Cooperative goal-satisfaction without communication in large-scale agent-systems

Onn Shehory and **Sarit Kraus**[1]

Abstract. A framework for cooperative goal-satisfaction in large-scale environments is presented in this paper, focusing on a low complexity physics-oriented approach. The multi-agent systems with which we deal are modeled by a physics-oriented model. According to the model, agent-systems inherit physical properties, and therefore the evolution of the computational systems is similar to the evolution of physical systems. To enable implementation of the model, we provide a detailed algorithm to be used by a single agent within the system. The model and the algorithm are appropriate for large-scale Distributed Problem Solver systems, in which agents try to increase the benefits of the whole system. The complexity is very low, and in some specific cases it has proven to be optimal. The analysis and assessment of the algorithm are performed via the well-known behavior and properties of the physical system which models the computational system.

1 INTRODUCTION

Multi-agent systems are designed in order to satisfy goals. Goal-satisfaction may require cooperation among the agents, but cooperative goal-satisfaction may be beneficial even if the agents can perform goals by themselves. Traditional task-allocation methods [13] require coordination via communication [3]. However, in very large agent-communities there usually cannot be direct connection between all of the agents, as such a connection is too costly. Therefore, when the number of agents increases, the complexity of most of the cooperation methods becomes unbearable. To resolve the computational explosion of cooperation mechanisms in large-scale agent-environments, a different approach must be considered.

In this paper we apply methods from classical mechanics to model large-scale agent-systems. The physics-oriented methods are used to construct a beneficial cooperative goal-satisfaction algorithm to be used by the single agent within the system. Although there are many differences between particles and computational systems, we will show that using the physics-oriented approach results in a model that enables feasible cooperation and goal-satisfaction in very large agent-systems. In addition, the presented approach reduces the complexity of the cooperation mechanism and provides beneficial results.

This research discusses systems of agents that have goals which they must fulfill and resources that they can use in order to fulfill these goals. The fulfillment of goals is done either by single agents or by groups of agents, via cooperation. We focus on cases where autonomous agents work as a Distributed Problem Solvers (DPS) system and try to increase the common outcome of the system. A variety of algorithms for agent-cooperation as a DPS system have been presented [2]. Here we are concerned with very large systems of agents where communication is limited and negotiation processes for establishing cooperation are too complex. The solution that we provide for this problem is based on a physics-oriented approach which enable cooperation without negotiation and with limited communication [4]. We emphasize that our model provides a good solution for cases wherein many agents and many goals are present.

Example 1.1 *A simple example is a system in which agents must block holes of various sizes in a planar surface. Each hole to be blocked is a goal, and the filling for blocking holes is the resource of the agents. The purpose of the system is to block as much hole-area as possible. Some holes cannot be blocked by a single agent and cooperation is necessary.*

1.1 Assumptions, notations and concepts

We assume that the agents with which we deal have the ability to perceive the virtual[2] displacement in the goal-space, and can perceive the properties of other adjacent agents and goals. This may be done by sensors integrated into the agents. We also assume that each agent knows about the types of resources that other agents may have, but may be uncertain as to the particular resource-holdings of any other individual. These two assumptions are necessary since the agents are expected to propagate from state to state within the goal-space according to the properties of the surrounding agents, goals and obstacles. In order to enable such propagation, some knowledge regarding neighbours is necessary. We assume that each agent has a performance capability that can be measured using standard measurement units. The standard measurement will be used as a quantitative way of measuring the agents' success in fulfilling goals. In addition, we assume that there is a scaling method which is used to represent the displacements of the agents in the goal-space and to evaluate the mutual distances between goals and agents within this space. This assumption is necessary since virtual distances (or physical distances) are a significant factor in the model we present. We assume that goal-satisfaction can be achieved progressively. That is, a goal may be partially satisfied at one instant, and its remaining non-satisfied part may be complete at another point in time.

