Swarm Intell
DOI 10.1007/s11721-008-0019-z

Biologically inspired redistribution of a swarm of robots among multiple sites

M. Ani Hsieh · Ádám Halász · Spring Berman · Vijay Kumar

Received: 3 December 2007 / Accepted: 31 July 2008

Abstract We present a biologically inspired approach to the dynamic assignment and reassignment of a homogeneous swarm of robots to multiple locations, which is relevant to applications like search and rescue, environmental monitoring, and task allocation. Our work is inspired by experimental studies of ant house hunting and empirical models that predict the behavior of the colony that is faced with a choice between multiple candidate nests. We design quorum based stochastic control policies that enable the team of agents to distribute themselves among multiple candidate sites in a specified ratio, and compare our results to the linear stochastic policies described in (Halasz et al., in Proceedings of the International Conference on Intelligent Robots and Systems (IROS'07), pp. 2320–2325, 2007). We show how our quorum model consistently performs better than the linear models while minimizing computational requirements and now it can be implemented without the use of inter-agent wireless communication.

Keywords Robot swarms · Task allocation · Decentralized control · Bio-inspired control

1 Introduction

Consider a collection of tasks each to be executed at different locales. We address the problem of allocating a homogeneous swarm of robots among the locations to accomplish these

We gratefully acknowledge the support of NSF grants CCR02-05336 and IIS-0427313, and ARO Grants W911NF-05-1-0219 and W911NF-04-1-0148.

M.A. Hsieh (✉) · Á. Halász · S. Berman · V. Kumar
GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104-6228, USA
e-mail: mya@grasp.upenn.edu

Á. Halász
e-mail: halasz@grasp.upenn.edu

S. Berman
e-mail: spring@grasp.upenn.edu

V. Kumar
e-mail: kumar@grasp.upenn.edu

tasks. This problem is relevant to numerous applications such as the surveillance of multiple buildings, large scale environmental monitoring or providing aerial coverage for ground units. In these applications, robots must have the ability to distribute themselves among various locations/sites and they must also have the ability to autonomously redistribute to ensure task completion at each site, which may be affected by robot failures or changes in the environment. Additionally, in applications such as mining and search and rescue, inter-agent wireless communication may be unreliable or non-existent. As such, it makes sense to consider the development of a (re)distribution strategy that requires little to no inter-agent wireless communication.

This is similar to the task/resource allocation problem where the objective is to determine the optimal assignment of robots to tasks. In the multi-robot domain, existing methods for the solution of these combinatorial optimization problems often reduce to market-based approaches (Lin and Zheng 2005; Guerrero and Oliver 2003; Jones et al. 2006) where robots must execute complex bidding schemes to determine the appropriate allocation based on the various perceived costs and utilities. While market-based approaches have gained much success in various multi-robot applications (Dias et al. 2006; Vail and Veloso 2003; Gerkey and Mataric 2002; Jones et al. 2007) and can be further improved when learning is incorporated (Dahl et al. 2006), these methods often scale poorly in terms of team size and number of tasks (Dias 2004; Golfarelli and Maio 1997).

In nature, we often see complex group behaviors arise from biological systems composed of large numbers of organisms that individually lack the communication and computational capabilities required for centralized control. Such decentralized consensus-building behaviors are observed in a variety of social organisms, including ants (Pratt et al. 2002), honeybees (Britton et al. 2002), and cockroaches (Ame et al. 2006), and have inspired much research on the development of self-organized task allocation strategies for multi-robot systems. A study on the effects of group size on task allocation in social animal groups, using both deterministic and stochastic swarm models, concluded that larger groups tend to be more efficient because of higher rates of information transfer (Pacala et al. 1996). A similar analysis on robotic swarm systems is described by Krieger et al. (2000), where experimental results showed the effects of group size and recruitment on collective foraging efficiency. Another study on foraging by Labella et al. (2006) shows how division of labor and greater efficiency can be achieved in a robot swarm via simple adaptation of a transition probability based on local information. A common technique in these types of allocation approaches is to employ threshold-based response, in which a robot engages in a task when a stimulus or demand exceeds its response threshold. A comparison of market-based and threshold-based allocation methods has shown that threshold methods outperform traditional market-based approaches when information is inaccurate (Kalra and Martinoli 2006). Agassounon and Martinoli (2002) take a step further to analyze and compare the efficiency and robustness of three threshold-based allocation algorithms for multi-agent systems.

Other recent task allocation strategies specific to multi-robot applications include the work by Shen and Salemi (2002) where the allocation problem is formulated as a Distributed Constraint Satisfaction Problem. This approach requires the explicit modeling of tasks, their requirements, and robot capabilities which makes implementation for large populations difficult. Milutinovic and Lima (2006) maximize robot occupation at a desired position using a centralized optimal control policy where the swarm is modeled via a partial differential equation. Similar works that focus on deriving a continuous model to describe the swarm include (Martinoli et al. 2004) and (Lerman et al. 2006). In these works, a continuous model with a high degree of predictive power is obtained by defining individual robot controllers and averaging their performance. Martinoli et al. (2004) models collaborative stick-pulling

while Lerman et al. (2006) considers an adaptive multi-foraging task. In (Lerman et al. 2006), the task is modeled as a stochastic process that does not involve explicit communication or global knowledge; however, the only way to control robot task reallocation is to modify the task distribution in the environment.

In this paper, we present a bio-inspired approach to the deployment of a robot swarm to multiple sites that is decentralized, robust to changes in team size, and requires no explicit inter-agent wireless communication. Our work draws inspiration from the process through which an ant colony selects a new home from several sites using simple stochastic behaviors that arise from local sensing and physical contact (Franks et al. 2002; Pratt 2005). Similar to (Martinoli et al. 2004; Lerman et al. 2006), our proposed strategy employs a multi-level representation of swarm activity. In contrast to these previous works, we apply a "top-down" design approach where decentralized agent level controllers are obtained by modeling and designing the desired outcome for the group. We build on (Berman et al. 2006, 2007; Halasz et al. 2007) and propose a wireless communication-free quorum sensing mechanism to speed up the allocation process. Furthermore, we address the problem of performance specification and compare the performance of our quorum approach to the linear stochastic models proposed in (Halasz et al. 2007).

This paper is organized as follows: We begin with some background in Sect. 2. In Sect. 3 we formulate the problem and discuss the stability and convergence properties of our controllers in Sect. 4. We describe our methodology in Sect. 5 and present our simulation results in Sect. 6. Section 7 provides a discussion of our results and some considerations for physical implementation of our approach. We conclude with a discussion of directions for future work in Sect. 8.

2 Background

In biology, decentralized consensus building behaviors can be seen in the group dynamics of ants (Pratt et al. 2002), honeybees (Britton et al. 2002), and cockroaches (Ame et al. 2006) to name a few. *Temnothorax albipennis* ants engage in a process of collective decision-making when they are faced with the task of choosing between two new nest sites upon the destruction of their old nest (Franks et al. 2002). This selection process involves two phases. Initially, undecided ants choose one of the sites quasi-independently, often after visiting both. A "recruiter" ant, that is, one who has chosen one of the two candidate sites, returns to the old nest site and recruits another ant to its chosen site through a *tandem run*. The path to the new site is then learned by the "naive" ant by following the recruiter ant. Once the path is learned, the naive ant becomes a recruiter for the particular candidate site; in essence, showing preference for the particular site. In such fashion, ants with preferences for different sites recruit naive ants from the old site to their preferred site. Once a critical population level, or *quorum*, at a candidate site has been attained, the convergence rate to a single choice is then boosted by recruiters, who instead of recruiting naive ants via tandem runs, simply transport naive ants from the old nest to the new one. In this second phase, the "recruitment" rate is significantly faster, and, while completely decentralized, results in the formation of a consensus about which of the two candidate sites to select.

