Chapter 1

Ant-Inspired Allocation: Top-Down Controller Design for Distributing A Robot Swarm among Multiple Tasks

Spring Berman, Ádám Halász and M. Ani Hsieh

aindx]Berman, S. aindx]Halász, Á. aindx]Hsieh, M. A.
GRASP Laboratory,
University of Pennsylvania, Philadelphia, PA 19104,
{*spring,halasz,mya*}*@seas.upenn.edu*

We present a decentralized, scalable, communication-less approach to the dynamic allocation of a swarm of homogeneous robots to multiple sites in specified fractions. This strategy has applications in surveillance, search-and-rescue, environmental monitoring, and other task allocation problems. Our work is inspired by an experimentally based model of ant house-hunting, a decentralized process in which a colony attempts to emigrate to the best nest site among several alternatives. In our approach, we design a continuous model of the swarm that satisfies the global objectives and use this model to define stochastic control policies for individual robots that produce the desired collective behavior. We define control policies that are derived from a linear continuous model, a model that includes navigation delays, and a model that incorporates switching behaviors based on quorum sensing. The stability and convergence properties of these models are analyzed. We present methods of optimizing a linear model for fast redistribution subject to a constraint on inter-site traffic at equilibrium, both with and without knowledge of the initial distribution. We use simulations of multi-site swarm deployments to compare the control policies and observe the performance gains obtained through model optimization, quorum-activated behaviors, and accounting for time delays.

1.1. Introduction

Advances in embedded processor and sensor technology that have made individual robots smaller, more capable, and less expensive have also enabled the development and deployment of teams of robotic agents. While multi–robot systems may seem like a recent paradigm shift, distributed ro-

botics research had its genesis in the early 1980's, when the initial focus was primarily on the control and coordination of multiple robotic arms.[1] The unrelenting progression of Moore's law in the 1990's with improvements in sensor and actuation technology, coupled with ubiquitous wireless communication, made it possible to create and deploy teams of robots numbered in the tens and potentially hundreds and thousands. However, as team size increases, it becomes difficult, if not impossible, to efficiently manage or control the team through centralized algorithms or tele–operation. Accordingly, it makes sense to develop strategies in which robots can be programmed with simple but identical behaviors that can be realized with limited on–board computational, communication, and sensing resources.

In nature, the emergence of complex group behaviors from simple agent behaviors is often seen in the dynamics of bee[2] and ant[3] colonies, bird flocks,[4] and fish schools.[5] These systems generally consist of large numbers of organisms that individually lack either the communication or computational capabilities required for centralized control. As such, when considering the deployment of large robot teams, it makes sense to consider such "swarming paradigms" where agents have the capability to operate asynchronously and determine their trajectories based on local sensing and/or communication. One of the earliest works to take inspiration from biological swarms for motion generation was presented in 1987 by Reynolds,[6] who proposed a method for generating visually satisfying computer animations of bird flocks, often referred to as *boids*. Almost a decade later, Vicsek *et al.* showed through simulations that a team of autonomous agents moving in the plane with the same speeds but different headings converges to the same heading using nearest neighbor update rules.[7] The theoretical explanation for this observed phenomenon was provided by Jadbabaie *et al.*,[8] and Tanner *et al.* extended these results to provide detailed analysis of the stability and robustness of such flocking behaviors.[9] These works show that teams of autonomous agents can stably achieve consensus through local interactions alone, i.e. without centralized coordination, and have attracted much attention in the multi-robot community.

We are interested in the deployment of a swarm of homogeneous robots to various distinct locations for parallel task execution at each locale. The ability to autonomously distribute a swarm of robots to physical sites is relevant to many applications such as the surveillance of multiple buildings, large-scale environmental monitoring, and the provision of aerial coverage for ground units. In these applications, robots must have the ability to distribute themselves among many locations/sites as well as to autonomously

redistribute to ensure task completion at each site, which may be affected by robot failures or changes in the environment.

This problem of (re)distribution is similar to the task/resource allocation problem, in which the objective is to determine the optimal assignment of robots to tasks. Such combinatorial optimization problems are known to scale poorly as the numbers of agents and tasks increase. Furthermore, in the multi–robot domain, existing methods often reduce to market–based approaches[10–12] in which 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,[13–16] the computation and communication requirements often scale poorly in terms of team size and number of tasks.[17,18] Therefore, as the number of agents increases it is often impractical, if not unrealistic, to expect small, resource-constrained agents to always have the ability to explicitly communicate with other team members, especially those physically located at other sites, *e.g.* in mining and search-and-rescue applications. For this reason, we are interested in developing a decentralized strategy to the allocation problem that can lead to optimal outcomes for the population in a manner that is efficient, robust, and uses *minimal* communication.

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 by which an ant colony selects a new home from several sites using simple stochastic behaviors that arise from local sensing and physical contact.[3,19] While there are many existing bio–inspired swarm coordination strategies,[8,9,20–23] group behaviors are often obtained from the synthesis of a collection of individual agent behaviors. Rather than follow these bottom–up approaches to group behavior synthesis, we propose a methodology that enables the design of group behaviors from global specifications of the swarm which can then be realized on individual robots. In other words, we provide a top–down approach to group behavior synthesis such that the resulting agent closed–loop control laws will lead the population to behave in the prescribed manner.

1.2. Background

1.2.1. *Related Work*

Social insect colonies have inspired much research on the development of self-organized task allocation strategies for multi-robot systems. In these decentralized strategies, robots switch between action states based on environmental stimuli and, in some cases, interactions with other robots. A study on the effects of group size on task allocation in social animal groups concluded, using deterministic and stochastic swarm models, that larger groups tend to be more efficient because of higher rates of information transfer.[24] A similar analysis on robotic swarm systems used the implementation of ant-inspired algorithms to demonstrate the effect of group size and recruitment on collective foraging efficiency.[25] Another study on foraging showed 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.[26]

A common technique in these types of allocation approaches is to employ a threshold-based response, in which a robot becomes engaged in a task, deterministically or probabilistically, when a stimulus or demand exceeds its response threshold. A comparison of market–based and threshold methods for task allocation has shown that threshold methods perform more efficiently when information is inaccurate.[27] The efficiency and robustness of three threshold–based allocation algorithms have been analyzed and compared for an object aggregation scenario.[28] One task allocation strategy, based on a model of division of labor in a wasp colony, minimizes global demand by modeling agents' preferences for particular tasks as stochastic, threshold-based transition rates that are refined through learning.[29]

Other recent work on distributed task allocation focuses on deriving a continuous model of the system with a high degree of predictive power by defining individual robot controllers and averaging their performance. In these "bottom-up" methods, the main challenge is to derive an appropriate mathematical form for the task transition rates in the model.[30] The rates are computed from physical robot parameters, sensor measurements, and geometrical considerations, under the assumption that robots and their stimuli are uniformly spatially distributed. The resulting continuous model is validated by comparing steady-state variables and other quantities of interest to the results of embodied simulations and experiments. The applications that have been modeled include collaborative stick-pulling,[20] object

clustering,[21] and adaptive multi-foraging.[22] In the last application, which uses a swarming paradigm similar to the one we consider, the task is modeled as a stochastic process and it involves no explicit communication or global knowledge. However, the only way to control robot task reallocation is to modify the task distribution in the environment.