To present our model, we review concepts and notations from physics. The displacement vector of a particle i is denoted by $\vec{r}_i$. $\vec{v}_i$ denotes the velocity, and $\vec{a}_i$ denotes the acceleration. The kinetic energy of a particle i is represented by k_i, and the potential is represented by V. The potential is a spatial function and therefore is

[1] Dept. of Math and Comp. Science, Bar Ilan Univ., Ramat Gan, 52900 Israel, {shechory, sarit}@bimacs.cs.biu.ac.il, Tel: +972-3-5318863. Kraus is also affiliated with the Inst. for Advanced Comp. Studies, Univ. of Maryland.
This work is supported in part by the NSF under grant No. IRI-9423967 and the Israeli Science Ministry grant No. 6288.

[2] Since the goal-space is not necessarily physical, we do not assume physical distances and therefore call them virtual.

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

sometimes called a field of potential or a potential-well. Forces can be derived from the potential. Each particle i's mass is denoted by m_i, its displacement is denoted by the displacement vector $\vec{r}_i$, its momentum by $\vec{p}_i$ and the force that acts on it is denoted by $\vec{F}_i$.

1.2 Adapting physics to DAI

DAI	Physics
identifying the environments where physics-oriented models are appropriate; matching particle properties to agents/goals	locating particle models and their properties
selecting the matter-states that can be used to model automated-agents' systems.	identifying states of matter and the particle behavior within
developing algorithms for agents' goal-satisfaction; adjusting to the physics system for validity of the algorithm	using mathematical formulation to predict and describe the properties and evolution of the selected particle model
analysis of the complexity and properties of the algorithm	theoretical and simulation-based analysis of physical particle systems behavior

Table 1. Distributed AI and Physics for DPS environments

In the DAI environments that we consider, there is a large set of agents and a large set of goals they need to satisfy. Each agent has capabilities and should move toward satisfying goals. We use a physics model that consists of particles which represent the agents and the goals, and to develop a distributed cooperative goal satisfaction mechanism. The first step in applying the physics model to DAI is the match between particles and their properties, agents and their capabilities, and goals and their properties (see table 1). The next step is to identify the state of matter for modeling a community of agents and goals. The mathematical formulation that is used by physicists either to describe or to predict the properties and evolution of particles in these states of matter, will serve as the basis for the development of algorithms for the agents. However, several modifications of the physics model are necessary to provide an efficient algorithm for automated agents.

In our model, agents and goals are modeled by dynamic particles and static particles, respectively. The match between particle properties and agent/goal properties is described in table 2. We model goal-satisfaction by a collision of dynamic particles with static particles. However, the properties of particle-collisions are different from the properties of goal-satisfaction and several adjustments are needed in order to provide the agents with efficient algorithms. These modifications are described in detail in this paper.

1.3 Related work

The path-planning and robot-navigation (PPRN) research[3] have used potential fields as a means for planning the path, e.g., in [16, 8, 7]. This approach appears to be closely related to our research. However, there are several significant differences between the path-planning and robot-navigation problem and our task-allocation and agent-coordination problem, and the techniques that are used for solving these problems. The differences are as follows:

1. In PPRN research, the main objective is planning an optimal path for the robots to navigate from an initial location to the destination.

[3] A comprehensive overview of these can be found in [9]

Automated Agents	Physics Model
community of agents satisfying goals	non-ionic liquid system
agent	dynamic particle
goal	static particle
agent's capabilities	particle's mass
agent's (virtual) location in agents-goals space	location of particle
goal satisfaction	static-dynamic collision
algorithm for goals allocation	formal method for calculating the evolution of displacement

Table 2. The match between the physics model components and the large-scale automated agents environments

Our main objective however is to solve a task allocation problem with aspects of agent-coordination in a multi-agent environment.