In this natural system, the ants exhibit a small number of distinct, simple behaviors that derive from their allegiance to one of the nests. During the selection process ants make a transition spontaneously between different allegiances at well defined and experimentally measurable rates. The pattern of transition rates, which ultimately determine the average propensity of individual ants to switch behaviors, ensures that the higher quality nest is generally chosen and that no ants are stranded in the worse nest. Figure 1 shows a schematic of

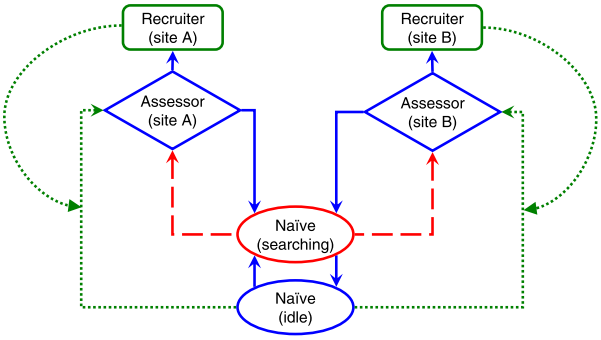

Fig. 1 Switching pattern between the different behaviors exhibited by house hunting ants when choosing between two new nest sites. *Solid lines* represent choices, such as deciding on a site versus searching further; *dashed lines* represent externally influenced outcomes, such as finding a new site; *dotted lines* represent cooperation, such as tandem runs

the switching pattern between the different behaviors exhibited by the Temnothorax albipennis or "house hunting ant". Unlike other ant species, these ants do not employ pheromones as a means of inter-agent communication in the house hunting process,[1] rather the group relies heavily on physical interactions between neighbors. While it has been shown that quorum activated recruitment significantly speeds up convergence to the desired new site (Pratt 2005, Berman et al. 2006, 2007), the exact motivation for the quorum mechanism by the ants is not well understood.

A model for this ant "house hunting" process was initially presented and analyzed in (Berman et al. 2006). In a step away from the ant behavior, Berman et al. (2007) showed how the model in (Berman et al. 2006) can be modified to converge to a state at which *both* new sites are occupied at a pre-specified ratio. This was then further extended to consider the problem of distributing N agents across M distinct locations in (Halasz et al. 2007). The result is a decentralized algorithm for the allocation problem based on the general notion of a quorum.

While Halasz et al. (2007) have been successful in the modeling, analysis, and extension of the ant house hunting model to the problem of allocating a robot swarm to multiple sites, they do not address the problem of performance specification. In this work, we build on (Halasz et al. 2007) and define metrics for the optimal (re)deployment strategy. Additionally, we consider a quorum strategy to improve the performance of the models proposed in (Halasz et al. 2007) based on our defined metrics. Unlike (Halasz et al. 2007), our quorum model results in a decentralized algorithm that can be implemented purely based on local sensing without requiring explicit wireless communication among agents and across sites. A brief summary of the models presented in (Halasz et al. 2007) and the formulation and analysis of our quorum model are described in the following sections.

3 Problem statement

3.1 Definitions

Consider N agents to be distributed among M sites. We denote the number of agents at site $i \in \{1, \ldots, M\}$ at time t by $n_i(t)$ and the desired number of agents at site i by $\bar{n}_i$. We define the population fraction at each site at time t as $x_i(t)$ where $x_i(t) = n_i(t) / \sum_j \bar{n}_j$ for all

[1] A possible explanation is that this would be impractical due to the small size of their colonies (Franks et al. 2002).

$j = 1, \ldots, M$. Then the system state vector is given by $\mathbf{x} = [x_1, \ldots, x_M]^T$. For some initial distribution of the agents given by $\{n_i\}^0$, $i = 1, \ldots, M$, we denote the target configuration as a set of population fractions to occupy each site given by

$$\bar{x}_i = \frac{\bar{n}_i}{\sum_{j=1}^{M} \bar{n}_j} \quad \forall\ j = 1, \ldots, M\ . \tag{1}$$

We note that a specification in terms of fractions rather than absolute agent numbers is practical for scaling as well as for applications where losses of agents to attrition and breakdown are common. Furthermore, such a specification enables us to address the controller synthesis problem by defining the desired group outcome at the population level rather than at the individual level. The task, then, is to redeploy the swarm of N robots to the desired configuration, given by $\bar{\mathbf{x}}$, starting from the initial distribution, $\mathbf{x}(0)$ which will be denoted as $\mathbf{x}^0$, as fast as possible while minimizing inter-agent communication.

We model the interconnection topology of the M sites via a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of vertices, $\mathcal{V}$, represents sites $\{1, \ldots, M\}$ and the set of edges, $\mathcal{E}$, represents physical "one-way" roads between sites. We say two nodes $i, j \in \{1, \ldots, M\}$ are *adjacent*, $i \sim j$, if a one-way road exists for agents to travel from i to j, and we represent this relation by the ordered pair, (i, j), such that $(i, j) \in \mathcal{V} \times \mathcal{V}$, with the set $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} | i \sim j\}$. We assume the graph $\mathcal{G}$ is a strongly connected graph, that is, a directed path exists for any $(u, v) \in \mathcal{V} \times \mathcal{V}$. Here, a *path* from site i to site j is defined as a sequence of vertices $\{v_0, v_1, \ldots, v_p\} \in \mathcal{V}$ such that $v_0 = i$, $v_p = j$ and $(v_{k-1}, v_k) \in \mathcal{E}$ where $k = 1, \ldots, p$. An example of such a graph is shown in Fig. 3. Similarly, when considering the more general task allocation problem, $\mathcal{G}$ should encode the precedence constraints between a collection of tasks represented by $\mathcal{V}$. In such a scenario, $\mathcal{E}$ would then represent the sequential relationships between the individual tasks.

We assign every edge in $\mathcal{E}$ a constant *transition rate*, $k_{ij} > 0$, where k_{ij} defines the transition probability per unit time for one agent to go from site i to site j. Here k_{ij} is essentially a stochastic transition rule and in general $k_{ij} \neq k_{ji}$. In addition, we assume there is a $k_{ij}^{\max}$ associated with every edge $(i, j) \in \mathcal{E}$ which represents the maximum capacity for the given edge. While it is possible to assign nonlinear transition rates, our choice of constant k_{ij}s will allow us to model the transition of agents between sites as Poisson processes to enable our top-down design methodology. Lastly, we define the flux from site i to site j, denoted by ϕ_{ij}, as the fraction of all agents per unit time going from i to j and denote by τ_{ij} the time required to travel from site i to site j.

Central to our method is the use of stochastic switching between simple behaviors at the agent level to generate complex group behaviors. Thus, we assume agents have the ability to localize themselves within the given environment and are capable of executing individual tasks such as: navigation from one site to another and estimation of the population at a given site. Finally, we assume every agent has complete knowledge of $\mathcal{G}$ as well as all the transition rates k_{ij} and their corresponding $k_{ij}^{\max}$. This is equivalent to providing a map of the environment as well as a collection of "lower-level" controllers to every agent.

3.2 The baseline model

Our baseline strategy, first introduced in (Halasz et al. 2007), endows each agent with a small set of instructions based on the transition rates k_{ij} and achieves (re)deployment of the swarm of N robots to M sites using no explicit wireless communication. In the limit of large N, (re)distribution is achieved by modeling the swarm as a continuum rather than a collection

of individual agents and expressing the time evolution of the population fraction at site i by the linear ordinary differential equation (ODE):

$$\frac{dx_i(t)}{dt} = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}x_i(t). \tag{2}$$

This results in a system of equations for the M sites given by

$$\frac{d\mathbf{x}}{dt} = \mathbf{K}\mathbf{x} \tag{3}$$

with $\mathbf{K}_{ij} = k_{ji}$ for $i \neq j$ and $\mathbf{K}_{ii} = -\sum_{j=1,(j,i)\in\mathcal{E}}^{M} k_{ij}$. We note that the columns of $\mathbf{K}$ sum to 0 and since the number of agents is conserved, the system is subject to the following conservation constraint

$$\sum_{i=1}^{M} x_i(t) = 1. \tag{4}$$

This model results in agents transitioning between sites even at equilibrium, that is, when the net flux for each site is zero. This is because such a model forces a trade-off between maximizing the transition rates for fast equilibration and achieving long-term efficiency during equilibrium.