In the task allocation methods discussed, it is necessary to perform simulations of the system under many different conditions to investigate the effect of changing parameters. In contrast, our top-down synthesis approach allows us to *a priori* design the set of transition rates that will guarantee a desired system outcome by carrying out the design at the population level, which can then be used to synthesize individual robot controllers. This work draws inspiration from the process by which a *Temnothorax albipennis* ant colony selects a new home from several sites using simple behaviors that arise from local sensing and physical contact with neighbors. We describe this process in the following section.

1.2.2. *Ant House-Hunting*

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.[19] This selection process involves two phases. Initially, undecided ants choose one of the sites quasi-independently, often after visiting both. A recruiter ant, *i.e.* 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 "naive" ant learns the path to the new site by following the recruiter ant. Once the path is learned, the naive ant is highly likely to become a recruiter for the particular candidate site; in essence, showing a preference for this site. In such a fashion, ants with preferences for different sites recruit naive ants from the old nest 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 leading naive ants in tandem runs, simply transport them from the old nest to the new one. While it has been shown that this quorum-activated recruitment significantly speeds up convergence of the colony to the higher-quality site,[3,31,32] the exact motivations for the quorum mechanism by the ants are not well understood. However, the process itself is completely decentralized and results in the formation of a consensus by the colony on which of the two candidate sites to select as its new home.

This "house-hunting" process has been modeled from experimental observations for the case of two available nest sites of differing quality.[3,19] During the selection process, ants transition spontaneously but at well-defined and experimentally measurable rates between behaviors. The pattern of transition rates, which determines the average propensity of individual ants to switch behaviors, ensures that the higher quality nest is chosen and that no ants are stranded in the worse nest. This outcome has been observed to be robust to both environmental noise and to changes in the colony population.

In our initial studies on applying the ant-inspired swarm paradigm to multi-robot task allocation, we synthesized robot controllers to produce antlike house-hunting activity that resulted in convergence to the better of two sites.[31] We then explored the idea of controlling the degree of distribution; we altered the house-hunting model with realistic ant behaviors such that the synthesized controllers cause the swarm to split between the two sites in a predetermined ratio.[32] In this chapter, we extend this concept to the problem of distributing robots among many sites in predefined fractions. We abandon the ant roles in favor of a simpler set of tasks defined as site occupation. Furthermore, we address the problem of performance specification and compare the performance of the various proposed models.

In the remainder of this chapter, we provide a detailed description and analysis of the various models that can be used to represent the swarm at the population level. We formulate the problem in Section 1.3 and discuss the stability of our derived controllers in Section 1.4. Sections 1.5 and 1.6 summarize our design methodology and the derivation of agent–level controllers from our proposed population models. We present our simulation results in Section 1.7, which is followed by a brief discussion in Section 1.8. We conclude with some final thoughts in Section 1.9.

1.3. Problem Statement

We are interested in the distribution of a large population of N homogeneous agents among M sites. We begin with a brief summary of relevant definitions and a detailed discussion of the various models/abstractions used to represent the swarm of agents.

1.3.1. *Definitions*

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 specify that $\bar{n}_i > 0$ at each site. The population fraction at each site at time t is defined as $x_i(t) = n_i(t)/N$, and the system state vector is given by $\mathbf{x}(t) = [x_1(t), \ldots, x_M(t)]^T$. We denote the desired final distribution of the swarm as

$$\bar{x}_i = \bar{n}_i/N \qquad \forall\ i = 1, \ldots, M\ . \tag{1.1}$$

A specification in terms of fractions rather than absolute agent numbers is practical for scaling as well as for potential applications where losses of agents to attrition and breakdown are common. The task is to redeploy the swarm of robots from an initial distribution $\mathbf{x^0}$ to the desired configuration $\bar{\mathbf{x}}$ while minimizing inter-agent communication.

To model the interconnection topology of the M sites, we use a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that the set of vertices, $\mathcal{V}$, represents sites $\{1, \ldots, M\}$ and the set of edges, $\mathcal{E}$, represents physical routes between sites. Two sites $i, j \in \{1, \ldots, M\}$ are *adjacent*, $i \sim j$, if a route exists for agents to travel directly from i to j. We represent this relation by (i, j) such that the edge set is defined as $\mathcal{E} = \{(i, j) \in \mathcal{V} \times \mathcal{V} | i \sim j\}$. We assume that the graph $\mathcal{G}$ is *strongly connected*, *i.e.* a path exists for any $i, j \in \mathcal{V}$. 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 Figure 1.3.

We consider $\mathbf{x}(t)$ to represent the distribution of the state of a Markov process on $\mathcal{G}$, for which $\mathcal{V}$ is the state space and $\mathcal{E}$ is the set of possible transitions. We assign to 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 at site i to go 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 (*i.e.*, road). Lastly, we define the flux from site i to site j, denoted by ϕ_{ij}, as the fraction of agents per unit time moving from i to j and denote the time required to travel from site i to site j as τ_{ij}.

For a graph that is strongly connected with bidirectional edges ($(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$) but not necessarily fully connected, we consider the case of having a *reversible* Markov process on the graph. A reversible Markov process with respect to $\bar{\mathbf{x}}$ satisfies the *detailed balance equations*:

$$k_{ij}\bar{x}_i = k_{ji}\bar{x}_j\ , \quad (i, j) \in \mathcal{E}\ . \tag{1.2}$$

In this case, we can define a strongly connected, *undirected* graph $\mathcal{G}_u = (\mathcal{V}, \mathcal{E}_u)$ that corresponds to $\mathcal{G}$. $\mathcal{E}_u$ is the set of unordered pairs (i, j) such that the ordered pair (i, j) (and thus also (j, i)) is in the edge set $\mathcal{E}$ of graph $\mathcal{G}$. Each edge $(i, j) \in \mathcal{E}_u$ is associated with a weight w_{ij}, defined as

$$w_{ij} = k_{ij}\bar{x}_i = k_{ji}\bar{x}_j, \quad (i, j) \in \mathcal{E}_u \ . \tag{1.3}$$

We assume that 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 to every agent. Finally, we assume that agents have the ability to localize themselves within the given environment.

