Newsha Khani[1]
Department of Aerospace
and Mechanical Engineering,
University of Southern California,
3650 McClintock Avenue, OHE430,
Los Angeles, CA 90089-1453
e-mail: nkhani@usc.edu

James Humann
Department of Aerospace
and Mechanical Engineering,
University of Southern California,
3650 McClintock Avenue, OHE430,
Los Angeles, CA 90089-1453
e-mail: humann@usc.edu

Yan Jin
Department of Aerospace
and Mechanical Engineering,
University of Southern California,
3650 McClintock Avenue, OHE430,
Los Angeles, CA 90089-1453
e-mail: yjin@usc.edu

Effect of Social Structuring in Self-Organizing Systems

Dealing with unforeseeable changing situations, often seen in exploratory and hazardous task domains, requires systems that can adapt to changing tasks and varying environments. The challenge for engineering design researchers and practitioners is how to design such adaptive systems. Taking advantage of the flexibility of multi-agent systems, a self-organizing systems approach has been proposed, in which mechanical cells or agents organize themselves as the environment and tasks change based on a set of predefined rules. To enable self-organizing systems to perform more realistic tasks, a two-field framework is introduced to capture task complexity and agent behaviors, and a rule-based social structuring mechanism is proposed to facilitate self-organizing for better performance. Computer simulation-based case studies were carried out to investigate how social structuring among agents, together with the size of agent population, can influence self-organizing system performance in the face of increasing task complexity. The simulation results provide design insights into task-driven social structures and their effect on the behavior and performance of self-organizing systems.
[DOI: 10.1115/1.4032265]

1 Introduction

Adaptability is needed for systems to operate in harsh and unpredictable environments where it is impossible for the designer to conceptualize every possible incident or predict details of changing functional requirements. Example scenarios include exploratory missions and system deployment in hazardous environments. An alternative approach for developing adaptive engineered systems is to embrace complexity, like biological systems, in which local interactions are not completely known but follow a set of rules and regulations, allowing the system-level function to *emerge* from these local interactions. Following this idea, a cellular self-organizing (CSO) systems approach has been proposed [1–4]. A CSO system is composed of multiple homogeneous or heterogeneous mechanical cells or agents that can be a small functional component or a robot. Each agent is equipped with needed sensors and actuators and encoded with system design-DNA (dDNA) containing the information that specifies cellular behavior. Agents interact with their task environment and with each other, leading to self-organizing emergent behavior and functionality at the system level. To facilitate agents' interactions with the task environment, a task field-based regulation (FBR) mechanism has been developed [3].To explore agent interactions, a cohesion, avoidance, alignment, randomness, and momentum parametric model has been examined [2,5].

The self-organizing behavior of the current CSO systems is caused by each agent's transforming the task environment into a task field in which it finds its optimal location, similar to an organism finding a desirable environment. The system-level task is completed by the collective effort of the agents' individually seeking their optimal locations. This distributed and self-interested approach allows for *flexibility* to cope with changing tasks, *robustness* to deal with a changing environment, and *resilience* to still operate event after system damage or malfunction.

The current FBR approach to self-organization [4,6] has limited capabilities because it does not directly address the interaction between agents with respect to the task context, leaving the power of agents' self-organized structures underutilized. Social structures play an important role in solving collective tasks [7,8]. In this research, we explore a dynamic social structuring approach to enhance the functionality of self-organizing systems. More specifically, we attain dynamic social structuring among agents by introducing both general and context-based social rules and devise a social rule based regulation (SRBR) for agents to choose their actions. To facilitate the SRBR, we introduce the concept of "social field" in addition to the current "task field." In SRBR, an agent's behavior is adjusted based on its perceived social field. Social rules can be designed to increase system functionality by resolving possible conflicts. However, the introduction of social interactions leads to its own set of challenges: (1) different social interaction rules may be necessary for large and small sets of agents, respectively; (2) certain rules may increase system reliability, at the cost of efficiency; and (3) the percentage of agents who follow or ignore the rules may also impact the system performance depending on the task complexity and the size of the agent population. The goal of this research is to investigate the interplays among *social interaction rules* (i.e., social structuring), *agent population size*, *task complexity*, and *system performance,* measured by *quality* and *efficiency*, with an intent to generate design insights for developing self-organizing systems, as illustrated in Fig. 1.

In the rest of this paper, we first review the related work in Sec. 2, and then, in Sec. 3, discuss an emergence-based approach to attain system complexity. Sections 4 and 5 introduce our dynamic social structuring concepts and present our social rule based

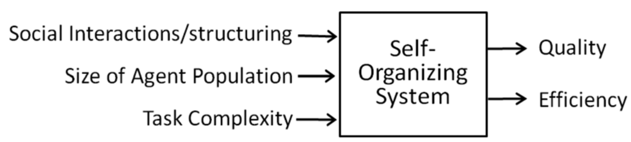

Fig. 1 Interplay among social structuring, size of agent population, task complexity, and system performance

[1]Corresponding author.

Contributed by the Design Theory and Methodology Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received March 12, 2015; final manuscript received November 20, 2015; published online February 23, 2016. Assoc. Editor: Andy Dong.

behavior regulation approach, respectively. In Sec. 6, we demonstrate the effectiveness of our approach through simulation-based case studies. Section 7 draws conclusions and points to future research directions.

2 Related Work

In the field of engineering design, design for adaptability and design of reconfigurable systems have been investigated in the past decade. In their work focusing on vehicle design, Ferguson and Lewis [9] introduced a method of designing effective reconfigurable systems, which focuses on determining how the design variables of a system change as well as investigating the stability of a reconfigurable system through the application of a state-feedback controller. Martin and Ishii [10] proposed a design for variety approach that allows quick reconfiguration of products and aim to reduce time to market by addressing generational product variation. Indices have been developed for generational variance to help designers reduce the development time of future evolutionary products. In addition to developing design methods for reconfigurable systems, various reconfigurable robotic systems have been developed mostly by computer scientists. Rus and Vona [11–13] at MIT introduced numerous different modular robotic systems. Fukuda and Nakagawa [14] developed a dynamically reconfigurable robotic system (DRRS). Unsal et al. [15] focused on creating systems of simplistic "i-Cubes," which can attach to one another [16–18]. PolyBot [18] acquired prominence by being the first robot that "demonstrated sequentially two topologically distinct locomotion modes by self-configuration." SuperBot [19] is composed of a series of homogeneous modules each of which has three joints and three points of connection. Control of SuperBot is naturally inspired and achieved through a "hormone" control algorithm. In the robotics literature, extensive work has been conducted on distributed approaches and cooperation among robots [20–23].

Despite the implicit and informal nature of some multi-agent relations, all multi-agent systems possess some form of organization. For a distributed system with a specific task, an organized way of sharing information among agents can be very helpful. Organizational oriented design has shown to be effective and is typically used to achieve better communication strategies [24]. Researchers have suggested that there is no single type of organization that is a best match for all circumstances [25–27]. It has been proved that the behavior of the system depends on shape, size, and characteristics of the organizational structure [26,28–33] and strategies of coordination among agents [34,35].

Our previous work on CSO systems has provided useful insights into understanding self-organizing systems and introducing nature-inspired design concepts. A CSO system is based on some key concepts including a dDNA capturing all design information in a bit-string [1], parametric behavioral models for agent control and optimization [36,37], and an FBR mechanism [4].

The current FBR approach is fully distributed since every agent works on its own without concerning other agents. From a multi-agent systems perspective, the full distribution represents a level of disorder that has two important implications. First, the disorder means limited functional capabilities. While task FBR allows individual actions to collectively contribute to the overall task for simple task domains, it lacks ways to ensure similar system performance when tasks become more complex. Second, the disorder, on the other hand, provides an opportunity for us to infuse order into the system and therefore increase the overall system capability. The question is: how can we devise such order so that we can "control" the level of order for the best balance of system adaptability and functionality?

3 System Complexity by Emergence

As mentioned above, a system needs to possess a certain level of complexity in order to deal with tasks with a corresponding level of complexity [38]. Furthermore, it has been demonstrated that a system with higher physical complexity is more adaptable because the higher-level diversity permits satisfaction of changing constraints [39].

Following Huberman and Hogg [39], we consider the complexity spectrum of engineered systems over order and disorder to be bell shaped, as illustrated in Fig. 2. A single solid object, such as a metal shaft, has complete order, as indicated in point (*a*) in Fig. 2; it has close to zero complexity and can deal with very simple tasks, such as hitting something or transmitting rotation without changing speed. By increasing the number of dedicated components and introducing interactions between them, the order decreases in the sense that the system can be in a range of various possible states. Such systems can be a simple gearbox or a more complex automatic transmission system. Although this "complexity by design" approach (from (a)–(c) in Fig. 2) has been the mainstream approach for developing complex engineered systems and has been highly effective, if pushed too far, the unintended and unknown interactions among the components may become unmanageable and disastrous.

