Scalable Potential-Field Multi-Agent Coordination In Resource Distribution Tasks

Steven de Jong, Karl Tuyls
Institute of Knowledge and Agent Technology,
Universiteit Maastricht, P.O. Box 616, 6200 MD,
Maastricht, The Netherlands
{steven.dejong, k.tuyls}@cs.unimaas.nl

Tsuyoshi Hashimoto, Hiroyuki Iida
School of Information Science, Japan Advanced
Institute of Science and Technology, 1-1,
Asahidai, Nomi, Ishikawa 923-1292, Japan
{t-hashi, iida}@jaist.ac.jp

ABSTRACT
In this paper, we look at a scalable, generic methodology for coordination in large, embodied multi-agent systems (MAS), operating in noisy, real-time environments. More specifically, we look at real-world task assignment problems and use the testbed of robots performing resource distribution in large storage facilities. The main problem when using MAS for task assignment problems in real-world environments is the limited scalability of traditional MAS approaches such as centralized planning.

As a first step toward a generic methodology for embodied, distributed MAS, we analyse the behavior and the scalability of an embodied MAS which is driven by simulated potential fields. Resources in a storage facility emit a certain simulated potential while robots emit an opposite potential. Applying principles inspired by physics to local behavior, idle robots move toward resources but away from other robots. Whenever a resource appears, a central agent system registers this appearance and creates a plan for a robot to pick up the resource and deliver it to some destination area. We show that this system is highly scalable with respect to both the environment and the number of robots, while maintaining functionality, adaptivity and robustness. Furthermore, the integration of planning prevents local optima and increases fairness.

In future work, the central agent system will be replaced by a distributed system, such as a sensor network, to further increase scalability, adaptivity and robustness of the system.

1. INTRODUCTION
Recent years have seen increasing interest and advances concerning coordination of agents in distributed systems in both the agent-oriented software engineering (AOSE) community and the learning and adaptivity research community [4, 16, 18, 21]. Coordination of a large group of agents in a multi-agent system (MAS) is still an open issue in noisy, real-time, cooperative and competitive environments especially because many existing methods do not scale well with an increasing number of agents or increasing environmental complexity [18].

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

On the one hand, research on agent coordination from an AOSE perspective has mainly focused on core topics in the design of the software architecture and the role of the environment in MASs. On the other hand, research from the learning perspective has mostly focused on the fundamental question of how agents can learn to optimize their utility at the individual level and still coordinate their actions as a collective at the group level, i.e. showing global desired emergent behavior. As both lines of research meet each other along the ideas of swarm intelligence and stigmergy (the indirect communication taking place among individuals in social insect societies [10]), it is important to observe that our work is situated in the second perspective, i.e. the area of adaptive and learning agents. Most work so far in the learning agents area has focused on stateless decision games or matrix games from (evolutionary) game theory [11, 13]. Different approaches, varying from joint action learners to local Q-learners, have achieved satisfying results in game theoretic settings. For an overview, see [16, 18].

Recently, research interest has been shifting to more challenging real-world coordination problems for (learning) agents, i.e., having multiple states (multi-stage problems). Task assignment problems in grid worlds, i.e., the efficient distribution of a set of tasks among a group of agents (and the completion of these tasks) in highly dynamic grid environments, is one such important real-world multi-state optimization problem. Insights gained from state-of-the-art results in game theoretic matrix games, indicate that few solutions for this type of problem scale well with an increasing complexity of the problem domain and the number of agents. In particular, centralized planning approaches can quickly reach a point where the design of satisfying solutions becomes too complex and intractable. For this reason, a so-called distributed MAS, in which different adaptive agents of the system contribute to the emergent solution of the entire problem, is a conceptually attractive methodology. This emergent behavior depends on the notion that the usefulness of the collective (or system) is expected to increase as the individual agents optimise their own or local behavior.