Remark 1 In this work, we could model the system as a traditional linear model of the form $\dot{x} = Ax + Bu$ and determine a suitable feedback of the form $u = -Fx$ to ensure a stable closed-loop system, $\dot{x} = (A - BF)x$, whose equilibrium matches the desired distribution $\mathbf{x}^0$. Instead, we consider the design of the transition rates such that $\mathbf{K}$ determines the closed-loop dynamics of the system and gives the desired equilibrium point, in other words, $\mathbf{K} = A - BF$.

The above model assumes agents switch instantaneously from one site to another. In real cases, it takes a finite time τ_{ij} to travel between sites i and j (or to switch between two tasks). We note that the loss of agents at a site due to transfers to other sites is immediate, while the gain due to incoming agents from other sites is delayed. As such, the above model can be extended to take into consideration the time needed to travel between sites by converting (2) into a delay differential equation given by

$$\frac{dx_i(t)}{dt} = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t-\tau_{ji}) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}x_i(t). \tag{5}$$

Similar to (2), agents in this model will also move between sites at equilibrium. However, because of the time delays, at equilibrium the system will always result in a finite number of agents *en route* between sites. Furthermore, the fraction of agents en route versus the fraction at sites increases as the time delays increase. Thus, the conservation equation for this system is given by

$$\sum_{i=1}^{M} n_i(t) + \sum_{i=1}^{M}\sum_{\forall j|(j,i)\in\mathcal{E}} k_{ij}\tau_{ij}n_i(t) = N, \tag{6}$$

where the first term gives the number of agents at sites and the second term gives the number of agents on all the one-way roads.

We note that while we specify the transition rates at the swarm level, these transition rates, in fact, result in a set of stochastic switching rules that dictate how individual agents should switch from one controller to another given their current state. In other words, at the agent level, the transition rates give the transition probabilities per time unit for an agent to switch away from its current behavior. In the above models, this is equivalent to switching occupancy from site i to site j and vice versa.

It was shown in (Halasz et al. 2007) that, while these models accomplish the multi-site deployment task, the solutions are relatively inefficient since the rate of convergence of the system to the desired configuration depends on the magnitude of the transition rates. While large transition rates ensure fast convergence, they result in many idle trips once the design configuration is achieved. In actual robotic systems, the extraneous traffic resulting from the movement between sites at equilibrium comes at a significant cost. As such, an *optimal deployment strategy* is defined as the choice of $\mathbf{K}$ that simultaneously maximizes convergence towards the desired distribution while minimizing the number of idle trips at equilibrium. In other words, an *optimal transition matrix*, denoted by $\mathbf{K}_*$, is one that can balance short term gains, in this case fast convergence, against long term losses or idle trips at equilibrium.

Lastly, (2) and (5) can be equivalently written as

$$\frac{dx_i(t)}{dt} = \sum_{\forall j|(j,i)\in\mathcal{E}} \phi_{ji}(t) - \sum_{\forall j|(i,j)\in\mathcal{E}} \phi_{ij}(t)$$

and

$$\frac{dx_i(t)}{dt} = \sum_{\forall j|(j,i)\in\mathcal{E}} \phi_{ji}(t-\tau_{ji}) - \sum_{\forall j|(i,j)\in\mathcal{E}} \phi_{ij}(t),$$

respectively.

3.3 The quorum model

In this model, we employ the general concept of quorum and define the *quorum* as a threshold occupancy such that a site that is above the quorum will always transfer agents to adjacent sites at an increased rate. In addition to the assumptions discussed in Sect. 3.1, we will also assume the quorum at each site can be estimated by each agent based on the rate of encounter with other agents at the site (Pratt 2005).

Each site i is then characterized by a quorum, q_i, a threshold number of agents which we specify as a fraction of the design occupancy $\bar{x}_i$. If site i is above the quorum, the transition rate from i to an adjacent site j can be automatically set to either a multiple of the existing transition rate, αk_{ij}, with $\alpha > 0$ chosen to satisfy $\max \alpha k_{ij} < \min k_{ij}^{\max}$, or simply the maximum transition rate, $k_{ij}^{\max}$. We refer to such an edge as *activated*. This is maintained until x_i drops below q_i. Lastly, we will require the following balancing condition for our quorum model:

$$\bar{x}_i k_{ij} = \bar{x}_j k_{ji} \quad \forall (i,j) \in \mathcal{E}. \tag{7}$$

Then the differential equation model with quorum is

$$\frac{dx_i(t)}{dt} = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji} x_j(t) - \sum_{\forall i|(i,j)\in\mathcal{E}} \phi_{ij}(t), \tag{8}$$

where the rates ϕ_{ij} are given by:

$$\phi_{ij}(t) = k_{ij}x_i(t) + \sigma_i(x_i, q_i)(\alpha - 1)k_{ij}x_i(t) \tag{9}$$

or

$$\phi_{ij}(t) = k_{ij}x_i(t) + \sigma_i(x_i, q_i)\big(k_{ij}^{\max} - k_{ij}\big)x_i(t), \tag{10}$$

depending on the choice of edge activation and where $\sigma_i \in [0, 1]$ is the analytic switching function given by

$$\sigma_i(x_i, q_i) = \big(1 + e^{\gamma(q_i - \frac{x_i}{\bar{x}_i})}\big)^{-1}. \tag{11}$$

We note as $(q_i - x_i/\bar{x}_i) \to -\infty$, $\sigma_i \to 1$ with $\sigma_i \to 0$ otherwise. Furthermore, $\gamma \gg 1$ is a constant such that $\sigma_i \approx 1$ when $x_i/\bar{x}_i = q_i + \epsilon$ where $\epsilon > 0$ is small. This is similar to threshold methods described by Bonabeau et al. (1997) and Agassounon and Martinoli (2002).

To incorporate time delays due to nonzero quorum estimation time and nonzero travel time between sites, we formulate (8) as a linear time-delayed differential equation similar to (5). Thus, the quorum with time delay model is given by

$$\frac{dx_i(t)}{dt} = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t - \tau_{ji} - \tau_{E_j}) - \sum_{\forall j|(i,j)\in\mathcal{E}} \phi_{ij}(t - \tau_{E_i}), \tag{12}$$

where τ_{E_i} denotes the time required to estimate the quorum at a given site i. Similar to (9) and (10), the rates ϕ_{ij} are given by:

$$\phi_{ij}(t) = k_{ij}x_i(t - \tau_{E_i}) + \sigma_i(x_i, q_i)(\alpha - 1)k_{ij}x_i(t - \tau_{E_i})$$

or

$$\phi_{ij}(t) = k_{ij}x_i(t - \tau_{E_i}) + \sigma_i(x_i, q_i)\big(k_{ij}^{\max} - k_{ij}\big)x_i(t - \tau_{E_i}),$$

depending on the choice of edge activation.

4 Analysis

In this section we consider the uniqueness and stability properties of the equilibrium points, that is, the desired final configurations, for the models described in the previous section. We begin by stating our result for the baseline model described by (3) for the sake of completeness and refer the interested reader to (Halasz et al. 2007) for the proof.

Theorem 1 *For a strongly connected graph $\mathcal{G}$, the system (3) subject to (4) has a unique stable equilibrium point.*

We now show that the quorum model described by (8) for all i has a stable equilibrium point that satisfies the desired specifications (1). We begin with the edge activation scheme given by (9). For simplicity we assume $q_i = q$ for all i. Consider the following candidate Lyapunov function given by

$$V = \sum_{i=1}^{M} \frac{x_i^2}{2\bar{x}_i}. \tag{13}$$

Theorem 2 *The system defined by* (8) *for* $i = 1, \ldots, M$ *for all* $(i, j) \in \mathcal{E}$ *with condition* (7) *and the conservation constraint* (4), *converges asymptotically to* $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_M]^T$, *defined by specification* (1).