An alternative approach for designing complex engineered systems is to start from completely disorganized simple agents as indicated by point (b) in Fig. 2. While the completely disordered agents cannot perform any task—not even hitting something—introducing order into the system can potentially lead to a functional system (from (b) and (c) in Fig. 2). Many natural systems such as ant colonies achieve complexity this way.

Our research on self-organizing systems takes a "by emergence" approach, in which small groups of agents organize themselves locally, but the system functionality emerges globally from these interactions. Although currently uncompetitive with traditional approaches for systems design, this indirect approach presents an alternative strategy for developing complex engineered systems. Since by emergence does not require explicit knowledge of specific interactions among agents, the issue of unintended complex interactions mentioned above can be avoided. Furthermore, this approach may fundamentally expand the conceptualization of engineered systems by bringing biological developmental concepts into mechanical system design.

In our previous work, a task field-based behavior regulation mechanism has been developed to allow agents to self-organize into functional systems [4]. Although this limited order was effective for completing "pushing box" tasks, it was not enough for tasks involving both "pushing" and "rotating" the box. To further increase the level of order, in this research we introduce the concept of "social structuring" to capture explicit and direct interactions among agents and apply "social rules" to facilitate dynamic social structuring.

4 Task Complexity and Social Complexity

To investigate the relationship between task complexity and system complexity, we need to measure them. Depending on the

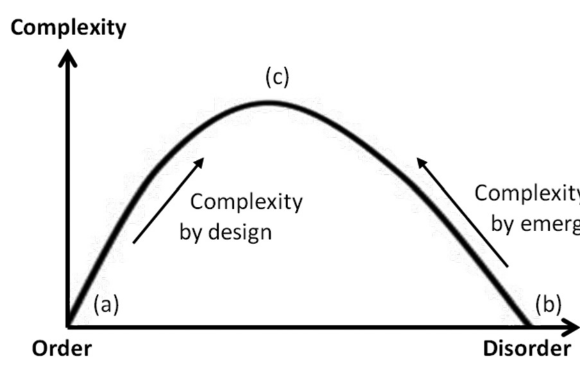

Fig. 2 Hypothetical system complexity over order–disorder spectrum (adapted from Huberman and Hogg [39])

purpose, there have been various measures of task complexity [40,41]. Inspired by Wood [40], in our research we analyze tasks with respect to the number of functions involved. Typically, a function can be represented as a <verb> <object> pair (e.g., <push><box>). As the number of distinct verbs (i.e., actions) and the number of distinct objects associated with a task increase, agents need to be more knowledgeable and skillful to perform the task. This view of task has led us to define task complexity based on three components: *action complexity*, *object complexity*, and *dynamic complexity*.

A general way to include action complexity is to sum distinct verbs used to describe a task. For a given task, in addition to the number of actions there can be various relationships between these actions (e.g., parallel, sequential, or specific delay) that must be maintained, e.g., through coordination, for the completion of the task. We have

Definition 1. *Action and Action Relation Complexity:* $\mathrm{AC} = |V| + \sum_{i\epsilon V} \sum_{j\epsilon V} r\mathrm{Act}_{ij}$

where $V = \{v_1,\ldots, v_n\}$ *is the set of all distinguished actions, and* $r\mathrm{Act}_{ij}$ *is an action relation between actions (verb) i and j, which can be sequential or reciprocal.*

As one of the main elements of task definition, objects involved in the task environment can play an important role in changing the complexity of the task. In addition to the number of objects, the characteristics of the objects involved in a task, such as shape, dimension, and mass, also contribute to the task complexity. Therefore, the number of parameters used to describe the distinctive objects can be used to define the *object complexity* of the task. The more parameters there are, the higher the complexity level is. In addition, the relationships between objects may pose constraints to possible actions and add more complexity to the task environment. Therefore, we define object complexity of a task as

Definition 2. *Object Complexity:* $\mathrm{ObC} = \sum_{i\,\epsilon\, O} |P_i| + \sum_{i\,\epsilon\, O} \sum_{j\,\epsilon\, O} \mathrm{ORC}_{ij}$

where O is the set of objects (i.e., types of objects) involved in the task environment; ORC_{ij} *is the object relation complexity between objects i and j, which is introduced as part of the task definition; and* $|P_i|$ *is the number of parameters for describing object i.* P_i *is a set of attribute and value pairs:* $P_i = \{(a_{i1}, v_{i1}), (a_{i2}, v_{i2}), \ldots, (a_{\mathrm{in}}, v_{\mathrm{in}})\}$

It is worth mentioning that since all the relevant objects in the operation environment are included in Definition 2, the task complexity also covers the complexity of the environment at any given time. Another aspect of complexity deals with the changing environment. Depending on the degree of variation, an agent's behavior may need to be adjusted. This behavior may entail different actions or responses to different objects. Therefore, we can capture such *dynamic complexity* by calculating the sum of differences across a certain time period for the aforementioned three complexity components if either the environment or task requirement changes, as described in Ref. [40]. We have

Definition 3. *Dynamic Complexity:* $\mathrm{DC} = \sum_t^T |\mathrm{AC}_{t+1} - \mathrm{AC}_t| + |\mathrm{ObC}_{t+1} - \mathrm{ObC}_t|$

The overall task complexity is a weighted sum of these complexity measures.

Definition 4. *Task Complexity:* $\mathrm{TC} = W_{\mathrm{AC}}\mathrm{AC} + W_{\mathrm{ObC}}\mathrm{ObC} + W_{\mathrm{DC}}\mathrm{DC}$

where W_{AC}*,*W_{ObC}*, and* W_{DC} *are the weights assigned to each complexity measure.*

Examples of how these complexity measures are applied and computed are given in the case study section.

We have introduced the concepts of agent and their *states*, *actions*, and *behaviors* in our previous work [4]. In this section, we introduce agent complexity, social structure, and social complexity.

Focusing on the physical and effective features [39,42], we consider the complexity of an agent in terms of its number of performable actions, number of decision-making behaviors, and communication capacity (e.g., range and number of channels). We have

Definition 5. *Individual Agent Complexity:* $C_{\mathrm{agent}_i} = N_a + N_b + C_{\mathrm{Com}}$

where N_a *is the number of actions,* N_b *is the number of behaviors, and* C_{com} is the communication capacity.

Individual agent complexity is usually only a small part of the overall complexity of a highly complex system. If the agents of a system are all independent from each other, then the system-level complexity can be a simple summation of individual agent complexities. Increasing the emergent complexity of a multi-agent system requires devising order within the system, as indicated in Fig. 2. In this research, we devise order by introducing social structures among agents. More specifically, we apply graph theory principles to capture the interactions among agents.

Assume G is a set of all possible graphs that can be formed by N agents $Ag = \{a_1, a_2,\ldots, a_N\}$. Then, we define:

Definition 6. *Social Structure:* $G(t) = (N, E(t))$*,*

where N is the number of agents, and E(t) is the links of interactions/relations between agents at time t.

As shown above, social structure $G(t)$ is a function of time and is directly dependent on the evolution of agents' interactions. For simplicity, we assume that agents are constant nodes in the graph. It is the edges between the nodes that change over time, resulting in a dynamic structure. The ideal situation is to keep the topology of agents frozen throughout much of the process but to adapt swiftly when the task and/or environment changes.

In this research, the social structure represented as a connectivity graph is realized by defining social rules that specify how agents interact with each other. These social rules can be general (e.g., "move in a similar direction with neighbors") or task specific (e.g., "move closer to neighbors on the edge of a box"). We define the social complexity of the system based on the connectivity graph that originates from social rules. This type of graph complexity is notably similar to the complexity measures defined in molecular chemistry [43–45]. Vertex degree magnitude-based information content I_{vd} has been validated as a measure of network complexity. It is based on Shannon's information theory and defines information as the relative measure between a system's entropy and the maximum possible entropy of a hypothetical system of the same size [46–48].

Definition 7. *Social Complexity:* $\mathrm{SC} = \sum_{i=1}^N d_i \log(d_i)/N$

where d_i *is the degree of each node i (how many other agents are communicating with agent i).*

5 Social Rule Based Behavior Regulation

In this research, we explore ways to facilitate emergence of order and therefore complexity so that a self-organizing system can deal with more complex tasks. More specifically, we want to devise dynamic structuring methods that can help guide self-organization of agents. We take *a social rule based behavior regulation* approach and explore various local and bottom-up social relations to achieve dynamic social structuring.