Usually, individual agents in MAS literature contribute to some part of the collective through their individual actions. The joint actions of all agents then lead to some reward or utility from the outside world. To enable local learning, this reward has to be divided among the individual agents where each agent aims to increase its received reward. However, unless special care is taken with respect to how reward is assigned, there is a risk that agents in the collective work at cross-purposes. For example, agents can reach sub-optimal solutions by competing for scarce resources or by inefficient task distribution among the agents as each of them only considers its own goals (e.g., a Tragedy of the Commons [9]),

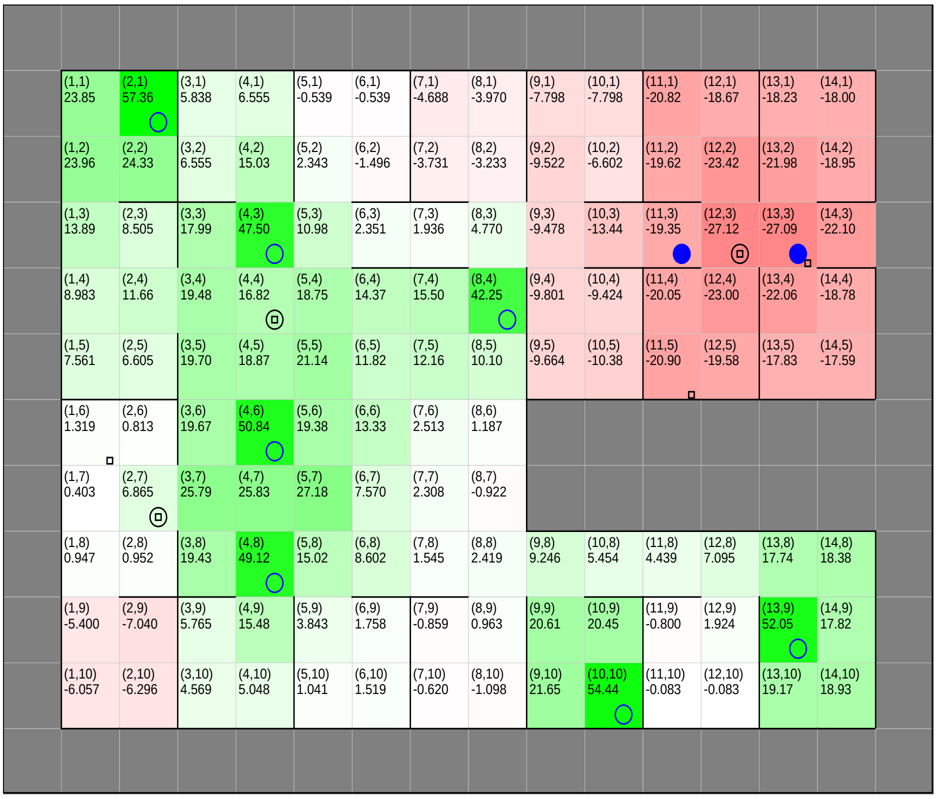


Figure 1: An example grid environment with various walls. Potentials are visualized by color and denoted in each cell, below the cell's coordinate. Gray cells lie outside the environment. Robots are visualized using circles (outlined for idle robots, filled for robots that are going to a resource). Resources are visualized using small black boxes. It can be seen that some resources are currently held by a robot (e.g., at (4,4)). Due to the negative potential in the upper right of the environment, some robots in the center area will automatically start moving in that direction.

or by policy oscillations [14]. In general, the performance of distributed MASs is difficult to predict. Furthermore, the outcome of the learning process can also be strongly dependent on the parameters of the learning algorithms used.

In this work, we analyse a scalable, nature-inspired methodology to deal with multi-state task assignment problems, based on the ideas of potential fields from physics, pheromones from biology and swarm intelligence from computer science.[1]

We investigate a specific class of multi-state task assignment problems, i.e. *resource distribution* in large storage facilities. Robots working in these facilities have to co-operate in a complex, dynamical environment and deal with unknown resource transportation requests, unexpected events and robot breakdowns. Thus, any control system for resource distribution should not only be functional, but also adaptive, robust and scalable. Following the classification of multi-robot systems (MRS) as discussed in [8], we are interested in cooperative, strongly coordinated systems performing so-called 'foraging' and 'object transportation' tasks. However, our current research does not focus on the physical difficulties involved in carrying out object transportation tasks.