Proof We begin by showing the system is stable. First, note that V is a radially unbounded function of $\|\mathbf{x}\|$. We define the net flux from site i to site j as $\Phi_{ij} = -\phi_{ij} + k_{ji}x_j$. We note that $\Phi_{ij} = -\Phi_{ji}$ and by design $\Phi_{ij} = -\phi_{\max} + k_{ji}x_j < 0$ if $x_i/\bar{x}_i > q$ and $x_j/\bar{x}_j < q$ and $\Phi_{ij} = -k_{ij}x_i + \phi_{\max} > 0$ if $x_i/\bar{x}_i < q$ and $x_j/\bar{x}_j > q$. If both sites are above quorum, then Φ_{ij} simplifies to

$$\Phi_{ij} = \alpha(-k_{ij}x_i + k_{ji}x_j) \ .$$

Using (7), the above equation can be rewritten as

$$\begin{aligned}\Phi_{ij} &= \alpha\left(-k_{ij}x_i + \frac{\bar{x}_i}{\bar{x}_j}k_{ij}x_j\right) \\ &= \alpha k_{ij}\bar{x}_i\left(-\frac{x_i}{\bar{x}_i} + \frac{x_j}{\bar{x}_j}\right).\end{aligned}$$

The above relationship holds when both sites are below quorum except $\alpha = 1$.

Consider the time derivative of the Lyapunov function given by (13)

$$\begin{aligned}\frac{dV}{dt} &= \sum_{i=1}^{M}\frac{x_i}{\bar{x}_i}\frac{dx_i}{dt} \\ &= \sum_{i=1}^{M}\frac{x_i}{\bar{x}_i}\left(\sum_{\forall j|(j,i)\in\mathcal{E}}\phi_{ji}(t) \ - \sum_{\forall i|(i,j)\in\mathcal{E}}\phi_{ij}(t)\right) \\ &= \sum_{\forall i|(i,j)\in\mathcal{E}}\frac{1}{2}\left(\frac{x_i}{\bar{x}_i} - \frac{x_j}{\bar{x}_j}\right)\Phi_{ij}.\end{aligned}$$

By design, if $\Phi_{ij} < 0$, then $x_i/\bar{x}_i > x_j/\bar{x}_j$ and similarly, if $\Phi_{ij} > 0$, then $x_i/\bar{x}_i < x_j/\bar{x}_j$. In the event sites i and j are both above quorum, Φ_{ij} will be opposite in sign to $(x_i/\bar{x}_i - x_j/\bar{x}_j)$. As such,

$$\frac{dV}{dt} = \sum_{\forall j|(i,j)\in\mathcal{E}}\frac{1}{2}\left(\frac{x_i}{\bar{x}_i} - \frac{x_j}{\bar{x}_j}\right)\Phi_{ij} < 0.$$

Thus, the time derivative of the Lyapunov function is always negative and so the system is stable. To show that the equilibrium point is given by (1), consider the set of equilibrium states $\mathbf{x}^e$ satisfying (4), such that $\frac{dV}{dt} = 0$. The time derivative of the Lyapunov function evaluates to zero when all $\Phi_{ij} = 0$ or when $x_i = \bar{x}_i$ for all i. By design $\Phi_{ij} \neq 0$ for all $(i, j) \in \mathcal{E}$ whenever $x_i/\bar{x}_i \neq x_j/\bar{x}_j$, thus $(\frac{x_i}{\bar{x}_i} - \frac{x_j}{\bar{x}_j})\Phi_{ij} < 0$ for all i, j. And the only stable equilibrium is when (1) is satisfied. Thus, the system converges asymptotically to (1). □

Similarly, one can show that the quorum model (8) with edge activation scheme (10) is also in fact stable. However, rather than treating the system as a single continuous system described by (8), we can treat the system as a hybrid system with one mode described by (8) and the other described by (2). The system is in the quorum mode if $x_i > q_i$ for some i

and in the linear mode if $x_i < q_i$ for all i. For simplicity we assume $q_i = q$ for all i and $k_{ij}^{\max} = k^{\max}$ for all $(i, j) \in \mathcal{E}$. Next, consider the following function:

$$W_q = \sum_{i=1}^{M} \max\{x_i - q\bar{x}_i, 0\}, \tag{14}$$

where W_q denotes the fraction of the population that is operating in quorum mode. We note that agents who transition between quorum sites have no net effect on W_q while the flux between sites in quorum and sites not in quorum do. Using nonsmooth analysis, one can show that the time rate of change of W_q is in fact always negative by design. This means that the number of agents operating under quorum mode is always decreasing. Additionally, let W_l denote the occupancy of the most populated site in the linear mode,

$$W_l = \max_i \frac{x_i}{q}. \tag{15}$$

Similarly, one can show that the time rate of change of W_l is always decreasing since $\max_i(x_i/q_i) \geq (x_j/q_j)$ for all j. This means that even in the linear mode, the occupancy at the most populated site always decreases. Therefore, as the system switches between the quorum and linear modes, the system is always exiting each mode at a lower energy state than when it first entered it. Furthermore, once the system enters the linear mode it will not return to the quorum mode and thus the quorum model (8) with edge activation scheme (10) is in fact stable.

We can view the time-delayed quorum model (12) as an abstraction of a more realistic model in which delays are random variables with a certain distribution. To prove stability for this system, we can exploit the fact that a delay differential equation model with gamma-distributed delays can be converted into an ODE model by representing each transition associated with a delay (for instance, each edge in the graph) as a finite sequence of "dummy sites". This approach is similar to the *linear chain trick* (MacDonald 1978). However, it is possible that the addition of the time delays will lead to significant chattering of the system, in other words, cyclic switching between states. This is a topic for future work.

5 Methodology

In this section we describe our methodology for obtaining $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x}^0)$ using our baseline model and our simulation methodology which is based on the abstraction procedure in (Gillespie 1976, 2007). In general, determining an optimal transition matrix $\mathbf{K}$ that satisfies both the short and long term restrictions is not trivial. In addition, given the same set of short and long term requirements, one can either determine a transition matrix that is optimal for the entire domain of initial distributions or a transition matrix that is optimal for the given initial distribution, $\mathbf{x}^0$. While a general optimal $\mathbf{K}$, denoted by $\mathbf{K}_*$, that is suitable for a large range of initial distributions may seem attractive, this may be computationally expensive. On the other hand, an optimal $\mathbf{K}$ specifically tailored for a given initial configuration, denoted by $\mathbf{K}_*(\mathbf{x}^0)$, requires a method for obtaining the initial state of the system which in turns is equivalent to some global knowledge of the system state. While this may seem trivial in certain situations, it may prove to be impossible for others, such as a swarm of robots deployed within a mine or in the ocean.

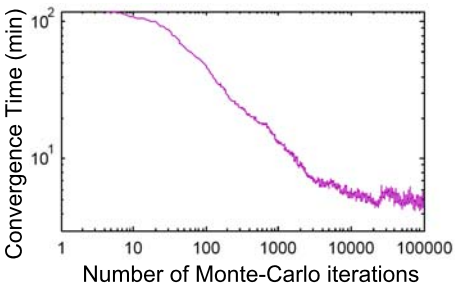

Fig. 2 This plot shows the convergence time for each configuration during a random walk in **K** space where all **K** configurations satisfy the equilibrium and size constraints in the stochastic optimization of the transition rates. The walk is biased with the convergence time so lower times are eventually found

5.1 Optimal design of transition rates

Since our baseline model is linear, we determine $\mathbf{K}_*(\mathbf{x}^0)$ by using (3) to calculate the convergence time to a set number of misplaced agents or agents in transitions, in closed form. This is achieved by decomposing **K** into its normalized eigenvectors and eigenvalues and mapping (3) into the space spanned by its normalized eigenvectors. Thus, given an initial state, $\mathbf{x}^0$, we can apply the appropriate transformation and compute the new state $\mathbf{x}(t)$ using the matrix exponential of the matrix of eigenvalues multiplied by time. After applying the appropriate transformations, we can then calculate the difference in the number of misplaced agents between this new configuration and the desired one, $\bar{\mathbf{x}}$. Since model (3) is stable (Halasz et al. 2007), the number of misplaced agents always decreases monotonically over time. Therefore, a Newton optimization scheme is used to calculate the exact time when the number of misplaced agents is reduced to 10% of its initial value. Figure 2 shows the convergence times for the stochastic optimization of the transition rates.

