specified by their users [4]. The details of how to meet those objectives should be left to the systems themselves. Therefore, autonomic systems are required to self-optimize, self-heal, self-protect, and self-configure in order to meet their objectives, even in the face of changing operating conditions.

Enabling autonomic behaviour is particularly challenging in decentralized autonomic systems [9], where central control is not tractable, due to the large number and geographical dispersion of the entities involved. Important examples of such systems that could benefit from autonomic management include large-scale critical infrastructures (transportation networks, electricity, gas, and water supply etc), as well as numerous other applications that involve, for example, scheduling, task allocation, routing, or load balancing, such as global supply chain management [6].

A decentralized autonomic system can be implemented as a group of autonomous agents, as the characteristics of agents map to those required by autonomic elements [4]. These agents should have the ability to self-organize based only on local actions and interactions, so that the global behaviour of the system, required to meet its global objectives, emerges from the agents' local behaviours. A number of techniques have been used to support such self-organization to implement autonomic characteristics, e.g. ant-colony optimization in load balancing [5] and digital evolution in autonomous robot navigation [2].

These techniques have mostly been used to implement systems with only a single explicit objective (i.e., a single policy) while in reality management of most such systems requires optimization towards multiple, often conflicting objectives. For example, critical infrastructures need to deal with sudden surges in demand, or drops in supply to the different areas of the system, while maintaining high quality of service and steady supply to prioritized users. This leads to different parts of the system implementing different policies, resulting in the heterogeneity of the agents of which the system is composed. The coordination required for self-organization is particularly challenging in such heterogeneous environments. Agents might be required to contribute to the implementation of other agents' policies that might be of a higher priority for the system, but might not be aware of what those policies are. Additionally, due to this lack of awareness of other agents' policies, an agent might not be able to exchange meaningful and relevant information with other agents, which might be required for coordination.

2. DISTRIBUTED W-LEARNING

To address multi-policy self-optimization in decentralized autonomic systems, we have developed an RL-based algorithm, Distributed W-Learning (DWL), which has been inspired by W-Learning [3]. RL-based techniques have already been shown to be suitable for the implementation of autonomic systems [7], as well as suitable for large-scale dynamic environments (e.g. urban traffic control (UTC) [1]), as they do not require domain knowledge or predefined environment models. DWL enables agents to engage in cooperation with their neighbours in order to decide on the most suitable actions to take in order to meet system policies. Collaboration is enabled regardless of agent and policy heterogeneity, and the lack of central control or global knowledge is addressed by making all required actions and interactions local.

In DWL, as well as learning Q-values [8] and W-values [3] for all of their local policies, agents also learn Q-values and
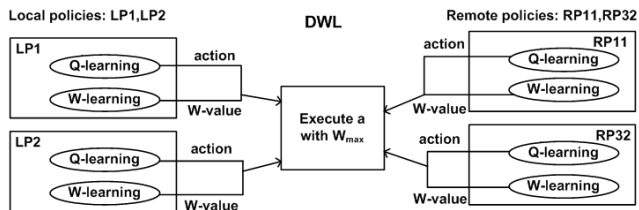
Figure 1: DWL action nomination

W-values for all of the policies that their immediate neighbours are implementing. At each time step, each agent considers W-values for the current state of each of its own and its immediate neighbours' policies. If any of the immediate neighbours' policies has a higher W-value than the agent's local W-values, the action suggested by that neighbour can be executed. The neighbours' W-values are multiplied by a cooperation coefficient C, to enable a local agent to give a varying degree of importance to the neighbours' action suggestions. C can range from fully non-cooperative, C=0, where an agent does not consider neighbours' suggestions at all, to fully cooperative, C=1, where the neighbours' suggestions matter as much as local ones. W-values learnt by an agent reflect the dependencies between the agent's own policies, as well as potential dependencies between the agent's actions and its neighbours' performance. Agents can then exploit this knowledge to improve not just their own performance but the performance of their neighbours' as well.

3. EVALUATION OF DWL

We have evaluated DWL in a simulation of a UTC system, a canonical example of a decentralized autonomic system. In our experiments we deploy two policies, one that optimizes general traffic flow (GWO), and the other that prioritizes public transport vehicles (PTO). Both policies are deployed separately, as well as using two multi-policy scenarios: a non-collaborative scenario, where each agent implements action-selection using W-Learning [3] (C=0), and a fully-collaborative scenario (C=1), where each agent implements action-selection taking its neighbour's preferences into account.

We simulate 750 minutes of car and public transport vehicle traffic on a road network corresponding to inner city Dublin, consisting of 62 signaled junctions. GWO is implemented on all junctions, while PTO is implemented only on 47 junctions that are positioned on bus routes. Experiments were performed for two traffic loads: 35,000 vehicles and 60,000 vehicles.

4. RESULTS

Our experiments indicate that DWL is a feasible algorithm for multi-policy multi-agent optimization in decentralized autonomic systems. Cooperative DWL scenarios outperform non-cooperative W-Learning scenarios, highlighting dependencies between the agents and DWL's capability to exploit them. DWL multi-policy scenarios outperform single-policy scenarios, highlighting the dependency between the multiple policies deployed, and the need to address them simultanously.

5. CONCLUSIONS AND FUTURE WORK

This paper addressed the problem of multi-policy multi-agent optimization in decentralized autonomic systems. We have presented Distributed W-Learning (DWL), an algorithm that enables collaborative optimization towards multiple policies on multiple agents simultaneously. Our initial results show that collaborative multi-policy DWL implementation outperforms corresponding non-collaborative and single-policy scenarios. We plan to further investigate the applicability of DWL in UTC and other application areas, as well as to investigate the impact of agents engaging in varying levels of collaboration in DWL.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] ABDULHAI, B., PRINGLE, R., AND KARAKOULAS, G. Reinforcement learning for the true adaptive traffic signal control. *Journal of Transportation Engineering 129*, 3 (May/June 2003), 278–285.

[2] GOLDSBY, H. J., CHENG, B. H. C., MCKINLEY, P. K., KNOESTER, D. B., AND OFRIA, C. A. Digital evolution of behavioral models for autonomic systems. In *ICAC '08*, IEEE Computer Society, pp. 87–96.

[3] HUMPHRYS, M. *Action Selection methods using Reinforcement Learning.* PhD thesis, University of Cambridge, 1996.

[4] KEPHART, J. O., AND CHESS, D. M. The vision of autonomic computing. *Computer 36*, 1 (Jan 2003), 41–50.

[5] MONTRESOR, A., MELING, H., AND BABOGLU, O. Messor: Load-balancing through a swarm of autonomous agents. In *AP2PC '02* (Bologna, Italy), G. Moro and M. Koubarakis, Eds., no. 2530 in Lecture Notes in Artificial Intelligence, Springer-Verlag, pp. 125–137.

[6] OLEARY, D. Supporting decisions in real-time enterprises: autonomic supply chain systems. *Information Systems and E-Business Management 6*, 3 (June 2008), 239–255.

[7] TESAURO, G. Reinforcement learning in autonomic computing: A manifesto and case studies. *IEEE Internet Computing 11*, 1 (2007), 22–30.

[8] WATKINS, C. J. C. H., AND DAYAN, P. Technical note: Q-learning. *Machine Learning 8*, 3 (May 1992), 279–292.

[9] WOLF, T. D., AND HOLVOET, T. *Autonomic Computing: Concepts, Infrastructure, and Applications.* CRC Press, 2007, ch. A Taxonomy for Self-* Properties in Decentralized Autonomic Computing.