We divide deficiency due to disorganization into two categories: *conflict* deficiency and *opportunity-loss* deficiency. For simple tasks (e.g., pushing a box to a destination in an open space), where an individual agent's goal is mostly consistent with the system goal, the agents' effort can additively contribute to the system's overall function. When tasks become more complex, conflicts between agents' actions (e.g., pushing box in opposite directions due to space constraints) may occur more often. Furthermore, while more cooperation opportunities (e.g., pushing the box in opposite directions at different locations in order to rotate a box) may present themselves, the lack of coordinated individual actions may leave many opportunities unutilized.

In order to minimize conflicts between agents and exploit cooperation opportunities, social rules and social relations can play an important role. A social rule is a description of a behavioral relationship between two encountering agents that can be used by the agents to modify their otherwise individually determined actions.

Two agents acting on a given social rule are described as being engaged in a social relationship. Based on Definition 7, when agents are engaged in social relations by following social rules, social structures emerge, leading to more order and potentially higher system complexity.

To avoid conflicts and promote cooperation, social rules can be defined to specify which actions should be avoided and which actions are recommended under certain conditions. Rules, as also mentioned in Ref. [49], are sets of coded restrictions. We have

DEFINITION 8. *Social Rule: sRule = <C, ForbA, RecA>*

where C is a condition specifying a set of states, ForbA are the forbidden actions for the specified states, and RecA are the recommended actions.

Social rules defined above introduce relations among encountering agents. It is conceivable that when an agent encounters neighbors, and those neighbors encounter their neighbors, and so on, the cascading effect may lead to a large-scale network structure. The distribution of such a network can be defined as a *social field* in which every agent has its own position and the awareness of the social field allows agents to reach beyond their immediate neighbors. We have

DEFINITION 9. *Social Field: sField = FLDs (sRule)*

where FLDs is the field formation operator, and sRule is a social rule.

Social field adds another layer to the design of self-organizing systems as a helpful mechanism to secure synergy in the system. We will explore its full implication in future research. In this research, the focus is on allowing agents to adjust their otherwise independent agent behavior by applying social rules in the context of their neighboring agents. This social rule based behavior regulation can be defined as follows:

DEFINITION 10. *Social Rule Based Behavior Regulation:*

$$\mathrm{SocSat}_{\mathrm{Beh}_i} = \mathrm{SRBR}(\mathrm{Sat}_{\mathrm{beh}_i};\ \mathrm{SR}_i,\ \mathrm{NA}\ _i)$$

Where SRBR is the social FBR operator, $\mathrm{Sat}_{\mathrm{beh}_i}$ *is task field-only behavior satisfaction,* SR_i *is the set of social rules,* $\mathrm{NA}\ _i$ *is the set of encountering neighbor agents, and* $\mathrm{SocSat}_{\mathrm{Beh}_i}$ *is the socially regulated behavior satisfaction.*

The above is a general definition. To apply SRBR, an agent needs to (1) generate its independent satisfaction [50] behavior profile through an FBR operator, (2) identify and communicate with its neighbors, (3) possess social rules, (4) determine which rule to apply for the given situation, and (5) generate a new socially driven behavior. Each of the five steps can be task domain dependent. In Sec. 6 (Case Study), we discuss how these steps can be implemented and these concepts can be applied.

Definition 10 defines how an agent i will choose its behavior given social rules SRi, should it decide to adopt the rules. To explore the effect of social structuring with a varying percentage of rule adoption, we introduce an important concept called "social policy" defined as the rule adoption rate quantified as a percentage. A policy of 100% rule adoption rate means that all agents are required to adopt the social rules whenever they become applicable. A policy of 20% means that they can apply the social rules to only 20% of applicable cases randomly. The purpose of the policy is to control the level of agents' interactions. In this paper, we explore how various rule adoption rates can impact the performance of a given number of agents with a given set of social rules in different task contexts.

6 Case Study

The objective of our case study is to explore and demonstrate how social rule based behavior regulation can increase the order, and therefore potentially the complexity, of a self-organizing system and how this increased order is essential for dealing with more complex tasks. In addition, we also attempt to gain insights on how social structuring mechanisms, such as social rules and rule adoption policies, influence self-organizing system performance with respect to a varying number of agents and different levels of task complexity. Such insights will eventually be useful for providing guidance for designing self-organizing systems.

6.1 Tasks.

The box-moving task, illustrated in Fig. 3, has been used for the case study. Multiple agents intend to move the box to the destination or goal "*G*." Given that the pathway becomes narrower, the agents must rotate the box to horizontal, as it gets closer to the entrance of the narrow part. Further, there can be an obstacle "obs" on the way.

The specific functions involved in this task can be expressed as follows:

T1 = <Aim><Goal>
T2 = <Push><Box>to<Goal>
T3 = <Move Around><Box>
T4 = <Avoid><Wall>
T5 = <Avoid><Obstacle>

To facilitate task field-based behavior regulation, the task fields are defined to include an attraction field from the goal *G* and several repulsion fields from the "walls" as well as the obstacle obs if present, as indicated in Fig. 3. For the goal attraction field, a gravitylike field is applied, and for the walls and the obs, a gradient-based repulsion distribution is introduced to provide "warnings" of collision as agents get closer to them. The gradient distribution of the constraints (i.e., walls and obstacles) together with the sensory range of agents determines how far ahead the agents can predict the collision and find ways to avoid it. In this simulation-based study, *higher positions* in the field are more desirable to agents. With only task field-based behavior regulation, to move the box, an agent always tries to find a low field position around the box and from there to push the box toward a high field position which is often, but not always, the goal *G* position. However, when the social structuring and hence social field is considered, the agent behavior will be modified by the social rules described in Sec. 6.2 (Social Rules).

The object complexity is calculated for all the objects involved in this task. Descriptive complexity has been used that captures the amount of information required to characterize each object. Three parameters *x, y, r* have been used to describe the target and obstacles, with *(x, y)* being the coordinates of their center, and *r* the radius associated with their spread length. The characteristics of the box include its center location *(x, y)*, dimensions W_{Box} (width) and L_{Box} (length), and its orientation *g*. Thus, the object complexity sums up to 4 for this item. The walls on the sides can be described by *three points* and a line passing each point. Each point can be described with two-dimensional variables *(x, y)*, resulting in complexity of 6. Therefore, the total descriptive complexity of objects adds up to 13 for the no-obstacle case and 16 for the one-obstacle and two-obstacle cases.

The object relation complexity is calculated based on the representation of the problem and the distribution the objects in the environment. Agents' difficulty in moving the box through the environment depends on the distance between two encountering objects. In this case, the object relation complexity is defined to be proportional to the distance between the objects and the walls

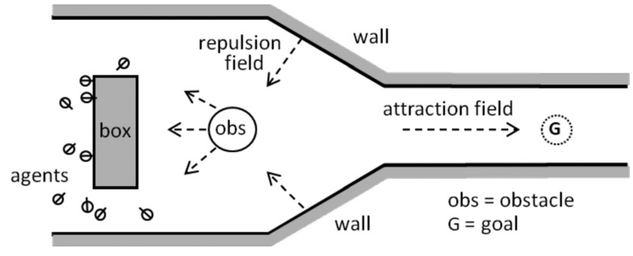

Fig. 3 Box-moving task used in case studies

that need to be avoided with respect to the minimum dimension of the box to be carried W_{Box} (i.e., its width). The average of object relation complexity in all possible paths is the final object relation complexity measure.

$$\text{ORC for each emergent path} = \sum_{\text{Obj} \in O} \frac{1}{\text{Dist}(\text{Obj}_i, \text{Obj}_j)/W_{\text{Box}}} \tag{1}$$

In the no-obstacle case, there is only a wall-to-wall relation and one path toward the goal, which leads to object complexity of 0.45 * (1/(5.794) + 1/(3.35)). In the one-obstacle case study, there exist two identical paths; each path has four areas between objects that include obstacle–obstacle, wall–wall, and wall–obstacle relations. There are two obstacle–wall distances (3 * W_{Box}, 2.67 * W_{Box}) and two wall–wall distances (5.8 * W_{Box} and 3.35 * W_{Box}) to pass, which add up to 1.154 for object relation complexity. Adding another obstacle provides two different possible paths to get to the goal. In the first scenario, there are two obstacle–wall distances (3 * W_{Box}, 2.26 * W_{Box}), two wall–wall distances (6 * W_{Box} and 3.353 * W_{Box}), and one obstacle–obstacle distance (3 * W_{Box}) to pass, resulting in 1.575, and another path of two obstacle–wall distances (3 * W_{Box} and 2.5 * W_{Box}) and two wall–wall distances (6 * W_{Box} and 3.353 * W_{Box}), which gives 1.184 for object relation complexity. The maximum object relation complexity becomes 1.575.