2. While we discuss the multi-agent case, with potentially hundreds of agents, the type of planning research that is involved with potential functions usually discusses the single robot case. In cases where the more-than-one robot case is discussed, the number of robots is considered very small as opposed to the agent-systems we discuss.
3. In cases where the PPRN research addresses the multi-robot planning problem, e.g. in [5, 10, 1, 15], the potential field concept is not employed. In such cases the behavior of groups and formations of robots are discussed, given a set of specific strategies according to which the robots act. Among these strategies you may find some in which robots follow a leader or some predefined geometric patterns. In our multi-agent model however the agents' strategy is based on the physical potential-well concept.
4. Although both the PPRN and our research use potential fields for problem representation and resolution, the type of potential functions and the way of using them is different, as follows:
   - The potential functions employed for path-planning are artificial. The only expected result from such potential functions is robot-motion according to the potential field gradient. In our research, we employ physical potential functions. This results in a physical behavior of the agents that act with respect to these functions. In particular, the use of the potential functions of particles in a non-ionic fluid results in a model that provides the single agent with an algorithm for reaching and performing goals within a large community of agents. In addition, the use of such potential functions enables the prediction of the bulk properties (i.e., the behavior) of the agent-system as a whole.
   - In the PPRN research, an attractive potential field (usually quadratic) is employed to lead the robot to the goal, and various shapes of repulsive potential fields are used to cause obstacle-avoidance. In our work, all of the entities –agents as well as goals– (and obstacles, if present) are modeled by the same type of potential function; i.e., by a physical potential-well.
   - The potential-wells in our model may change dynamically due to the fulfillment of goals and the expenditure of resources. This leads to a dynamically alternating potential field which results in a dynamic update of the agents' behavior. Dynamics of the potential field in PPRN research, when present, refer only to the change in the locations of robots and obstacles, and not to a change in the specific function that models a specific entity.
5. Another important difference of our research, as compared to the path-planning research, is that we do not restrict the model to physical trajectories – the model can be used for abstract motion.[4]

[4] The concept of abstract motion shall be explained in the next section.

In summary, the PPRN research with artificial potential functions discusses cases where a single robot or a small number of robots must navigate and locate their goals. Our approach is very different: we discuss cases of large-scale agent-systems, with many agents involved; cases where a specific agent does not have a specific goal towards which it must navigate; cases of cooperative goal satisfaction. These are not the aim of the PPRN research and therefore are not discussed in it.

Glance and Huberman [6] present a detailed physical formalism of the dynamics of the collective action of a system of individuals. In our work the main issue is the physical behavior of the single agent. Shoham and Tennenholtz [12] presented results of simulations that were performed in order to perceive the emergence of conventions in multi-agent systems. In our research, we discuss emergent cooperation and determine the social laws to be such – –physical laws– – that they will cause the emergent cooperation of the system when this cooperation is necessary. Mataric [10] proposes defining a set of basic interactions that will allow the simplification of group behavior analysis. In our work, we concentrate on the nature of the basic interactions and adopt the physical interactions among particles to model the interactions among agents and goals.

The issue of allocating agents to goals has widely been discussed among DAI researchers. The Contract Net Protocol [13] uses negotiation based on task announcements, bids and contracts for task allocation. While the CNP is based on the exchange of information, the model we present minimizes the transmitted information and thus enables large-scale systems to be efficient. A study of planning in large-scale agent-systems has been presented in [17]. In that research, the general-equilibrium approach from economics serves as the theoretical basis for the planning mechanism. We also discuss large-scale systems and apply an analytical model for designing the distributed planning mechanism, however we use a physics-oriented approach for DPS systems, not for competitive agents.

2 THE PHYSICS-ORIENTED APPROACH TO MODELING AGENTS

Classical mechanics provides a formal method for calculating the evolution of the displacement and the momentum of classical particles. For a particle i, the equations of motion are:

$$\vec{F}_i = m_i\ddot{\vec{r}}_i = m_i\vec{a}_i \quad \text{and} \quad \vec{p}_i = m_i\dot{\vec{r}}_i = m_i\vec{v}_i \tag{1}$$

The motion of a particle depends on the field of potential in which it moves and the force $\vec{F}_i = -m_i\vec{\nabla}_{r_i}V(\vec{r})$. The model we present entails treating agents, goals and obstacles as particles. That is, each agent will have its equations of motion and an initial state. Note that an agent's equations of motion do not necessarily entail real physical motion. The potential field in which an agent acts represents the goals and the other agents in the environment. Subject to the potential field, agents solve the equations of motion and, according to the results, progress towards the solution of goals and either cooperate or avoid conflicts with other agents. The cooperation and conflict-avoidance are emergent properties of our physics-oriented model.