To find the optimal **K** for the given initial configuration, $\mathbf{x}^0$, Metropolis optimization (Landau and Binder 2000) is used with the entries of **K** as the optimization variables. The objective function then is to minimize the convergence time subject to upper bounds on the number of idle trips at equilibrium and the individual transition rates, in other words, $k_{ij} \leq k_{ij}^{\max}$. A similar procedure can be used to obtain $\mathbf{K}_*$ by applying the optimization scheme for a large enough set of randomly selected initial configurations.

5.2 From individual transitions to ODEs

Our control policies, given by $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x}_0)$, were derived from the continuous baseline model (3) which assumes an infinitely large number of agents in a swarm. In practice, while the population size of a swarm is not infinitely large, it is often significant to render most agent-based simulations computationally costly. As such, it makes sense to develop an equivalent intermediate level of description, termed macro-discrete, as opposed to the micro-discrete (agent level) and macro-continuous (ODE) models, that will allow for relatively inexpensive simulations and retain some of the features of an agent-based simulation where the agents are assumed to be homogeneous.

The correspondence between a given ODE model and a set of individual stochastic transition rules is straightforward if we choose to implement the latter as Poisson transitions controlled by fixed transition rates. Consider two states, i and j, which can be either behaviors or could correspond to physically separate locations/sites. Assume that initially all agents are at site i and they all follow a stochastic transition rule by which they are to move to site j with rate $k = 1.0 \times 10^{-4}$ per second. At every iteration (assume one second per iteration for simplicity), each agent runs a random process with two possible outcomes, 0 or 1,

such that the probability of 1 is given by $k \times \Delta t = 1.0 \times 10^{-4}$. If the outcome is 1, the agent moves to site j, otherwise it stays at i. It is clear that, given a large number of agents, the number of agents remaining at site i after time t is well approximated by $N_A(t) = N_{\text{initial}} e^{-kt}$. Alternatively, instead of generating a random number each time, an agent could generate a random number $\hat{t}$ distributed according to the Poisson distribution

$$f(t) = \frac{1}{k} e^{-k\hat{t}} \tag{16}$$

and perform its transition at time $t = \hat{t}$. The two methods described above are mathematically equivalent in the limit of very short sampling times Δt. If the random number generators used by the agents are independent, then the individual transition times will be distributed according to the Poisson law (16) and the time dependence of the number of agents remaining at site i will approximate the continuous formula.

The idea for the macro-discrete model is as follows. Suppose that at $t = 0$ we have $N_i = 50$ agents at site i. The transition probability per unit time for *each* agent is k, so that the individual probability of transition between 0 and (an infinitesimally small) Δt is $\Delta p = k \Delta t$. The probability for *any one* of the N_i agents transitioning in the same time interval is $N_i k \Delta t$. Thus, the distribution of the time of the *first* transition is similar to that for a single agent, only with probability rate (or *propensity*) $N_i k$:

$$f(t, N_i) = \frac{1}{N_i k} e^{-N_i k t} \ . \tag{17}$$

We can then simulate the consecutive transitions in a single program, where we only follow *the number* of agents at each site, and generate transitions according to (17). Of course, once the first transition takes place, the number N_i is decreased by one unit $N_i' = N_i - 1$, and the next transition is generated using propensity $N_i' k$.

This is an illustration of the more general Gillespie algorithm (Gillespie 2007), where the system is described in terms of the number of agents at each site and transition times are generated consecutively using properly updated propensities. We stress that this method is *mathematically equivalent* to an agent-based simulation where the swarm is assumed to be homogeneous and individual agents follow the respective Poisson transition rules. This method has the advantage of much faster execution compared to an agent-based simulation. We refer the interested reader to (Berman et al. 2006) for further discussion of this framework.

6 Simulations

We consider the deployment of 20,000 planar homogeneous agents to 42 sites, each executing controllers derived from the model described by (2) with a nonoptimal choice for $\mathbf{K}_u$, an optimal $\mathbf{K}_*$ that is independent of the initial configuration, and an optimal $\mathbf{K}_*(\mathbf{x}^0)$ that is specifically chosen for a particular initial configuration given by $\mathbf{x}^0$. We then compare the performance of the same system with agents executing controllers derived from the quorum model with the edge activation scheme given by (10).

For our simulations, $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x}^0)$ were obtained following the methodology described in Sect. 5. Both $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x}^0)$ were computed assuming the same upper bounds on the number of idle trips at equilibrium and the transition rates. In all models, we assume transitions between sites occur spontaneously based on the probability per time unit of transition

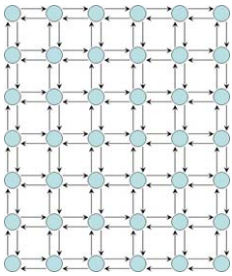

Fig. 3 The network of the 42 sites used in our simulations. The *arrows* denote the direction of travel that is allowed between adjacent sites

Table 1 Performance characteristics for the different transition matrices shown in Fig. 5

Choice of $\mathbf{K}$	Time units to 2/3 of initial deviation	Idle trip transition rate at equilibrium	Maximum k_{ij}
$\mathbf{K}_u$	21.81	1	1.35
$\mathbf{K}_*$	4.51	1	11.88
$\mathbf{K}_*(\mathbf{x}_0)$	1.84	1	6.94

Table 2 Performance characteristics for the quorum model with edge activation given by (10) vs. the baseline model with different transition matrices shown in Fig. 6

Choice of $\mathbf{K}$	Time units to 2/3 of initial deviation	Idle trip transition rate at equilibrium	Maximum k_{ij}
$\mathbf{K}_u$	179.3	1	0.396
$\mathbf{K}_{\max}(\mathbf{x}_0)$	14.18	12.64	12
Quorum with $\mathbf{K}_u$	19	1	12

to any adjacent site, k_{ij}. The interconnection topology of the sites is shown in Fig. 3 where each edge represents a one-way road connecting two sites.

For our simulation, agents are initially scattered at sites configured to form the number 0, and the task is to redistribute the team to another set of sites such that these sites form the number 8. Snapshots of the simulation are shown in Fig. 4 where sites are shown as circles. The circles are drawn to reflect the number of agents at each site. A larger circle represents a higher density.

While our focus is on the global design and properties of the swarm, our methodology takes into account the exact number of agents assigned to each site as well as the travel start and termination times for each individual traveler.

For our first set of simulations, the transition matrix $\mathbf{K}$ was set to $\mathbf{K}_u$ (a nonoptimal $\mathbf{K}$), $\mathbf{K}_*(\mathbf{x}^0)$ and $\mathbf{K}_*$. Here, agents switch between sites with controllers derived from (3) where quorum activation is absent. In all these simulations, the equilibrium transition rate from site i to site j was chosen to be 1 for $\mathbf{K}_u$, $\mathbf{K}_*(\mathbf{x}^0)$ and $\mathbf{K}_*$. This is equivalent to bounding the total number of idle trips at equilibrium. Similarly, $k^{\max}$ was set to 12 in choices for $\mathbf{K}_u$, $\mathbf{K}_*(\mathbf{x}^0)$ and $\mathbf{K}_*$.

Figure 5 shows the fraction of misplaced agents for the system, that is, the sum of the absolute values of the difference between the actual and desired fraction of agents at each site over the total, over time for the same system with three different choices of $\mathbf{K}$. It is not surprising that both $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x}^0)$ outperforms $\mathbf{K}_u$ in terms of convergence speed for the same bound on idle trips at equilibrium. The performance characterictics each $\mathbf{K}$ are summarized in Table 1.