Total complexity is the weighted sum of these complexity measurements. A weight of 1.0 for object and action complexity and 10.0 for object relation complexity has been used, leading to the total complexity of 24.7 for the "with wall" situation, as indicated in Table 1. Based on a similar calculation, the "wall + obs," "wall + two obs" situations have complexity values of 34.53 and 38.7, respectively, as shown in Table 1.

The system is composed of n agents: $A = \{a_i\}$ $(i = 1, \ldots, n)$. The initial positions of agents are randomly assigned but are always on the left side of the box. Guided by the task field of attraction and repulsion, each agent is supposed to contribute to the correct movement of the box in such a way that the emergent movement of the box is toward the goal. Although this strategy (i.e., "nonsocial") works well for open space with a few obstacles [3], when more constraints, such as walls and more obstacles, are added, new strategies (e.g., social structuring) are needed.

6.2 Social Rules. As mentioned above, social rules are designed to allow agents to avoid conflicts and/or to promote cooperation. In this case study, the social rules are set to provide guidance for agents to become aware of, and subsequently avoid, potential conflicts. Figures 4(*a*) and 4(*b*) illustrate possible force conflict and torque conflict, respectively, between agents i and j.

To facilitate the definition of social rules, we introduce the "box neighborhood" by defining six zones, as indicated in Fig. 4(*c*). Agents are aware of their location in terms of which zone they are in. Furthermore, they can broadcast their location information and field value to neighbor agents. We have the following communication rule:

Social rule 1 (communication rule): <condition: enter box neighborhood> ==> <recommended action: broadcast [location] and [field strength]>

Table 1 Complexity measures of various box-moving situations

Situation	0: Open space	1: With wall	2: With wall + one obs	3: With wall + two obs
Complexity	9.17	24.7	34.53	38.75

When an agent receives broadcast information from an agent in the neighborhood, it will attempt to determine if a force conflict or a torque conflict exists and then decide if it will take the recommended actions provided by the following two conflict avoidance rules:

Social rule 2 (force conflict rule): <condition: force conflict> ==> <forbidden action: push in opposite direction in opposite zone> & <recommended action: find a new location>

Social rule 3 (torque conflict rule): <condition: torque conflict> ==> <forbidden action: push in opposite direction in opposite zone> & <recommended action: move to next neighbor zone>

Agents have the option to ignore any or all of the above three rules depending on the "social rule adoption policy" described at the end of Sec. 5. When the probability for agents to follow the rules decreases, we say that the system is *less socially active* (i.e., weak social), and otherwise *more socially active* (i.e., strong social).

6.3 Experiment Design. Figure 5 illustrates the design of the simulation-based experiment. Two strategies were explored: with social structuring—i.e., agents make their behavioral decisions based on SRBR—and no social structuring—i.e., agents make their behavioral decisions based only on task FBR. These two strategies are captured by using a single-independent variable social rule adoption policy, with possible adoption probability values of 0–100%, in 10% increments, as indicated in Fig. 5. This approach is similar to modeling the centralization policy in an organization by counting the percentage of decisions reported to one's supervisor, i.e., 0% reporting means decentralized and 100% completely centralized [51].

The size of the agent population is another independent variable of the self-organizing systems. After experimenting with various numbers of agents, it has been found that the range between 7 and 15 agents is a productive range for the case study for the box-moving task.

Various combinations of social rule adoption policy and "number of agents" have been applied to different task complexity settings. As mentioned above, the box-moving tasks are categorized into three levels of complexity depending on how many obstacles are on the pathway. Level 1 corresponds to "with no obstacle," while levels 2 and 3 to "with one obstacle" and "with two obstacles," respectively.

For all settings, the system performance is measured by the following three-dependent variables:

- *Success rate*: The percentage of successful runs out of a total number of simulation runs, N_{simrun} (explained below). A simulation run is counted as successful if the agents move the box from its initial position to the goal within the time limit. For each run, the time limit is taken to be three times the amount of time for successful runs that used a 100% rule adoption strategy.
- *Time duration*: The number of simulation time steps it takes to move the box from its initial position to the goal position.
- *Total effort*: The total number of steps (i.e., unit-width) traveled by all agents during a simulation run. At each time step, agents can move toward their destination one unit-width. For scale, the box's width is five unit-widths.

Our multi-agent simulation system was developed based on the NetLogo platform [52], a popular tool used by researchers of various disciplines. To maintain the statistical significance of simulation results, all resulting data points are the mean values of N_{simrun} simulations runs. The number of N_{simrun} varied between 50 and 300, depending on the variability of simulation settings. To compare the results of two settings, a t-test was carried out for the two mean values to be compared. The number of simulation runs was increased (with increment of 50) to the point at which the $p \leq 0.05$ was achieved. This number is then set as the total number of

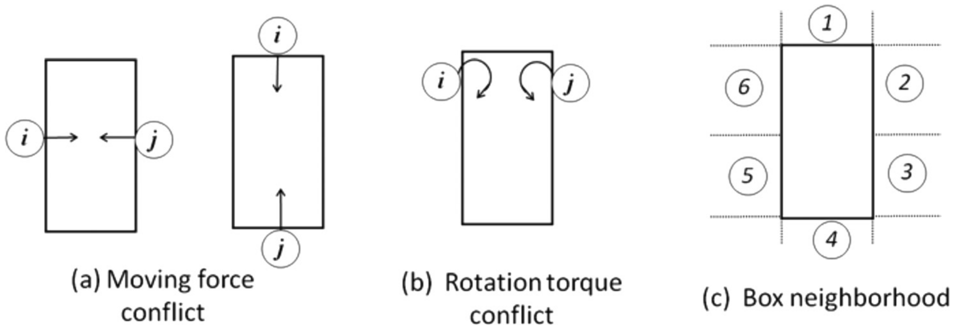

Fig. 4 Possible conflicts of agents *i* and *j* and box neighborhood: (*a*) moving force conflict, (*b*) rotation torque conflict, and (*c*) box neighborhood

simulation runs, N_{simrun}, for these settings. Because the variances of simulation results are smaller for simpler tasks and greater for more complex ones, we have set $N_{simrun} = 50$ for level 1 task simulations, $N_{simrun} = 100$ for level 2 task simulations, and $N_{simrun} = 300$ for level 3 task simulations. For example, when verifying the effect of social structuring on total effort in one-obstacle situation, an independent-samples t-test was used for two data sets of 0%-social and 100%-social adoption rates, $t(99) = 2.0255$, p-value $= 0.02208$, confirming the difference of 19,000 unit-width (i.e., more effort with the 0%-social setting).

6.4 Results and Discussion. The simulation-based studies were carried out against three levels of task complexity: level 1 (with only wall), level 2 (with wall + one obstacle), and level 3 (with wall and two obstacles). For each task complexity level, we vary the social rule adoption policy and number of agents involved. The performance measures of these simulation runs (i.e., success rate, total effort, and time duration) are then plotted for evaluation and comparison.

6.4.1 Task Complexity = Level 1. Figure 6 illustrates a series of screenshots of a typical simulation run where agents are represented as green squares who are moving a large brown box through a pathway to the goal (white circles).

Since the tasks with level 1 complexity are relatively simple, all the simulation runs were successful. Therefore, the success rates for "without social" (i.e., 0% social) and "with social" (10% through 100% social) cases are all 100%.

Figures 7 and 8 illustrate the comparisons of "total effort" and "duration time," respectively, for various social rule adoption policies for tasks of level 1 complexity with a varying number of agents. From Fig. 7, it can be seen that for the situations without social structuring (i.e., 0% social), the total effort is much higher than all other situations For example, when verifying the effect of social structuring on total effort, an independent-samples t-test was used for two data sets of 0%-social and 100%-social adoption rates, with p-value $= 0.003$ followed by a Tukey HSD test with adjusted p-value $= 0.0001246$. In these cases, the agents have to rely on trial-and-error to complete the task, leading to increased effort. Similar results can be observed for time duration (Fig. 8), although when the number of agents increases, the duration advantage of social structuring gradually diminishes. When the number of agents is small, e.g., 7, stronger social structuring is more preferable, with a p-value < 0.05 for both t-test and Tukey HSD test (e.g., adjusted p-value $= 0.021859$ for duration comparison of 10%-social and 100%-social adoption rate) indicating statistically significant differences. In these cases, the system complexity is limited by the small number of agents. Adding more social structuring can increase the level of system complexity, making the system more efficient. Another way to increase system complexity is to increase the number of agents. That is potentially the reason why the effect of stronger social structuring becomes less obvious for the situations with a larger number of agents, as shown in Figs. 7 and 8.