1.3.2. *Linear Model*

Our baseline strategy[33] endows each agent with a small set of instructions based on the transition rates k_{ij} and achieves (re)deployment of the swarm among the sites using no explicit wireless communication. Rather than model the system as a collection of individual agents, this strategy models the swarm as a continuum. In the limit of large N, the time evolution of the population fraction at site i is given by a linear equation, the difference between the total influx and total outflux at the site:

$$\dot{x}_i(t) = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}x_i(t) \ . \tag{1.4}$$

The system of equations for the M sites is given by the linear model

$$\dot{\mathbf{x}} \ = \ \mathbf{K}\mathbf{x} \tag{1.5}$$

where $\mathbf{K} \in \mathbb{R}^{M\times M}$. The entries of $\mathbf{K}$ are defined as $\mathbf{K}_{ij} = k_{ji}$, $j \neq i$, if $(j, i) \in \mathcal{E}$ and 0 otherwise, and $\mathbf{K}_{ii} = -\sum_{(i,j)\in\mathcal{E}} k_{ij}$. We note that the columns of $\mathbf{K}$ sum to 0. Additionally, since the number of agents is conserved, the system is subject to the conservation constraint:

$$\sum_{i=1}^{M} x_i(t) \ = \ 1 \ . \tag{1.6}$$

In general, for some desired final distribution $\bar{\mathbf{x}}$, the entries of $\mathbf{K}$ are chosen such that in the limit $t \to \infty$, $\mathbf{x}(t) \to \bar{\mathbf{x}}$. This, in turn, results in agent level closed–loop controllers given by $\mathbf{K}$ that encode the set of agent–level instructions necessary for (re)deployment of the swarm to the various sites.

1.3.3. *Time–Delayed Model*

The linear model assumes that agents instantaneously switch from one site to another. In practice, 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. The linear model can be extended to take into consideration the time needed to travel between sites by converting (1.4) into a delay differential equation (DDE),

$$\dot{x}_i(t) = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t-\tau_{ji}) - \sum_{\forall j|(i,j)\in\mathcal{E}} k_{ij}x_i(t) \tag{1.7}$$

for $i = 1, ..., M$.

Unlike (1.4), the time delays in (1.7) will always result in a finite number of agents *en route* between sites at equilibrium. Furthermore, the fraction of agents en route versus the fraction at sites increases as the time delays increase. Let $n_{ij}(t)$ be the number of robots traveling from site i to site j at time t and $y_{ij}(t) = n_{ij}(t)/N$. Then the conservation equation for this system is:

$$\sum_{i=1}^{M} x_i(t) + \sum_{i=1}^{M} \sum_{\forall j|(i,j)\in\mathcal{E}} y_{ij}(t) \;=\; 1\ . \tag{1.8}$$

1.3.4. *Quorum Model*

In the linear and time-delayed models, agents will move between sites even at equilibrium, when the net flux at each site is zero. This is because these models force a trade-off between maximizing the link capacities for fast equilibration and achieving long-term efficiency at equilibrium. In this section, we propose an extension to model (1.5) to enable fast convergence to the desired state using switching behaviors based on quorum sensing. We define the *quorum* as a threshold occupancy at a site; when a site is above quorum, the agents there will travel to adjacent sites at an increased rate. In addition to assuming that agents have a map of the environment and know $\mathcal{G}$, $\mathbf{K}$, and each k_{ij}^{max}, we assume that they can detect a quorum at a site based on their encounter rate with other agents at the site.[34]

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 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* and activation is maintained until x_i drops below q_i. We assume that the site graph $\mathcal{G}$ has bidirectional edges and that there is a reversible Markov process on the graph, so that equations (1.2) hold.

The differential equation model with quorum is

$$\dot{x}_i(t) = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t) - \sum_{\forall i|(i,j)\in\mathcal{E}} \phi_{ij}(t), \tag{1.9}$$

where the fluxes ϕ_{ij} are defined as

$$\phi_{ij}(t) = k_{ij}x_i(t) + \sigma_i(x_i, q_i)(\alpha - 1)k_{ij}x_i(t) \tag{1.10}$$

or

$$\phi_{ij}(t) = k_{ij}x_i(t) + \sigma_i(x_i, q_i)\left(k_{ij}^{max} - k_{ij}\right)x_i(t) \tag{1.11}$$

depending the choice of edge activation. The function $\sigma_i \in [0, 1]$ is an analytic switch given by

$$\sigma_i(x_i, q_i) = \left(1 + e^{\gamma(q_i - x_i/\bar{x}_i)}\right)^{-1}, \tag{1.12}$$

where $\gamma \gg 1$ is a constant. We note that as $(q_i - x_i/\bar{x}_i) \to -\infty$, $\sigma_i \to 1$ with $\sigma_i \to 0$ otherwise. The constant γ is chosen 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 in[29] and.[28]

To incorporate time delays due to non–zero quorum estimation time and non–zero travel time between sites, we can formulate (1.9) as a delay differential equation:

$$\dot{x}_i(t) = \sum_{\forall j|(j,i)\in\mathcal{E}} k_{ji}x_j(t - \tau_{ji} - \tau_{E_j}) - \sum_{\forall j|(i,j)\in\mathcal{E}} \phi_{ij}(t - \tau_{E_i}), \tag{1.13}$$

where τ_{E_i} denotes the time required to estimate the quorum at site i. The fluxes ϕ_{ij} are given by equation (1.10) or (1.11) depending on the choice of edge activation.

1.4. Analysis

In this section we consider the uniqueness and stability properties of each model's equilibrium point, which is the desired final distribution.

1.4.1. *Linear Model*

We state our first theorem for the linear model and provide a brief sketch of the proof.

Theorem 1.1.[33] *For a strongly connected graph $\mathcal{G}$, the system (1.5) subject to (1.6) has a unique, stable equilibrium point.*

Proof. The proof of this theorem can be easily constructed in two parts. The first part builds on the fact that the rank of $\mathbf{K}$ is $(M-1)$ since the columns of the matrix $\mathbf{K}$ sum to 0. Furthermore, the vector $\mathbf{1}$ exists in the nullspace of $\mathbf{K}^T$. From here we can conclude that the system $\mathbf{Kx} = \mathbf{0}$ subject to (1.6) has a unique equilibrium point. Next, to show that the equilibrium point is in fact a stable one, consider the matrix given by $\mathbf{S} = (1/s)(s\mathbf{I} + \mathbf{K}^T)$, where $s > 0$ and $\mathbf{I}$ is the $M \times M$ identity matrix. We note that for large enough s, $\mathbf{S}$ is a Markov matrix with nonnegative entries which then allows us to prove that $\mathbf{K}^T$ is negative semi–definite. We refer the interested reader to[33] for more detailed exposition. □

1.4.2. *Time-Delayed Model*

We can arrive at a similar conclusion for the time–delayed model (1.7) if we view this model as an abstraction of a more realistic one in which the time delays, τ_{ij}, are random variables from some distribution. If we approximate the probability density function of these delays as a gamma distribution, then the DDE model given by (1.7) can be transformed into an ordinary differential equation (ODE) model of the form (1.4).