Our research is divided in two distinct phases. In the first phase, we focus on scalability with respect to the number of robots and the size of the environment, allowing a central control approach. In the second phase, we study how we can further increase adaptivity and robustness by introducing distributed control. We believe that it is important to clearly divide both complex tasks, since the large number of different effects playing a role in both of these tasks could make results difficult to interpret. To investigate the first phase, we have developed a system in which the behavior of a large group of simple robots is guided by a central agent. The central agent (which for the robots essentially is part of the environment) monitors the environment, creates simulated potential fields and performs straightforward route planning tasks for robots that have to fetch and/or drop resources. The central agent is capable of replanning in case of failure (e.g., a robot got off-track).

In this paper, we will discuss this first phase simulator and various experiments, focusing on scalability. The second phase will not be addressed yet. The remainder of this paper is organized as follows. Section 2 gives an overview of related work and the opportunities and motivations for our own research. Section 3 discusses our methodology in combination with our simulated environment; Section 4 presents some initial experiments and results; in Section 5, we conclude and look at future work.

[1] Since the first two ideas are in essence the same apart from minor differences such as evaporation (which is observed in pheromones but not in potential fields), we will use the term potential fields in the remainder of this paper.

2. RELATED WORK

The idea of applying potential fields to the domain of *embodied MASs* (i.e., MASs with physical agents such as robots) is obviously not new; see, for example, [3, 5, 6, 19, 22]. Similar techniques have been shown to work properly on real robots (e.g., [17]). Although results are very appealing, they originate from research that mostly follows either an engineering perspective or a perspective purely committed to one technique. We are aiming at the development of a clear and powerful methodology for coordination in embodied MAS in general, instead of engineering a satisfying solution for a typical problem case or committing to a single technique. As the state of the art points out, such a methodology is currently lacking (as mentioned in [22]). This paper presents a first step toward this goal. This leads to a different focus in our research and two important differences between this work and earlier work.

First, for this type of research, it is very important to thoroughly examine the principles and effects underlying certain design choices (e.g., potential propagation functions, distributed control, adaptation of environmental properties) both empirically and theoretically. In related work, such an analysis is usually missing or incomplete (with fine exceptions, e.g. [19]). For instance, in many papers, the problem of local optima in potential fields is mentioned, but hardly ever elegantly solved [22]. In order to overcome such well-known problems of existing approaches, we are especially interested in investigating whether certain current claims on the integration of various approaches indeed hold. For example, Weyns et al. [22] point out that the integration of straightforward planning methods in a potential-field-based system is difficult. However, we show that this is not necessarily the case and that it can contribute to performance and fairness when local optima occur.At the empirical level, our initial experiments illustrate that (1) design choices regarding small details can have large effects on the success of the system as a whole; (2) the first version of our methodology leads to a highly scalable (and still functional) solution for resource allocation and resource distribution, and (3) the effects of local optima in our system are reduced, which increases fairness (i.e., all resources have to wait for a reasonable amount of time). At the theoretical level, we are especially interested in evolutionary game theory, which has shown to offer powerful tools (for instance evolutionary stability criteria) with which to understand complex interactions and emergent behavior in systems with a large number of agents.

A second difference in focus between existing work and our work is caused by our interest in the role of the environment in embodied, distributed MAS which uses nature as a source of inspiration. Even though many authors claim to use methods derived from biology (e.g., swarm intelligence, stigmergy), the agents are only loosely based on their biological counterparts. For example, in earlier work, agents are equipped with memory buffers [22] or the ability to perform simulated movement instead of actual movement [5], or they contain internal state machines [15]. Obviously, although such features make the agents much stronger, they also make them less generic and less biologically plausible. In our opinion, the (global) behavior of the environment must be emphasized and studied in order to facilitate drastic simplification of the (local) behavior of the agents inhabiting it. Therefore, we look at possibilities to build adaptivity into the environment or the agent(s) representing it, in order to make local behavior lead to desired global behavior. Some research has been conducted in the area of combining potential-based approaches with adaptive techniques, for example Reinforcement Learning (e.g., [15]). However, we will not take an explicit Reinforcement Learning approach, in which the different agents are equipped with a learning algorithm as for instance Q-learning, but instead we investigate learning capacities for the *environment*, in order to optimally guide the agents to the local and collective solution of the problem. In this paper, we manually adapt environmental properties, to illustrate that autonomous environmental adaptation can definitely contribute to performance.