An appropriate physical system must consist of a potential that, when adapted to the agent-model, will lead the agents to successful and beneficial goal-satisfaction. The liquid model is most appropriate for our systems. As opposed to the solid state, a liquid system can evolve from its initial state into new, different states. However, unlike the gas state, a liquid is dense enough to cause frequent interaction among its particles. In non-ionic liquids the mutual potential among each pair of particles is proportional to r^{-n} for any $n > 1$. We have chosen a typical potential of a particle i in a non-ionic liquid corresponding to its distance r_{ij} from particle j, the Lennard-Jones potential:

$$V(\vec{r})_{LJ_{ij}} = 4\varepsilon\left(\frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^{6}}\right) \tag{2}$$

This potential diminishes after a short distance, thus implying that the interaction between the particles in the system is limited to short distances.

3 THE PHYSICS-AGENT-SYSTEM (PAS) MODEL

The DPS system with which we deal is modeled by a set of particles and a potential field. The agents in the DPS system are modeled by dynamic particles and their potential-wells. The goals and the obstacles are modeled by static particles which are represented by fixed potential-wells. The superposition of the potential-wells of the particles, either agents or goals and obstacles, constructs a potential field. The particles move according to the field of potential and their own properties.

In the PAS model, the agent's capability of satisfying goals is represented by the mass of the particle that models it, and therefore by the potential-energy $k = mv^2/2$, which is a product of the mass, as well. Particles with a greater potential-energy model agents that can satisfy larger or more difficult goals and sub-goals. This means that a greater mass of a dynamic particle that models an agent (other properties remaining constant, and thus causing a greater potential-energy), entails a larger capability of goal-satisfaction by the agent. The mass of a fixed particle (which models a goal or an obstacle) represents the size of the goal or the obstacle. This means that in order to satisfy a greater goal, which is modeled by a particle with a greater mass, more efforts are necessary on the part of the agents.

The displacement vector of a particle $\vec{r_i}$ models the displacement of the agent in the goal-space. According to the virtual displacement of an agent, its distances from other agents, goals and obstacles can be calculated. The potential is calculated according to these distances. The momentum vector $\vec{p_i}$ of particle i represents its physical velocity and is used for the calculation of the kinetic energy. In the PAS model, the velocity of a dynamic particle represents the rate of movement towards the satisfaction of a goal or a part of a goal.

Example 3.1 *Recall of the example in section 1. According to PAS, each hole is modeled by a particle, represented by a potential-well. The hole-size is represented by the mass. Due to the physical nature of the potential-well, particles surrounding a well with a greater mass will experience stronger attraction to the well. This property is appropriate for our purposes. The agents in this system are also modeled by particles, represented by potential-wells which are free to move. The agents have a variety of masses which represent their various abilities to block holes.*

3.1 Virtual motion towards goal-satisfaction

In the physical world, the motion of particles is caused by the mutual attraction between them. In the agents' system, the agents calculate the attraction and move according to the results of these calculations. The reaction of a particle to the field of potential will yield a change in its coordinates and energies. In our model, each agent will calculate the effect of the potential field on itself by solving a set of differential equations. According to the results of these calculations, it will move to a new state in the goal-domain (section 3.3).

The steep decay of the potential function beyond a short distance from the center of the potential-well results in derived weak forces and negligible interaction. Physicists have shown that when the long-distance interactions are neglected, the results of simulations still agree with theoretical statistical-mechanics and thermodynamics [14, 11]. Therefore, it is common to cut off the range of interaction by cutting off the potential function after it diminishes to from 1 to 10% of its maximal value. The radius of interaction (and of the cut-off) is denoted by r_I.

Agents will use numerical integration to solve the equations of motion that they must solve, with respect to time. The integration must be iterated frequently and performed with small time-steps dt. We determine the size of the time differential dt relying on the experience gathered in physics simulations [11]: we demand that a typical particle in the model will pass a distance of r_0 in $\sim$ 10 time-steps dt. This requirement implies that the average velocity $\overline{v}$ of a particle (at its initial displacement) directly affects dt by the relation $dt = r_0/\overline{v}$.