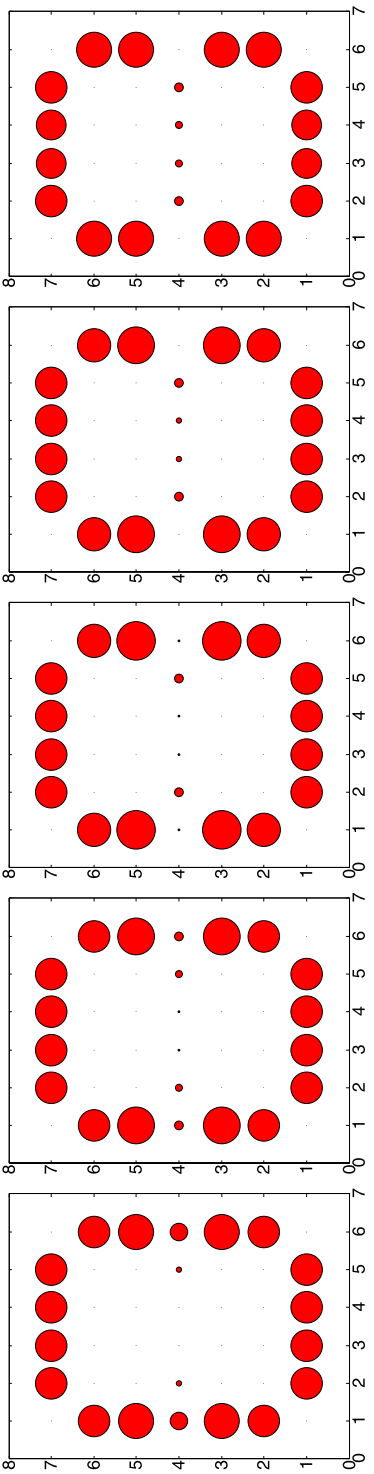

Fig. 4 From *top left* to *bottom right*. Snapshots from a simulation of 20,000 agents. The initial configuration forms the number 0 with the design specification for the number 8. This simulation is based on the model given by (3) with non-optimal choice for **K**

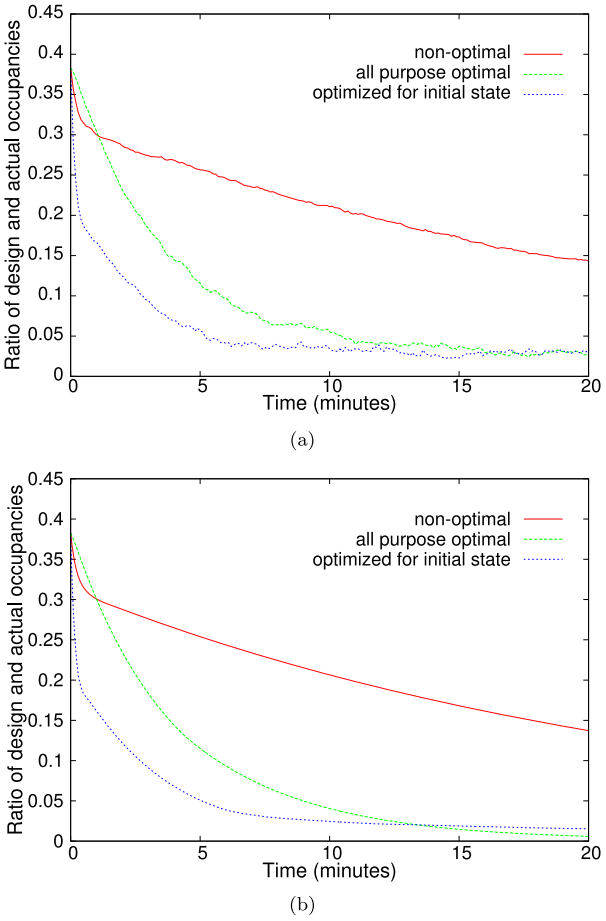

Fig. 5 Fraction of misplaced agents over time for the system model given by (3) for three choices of $\mathbf{K}$: $\mathbf{K}_u$, $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x}_0)$. (**a**) Stochastic simulation. (**b**) Differential equation simulation

Figure 6 shows the number of misplaced agents for the system over time for the baseline model (3) with k_{ij} chosen from $\mathbf{K}_u$ and $\mathbf{K}_{\max}(\mathbf{x}_0)$ and the quorum model (8) with edge activation scheme given by (10) with k_{ij} chosen from $\mathbf{K}_u$. Here, $\mathbf{K}_{\max}(\mathbf{x}^0)$ is the optimal transition matrix given an initial configuration $\mathbf{x}^0$ subject solely to constraints $k_{ij} \leq k^{\max}$ with no constraints on idle trips at equilibrium. This means that $\mathbf{K}_{\max}(\mathbf{x}^0)$ is the optimal transition matrix with respect to convergence speed. In these simulations we see that the quorum model allows us to maximize transient transfer rates between sites without sacrificing the limit on the number of idle trips at equilibrium. Table 2 summarizes the different properties of the three systems shown in Fig. 6.

A second deployment of 20,000 agents to 42 sites is shown in Fig. 7. For this simulation, agents are initially scattered at the four corner sites, and the task is to redistribute the team to another set of sites such that these sites form the letter S. Snapshots of the simulation are shown in Fig. 7 where agents are denoted by dots arranged into circles at each site.

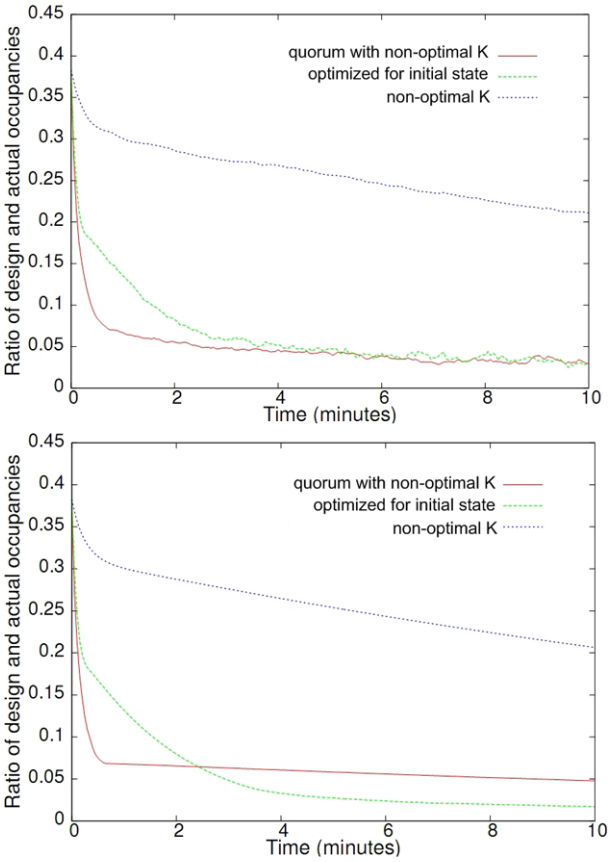

Fig. 6 Fraction of misplaced agents over time for the system model given by (2) with $\mathbf{K}_*(\mathbf{x}_0)$, $\mathbf{K}_{\max}$, and the quorum model given by (8) with k_{ij} given by $\mathbf{K}_u$. (**a**) Stochastic simulation. (**b**) Differential equation simulation

7 Discussion

7.1 Results

In biology, complex group behaviors often arise from numerous interactions of simple individual behaviors. In (Berman et al. 2006, 2007), it was shown how the differential equations of the global evolution of the swarm stem from simple, stochastic switching rules executed by the individuals. These rules are equivalent to the transition rates k_{ij} encoded in $\mathbf{K}$.

In (Halasz et al. 2007), we applied our switching paradigm to a multi-site system, where a large swarm of agents could switch between 42 possible "sites" and showed how a simple set of switching rules can be used to design a decentralized strategy for switching between complex configurations. The "sites" used in our model can be interpreted as behaviors, physical locations, or even internal states. The formalism scales well with the number of sites and agents, and thus can be applied to many different situations.