To achieve this, we replace each edge (i, j) with a directed path composed of a sequence of dummy sites, $u = 1, ..., D_{ij}$. Assume that the dummy sites are equally spaced; then τ_{ij}/D_{ij} is the deterministic time to travel from dummy site $u \in \{1, ..., D_{ij}\}$ to its adjacent site. The rate at which an agent transitions from one dummy site to the next is defined as the inverse of this time which we denote by $\lambda_{ij} = D_{ij}/\tau_{ij}$. The number of transitions between two adjacent dummy sites in a time interval of length t has a Poisson distribution with parameter $\lambda_{ij}t$. If we assume that the numbers of transitions in non-overlapping intervals are independent, then the probability density function of the travel time T_u between dummy sites u and $u+1$ is given by

$$f(t) = \lambda_{ij} e^{-\lambda_{ij} t} \ . \tag{1.14}$$

Let $T_{ij} = \sum_{u=1}^{D_{ij}} T_u$ be the total time to travel from site i to site j. Since $T_1, ..., T_{D_{ij}}$ are independent random variables drawn from the common den-

sity function (1.14), their sum T_{ij} follows the gamma density function[35]

$$g(t) = \frac{\lambda_{ij}^{D_{ij}} t^{D_{ij}-1}}{(D_{ij}-1)!} e^{-\lambda_{ij} t} \tag{1.15}$$

with expected value $E(T_{ij}) = \tau_{ij}$ and variance $Var(T_{ij}) = \tau_{ij}^2/D_{ij}$. Thus, in an equivalent ODE model, the average travel time from site i to site j is always the delay τ_{ij} from the DDE model, and the number of dummy sites D_{ij} can be chosen to reflect the travel time variance. Lastly, note that $Var(T_{ij}) \to 0$ as $D_{ij} \to \infty$.

Therefore, the equivalent ODE model for the system (1.7) with gamma-distributed time delays can be obtained by replacing each $\dot{x}_i(t)$ with the following set of equations:

$$\begin{aligned}
\dot{x}_i(t) &= \sum_{j|(j,i)\in\mathcal{E}} \lambda_{ji} y_{ji}^{(D_{ji})}(t) \; - \sum_{j|(i,j)\in\mathcal{E}} k_{ij} x_i(t) \; , \\
\dot{y}_{ij}^{(1)}(t) &= k_{ij} x_i(t) \; - \; \lambda_{ij} y_{ij}^{(1)}(t) \; , \\
\dot{y}_{ij}^{(m)}(t) &= \lambda_{ij} \left(y_{ij}^{(m-1)}(t) \; - \; y_{ij}^{(m)}(t) \right) \; , \\
& \quad m = 2, ..., D_{ij} \; ,
\end{aligned} \tag{1.16}$$

where $y_{ij}^{(l)}(t)$ denotes the fraction of the population that is at dummy site $l \in \{1, ..., D_{ij}\}$, λ_{ij} denotes the transition rates between the dummy sites, and $(i,j) \in \mathcal{E}$. Figure 1.1 illustrates how an edge from model (1.4) is expanded with dummy states $y_{ij}^{(l)}$ in (1.16).

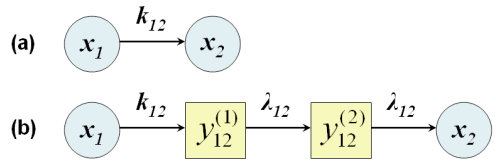


Fig. 1.1. A labeled edge $(i,j) = (1,2)$ that consists of (a) the physical sites, corresponding to model (1.4), and (b) both physical and dummy sites (for $D_{12} = 2$), corresponding to (1.16).

From here, it is easy to see that the equivalent ODE model (1.16) is in fact an expanded version of the linear model (1.4) whose interconnection topology can also be modeled as a directed graph, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \{1, ..., M'\}$ and $\mathcal{E}' = \{(i,j) \in \mathcal{V}' \times \mathcal{V}' \mid i \sim j \}$ with $M' = M + \sum_{i\sim j} D_{ij}$. Since $\mathcal{G}$ is strongly connected, so is $\mathcal{G}'$. Furthermore, the system (1.16) is subject to a similar conservation constraint as (1.6), where the total

number of agents is conserved across all real and dummy sites. We refer to this system as the *chain model*, since it incorporates a chain of dummy sites between each pair of physical sites. We will refer to the corresponding linear model (1.5) (the system without dummy sites) as the *switching model*, since it describes a system in which agents switch instantaneously between sites. This approach is similar to the "linear chain trick" used by MacDonald[36] to transform a system of integro-differential equations with gamma-distributed delays into an equivalent system of ODE's.

Theorem 1.2.[37] *If the graph $\mathcal{G}$ is strongly connected, then the chain model has a unique, stable equilibrium.*

Proof. Since the system can be represented in the same form as (1.5) subject to (1.6), Theorem 1.1 can be applied to show that there is a unique, stable equilibrium. □

1.4.3. *Quorum Model*

In this section we state our final theorem, which concerns stability of the quorum model given by (1.9). We begin with the edge activation scheme given by (1.10) and assume that $q_i = q$ for all i. Consider the following candidate Lyapunov function given by

$$V = \sum_{i=1}^{M} \frac{x_i^2}{2\bar{x}_i}. \tag{1.17}$$

Theorem 1.3.[38] *The system defined by equation* (1.9) *for $i = 1, \ldots, M$ subject to condition* (1.2) *and the conservation constraint* (1.6) *converges asymptotically to $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_M]^T$, defined by the specification* (1.1).