3.2 Collision and goal-satisfaction

The dynamics of the physical system which models the computational system leads to collisions between particles. Two types of collisions are possible: a collision between two dynamic particles, which we denote by DDC, and a collision between dynamic and static particles, denoted by SDC. In our model, the DDC represents the interaction between two agents. In order to prevent situations where agents overlap, the particles that model the agents have a mutual repulsion. The decision on which agents shall perform a specific goal will emerge from the repulsion. Dynamic particles that model agents shall have a potential that consists of a dominant repulsive component.

The SDC represents agent-goal interaction. In such interactions we would like the static particle that models the goal to attract the dynamic particle that models the agent. Adopting physical concepts, we use the notion of typical radius to specify the point from which the particle starts the collision. A typical radius σ of a particle is usually taken to be the distance from its center to the point wherein the force is zero. A simple calculation yields $\sigma = 4\varepsilon\sqrt[6]{2}$. An SDC occurs when a dynamic particle is in the vicinity of a static particle. Vicinity here means that the distance between them is a few typical radii (r_0).

The goal-satisfaction is performed during the collision. An agent that reaches a goal may either completely or partially satisfy it. In both cases, the model requires a reduction in the magnitude of the goal. This implies that the mass of the modeling particle shall be reduced, but mass-reduction is not a physical property of such a collision. Therefore, some modifications of the model shall be done, as long as they do not affect the general evolution of the system. This will be possible if the model consists of a scheme for a temporal partition of the evolution of the system. This means that the evolution of the system will be partitioned into several time segments (different from dt, much longer), and in each temporal segment the physical evolution of the system will not depend on the other segments.

Our model requires the adjustment of the physical collision time to the goal-satisfaction time. The central repulsive part of the potential-well will cause a gradual relaxation of the particle that will have reached the well, until the particle stops. Time is required for the relaxation process, which will model the goal-satisfaction time.

3.3 A protocol for the single agent

In order to cause evolution of the system towards goal-satisfaction, each agent uses the information that it can gather by observation (e.g., via sensors) about its neighboring agents and goals and regarding its previous state. According to this information, the agent will construct the local field of potential and solve the equations of motion. The results of the equations of motion will enable the agent to decide what its next step towards goal-satisfaction will be. The exact detailed algorithm for the single agent i is as follows:

Loop and perform the goal-reaching and goal-satisfaction processes until the resources necessary for satisfying goals have been depleted or no goals within the interaction range r_I have been observed for several time-segments.

Goal-reaching process

1. Advance the time counter t by dt.
2. Locate all of the agents and goals within the range r_I, the predefined interaction distance. Denote the distance to any neighbouring entity j by r_{ij}.
3. Calculate the mutual Lennard-Jones potential (using equation 2) with respect to each of the agents and goals within the range.
4. Sum over all of the pairwise potentials $V_{LJ}(r_{ij})$ and calculate the gradient of the sum to derive the force F_i.
5. Using F_i and the previous state $\vec{r_i}(t-dt), \vec{p_i}(t-dt)$, solve the equations of motion as described in section 2, in equations 1.
6. The results of the equations of motion will be a new pair $\vec{r_i}(t), \vec{p_i}(t)$. Move to the new state that corresponds to the displacement $\vec{r_i}(t)$.
7. At each time-step, after moving to a new state, calculate the new kinetic energy and potential according to the new coordinates $\vec{r_i}(t), \vec{p_i}(t)$.
8. If your distance from the center of a particle that models a goal is greater than r_0, return to step 1. Otherwise, start the goal-satisfaction process.

The goal-satisfaction process

After reaching a goal, the agent must satisfy all or at least parts of it:

- Move into the potential-well that models the goal according to the physical properties of the entities involved in the process and perform the goal.
- Subtract the mass of the particle that models the agent from the mass of the particle that models the goal. Return to step 1.

The iterative method which we propose leads to a gradual reduction in the amount and size of the goals to be satisfied, and will lead finally, to completion of the goals.