In this work, we build on the deployment scheme presented in (Halasz et al. 2007) and propose a quorum strategy for the (re)distribution of a swarm of autonomous agents to a

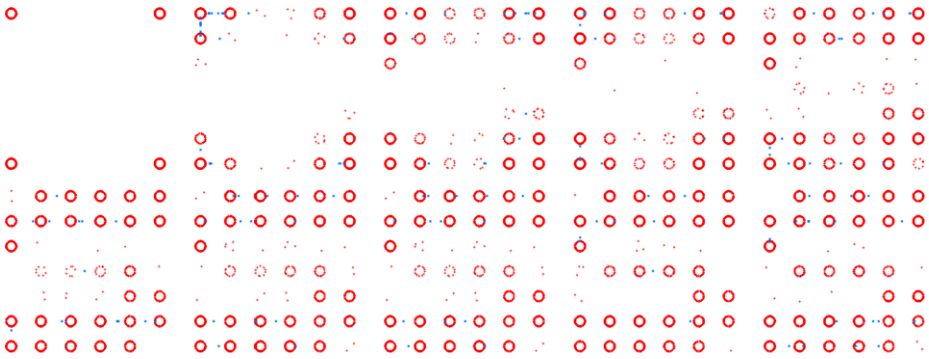

Fig. 7 From *top left* to *bottom right*. Snapshots from a simulation of 20,000 agents. Agents are initially located at the four corner sites and distribute themselves to form the design specification for the letter S. This simulation is based on the quorum model given by (8)

large number of sites without explicit inter-agent wireless communication. Specifically, our quorum based (re)distribution is achieved by setting quorum to 5% above the desired design occupancy for a site. The choice of the underlying baseline model and the rules applied in the above-quorum regime are independent of each other. The only requirement is that the baseline model, that is, the below quorum model, must obey (7) and that the quorum transition rates must be higher than the highest transition rate of the baseline model for the same link. In addition, we showed that the quorum model does, in fact, converge to the desired state (in the absence of time delays). However, it is not mathematically obvious why it provides a faster convergence rate as compared to the baseline model. Lacking a first-principles justification of our quorum approach, we have shown numerically that indeed, this heuristic is significantly more efficient than any comparable baseline model with fixed $\mathbf{K}$ and it approaches the best available performance under the given circumstances.

To enable comparison between the various models we defined two performance metrics: fast convergence and bounds on idle trips at equilibrium. Although the exact motivations for these quorum strategies in ants have not been clearly understood (Pratt et al. 2002; Pratt 2005), one explanation is that the quorum mechanism is used to secure the choice of a candidate new nest site for ants by speeding up convergence. This can be seen from our simulations which indicate that the quorum strategy, with a nonoptimal choice for $\mathbf{K}$, consistently outperforms the baseline models regardless of whether $\mathbf{K}$ is chosen to be $\mathbf{K}_*(\mathbf{x}^0)$ or $\mathbf{K}_*$. While we have only presented results for one pair of initial and final configurations, our exploration of a varied set of initial and final conditions supports this finding.

Lastly, one can argue that as processors become faster, the computational costs associated with determining $\mathbf{K}_*$ or $\mathbf{K}_*(\mathbf{x}^0)$, a priori, become trivial. However, this does not address the question of how best to obtain $\mathbf{K}_*$ since the notion of globally optimal convergence is not well defined and its proper definition is likely to be sensitive to many implementation issues such as the magnitude of uncertainties. Further, the cost of computation, per se, is likely to be dwarfed by the sensing and communication costs associated with estimating $\mathbf{x}^0$, should $\mathbf{x}^0$ be unknown. As such, the quorum strategy becomes extremely attractive since it can be implemented with minimal computation and communication resources.

7.2 Towards physical implementation

In this section we provide a brief description of the implementation of our decentralized controllers at the agent level and discuss the effects of noisy sensors and actuators on the performance of our methodology.

At the agent level, the controllers are implemented based on the simulation methodology described in Sect. 5.2. Rather than have every agent generate a random number to determine whether it should switch from one site to another, we use the alternative approach where each agent computes the time to switch to another site. Each time an agent arrives at site i, it generates a random number $\hat{t}_{s_j}\ \forall j$ such that $i \sim j$, distributed according to the Poisson distribution given by (16) with the appropriate k_{ij}. Then, $t = \min_j \hat{t}_{s_j}$, gives the time when the agent switches from the current site i to the neighboring site j. The implementation of the controller is provided in Algorithm 1.

To incorporate the quorum mechanism, we assume agents have the ability to determine whether a site is either above or below the quorum based on encounter rates with other agents. Thus, once an agent arrives at a site i, it first estimates the quorum level and then executes a similar procedure to determine the time it should transition to a neighboring site. The quorum controller is provided in Algorithm 2. We note that, while the transition times in our algorithms are computed based on each agent's internal clock, the implementation of these controllers does not require the synchronization of all clocks as long as the transition times are distributed according to the Poisson law (16).

In physical systems, inaccuracies due to navigation are common since sensors and actuators are generally noisy. A natural question that arises is the extent to which our approach can be generalized to real-world situations where nonlinear effects cannot be ignored. In this

Algorithm 1 Agent-level controllers derived from the linear model

Given a strongly connected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ s.t. $u \in \mathcal{V}$, and $\mathbf{K}$
Compute $\hat{t}_{s_j}$ from (16) with $k = k_{ij}\ \forall j : i \sim j$
if *currentTime* $\leq \min_j t_{s_j}$ **then**
  Stay at current site
else
  Switch to site $k = \arg\min_j t_{s_j}$ at *currentTime* $= \min_j t_{s_j}$
end if

Algorithm 2 Agent-level controllers derived from the quorum model

Given a strongly connected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ s.t. $u \in \mathcal{V}$, and $\mathbf{K}$
Estimate the current occupancy level at site i given by $x_i/\bar{x}_i$
if $x_i/\bar{x}_i < q_i$ **then**
  Compute $\hat{t}_{s_j}$ from (16) with $k = k_{ij}\ \forall j : i \sim j$
else
  Compute $\hat{t}_{s_j}$ from (16) with $k = k_{ij}^{\max}\ \forall j : i \sim j$
end if
if *currentTime* $\leq \min_j t_{s_j}$ **then**
  Stay at current site
else
  Switch to site $k = \arg\min_j t_{s_j}$ at *currentTime* $= \min_j t_{s_j}$
end if

work, we employ linear transition rates that allow us to model the transition of agents between sites as Poisson processes, which in turn result in a set of switching rules for individual agents. In cases where the nonlinearities can be well quantified, there is no obstacle to deriving a top-level differential equation that has the appropriate nonlinear terms. This, of course, would require a method of mapping the swarm level nonlinear feedback controllers to a set of switching rules that can be executed by individual agents. In the case of linear transition rates, there is a one-to-one correspondence between the swarm and agent level transition matrices which may no longer hold when nonlinearities are introduced. Nonetheless, there are many analysis methods that can be applied to the resulting nonlinear differential systems that may potentially be orders of magnitude more efficient than dealing with the agent-level models and this is clearly a direction for future work.

In the presence of weaker nonlinearities, such as the effect of traffic on transit time, a more careful analysis can be achieved. Such inaccuracies can be captured via our delayed differential models, (5) and (12), where the time delays due to navigation and/or quorum estimation are assumed to be constant (Halasz et al. 2007). As discussed in Sect. 4, these delayed differential models are in fact an abstraction of more realistic models where the delays are modeled as random variables that follow some distribution. In the work by Berman et al. (2008), an empirical distribution of the transit times are derived for a given level of traffic and modeled as an expanded ODE where each edge in the interconnection graph is represented by a sequence of fictitious sites. In other words, we can convert the continuous time-delayed models into an equivalent baseline model with no time delays by representing each edge with a finite sequence of "dummy sites". The optimality of the transition rates derived from this expanded model is then improved using successive approximations. This equivalence can be exploited when optimizing $\mathbf{K}_*$ as long as the inaccuracies introduced by the physical system can be mapped to stochastic time delays.