The change of social complexity during runtime in a typical simulation run with a social structuring strategy and 12 agents is shown in Fig. 9. As shown in the figure, social complexity increases when agents start to communicate with each other by following *social rule 1* and help each other by following social rules 2 and 3 when rotating the box in the middle of the process. Social complexity through social structuring varies over time; it increases when needed by the task situation (e.g., rotating the box) and decreases when the situation is resolved. This task-driven variability is the key difference from the agent complexity obtained through adding more agents. While adding more agents somehow relies on "randomness" to complete the task inefficiently, the social rule based self-organization builds competence through explicit structuring.

6.4.2 Task Complexity = Level 2. To further explore how more complex tasks demand social structuring, we carried out simulations for level 2 (i.e., "with wall + one obstacle") tasks. The task complexity measure for this situation is 34.57 (see Table 1), higher than the with wall situation.

Figure 10 shows the success rate for the simulations with different settings. While the simulations without any social structuring (i.e., 0% social in Fig. 10) had about 50% and less than 10% success rates for the 11-agent and 13-agent cases, respectively, the simulations with the strongest social structuring (i.e., 100% social in Fig. 10) always had 100% success rate. The increased level of task complexity demands a higher level of complexity of the system. When the number of agents is small (i.e., 7 and 9 in Fig. 10),

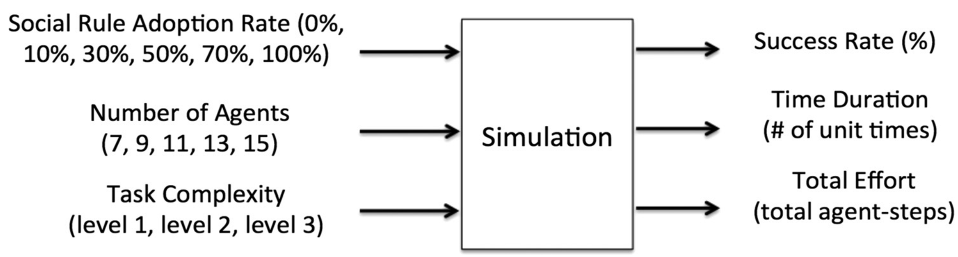

Fig. 5 Experiment design with three-independent variables and three-dependent variables

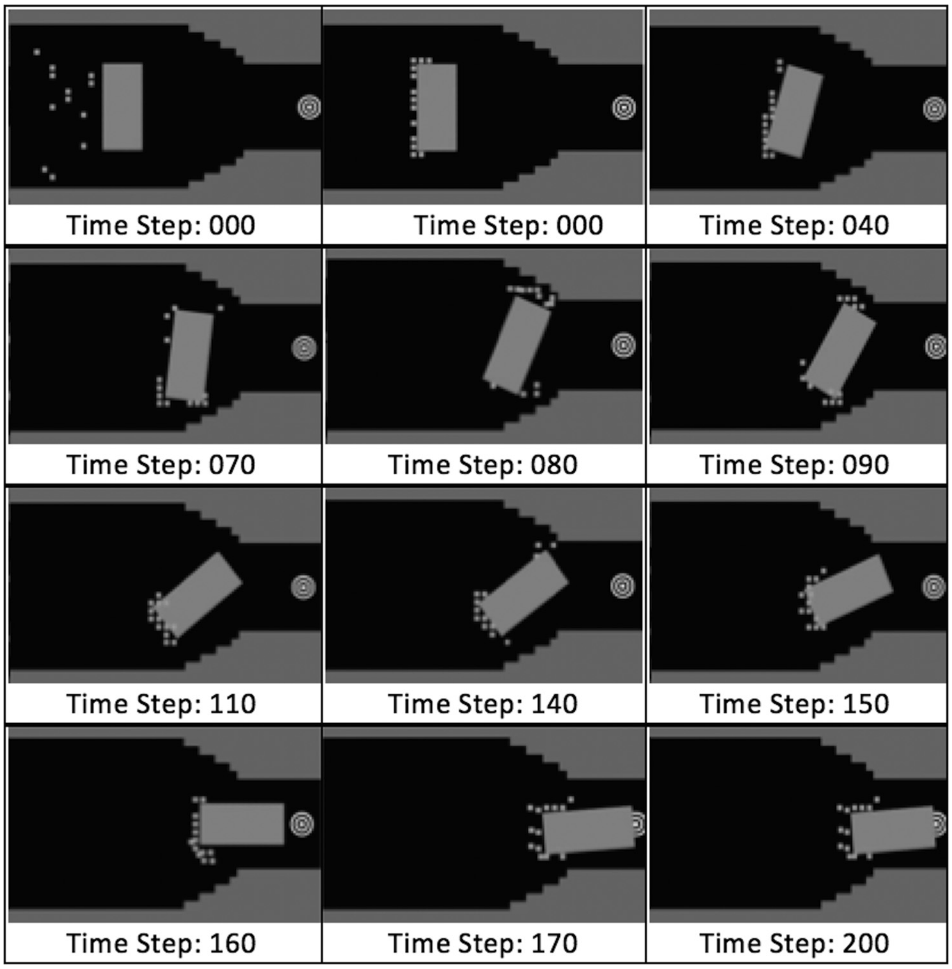

Fig. 6 Screenshots of a typical simulation run for the with wall task

the weak social structuring has driven the success rate lower than the no-social structuring cases. To statistically verify the difference of success rate between 0%-social and 10%-social adoption rate settings, for example, 20 data points of each social adoption rate setting were created. Each data point was the success rate over 100 simulation runs. A t-test was conducted with p-value $= 1.091 \times 10^{13}$ followed by adjusted p-value of 0 for Tukey HSD test that further illustrates the clear difference. This result indicates that having agents only occasionally follow the social rules in the small number of agent cases can be ineffective because in these cases partial rule-following can cause counterproductive incomplete coordination among agents. We use the phrase "weak social disadvantage" to indicate the ineffectiveness due to the combination of a small number of agents working on complex tasks with only weak social structuring. Figure 10 also shows that the stronger social structuring is more favorable for the cases of smaller numbers of agents.

It can also be seen in Fig. 10 that the success rate drops to its minimum when the number of agent is 13 and then picks up after that (e.g., with the same method mentioned in the last paragraph, the success rate difference between 0%-social and 13-agent and 0%-social and 11-agent settings was verified with

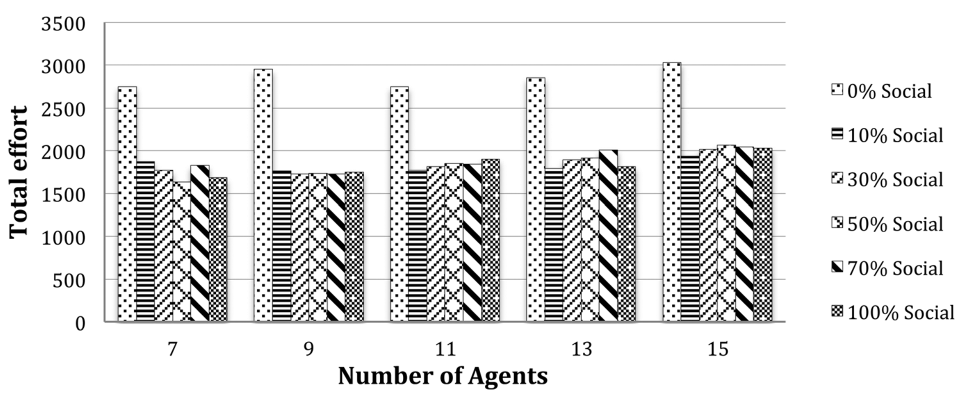

Fig. 7 Total effort comparison for various social rule adoption policies for the with wall task with varying number of agents

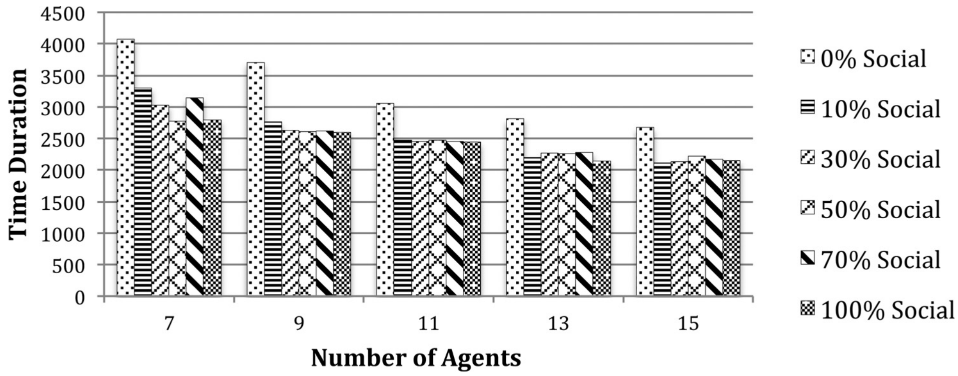

Fig. 8 Time duration comparison for various social rule adoption policies for the with wall task with varying number of agents