4 COMPLEXITY

The time that each agent consumes at each time-segment for calculating its propagation in the goal-domain is equal to the number of time-steps dt that comprise the time-segment, multiplied by the time necessary for the calculations within the time-step. We denote the number of agents by N, the number of goals by G, the total area of the goal-domain by S, the density by n and the average virtual distance between agents and goals by d. The average numbers of agents and goals within the range of interaction are given by:

$$N_I = \frac{N}{N+G} n\pi r_I^2 \text{ and } G_I = \frac{G}{N+G} n\pi r_I^2 \tag{3}$$

When the size S of the goal-domain is fixed, in can be shown that the single-agent time-consumption C_t per time-segment is:

$$C_t \sim (N_I + G_I)\text{Itr} \sim \sqrt{N+G} \tag{4}$$

However, it may be that both the number of the agents and the size of the system have simultaneously been increased. In such cases we can derive

$$C_t \sim (N_I + G_I)\text{Itr} \sim \text{const} \tag{5}$$

The number of time-segments for satisfying all of the goals will be, in the worst case, $O(G)$. However, since all of the agents are working simultaneously on goal-reaching, Itr usually decreases. This reduction is bounded by $\frac{G}{N}$, which is the optimal number of time-segments for satisfying all of the goals when $N < G$. When an agent has reached a goal, other agents have also progressed toward goals. Therefore in the next time-segment they will have to move only the remaining distance. The sum of the distances during the whole goal-satisfaction process is proportional to $\frac{G}{N}\sum_{i=1}^{G}\frac{1}{x^i}$. This sum has a constant upper bound equal to $\frac{1}{x-1}$. Resulting from this sum are both the convergence of the algorithm to a solution and the expected complexity of the general procedure of goal-reaching. In cases where equation 4 holds, this complexity is $O(G\sqrt{N+G})$ in the worst case and $O(\frac{G}{N}\sqrt{N+G})$ in the average case. In cases where equation 5 holds, the complexity is $O(G)$ in the worst case and $O(\frac{G}{N})$ in the average case. In a case that $N > G$, the complexity is $O(\sqrt{N+G})$ when equation 4 holds and $O(1)$ when equation 5 holds.

We discuss hereby the effect of the time necessary for goal-satisfaction on the performance of the algorithm. We shall differentiate between cases subject to the following parameters: size of goals; size of agents; time consumed for goal-reaching (denoted by t_r); time consumed for goal-satisfaction (denoted by t_g). In cases where all of the goals are of the same size and all of the agents are such that each agent satisfies a goal with the same efforts and time consumption, the complexity of the algorithm for the single agent is $O(\frac{G}{N})$ in the average case and $O(G)$ in the worst case. However, t_g and t_r affect the multiplying constant. This has a significant effect on the complexity in the case that $t_g \gg t_r$, where the multiplier is $\frac{t_g}{t_r}$.

When we have goals of various sizes, we must modify the analysis above. We denote the goal size by S_g and the agent size by S_a. The number of goals per agent μ is given by $\overline{\mu_i} = \frac{\sum_j S_{g_j}}{S_{a_i}\sum_k S_{a_k}}$ and $\mu_i^w = \frac{\sum_j S_{g_j}}{S_{a_i}}$ in the average and worst cases, respectively. The ratio-bound in this case is $2/(1-\delta)^\mu$, where μ is the number of goals that an agent has approached during the goal-satisfaction process and $\delta \ll 1$. To summarize the case where $t_g \ll t_r$, the complexity is the number of goals per agent multiplied by the ratio-bound, i.e., $2\mu/(1-\delta)^\mu$. This analysis holds for the case where $t_g \sim t_r$, as well.

In case that $t_g \gg t_r$, since the time consumption for goal-satisfaction is much greater than the goal-reaching time, we shall ignore the latter. The goal-satisfaction time of an agent is the sum of the time-periods consumed for the satisfaction of goals by this agent. This sum depends on the goal allocation, which depends on the size of the agent and the sizes of the goals. Therefore, it will be $\frac{t_g}{t_r}\mu$.