Proof. We first show that the system is stable. We note that V is a radially unbounded function of $\|\mathbf{x}\|$. The net flux from site i to site j is defined as $\Phi_{ij} = -\phi_{ij} + k_{ji}x_j$. Note that $\Phi_{ij} = -\Phi_{ji}$. 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 (1.2), the above equation can be rewritten as

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

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

Consider the time derivative of the Lyapunov function (1.17):

$$\frac{dV}{dt} = \sum_{i=1}^{M} \frac{x_i}{\bar{x}_i} \frac{dx_i}{dt} = \sum_{\forall i | (i,j) \in \mathcal{E}} \frac{1}{2} \left(\frac{x_i}{\bar{x}_i} - \frac{x_j}{\bar{x}_j} \right) \Phi_{ij} \ . \tag{1.18}$$

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 that sites i and j are both above quorum, Φ_{ij} will be opposite in sign to $(x_i/\bar{x}_i - x_j/\bar{x}_j)$. Thus, by (1.18), 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 (1.1), consider the set of equilibrium states $\mathbf{x}^e$ satisfying (1.6), 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 $x_i/\bar{x}_i \neq x_j/\bar{x}_j$, so $\left(\frac{x_i}{\bar{x}_i} - \frac{x_j}{\bar{x}_j} \right) \Phi_{ij} < 0$ for all i, j. Thus, the only stable equilibrium is $\mathbf{x}^e = \bar{\mathbf{x}}$, so the system converges asymptotically to (1.1). □

Similarly, we can show that the quorum model (1.9) with edge activation scheme (1.11) is also stable. However, rather than view the system as a single system described by (1.9), we treat it as a hybrid system with one mode described by (1.9) and the other described by (1.4). The system is in the *quorum mode* when $x_i > q_i$ for some i, *i.e.* some sites are above quorum, and in the *linear mode* when $x_i < q_i \ \forall i$, where all sites are below quorum. For simplicity we assume that $q_i = q$ for all i and $k_{ij}^{max} = k^{max}$ for all $(i,j) \in \mathcal{E}$. Next, consider the following function:

$$W_q = \sum_{i=1}^{M} \max\{x_i - q\bar{x}_i, 0\} \tag{1.19}$$

where W_q denotes the fraction of the population that is operating in the quorum mode. We note that agents who transition between above-quorum sites have no net effect on W_q, while the flux between sites above quorum and sites below quorum does produce an effect. Using non–smooth analysis, one can show that the time rate of change of W_q is always negative by design. This means that the number of agents operating in the quorum mode is always decreasing. Additionally, let W_l denote the scaled occupancy of the most populated site in the linear mode,

$$W_l = \max_i \{x_i/q\} \ . \tag{1.20}$$

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 the system first entered it. Furthermore, once the system enters the linear mode, it will not return to the quorum mode and thus the quorum model (1.9) with edge activation scheme (1.11) is stable.

1.5. Design of transition rate matrix K

As mentioned previously, while these models accomplish the multi-site deployment task, the solutions can be relatively inefficient since the rate of convergence of the system to the desired configuration depends on the magnitudes of the transition rates k_{ij}. 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 come at a significant cost. In light of this trade-off, we define an *optimal deployment strategy* as a choice of $\mathbf{K}$ that maximizes convergence toward the desired distribution while bounding the number of idle trips at equilibrium. In other words, an *optimal transition rate matrix* is one that can balance short term gains, *i.e.* fast convergence, against long term losses, *i.e.* inter-site traffic at equilibrium.

1.5.1. *Linear Model*

In general, determining an optimal transition rate matrix 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 rate matrix that is optimal for the entire domain of initial distributions or one that is optimal for the given initial distribution, $\mathbf{x^0}$. In this section, we describe our methodology for obtaining a general optimal transition rate matrix $\mathbf{K}_*$ that is suitable for a large range of initial distributions and an optimal transition rate matrix $\mathbf{K}_*(\mathbf{x^0})$ that is specifically tailored for a given initial configuration.

While obtaining $\mathbf{K}_*$ may seem computationally expensive, this matrix can in fact be calculated with limited assumptions using convex optimization. Since system (1.5) is linear, the rate of convergence of $\mathbf{x}$ to $\bar{\mathbf{x}}$ is governed by the real parts of the eigenvalues of $\mathbf{K}$, of which one is zero and

the rest are negative by Theorem 1.1. The smallest nonzero eigenvalue of $\mathbf{K}$, denoted by $\lambda_2(\mathbf{K})$, governs the asymptotic rate of convergence of (1.5) to $\bar{\mathbf{x}}$. Thus, when designing $\mathbf{K}$, one way to produce fast convergence to $\bar{\mathbf{x}}$ is to maximize $Re(\lambda_2(-\mathbf{K}))$ (note that this quantity is *positive*) subject to the following constraint on the total flux (traffic) at equilibrium,

$$\sum_{(i,j)\in\mathcal{E}} k_{ij}\bar{x}_i \leq c_{tot} \ , \tag{1.21}$$

where c_{tot} is a positive constant.

To achieve this, we assume that $\mathcal{G}$ has bidirectional edges and that there is a reversible Markov process on the graph. The site topology can be modeled by a corresponding undirected graph $\mathcal{G}_u = (\mathcal{V}, \mathcal{E}_u)$, as described in Section 1.3.1. The problem of maximizing $Re(\lambda_2(-\mathbf{K}))$ subject to (1.21) can then be posed as

$$\begin{aligned} \min_{\mathbf{w}} \quad & \sum_{(i,j)\in\mathcal{E}_u} 2w_{ij} \\ \text{subject to} \quad & \mathbf{\Pi}^{-1/2}\mathbf{L}\mathbf{\Pi}^{-1/2} \succeq \mathbf{I} - \mathbf{v}\mathbf{v}^T \\ & \mathbf{w} \geq 0 \end{aligned} \tag{1.22}$$

where w_{ij} is the set of undirected edge weights, $\mathbf{v} = [(\bar{x}_1)^{1/2} \ ... \ (\bar{x}_M)^{1/2}]^T$, and $\mathbf{\Pi} = diag(\bar{\mathbf{x}})$. $\mathbf{L}$ is the $M \times M$ weighted Laplacian of $\mathcal{G}_u$; its (i,j) entries are defined as $\mathbf{L}_{ij} = -w_{ij}$ if $(i,j) \in \mathcal{E}_u$, $\mathbf{L}_{ii} = \sum_{(i,j)\in\mathcal{E}_u} w_{ij}$, and $\mathbf{L}_{ij} = 0$ otherwise. Here, the optimization variable is $\mathbf{w}$ and the transition rates k_{ij} are derived from $\mathbf{w}$ according to (1.3). This program is similar to program (11) in[39] for finding the fastest mixing reversible Markov process on a graph. The optimization problem can be further extended to more general strongly connected graphs.[40]

To find the optimal $\mathbf{K}$ for a given initial configuration $\mathbf{x^0}$, $\mathbf{K}_*(\mathbf{x^0})$, Metropolis optimization[41] is used with the entries of $\mathbf{K}$ as the optimization variables. The objective is to minimize the convergence time subject to upper bounds on the number of idle trips at equilibrium according to (1.21) and possibly on the transition rates, $k_{ij} \leq k_{ij}^{max}$. We use the linear model (1.5) to calculate the convergence time to a set fraction of misplaced agents $\Delta(\mathbf{x}, \bar{\mathbf{x}})$, the total disparity between the actual and desired population fractions at all sites, in closed form. This is achieved by decomposing $\mathbf{K}$ into its normalized eigenvectors and eigenvalues and mapping (1.5) into the space spanned by its normalized eigenvectors. Then, given an initial state $\mathbf{x^0}$, we can apply the appropriate transformation to compute the new state $\mathbf{x}(t)$ using the matrix exponential of the diagonal matrix of eigenvalues of $\mathbf{K}$

multiplied by time. We use $\mathbf{x}(t)$ to calculate $\Delta(\mathbf{x}, \bar{\mathbf{x}})$. Since (1.5) is stable by Theorem 1.1, $\Delta(\mathbf{x}, \bar{\mathbf{x}})$ always decreases monotonically with time, so a Newton scheme can be used to calculate the exact time when $\Delta(\mathbf{x}, \bar{\mathbf{x}})$ is reduced to 10% of its initial value. Figure 1.2 plots the convergence time for a sample stochastic optimization of the transition rates.