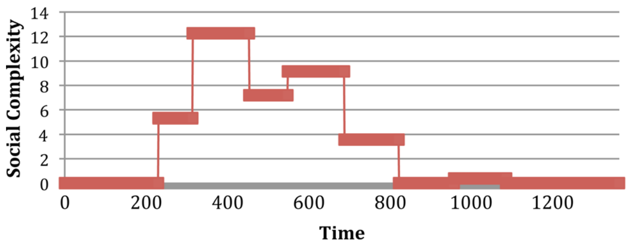

Fig. 9 Social complexity during the process of moving the box toward the goal with SRBR strategy and 12 agents

p-value $< 2.2 \times 10^{16}$ for t-test as well as adjusted p-value of 0 for Tukey HSD test. One plausible explanation is that more agents (until 13) caused more cancelations of effective random moves, leading to more failures. The further increase of the number of agents to 15 allowed opportunities for more random effective moves on top of the cancelations (e.g., t-test p-value $< 2.2 \times 10^{16}$ was achieved by comparing success rate of 0%-social and 13-agent and 0%-social and 15-agent settings and adjusted p-value of 0 in Tukey HSD test).

Figures 11 and 12 indicate the comparisons of the total effort and duration time for the successful simulation runs, respectively. Overall, the social structuring cases, from 10% to 100%, had better efficiency than the no-social structuring (i.e., 0% social) cases. However, there was a "singular" point, where the number of

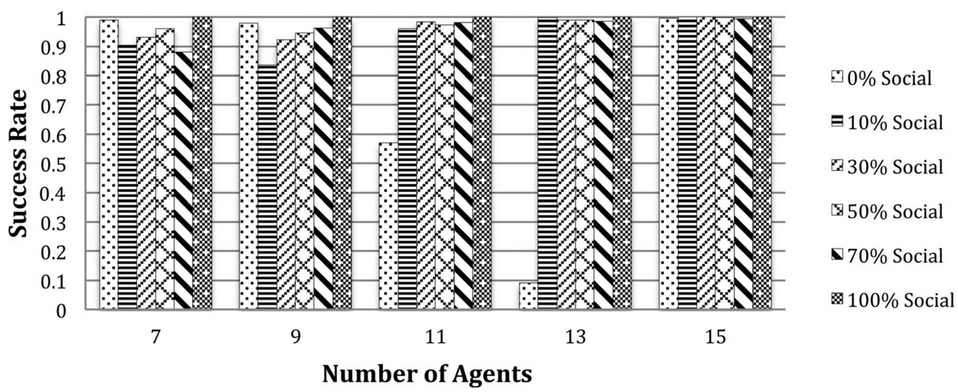

Fig. 10 Success rate comparison for various social rule adoption policies for the with wall + one obstacle task with varying number of agents

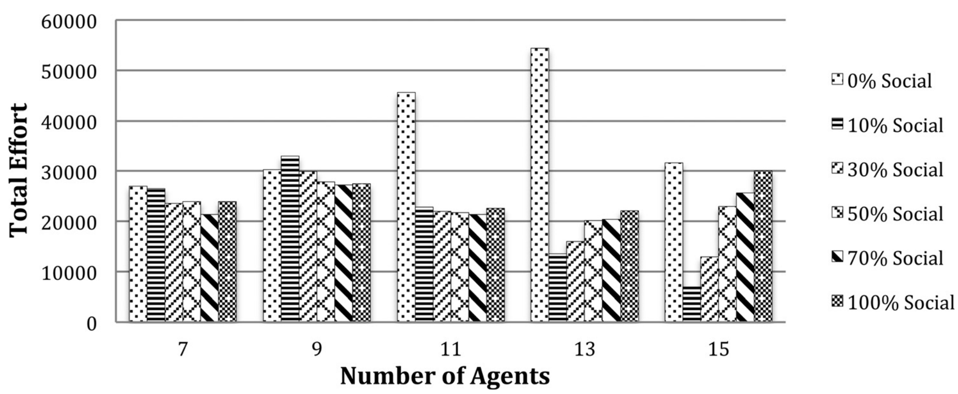

Fig. 11 Total effort comparison for various social rule adoption policies for the with wall + one obstacle task with varying number of agents

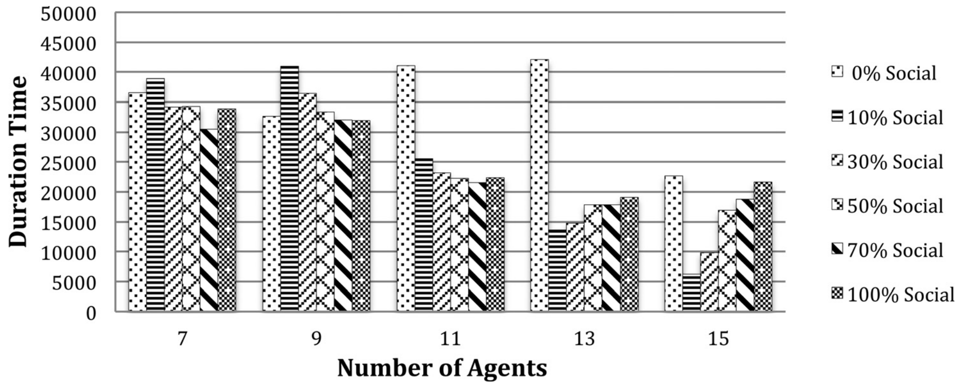

Fig. 12 Duration time comparison for various social rule adoption policies for the with wall- + one obstacle task with varying number of agents

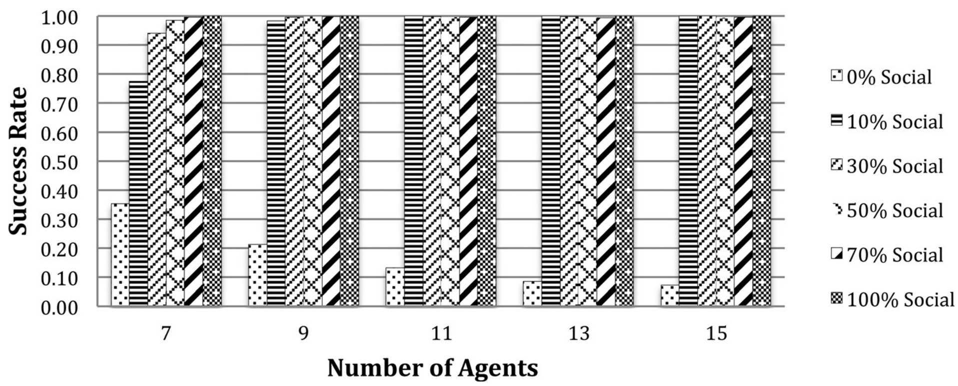

Fig. 13 Success rate comparison for various social rule adoption policies for the with wall- + two obstacles task with varying number of agents

agents is 9 and the efficiency of the no-social structuring case was either equal to or better than the social structuring cases (e.g., to analyze the duration time variance between 0%-social and 10%-social adoption rate, a t-test was conducted with p-value $= 0.0009876$ followed by adjusted p-value of 0.0019247 for Tukey HSD test that further demonstrates the definite difference). This seems to be the place where the weak social disadvantage was taking effect. Again, it can be more clearly seen that when the number of agents was under 11, the stronger social structuring helped improve the system efficiency (e.g., statistical p-value $= 0.04326$ was achieved in t-test accepting the alternative hypothesis of greater difference in means of 10%-social versus 100%-social and p-value $= 0.9567$ for rejecting the alternative hypothesis of having less difference in means of the exact same group set confirming the less efficiency in 10% social), while for 13-agent and 15-agent situations, stronger social structuring decreased the system efficiency (e.g., comparing total effort of 10% and 100%-social in 15-agent settings results in p-value $< 2.2 \times 10^{16}$ in t-test as well as adjusted p-value of 0 in Tukey HSD test demonstrating the recognizable difference, potentially due to the added structuring overhead).