5 CONCLUSION

The problem of the behavior of agents in very large agent-societies imposes difficulties that are hard to solve even when the proposed solutions are of low-order polynomial complexity. The approach which we present suggests a solution for this problem, since the complexity on the side of the single agent is very low and may even be $O(1)$. Such results are possible since we use a model whose behavior is already known. Therefore, we are not required to perform the numerous explicit calculations that would have otherwise been necessary.

The model we have presented, and the algorithm that enables the single agent to act according to the model, are methods with which the agents allocate themselves to goals in order to satisfy the goals. The agent-goal matching is an emergent result of the physics-oriented behavior of the agents. In cases where too many agents fit the requirements of the same goal, our model will disenable some of them from reaching the goal, via the property of mutual rejection. As we have shown, our algorithm leads to agent-goal allocation, it converges to a solution, its complexity is low and no communication is necessary. In addition to these properties, we have proven that the algorithm we provide performs relatively close to the optimum. Our method does not lead to the optimal allocation, but reaching an optimal allocation requires complete on-line information about all of the agents and goals comprising the system, and an exponential computation-time.

While common DAI algorithms must be checked for their validity either by a formal proof or by simulations, our model can rely on theoretical and experimental results that are already known from physics. According to these results, we can predict the evolution of the modeled agent-system, since it will evolve in the same manner as a corresponding physical system. The local interactions, which enable one to derive the global behavior of the system, assure a low computational complexity of the model. In very large-scale agent-systems, this approach provides a model that promises emergent cooperative goal-satisfaction activity. In addition, the properties of the system as a whole can be analyzed using concepts from statistical mechanics. The employment of such concepts enables us to derive the bulk properties of a system via the properties of its components. We leave this analysis for future work.

REFERENCES

[1] Q. Chen and J. Y. S. Luh. Coordination and control of a group of mobile robots. In *Proc. of the 1994 IEEE international conference on Robotics and Automation*, pages 2315–2320, 1994.

[2] E. H. Durfee. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston, 1988.

[3] E. H. Durfee, V.R. Lesser, and D. D. Korkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, 36(C):1275–1291, 1987.

[4] E. Ephrati, M. Pollack, and S. Ur. Deriving multi-agent coordination through filtering strategies. In *IJCAI95*, pages 679–685, 1995.

[5] D. W. Gage. Command control for many-robot systems. *Unmanned systems*, Fall:28–34, 1992.

[6] N. S. Glance and B. A. Huberman. The outbreak of cooperation. *Journal of Mathematical Sociology*, 17(4):281–302, 1993.

[7] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International J. of Robotics Research*, 5(1):90–98, 1986.

[8] P. Khosla and R. Volpe. Superquadratic artificial potentials for obstacle avoidance and approach. In *Proc. of the 1989 IEEE international conference on Robotics and Automation*, pages 1778–1784, 1988.

[9] J. C. Latombe. *Robot Motion Planning*. Kluwler Academic Pub., 1991.

[10] M. J. Mataric. Kin recognition, similarity, and group behavior. In *Proc. of the 15th Cognitive Science Society Conf.*, pages 705–710, 1993.

[11] D. C. Rapaport. Large-scale molecular dynamics simulation using vector and parallel computers. *Comp. phys. reports*, 9(1):1–53, 1988.

[12] Y. Shoham and M. Tennenholtz. Emergent conventions in multi-agent systems: initial experimental results and observations. In *Proc. of KR-92*, 1992.

[13] R. G. Smith. The contract net protocol: high-level communication and control in a distributed problem solver. *IEEE Transaction on Computers*, 29(12):1104–1113, 1980.

[14] C. Trozzi and G. Ciccotti. Stationary nonequilibrium states by molecular dynamics. II. Newton's law. *Phys. rev. A*, 29(2):916–925, 1984.

[15] P. Wang. Navigation strategies for multiple autonomous robots moving in formation. *Journal of Robotic Systems*, 8(2):177–195, 1991.

[16] C. W. Warren. Global path planning using artificial potential fields. In *Proc. of the 1989 IEEE international conference on Robotics and Automation*, pages 316–321, Scottsdale, AZ, 1989.

[17] M. P. Wellman. A general-equilibrium approach to distributed transportation planning. In *Proc. of AAAI-92*, pages 282–289, 1992.