3. METHODOLOGY AND SIMULATION ENVIRONMENT

In this section, we concisely explain our methodology for solving resource distribution problems, integrated in our simulation environment. As mentioned above, the key properties of our methodology (apart from functionality) are adaptivity, robustness and scalability. Adaptivity and robustness are realized primarily by using nature-inspired and environment-focused methods, e.g. potential fields [2, 12], which are simulated, maintained and communicated to robots using an auxiliary agent system.[2] Every resource that appears in our simulated system will emit a certain negative potential (proportional to its priority). Idle robots in turn emit positive potentials. Using adaptable potential propagation and enforcement methods, the negative and positive potentials spread over the environment and cause a force that attracts idle robots to resources, but repels them from each other. Once a resource is picked up by a robot, it immediately stops emitting potential. However, in the cell where the resource has been picked up, the potential decays only slowly. This leads to a *memory effect* in the environment, which contributes to optimal placement of idle robots if future resource locations conform more or less to historical locations. Other approaches (e.g., [6]) do not use such a memory effect.

Busy robots, i.e., robots that are heading toward a resource or a target location, are guided by straightforward path (re-)planning using Dijkstra's algorithm [7]. Here, using planning offers a few advantages over potential fields. Most importantly, using planning for the actual task assignment is beneficial for the system's performance in two ways. First, it helps to overcome the effects of local optima in potential fields; in purely field-based approaches, if multiple robots go after the same resource, none of them might be able to reach it because they are all trying to avoid one another and thus get stuck in local minima. In our case, whenever a robot is assigned to a resource, it immediately stops emitting potential. It will therefore not be repelled by other robots, nor disturb the potential field in areas it crosses while moving toward the resource. Second, in purely field-based approaches, remote areas often experience unfairly long waiting times [22]. With planning, we can make sure that the first resource to appear is also the first served, or take remoteness into account without increasing waiting times too much. Another advantage of planning is that it allows us to construct an optimal route for busy robots with linear time complexity (with respect to route length), as will be shown below. Finally, integrating planning behavior in our system is not difficult. It should be noted that this finding contradicts the view of Weyns et al. [22].

Since the potentials are directly related to the current state of the environment, we ensure that our system is indeed adaptive and robust (apart from possible breakdowns of our central agent, which will

[2] In the first phase of our research, these fields are maintained in a central agent (which essentially is an 'invisible' part of the environment). In the second phase, the central agent will be replaced by a distributed system, ensuring even more robust and adaptive operation. Both systems will use the same simple robot controllers, in line with our intention to develop robots with a biologically plausible complexity.

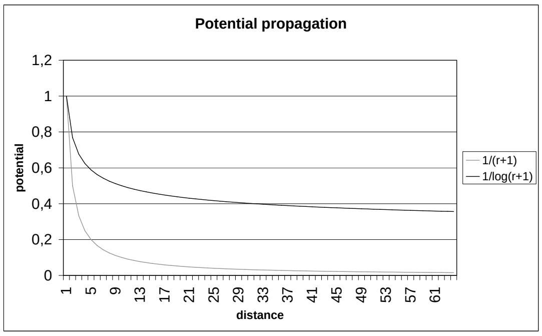


Figure 2: Potential propagation with inverse polynomial and logarithmic decay. In both cases, the potential decreases when the distance (in grid cells) increases. Inverse polynomial decay contains more directional information near the maximum, whereas logarithmic decay is able to distinguish direction further away from this maximum.

be addressed in the second phase of our research). It is adaptive because the potential field can change immediately whenever the environment changes. Due to a tunable memory effect, the field can either adapt fast and/or retain rich information about past experiences. Robustness is ensured by (1) spreading information over the environment (by potential propagation) and (2) the central agent's ability to replan in case a robot gets off track. In future work, we will support these claims with a set of experiments.

Scalability in our case should be ensured for two distinct entities, viz. environment (size) and robots (number). Many traditional MAS approaches unfortunately possess an exponential complexity with respect to these entities. In order to obtain a scalable approach, we need to ensure that it has at most a polynomial, and preferably even a linear, complexity here. In our case, both idle and busy robots are guided by a process for which the complexity is mostly related to the environment and only linearly dependent on the number of robots. This implies that our approach is indeed scalable with respect to the number of robots.