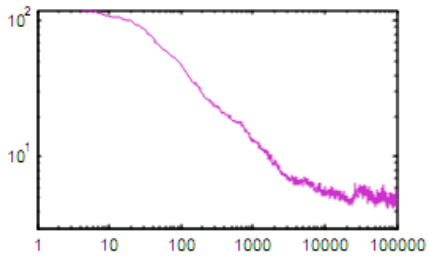


Fig. 1.2. Convergence time for stochastic optimization of the transition rates. The random walk in the space of all $\mathbf{K}$ configurations is biased with the convergence time so that lower times are eventually found. The horizontal axis is the number of the respective configuration.

1.5.2. *Time-Delayed Model*

The optimization methods described above can be extended to the time-delayed model (1.7) when it is represented as an equivalent ODE model as outlined in Section 1.4.2. We employ our multi–level swarm representation to empirically determine the travel time distributions for all pairs of sites. These distributions can be affected by road congestion, population levels at the target sites, and time spent on collision avoidance. Once a gamma distribution of the form (1.15) is fitted to the distribution for each edge, the resulting parameters D_{ij} and λ_{ij} give the number of dummy sites to be added and the transition rates between these dummy sites, respectively. This allows us to construct the chain model, which in turn allows for the optimization of the transition rates following the procedures outlined in the previous section.

1.6. Simulation Methodology

As in recent work on modeling and analyzing swarm robotic systems[20–22,30] we employ a multi-level representation of swarm activity. At the physical level, the swarm consists of a collection of individual robots whose dynamics are governed by their control laws. In our case, the control policies, *i.e.* $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x^0})$, are derived from the continuous linear model (1.5), 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 large enough to render most agent–based simulations 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.

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, which can either be behaviors or correspond to physically separate locations or sites. Assume that initially all agents are at site i and they all follow a stochastic transition rule by which they move to site j at 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 $k\Delta t = 1.0 \times 10^{-4}$. If the outcome is 1, the agent moves to site j; otherwise it stays at i. It is clear that, given a large population size, the number of agents remaining at site i after time t is well approximated by $n_i(t) = n_i(0)e^{-kt}$. Alternatively, instead of generating a random number each time, an agent could generate a random number T distributed according to the Poisson distribution

$$f(t) = \frac{1}{k} e^{-kt} \tag{1.23}$$

and perform its transition at time $t = T$. 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 (1.23) 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. We begin with a population of $n_i(0)$ 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 $n_i(0)$ agents to transition in the same time interval is $n_i(0)k\Delta t$. Thus, the distribution of the time of the first transition is similar to that for a single agent, only with probability rate (or *propensity*) $n_i(0)k$:

$$f(t, n_i(0)) = \frac{1}{n_i(0)k} e^{-n_i(0)kt} \quad . \tag{1.24}$$

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 (1.24). Of course, once the first transition takes place, the number $n_i(0)$ is decreased by one agent and the next transition is generated using propensity $(n_i(0) - 1)k$.

This is an illustration of the more general Gillespie algorithm,[42] in which 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 in which 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[31] for further discussion of this framework.

1.7. Results

We implemented simulations of multi-site deployment scenarios to compare the performance of different choices of $\mathbf{K}$ in terms of convergence to the desired distribution and the number of idle trips at equilibrium. We begin with the comparison of optimal and non–optimal $\mathbf{K}$ matrices and demonstrate the benefit of using quorum-activated transition rates.[38] Next, we show how a chain model, which takes into account the time delays due to navigation, performs better than the corresponding switching model.[37]

1.7.1. *Linear Model vs. Quorum Model*

We consider the deployment of $20,000$ planar homogeneous agents to 42 sites, each executing controllers derived from the model (1.5) with a non–optimal choice $\mathbf{K}_u$ for $\mathbf{K}$, an optimal matrix $\mathbf{K}_*$ that is independent of the initial configuration, and an optimal matrix $\mathbf{K}_*(\mathbf{x^0})$ that is specifically chosen for a particular initial configuration $\mathbf{x^0}$. 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 (1.11).

For our simulations, $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x^0})$ were obtained following the methodology described in Section 1.5.1. 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. The transitions between sites were simulated according to the methodology in Section 1.6. The interconnection topology of the sites is shown in Figure 1.3, in which each arrow represents a "one–way road" connecting two sites.

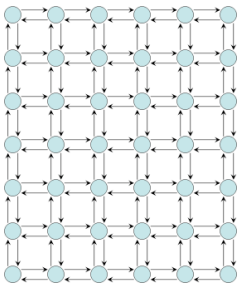


Fig. 1.3. The network of the 42 sites used in our simulations.

Agents are initially scattered at sites configured to form the number 0, and the task is to redistribute the swarm to another set of sites that form the number 8. While our focus is on the global design and properties of the swarm, our simulation methodology takes into account the exact number of agents assigned to each site as well as the travel initiation and termination times for each individual traveler. Snapshots of the simulation are shown in Figure 1.4, in which the red circles represent the number of the agents at each site. The larger the circle, the higher the agent population.

For the first set of simulations, the transition rate matrix $\mathbf{K}$ was set to $\mathbf{K}_u$, $\mathbf{K}_*$, and $\mathbf{K}_*(\mathbf{x^0})$. The agents switch between sites with controllers derived from (1.5), *i.e.* no quorum activation is used. In all of these simulations, the equilibrium initiation rate for a transition from site i to site j was set to 1. This is equivalent to bounding the total number of idle trips at equilibrium. Also, $k^{max} = 12$ in all simulations.