From Figs. 11 and 12, it can be seen that the system efficiency depends on the combination of the number of agents and the social rule adoption policy. The best strategy in this task situation

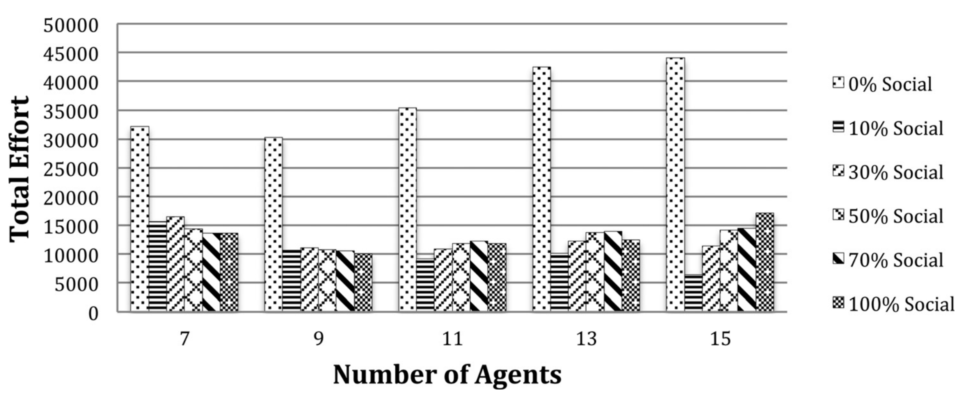

Fig. 14 Total effort comparison for various social rule adoption policies for the with wall- + two obstacles task with varying number of agents

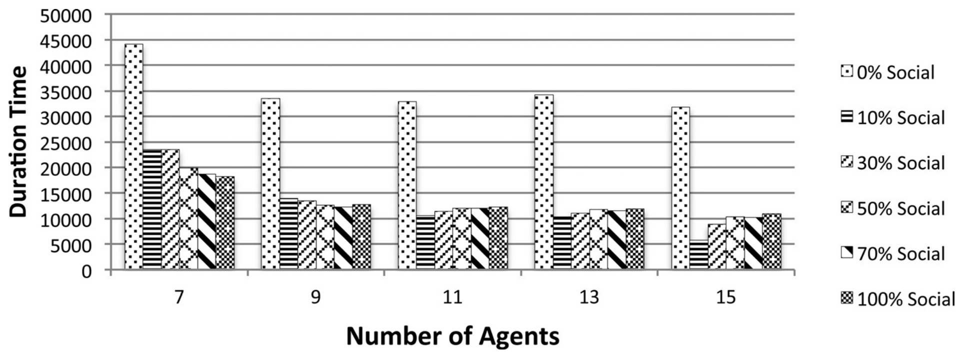

Fig. 15 Duration time comparison for various social rule adoption policies for the with wall + two obstacles task with varying number of agents

seems to be to use more agents (e.g., 15) and apply a weak social structuring policy (e.g., 10%).

6.4.3 Task Complexity = Level 3. In order to fully investigate the relationship between more complex tasks and the need for social structuring, we carried out simulations for the "with wall + two obstacles" tasks.

Figure 13 shows the success rate comparison of various social rule adoption policies and different numbers of agents. It can be seen from Fig. 13 that the success rate for the no social structuring (i.e., 0% social) decreased dramatically even with larger numbers of agents, due to the higher-level task complexity. The 100% social structuring policy, on the other hand, maintained a 100% success rate for all situations. For scenarios with fewer agents, the weaker social structuring policies had a less than perfect success rate (e.g., stats for 7 agents and adoption rate of 10% and 30% are p-value of 1.526×10^{8} in t-test and 0 adjusted p-value clearly shows the significant difference between success rates), similar to what we saw in Fig. 10. Figure 13 clearly indicates that highly complex tasks cannot be dealt with by simple (i.e., unstructured) systems.

Figures 14 and 15 show the comparisons of total effort and duration time, respectively. Overall, no-social structuring cases were less efficient. It can be seen from Figs. 13–15 that increasing the number of agents was not significantly helpful on its own. On the other hand, adding a small amount of social structuring (e.g., 10% social) could significantly improve the performance of the system. This phenomenon is more evident in more complex tasks (Figs. 13–15) than less complex ones (Figs. 9–11). Similar to Figs. 11 and 12, the results in Figs. 14 and 15 illustrate that for a smaller number of agents, stronger social structuring is preferable (e.g., the t-test for 10% and 100% social in 7-agent had p-value $< 2.2 \times 10^{16}$) and for a larger number of agents, the weaker one is significantly more efficient proved by t-test (p-value $< 2.2 \times 10^{16}$) and Tukey HSD test (p-value $= 0$) because stronger social structuring involves more overhead.

7 Conclusions

As tasks become more complex, engineered systems have been made more complex by moving from rigid and tightly organized formations toward those of more components and more interactions. A potential issue with this top-down or ordered-to-disordered approach is the unintended and unknown interactions that may cause failure of the whole system. An alternative approach is to start with simple and disorganized agents and then move bottom-up and from disordered to ordered by devising dynamic structures through self-organization. In this research, we explored the sources of task complexity by defining various complexity types and investigated how social rule based behavior regulation can be applied to allow dynamic social structures, and thus system complexity, to emerge from self-organizing agents. The case study results have demonstrated the effectiveness of our proposed approach and shed some useful insights on designing self-organizing systems.

The behavior of self-organizing systems becomes more chaotic when tasks are more complex. Our statistical analysis of the simulation results shows that the variance of simulation results for complex tasks is greater than that for simpler tasks. The system behavior could be very sensitive to even small changes in initial conditions. Increasing the system complexity by both adding more agents and devising social structuring could reduce the variability of the system behavior.

Adding more agents and devising social structure have different effects. Increasing system complexity can be achieved through adding more agents or devising social structures. However, the former has a strong effect only with relatively simple tasks. When tasks become more complex, adding agents alone may not cause a system to attain a 100% success rate and the efficiency of the successful runs can be very low. On the other hand, devising social structures can make the system more adaptable. Not only did the 100% social systems always achieve a success rate of 100% but their efficiency was also maintained despite changing task complexity and a varying number of agents. This result is consistent with [39] conjecture that higher structural complexity makes a system more adaptable.

The balance of task complexity, the number of agents, and social structuring is the key. When tasks become more complex, devising social structures alone may not be sufficient. Increasing the number of agents can make the social structuring more efficient as indicated in Figs. 11 and 14. Our simulation results have demonstrated that for complex tasks, it is more desirable to include a sufficient number of agents and devise relatively weak social structuring (see Figs. 11 and 12 and 14 and 15). In general, we believe that the balance between the task complexity, the number of agents, and strength of social structuring is the key for making self-organizing systems more effective and efficient.

Stronger social structuring is effective for a smaller number of agents. When the number of agents is small, there is a weak social disadvantage. That is, having agents only occasionally follow the social rules can be ineffective and inefficient, because in these cases, partial rule-following can cause counterproductive or incomplete coordination among agents. For more complex tasks, increasing the social rule adoption rate can enhance both success rate and efficiency for systems with a low number of agents.

Weaker social structuring is more effective for a larger number of agents. When the number of agents is large enough, weaker social structuring can be more efficient. Since rule-following may incur overhead, having more agents follow social rules can be costly from an efficiency perspective. This effect can be observed only when the number of agents is larger than a *transition* point or region (i.e., when the number of agents is around 11 for Figs. 10–15). Before this transition point, the weak social disadvantage is more notable.

There can be a singular number of agents where social structuring is neither effective nor efficient. For the task complexity of level 2, when the number of agents is around 9, the social structuring, especially the weaker social structuring, becomes more a problem than a solution for the system. This result suggests that potentially there are certain combinations of task complexity and a small number of agents for which the social structuring can be counterproductive. Further research is needed to confirm and clarify this phenomenon.

The results reported in this paper and the conclusions described above are limited to the box-moving tasks studied in this research. Our ongoing work investigates more types of tasks and explores various types of task complexity and social structuring mechanisms. More practical engineering tasks will be explored to make our self-organizing systems more functional and valuable. Furthermore, we also plan to explore how agent learning may impact the system performance.

Acknowledgment

This paper was based on the work supported in part by the National Science Foundation under Grant Nos. CMMI-0943997 and CMMI-1201107.