In order to facilitate the scalability of the environment, we impose a square grid structure on it. Each grid cell is relatively large (e.g., a square with sides of 4 meters). Due to walls, adjacent grid cells are not always neighboring. An example environment is illustrated in Figure 1 and example potential fields are illustrated in Figure 3. The grid structure has two obvious advantages. First, it makes both potential propagation and path planning calculations feasible even in large environments; after some initial (light) calculations, path planning can be performed in linear time (relative to path length) and potential propagation in quadratic time (relative to the number of grid cells), while preserving necessary information. Second, the grid enables us to abstract away some of the details of our robotic systems; we consider robots to have autonomous capacity to move to a neighboring grid cell and to pick up or drop a resource in a grid cell. In our simulated system, we use discretized time and assume that each action can be completed in exactly one time step.

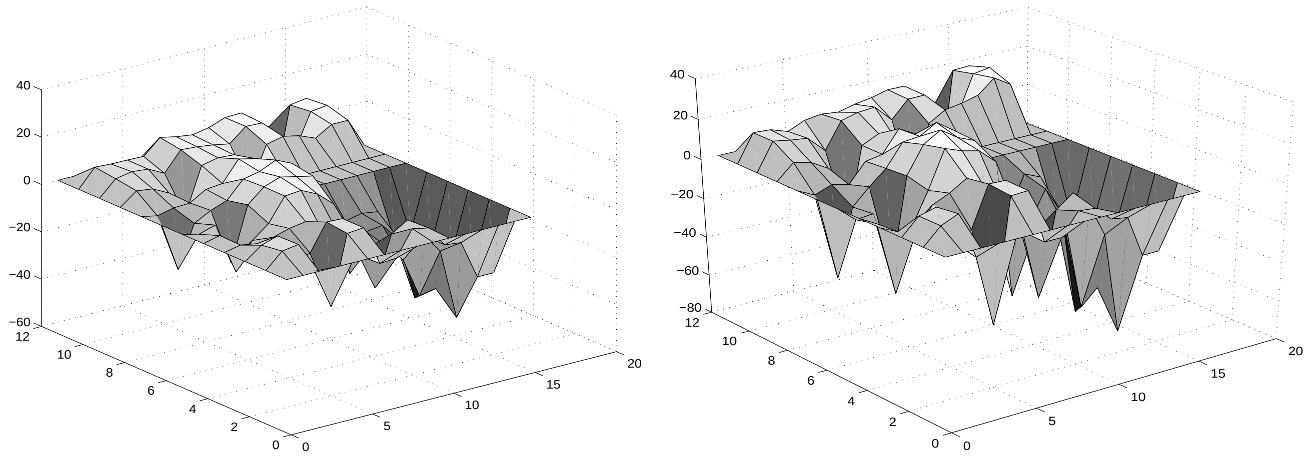
From the above, one might conclude that the grid actually makes the system polynomially time-complex with respect to the number of resources presented and the number of robots; after all, for each robot movement and for each resource appearance and disappearance, we have to recalculate the potential field, which is a polynomial process, related to the size of the environment. Furthermore, the number of robots needed is proportional to the resource appearance scenario and the size of the environment. However, the resulting polynomial complexity is obviously still due to the environment's size. In practical situations, this size should actually not be a bottleneck – when many cells are required, the problem is easily dividable into smaller subproblems by assigning restricted regions to robots and central agents. Additionally, practical limitations can make the environmental time complexity almost linear. For example, it is possible to impose a distance threshold on the potential propagation function used by the robots [3], since robots only need to be pushed away from each other when they are in each other's vicinity.

4. EXPERIMENTS

In our initial experiments, we focus on (1) the effects of selecting a potential propagation formula and a (simple) local behavior for robots and (2) empirical studies of the time complexity of our system. We do not yet focus on real applications of our proposed system. In future experiments, we will certainly do this. Furthermore, potential propagation and local behavior will be optimized using adaptive techniques such as reinforcement learning, and time complexity will be theoretically determined.