Figure 1.5 shows the fraction of misplaced agents over time for the three different choices of $\mathbf{K}$. It is not surprising that both $\mathbf{K}_*$ and $\mathbf{K}_*(\mathbf{x^0})$ outperform $\mathbf{K}_u$ in terms of convergence speed for the same bound on idle trips at equilibrium. Note that the stochastic runs fluctuate around the corresponding ODE simulations, which verifies that the transition rates designed using the continuous model produce similar system performance

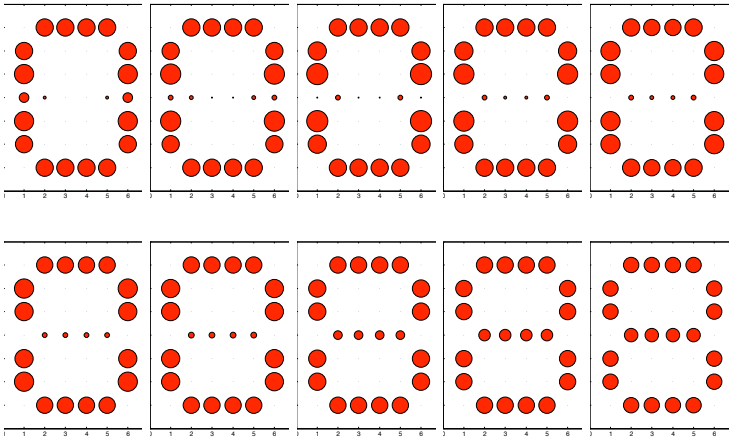
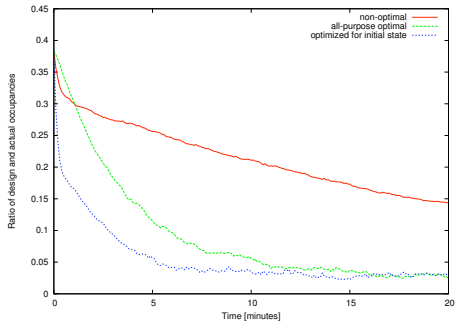


Fig. 1.4. Snapshots from a simulation of 20,000 agents, sequenced from top left to bottom right. The initial configuration forms the number 0, and the design specification is the number 8. This simulation is based on the linear model (1.5) with a non–optimal choice for $\mathbf{K}$.

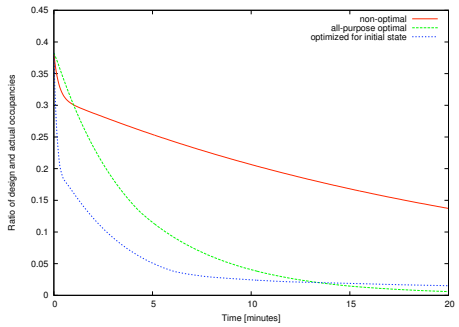
Choice of $\mathbf{K}$	Time units to 2/3 of initial deviation	Idle trip initiation rate at equilibrium	Maximum k_{ij}
$\mathbf{K}_u$	21.81	1	1.35
$\mathbf{K}_*$	4.51	1	11.88
$\mathbf{K}_*(\mathbf{x^0})$	1.84	1	6.94

when run on individual robots. The properties of each $\mathbf{K}$ are summarized in the first table.

Figure 1.6 shows the fraction of misplaced agents over time for the linear model (1.5) with $\mathbf{K}_u$ and two other choices of $\mathbf{K}$. The first, $\mathbf{K}_{max}(\mathbf{x^0})$, is the optimal transition rate 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 rate matrix with respect to convergence speed. The second choice of $\mathbf{K}$ is the quorum model (1.9) with below-quorum transition rates chosen from $\mathbf{K}_u$ and the edge activation scheme (1.11). 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. The second table summarizes the different properties of the three systems shown in



(a)



(b)

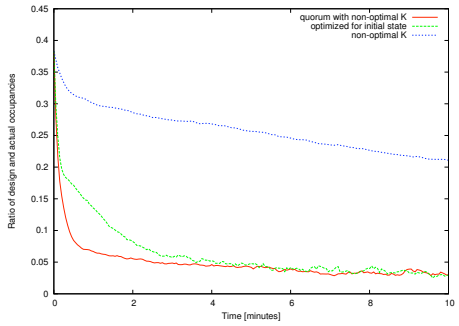
Fig. 1.5. Fraction of misplaced agents over time for system (1.5) with three choices of $\mathbf{K}$: $\mathbf{K}_u$ (red), $\mathbf{K}_*$ (green), and $\mathbf{K}_*(\mathbf{x^0})$ (blue). (a) Stochastic simulation. (b) Differential equation simulation.

Choice of $\mathbf{K}$	Time units to 2/3 of initial deviation	Idle trip initiation 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

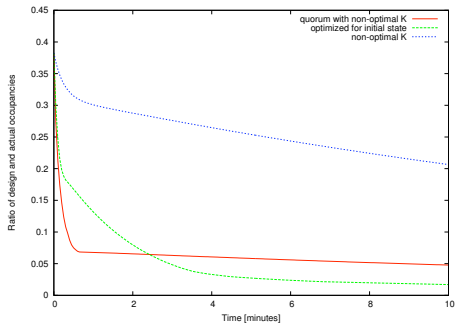
Figure 1.6.

1.7.2. *Linear Model vs. Time-Delayed Model*

To investigate the utility of the chain model in optimizing the transition rates, we simulated a surveillance task with transition rates from a linear



(a)



(b)

Fig. 1.6. Fraction of misplaced agents over time for system (1.5) with three choices of $\mathbf{K}$: the quorum model (1.9) with $\mathbf{K}_u$ and edge activation scheme (1.11) (red), $\mathbf{K}_{max}(\mathbf{x^0})$ (green), and $\mathbf{K}_u$ (blue). (a) Stochastic simulation. (b) Differential equation simulation.

model (1.5) (a "switching model"; see Section 1.4.2) and a chain model computed from travel time distributions. The workspace was chosen to be a four-block region on the University of Pennsylvania campus, shown in Figure 1.7. The task was for a collection of 200 robots to perform perimeter surveillance of four campus buildings, highlighted in light dashed lines in Figure 1.7. We used a graph $\mathcal{G}$ for these four sites with the structure in Figure 1.8. The robots are initially distributed equally between sites 3 and 4, and they are required to redistribute in equal fractions among all sites. The transitions between sites are simulated according to the methodology in Section 1.6.

Agents exhibit two types of motion: perimeter tracking and site-to-site navigation. An agent that is monitoring a building circulates around its perimeter, slowing down if the agent in front of it enters its sensing radius.

This results in an approximately uniform distribution of agents around the perimeter. This motion can be easily achieved with feedback controllers of the form given in;[43] in this simulation the agents simply aligned their velocities with the straight-line perimeters.