References

[1] Zouein, G., Chen, C., and Jin, Y., 2010, "Create Adaptive Systems Through 'DNA' Guided Cellular Formation," 1st International Conference on Design Creativity, pp. 149–156.
[2] Chiang, W., and Jin, Y., 2011, "Toward a Meta-Model of Behavioral Interaction for Designing Complex Adaptive Systems," ASME Paper No. DETC2011-48821.
[3] Jin, Y., and Chen, C., 2014, "Field Based Behavior Regulation for Self-Organization in Cellular Systems," *Design Computing and Cognition '12*, Springer, Dordrecht, The Netherlands.
[4] Chen, C., and Jin, Y., 2011, "A Behavior Based Approach to Cellular Self-Organizing Systems Design," ASME Paper No. DETC2011-48833.
[5] Humann, J., and Jin, Y., 2013, "Evolutionary Design of Cellular Self-Organizing Systems," ASME Paper No. DETC2013-12485.
[6] Jin, Y., and Chen, C., 2014, "Cellular Self-Organizing Systems: A Field-Based Behavior Regulation Approach," Artif. Intell. Eng. Des. Anal. Manuf., **28**(2), pp. 115–128.
[7] Simon, H. A., 1962, "The Architecture of Complexity," Proc. Am. Philos. Soc., **106**(6), pp. 467–482.
[8] Williams, E. L., 1981, *Thermodynamics and the Development of Order*, Creation Research Society Books, Norcross, GA.
[9] Ferguson, S., and Lewis, K., 2006, "Effective Development of Reconfigurable Systems Using Linear State-Feedback Control," AIAA J., **44**(4), pp. 868–878.
[10] Martin, M. V., and Ishii, K., 1997, "Design for Variety: Development of Complexity Indices and Design Charts," ASME Paper No. DETC97/DFM-4359.
[11] Rus, D., and Vona, M., 1999, "Self-Reconfiguration Planning With Compressible Unit Modules," IEEE International Conference on Robotics and Automation, Detroit, MI, Vol. 4, pp. 2513–2520.
[12] Rus, D., and Vona, M., 2000, "A Physical Implementation of the Self-Reconfiguring Crystalline Robot," IEEE International Conference on Robotics and Automation, ICRA'00, Vol. 2, pp. 1726–1733.
[13] Rus, D., and Vona, M., 2001, "Crystalline Robots: Self-Reconfiguration With Compressible Unit Modules," Auton. Rob., **10**(1), pp. 107–124.
[14] Fukuda, T., and Nakagawa, S., 1987, "A Dynamically Reconfigurable Robotic System (Concept of a System and Optimal Configurations)," Robotics and IECON'87 Conferences, pp. 588–595.
[15] Unsal, C., Kiliccote, H., and Khosla, P. K., 1999, "I (CES)-Cubes: A Modular Self-Reconfigurable Bipartite Robotic System," Proc. SPIE, **3839**, pp. 258–269.
[16] Prevas, K. C., Unsal, C., Efe, M. O., and Khosla, P. K., 2002, "A Hierarchical Motion Planning Strategy for a Uniform Self-Reconfigurable Modular Robotic System," IEEE International Conference on Robotics and Automation, ICRA'02, Vol. 1, pp. 787–792.
[17] Yim, M., 1993, "A Reconfigurable Modular Robot With Many Modes of Locomotion," International Conference on Advanced Mechatronics, pp. 283–288.
[18] Yim, M., Zhang, Y., and Duff, D., 2002, "Modular Robots," IEEE Spectrum, **39**(2), pp. 30–34.
[19] Shen, W. M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., and Venkatesh, J., 2006, "Multimode Locomotion Via SuperBot Reconfigurable Robots," Auton. Rob., **20**(2), pp. 165–177.
[20] Arkin, R. C., and Balch, T., 1998, "Cooperative Multiagent Robotic Systems," *Artificial Intelligence and Mobile Robots*, MIT/AAAI Press, Cambridge, MA.
[21] Brooks, R., 1986, "A Robust Layered Control System for a Mobile Robot," IEEE J. Rob. Autom., **2**(1), pp. 14–23.
[22] Mataric, M. J., 1997, "Behaviour-Based Control: Examples From Navigation, Learning, and Group Behaviour," J. Exp. Theor. Artif. Intell., **9**(2–3), pp. 323–336.
[23] Parker, L. E., 1998, "ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation," IEEE Trans. Rob. Autom., **14**(2), pp. 220–240.
[24] Horling, B., and Lesser, V., 2004, "A Survey of Multi-Agent Organizational Paradigms," Knowl. Eng. Rev., **19**(4), pp. 281–316.
[25] Thompson, J. D., 1967, *Organizations in Action: Social Science Bases of Administrative Theory*, Transaction Publishers, New Brunswick, NJ.
[26] Galbraith, J. R., 1977, *Organization Design*, Addison-Wesley, Reading, MA.
[27] Scott, W., 1992, *Richard: Organizations-Rational, Natural, and Open Systems*, Prentice Hall, New York.
[28] Durfee, E. H., Lesser, V. R., and Corkill, D. D., 1987, "Coherent Cooperation Among Communicating Problem Solvers," IEEE Trans. Comput., **100**(11), pp. 1275–1291.
[29] Horling, B., Mailler, R., and Lesser, V., 2004, "A Case Study of Organizational Effects in a Distributed Sensor Network," IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2004, Sept. 20–24, pp. 51–57.
[30] Matson, E., and DeLoach, S., 2003, "Using Dynamic Capability Evaluation to Organize a Team of Cooperative, Autonomous Robots," International Conference on Artificial Intelligence (IC-AI '03), Vol. 2, pp. 744–749.
[31] Barber, K. S., Goel, A., and Martin, C. E., 2000, "Dynamic Adaptive Autonomy in Multi-Agent Systems," J. Exp. Theor. Artif. Intell., **12**(2), pp. 129–147.
[32] So, Y., and Durfee, E. H., 1998, *Designing Organizations for Computational Agents, Simulating Organizations: Computational Models of Institutions and Groups*, MIT Press, Cambridge, MA.
[33] Brooks, C. H., and Durfee, E. H., 2003, "Congregation Formation in Multiagent Systems," Auton. Agents Multi-Agent Syst., **7**(1), pp. 145–170.
[34] Lesser, V. R., 1998, "Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture," Auton. Agents Multi-Agent Syst., **1**(1), pp. 89–111.
[35] Durfee, E. H., 2001, "Scaling up Agent Coordination Strategies," Computer, **34**(7), pp. 39–46.
[36] Humann, J., and Madni, A. M., 2014, "Integrated Agent-Based Modeling and Optimization in Complex Systems Analysis," Procedia Comput. Sci., **28**, pp. 818–827.
[37] Humann, J., Khani, N., and Jin, Y., 2014, "Evolutionary Computational Synthesis of Self-Organizing Systems," Art. Intell. Eng. Design, Anal. Manuf., **28**(3), pp. 259–275.
[38] Ashby, W. R., 1956, *An Introduction to Cybernetics*, Chapman & Hail, London.
[39] Huberman, B. A., and Hogg, T., 1986, "Complexity and Adaptation," Phys. Nonlinear Phenom., **22**(1), pp. 376–384.
[40] Wood, R. E., 1986, "Task Complexity: Definition of the Construct," Organ. Behav. Hum. Decis. Processes, **37**(1), pp. 60–82.
[41] Campbell, D. J., 1988, "Task Complexity: A Review and Analysis," Acad. Manage. Rev., **13**(1), pp. 40–52.
[42] Gell-Mann, M., 2002, "What is Complexity?," *Complexity and Industrial Clusters*, Physica-Verlag, Heidelberg, pp. 13–24.
[43] Bonchev, D. D., and Rouvray, D. H., 1991, *Chemical Graph Theory: Introduction and Fundamentals*, Vol. 1, Abacus Press, New York.
[44] Randi, M., and Plav, D., 2002, "On the Concept of Molecular Complexity," Croat. Chem. Acta, **75**(1), pp. 107–116.
[45] Bonchev, D., 1983, *Information Theoretic Indices for Characterization of Chemical Structures*, Vol. 5, Research Studies Press, Chichester, UK.
[46] Bonchev, D., 2003, "Shannon's Information and Complexity," *Complexity in Chemistry Introduction and Fundamental*, D. Bonchev, and D. H. Rouvray, eds., Taylor & Francis, London, pp. 157–187.
[47] Bonchev, D., 2003, "On the Complexity of Directed Biological Networks," SAR QSAR Environ. Res., **14**(3), pp. 199–214.
[48] Bonchev, D. D., and Rouvray, D., 2007, *Complexity in Chemistry, Biology, and Ecology*, Springer, New York.
[49] Zhang, C., Abdallah, S., and Lesser, V., 2008, "Efficient Multi-Agent Reinforcement Learning Through Automated Supervision," 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '08), Vol. 3, pp. 1365–1370.
[50] Gershenson, C., 2005, "A General Methodology for Designing Self-Organizing Systems," Preprint arXiv:Nlin0505009.
[51] Jin, Y., and Levitt, R. E., 1996, "The Virtual Design Team: A Computational Model of Project Organizations," Comput. Math. Organ. Theory, **2**(3), pp. 171–195.
[52] Wilensky, U., 2001, "Modeling Nature's Emergent Patterns With Multi-Agent Languages," EuroLogo, pp. 43–52.