Changing the potential propagation formula employed by the environment can have large effects on the local configuration of the environment, and thus on the actions of robots responding to this local configuration. For example, if we use inverse polynomial decay for our propagation (i.e., $1/r$, in accordance with potential propagation in physics [2, 12]), cells in the neighborhood of cells with high (absolute) potential will show a steep gradient and will therefore include rich directional information. However, the directional information in cells that are further removed is limited. With a logarithmic decay ($1/\log r$), the effects are exactly opposite, as illustrated in Figure 2. The effects of these potential propagation formulae are illustrated in Figure 3 for an actual, complex environment. It is clearly visible that the formulae, while leading to similar landscapes, lead to different local configurations and different amplitudes. Due to these differences, an inverse polynomial decay is more sensitive to local optima in configurations that are far from optimal, but converges better in configurations that are almost optimal already.

Changing the local behavior (movement) of robots can obviously also have a large influence on the system's global behavior. In our system, the only input a robot receives consists of the potentials of (1) itself, (2) its own grid cell and (3) the (at most eight) neighbor



(a) Logarithmic decay. (b) Inverse polynomial decay.

Figure 3: Logarithmic and inverse polynomial decay in the environment depicted in Figure 1. After placing 1,000 resources in the environment (with a potential of -1 each, leading to an environmental potential of -1,000), 12 robots are added randomly (with a potential of +83.3 each, leading to a robot potential of 1,000). The resulting potential field is shown for both decay formulae.

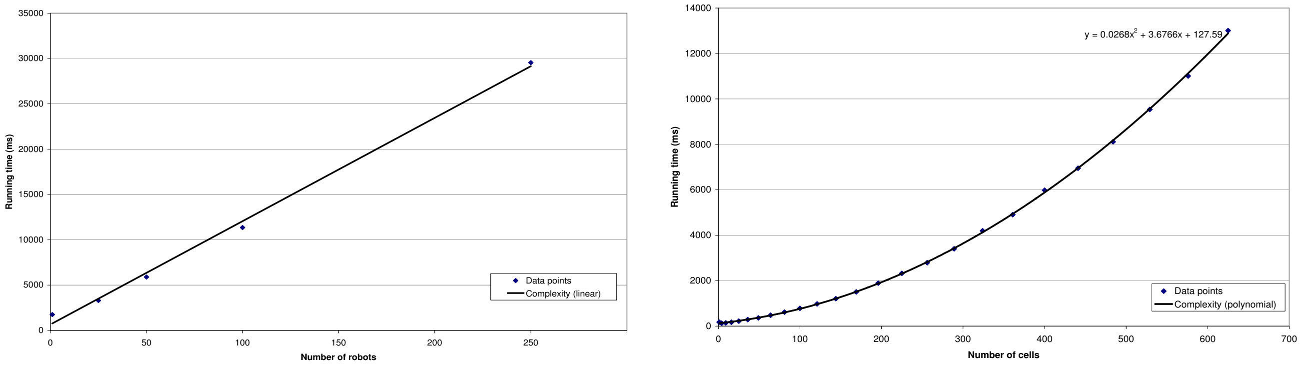
cells. Based on this input, many local behavior mechanisms can be devised; for example, greedy movement (the robot moves to the cell with the lowest potential), stochastic movement (the robot selects a cell with a probability equivalent to the cell's potential [6]), physically plausible movement directed by force vectors, or Boltzmann exploration [11].

In our experiments, we use both greedy and stochastic movement, both inverse polynomial and logarithmic decay and applied these to various environments. We observe that the system behaves in three distinct ways, regardless of the options chosen, viz. (1) when a resource appears, the nearest robot is assigned and brings the resource to its destination; (2) when a robot is (or becomes) idle, it automatically starts moving toward the area(s) where resources are most frequently offered; and (3) when no new resources are offered for a certain period of time, the robots are distributed over the environment in accordance with the distribution of past resource offerings. The first of these three behaviors is caused by the planning component integrated in the central agent. It ensures that every resource is actually picked up as quickly as possible, given the current configuration of robots. This automatically leads to a fair waiting time for each resource.