8 Conclusion

We have presented a bio-inspired approach to the (re)distribution of a swarm of robots among a set of available sites. Our methodology models the swarm as a hybrid system where agents switch between maximum transfer rates and constant transition rates dictated by the baseline model. Furthermore, our methodology models the swarm as a continuum in terms of population fractions which enables the synthesis of decentralized controllers with no communication. We then compared the performance of our quorum model to three versions of our baseline model and showed that our quorum model consistently outperforms the baseline, regardless of the choice of $\mathbf{K}$, in both time for convergence and number of idle trips at equilibrium.

One immediate direction for future work is the development of a more systematic method for the determination of an optimal transition matrix for a given network topology subject to an upper bound on the equilibrium idle trips for the baseline model. Additionally, we would like to incorporate agents' estimation of whether a quorum exists based on their own observations into our models. Lastly, we would like to extend our approach to incorporate nonlinear transition rules that may allow us to take inspiration from motifs in biomolecular networks and allow for interesting phenomena such as spontaneous switching as a result of an external input.

References

Agassounon, W., & Martinoli, A. (2002). Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proceedings of the first international joint conference on autonomous agents and multi-agent systems (AAMAS'02)* (pp. 1090–1097). New York: ACM.

Ame, J., Halloy, J., Rivault, C., Detrain, C., & Deneubourg, J.-L. (2006). Collegial decision making based on social amplification leads to optimal group formation. *Proceedings of the National Academy of Sciences of the USA*, *103*(15), 5835–5840.

Berman, S., Halasz, A., Kumar, V., & Pratt, S. (2006). Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system. In *Lecture notes in computer science*: *Vol. 4433*. *Swarm robotics* (pp. 56–70). Berlin: Springer.

Berman, S., Halasz, A., Kumar, V., & Pratt, S. (2007). Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. In *Proceedings of the 2007 IEEE international conference on robotics and automation (ICRA'07)* (pp. 2318–2323). Los Alamitos: IEEE.

Berman, S., Halasz, A., Hsieh, M. A., & Kumar, V. (2008). Navigation-based optimization of stochastic deployment strategies for a robot swarm to multiple sites. In *47th IEEE conference on decision and control*. Los Alamitos: IEEE.

Bonabeau, E., Sobkowski, A., Theraulaz, G., & Deneubourg, J.-L. (1997). Adaptive task allocation inspired by a model of division of labor in social insects. In D. Lundh, B. Olsson, & A. Narayanan (Eds.), *Biocomputing and emergent computation* (pp. 36–45). Singapore: World Scientific.

Britton, N. F., Franks, N. R., Pratt, S. C., & Seeley, T. D. (2002). Deciding on a new home: how do honeybees agree? *Proceedings of the Royal Society B: Biological Sciences*, *269*(1498), 1383–1388.

Dahl, T. S., Matarič, M. J., & Sukhatme, G. S. (2006). A machine learning method for improving task allocation in distributed multi-robot transportation. In D. Braha, A. Minai, & Y. Bar-Yam (Eds.), *Understanding complex systems: science meets technology* (pp. 307–337). Berlin: Springer.

Dias, M. B. (2004). *TraderBots: a new paradigm for robust and efficient multirobot coordination in dynamic environments*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Dias, M. B., Zlot, R. M., Kalra, N., & Stentz, A. T. (2006). Market-based multirobot coordination: a survey and analysis. *Proceedings of the IEEE*, *94*(7), 1257–1270.

Franks, N., Pratt, S. C., Britton, N. F., Mallon, E. B., & Sumpter, D. T. (2002). Information flow, opinion-polling and collective intelligence in house-hunting social insects. *Philosophical Transactions B: Biological Sciences*, *357*(1429), 1567–1584.

Gerkey, B. P., & Mataric, M. J. (2002). Sold!: Auction methods for multi-robot control. *IEEE Transactions on Robotics & Automation*, *18*(5), 758–768.

Gillespie, D. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*(4), 403–434.

Gillespie, D. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, *58*, 35–55.

Golfarelli, M., Maio, D., & Rizzi, S. (1997). Multi-agent path planning based on task-swap negotiation. In *Proceedings of the 16th UK planning and scheduling SIG workshop*. Durham: PlanSIG.

Guerrero, J., & Oliver, G. (2003). Multi-robot task allocation strategies using auction-like mechanisms. In *Artificial research and development, frontiers in artificial intelligence and applications* (Vol. 100, pp. 111–122). Amsterdam: IOS.

Halasz, A., Hsieh, M. A., Berman, S., & Kumar, V. (2007). Dynamic redistribution of a swarm of robots among multiple sites. In *Proceedings of the conference on intelligent robot systems (IROS'07)* (pp. 2320–2325). Los Alamitos: IEEE.

Jones, E. G., Browning, B., Dias, M. B., Argall, B., Veloso, M., & Stentz, A. T. (2006). Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *Proceedings of the 2006 IEEE international conference on robotics and automation (ICRA'06)* (pp. 570–575). Los Alamitos: IEEE.

Jones, E. G., Dias, M. B., & Stentz, A. (2007). Learning-enhanced market-based task allocation for oversubscribed domains. In *Proceedings of the conference on intelligent robot systems (IROS'07)* (pp. 2308–2313). Los Alamitos: IEEE.

Kalra, N., & Martinoli, A. (2006). A comparative study of market-based and threshold-based task allocation. In M. Gini & R. Voyles (Eds.), *Distributed autonomous robotic systems 7* (pp. 91–102). Minneapolis/St. Paul: Springer.

Krieger, M. J. B., Billeter, J.-B., & Keller, L. (2000). Ant-like task allocation and recruitment in cooperative robots. *Nature*, *406*, 992–995.

Labella, T. H., Dorigo, M., & Deneubourg, J.-L. (2006). Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, *1*(1), 4–25.

Landau, D. P., & Binder, K. (2000). *A guide to Monte-Carlo simulations in statistical physics*. London: Cambridge University Press.

Lerman, K., Jones, C. V., Galstyan, A., & Mataric, M. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, *25*(4), 225–242.

Lin, L., & Zheng, Z. (2005). Combinatorial bids based multi-robot task allocation method. In *Proceedings of the 2005 IEEE international conference on robotics and automation (ICRA'05)* (pp. 1145–1150). Los Alamitos: IEEE.

MacDonald, N. (1978). *Lecture notes in biomathematics: Vol. 27. Time-lags in biological models*. Berlin: Springer.

Martinoli, A., Easton, K., & Agassounon, W. (2004). Modeling of swarm robotic systems: a case study in collaborative distributed manipulation. *International Journal of Robotics Research: Special Issue on Experimental Robotics*, *23*(4–5), 415–436.

Milutinovic, D., & Lima, P. (2006). Modeling and optimal centralized control of a large-size robotic population. *IEEE Transactions on Robotics*, *22*(6), 1280–1285.

Pacala, S. W., Gordon, D. M., & Godfray, H. C. J. (1996). Effects of social group size on information transfer and task allocation. *Evolutionary Ecology*, *10*(2), 127–165.

Pratt, S. C. (2005). Quorum sensing by encounter rates in the ant temnothorax albipennis. *Behavioral Ecology*, *16*(2), 488–496.

Pratt, S., Mallon, E. B., Sumpter, D. J. T., & Franks, N. R. (2002). Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant leptothorax albipennis. *Behavioral Ecology and Sociobiology*, *52*, 117–127.

Shen, W.-M., & Salemi, B. (2002). Towards distributed and dynamic task reallocation. In M. Gini, W.-M. Shen, C. Torras, & H. Yuasa (Eds.), *Intelligent autonomous systems 7* (pp. 570–575). Marina del Rey: IOS.

Vail, D., & Veloso, M. (2003). Multi-robot dynamic role assignment and coordination through shared potential fields. In A. Schultz, L. Parker, & F. Schneider (Eds.), *Multi-robot systems* (pp. 87–98). Dordrecht: Kluwer.