Compared to greedy movement, stochastic movement leads to more physical exercise for idle robots – although this often leads to 'senseless' moving, it also contributes to less frequently occurring local optima. Comparing inverse polynomial and logarithmic decay, we observe that the latter creates more 'readable' results in complex environments since potentials are usually higher. However, inverse polynomial decay seems to work better when robots are equipped with stochastic movement and close to resources (based on our observations of robot behavior in many cases). We informally experimented on a large variety of environments to assess whether any of the design choices clearly outperformed the other ones, but found that this is not the case; each of these choices has its strengths and weaknesses. This shows that a system's performance can definitely benefit from adaptive (environmental) properties.

Increasing the number of robots and the size of the environment, we observe that the time complexity of this system conforms to linear bounds with regard to the number of robots (see Figure 4(a)) and to polynomial bounds with regard to the number of grid cells (see Figure 4(b)). It should be noted that the polynomial complexity with respect to the environment only starts playing a role in very large environments. In this case, it would be beneficial to divide the environment in various regions, where each region is controlled by a separate potential field. When we increase both the number of robots and the number of grid cells at the same time, we observe that the system conforms to polynomial bounds as well, as can be expected. Since most (if not all) multi-agent planners have an exponential time complexity with regard to the number of agents and/or the size of the environment, these are promising and encouraging results.

Considering functionality, we see that our approach does a balanced and thorough job. A large-scale comparison of the performance of our system with existing systems is beyond the scope of this paper. However, we can observe that all resources are (in due time) transported to the correct locations, even if we let robots make deliberate errors in the execution of plans (for example, they move to (3,4) instead of (4,3)) and also if we periodically introduce malfunctions in one or more idle robots (for example, they are not able to move at all). Our approach does not lead to isolated areas in the environment, i.e. remote areas in which long waiting times occur between resource appearance and processing, as reported in the paper of Weyns et al. [22]. Since our approach uses straightforward planning to assign and guide idle robots to newly appeared resources, waiting times are not dependent on the remoteness of a certain area, but rather on (1) the number of robots (with less robots, it happens more often that all robots are busy and a new resource has to wait until current tasks have been completed) and (2) the quality of potential field and local behavior (the better this quality, the greater the probability that idle robots stand close to a newly appeared resource). The performance of the system (as well as its scalability, as has been mentioned earlier) might be increased even further by cleverly partitioning the environment and assigning robots to the partitions instead of to the entire environment [1].



(a) Experiments in a constant environment with 1 to 250 robots. For clarity, many data points are omitted – they conform to the same linear trend line shown without a significant change in trend line accuracy.

(b) Experiments with 12 robots in an environment with 1 to 625 grid cells. Note that the trend line is polynomial; however, the polynomial factor is very small (0.0268) and not prominent in realistic situations (≤ 400 cells).

Figure 4: Results on experimental complexity analysis, repeated 100 times per experiment on a P4-2400. Data points represent average running times (in ms). Standard deviations are too small ($\leq 100ms$) to display.

5. CONCLUSIONS

In this work, we discussed our initial steps toward a generic methodology for large-scale multi-agent systems. Using nature-inspired methods such as potential fields in combination with straightforward planning, we developed a highly scalable, robust and adaptive system that operates with a central agent and is able to deal with large resource distribution tasks. In this paper, we focused on scalability, the system's time complexity and an overview of design considerations for potential field propagation and local robot behavior.

In future work, we will examine whether there are potential propagation functions and/or local robot behaviors that can outperform other choices in general. If this is not the case, mechanisms must be developed that enable the environment to adapt to changing circumstances to facilitate optimal performance. Additionally, a definition of 'outperforming' must be given; do we want the average waiting time for resources to be minimized, do we want to increase fairness, et cetera. We are planning to investigate how Homo Egualis can help here [20].

Furthermore, we will replace our central agent with a distributed control system, for instance a sensor network, thus ensuring even more robustness and adaptivity while maintaining the functionality and scalability of the current centralized system.

Acknowledgments. The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. It is partially funded by the Breedtestrategie programme of the Universiteit Maastricht. Part of the research is being carried out under the European research grant FP6–013569 NiSIS (Nature-inspired Smart Information Systems), running until March, 2008. Thanks go to Ben Torben-Nielsen for plotting various graphs and Jahn-Takeshi Saito for proof-reading. Finally, we would like to thank the anonymous reviewers for their constructive comments.

6. REFERENCES

[1] Mazda Ahmadi and Peter Stone. Continuous Area Sweeping: A Task Definition and Initial Approach. In *The 12th International Conference on Advanced Robotics*, July 2005.

[2] Marcelo Alonso and Edward J. Finn. *Fundamental University Physics*, volume 2. Addison-Wesley, 1980.

[3] S. Brueckner and H. Van Dyke Parunak. Multiple Pheromones for Improved Guidance. *Proceedings of Symposium on Advanced Enterprise Control*, 2000.

[4] S. A. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, and R. Nagpal (eds). Engineering Self-Organising Systems: Methodologies and Applications. In *LNCS Postproceedings of ESOA 2004*. Springer LNAI 3464, 2005.

[5] Sven Brueckner and H. Van Dyke Parunak. Digital Pheromones for Coordination of Unmanned Vehicles. In Weyns et al. [23], pages 246–264.

[6] Sven Brueckner and H. Van Dyke Parunak. Swarming Distributed Pattern Detection and Classification. In Weyns et al. [23], pages 232–245.

[7] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[8] A. Farinelli, L. Iocchi, and D. Nardi. Multirobot systems: A classification focused on coordination. *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 5:pp. 2015, 2004.

[9] Garreth Hardin. The Tragedy of the Commons. In *Science, Vol 162, pages 1243-1248*, 2004.

[10] O.E. Holland and C. Melhuish. Stigmergy, Self-organisation, and Sorting in Collective Robotics. *Artificial Life*, 5(2):173–202, 1999.

[11] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 1996.

[12] O. D. Kellogg. *Foundations of Potential Theory*. Dover, New York, 1953.

[13] M.L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning, p 157 - 163*, 1994.

[14] A. Nowe and K. Verbeeck. Distributed Reinforcement Learning, Loadbased routing a case study. In *Notes of the Neural, Symbolic and Reinforcement Methods for sequence learning workshop at IJCAI99*, 1999.

[15] Liviu Panait and Sean Luke. A Pheromone-Based Utility Model for Collaborative Foraging. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 36–43, Washington, DC, USA, 2004. IEEE Computer Society.

[16] Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[17] David Payton, Mike Daily, Regina Estowski, Mike Howard, and Craig Lee. Pheromone Robotics. *Auton. Robots*, 11(3):319–324, 2001.

[18] P.J. 't Hoen, K. Tuyls, L. Panait, S. Luke, and H. la Poutré. An Overview of Cooperative and Competitive Multiagent Learning. In K. Tuyls, P.J. 't Hoen, K. Verbeeck, and S. Sen, editors, *Learning and Adaptation in Multi-Agent Systems*, pages 1–50. Springer LNAI 3898, 2006.

[19] H. Van Dyke Parunak, Sven Brueckner, and John Sauter. Synthetic Pheronome Mechanisms for Coordination of Unmanned Vehicles. In *Proceedings of the First Intenational Joint Conference on Autonomous Agents and Multiagent Systems*, pages 448–450, Bologna, Italy, 2002. ACM Press, New York, NY.

[20] K. Verbeeck., J. Parent, and A. Nowé. Homo Egualis Reinforcement Learning Agents for Load Balancing. *First NASA Workshop on Radical Agents Concepts - to be published in Lecture Notes on Artificial Intelligence Springer Verlag*, 2003.

[21] D. Weyns, H. Van Dyke Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for Multiagent Systems State-of-the-Art and Research Challenges. In D. Weyns, H. Van Dyke Parunak, and F. Michel, editors, *Environments for Multi-Agent Systems*, pages 1–47. Springer LNAI 3374, 2005.

[22] Danny Weyns, Nelis Boucke, Tom Holvoet, and Wannes Schols. Gradient Field-Based Task Assignment in an AGV Transportation System. In *EUMAS 2005 - Proceedings of the Third European Workshop on Multi-Agent Systems, Brussels, Belgium, December 7-8, 2005*, pages 447–458, 2005.

[23] Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors. *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, volume 3374 of *Lecture Notes in Computer Science*. Springer, 2005.