To implement inter-site navigation, we first decomposed the free space into a tessellation of convex cells, shown in Figure 1.7. This results in a discrete roadmap on which shortest-path computations between cells can be easily obtained using any standard graph search algorithm. Each edge $(i, j) \in \mathcal{E}$ was defined as a path from building i to building j that begins at a distinct exit point on the perimeter of i and ends at an entry point on j. The exit and entry point for (i, j) were associated with the adjacent cells, and Dijkstra's algorithm was used to determine the shortest path between these cells, which consisted of a sequence of cells to be traversed by each agent moving from i to j. Navigation between cells was achieved by composing local potential functions such that the resulting control policy ensured arrival at the desired goal position.[44] These navigation controllers were then composed with repulsive potential functions to achieve inter-agent collision avoidance within each cell. Inter-site navigation was achieved by providing each agent *a priori* with the sequence of cells corresponding to each edge (i, j), and at each time step the agent computed the feedback controller to move from one cell to the next based on its current position and the set of agents within its sensing range.

We first used the Metropolis optimization of Section 1.5.1 to obtain an optimal switching model $\mathbf{K}_*^{sw}(\mathbf{x^0})$ (ignoring travel times). From the simulation using this $\mathbf{K}$, we collected $750 - 850$ travel times τ_{ij} per edge. We then fit a gamma distribution to the resulting histograms to obtain D_{ij} and λ_{ij} for each edge (Section 1.4.2); a sample fitting is shown in Figure 1.9. The average equilibrium traveler fraction from ~ 30000 data points was approximately 0.27. This fraction was used as a constraint in the optimization of the chain model $\mathbf{K}_*^{ch}(\mathbf{x^0})$ according to Section 1.5.2. The simulation was then run with the transition rates from $\mathbf{K}_*^{ch}(\mathbf{x^0})$.

The snapshots in Figure 1.10 illustrate the redistribution of the agents in equal fractions among the four sites for the chain model. The dark red agents have committed to traveling between two buildings; the light red agents are not engaged in a transition. Figure 1.11 shows that the traveler fraction for both models oscillates close to the mean switching model fraction, 0.27. Both models therefore have approximately the same equilibrium inter-site traffic. Figure 1.12 shows that the misplaced agent fraction of the chain model converges to 10% of its original value faster

than the switching model. This provides evidence that the chain model is a better approximation of the simulated surveillance system.

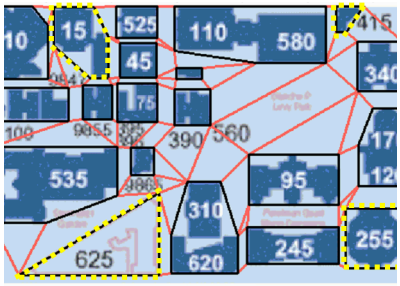


Fig. 1.7. Workspace with cell decomposition of the free space used for navigation.

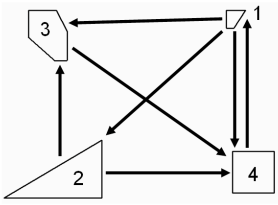


Fig. 1.8. Numbering and connectivity of surveyed buildings, which are highlighted in Figure 1.7.

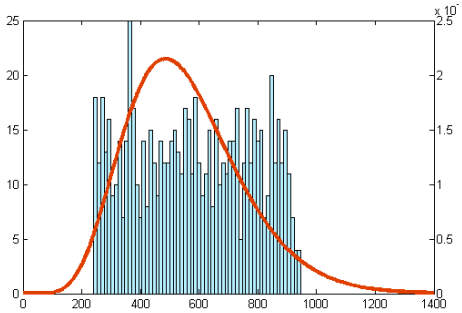


Fig. 1.9. Histogram of the travel times from site 1 to site 4 (758 data points) and the approximate gamma distribution. Based on this data, $D_{14} = 9$ and $\lambda_{14} = 0.0144$.

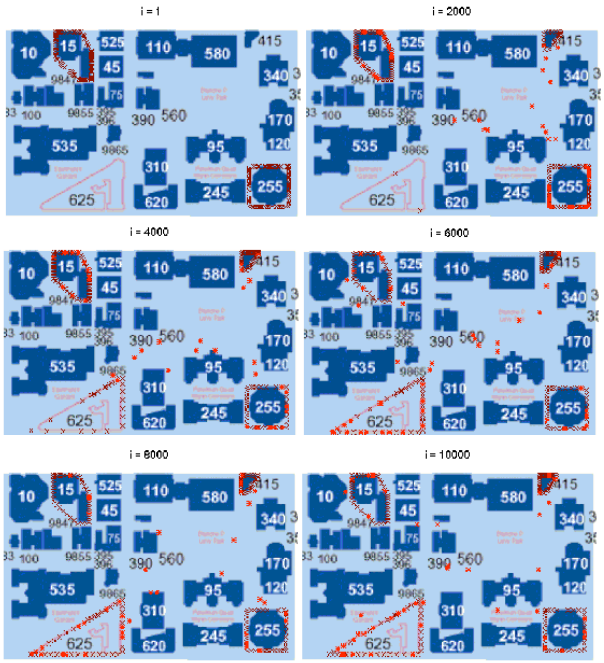


Fig. 1.10. Snapshots of the full chain model simulation at intervals of 2000 time steps.

1.8. Discussion

In biology, complex group behaviors often arise from the numerous interactions of individual agents that switch between a small set of "standard" behaviors. This is clearly seen in the house-hunting process of the *Temnothorax albipennis* ants. Additionally, it has been shown how the differential equations describing the global evolution of the swarm stem from simple, stochastic switching rules executed by the individuals.[31,32] These rules are equivalent to the transition rates k_{ij} encoded in $\mathbf{K}$. In this work, we built on the deployment scheme presented in[33] and independently considered the effects of a quorum–based strategy and the effects of travel time delays on the (re)distribution problem.

Our baseline model is a set of states that can be interpreted as physical sites or internal states/behaviors. As we demonstrated, this model forces a trade–off between fast convergence to the desired distribution and number of idle trips at equilibrium. Fortunately, under certain circumstances, the

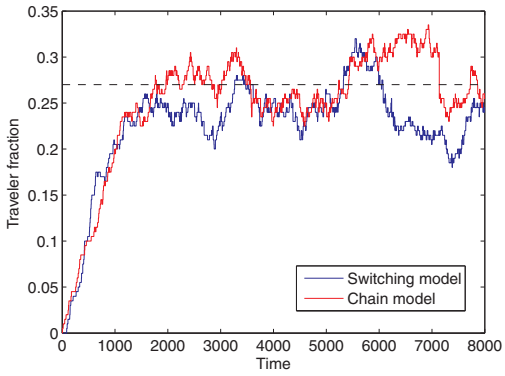


Fig. 1.11. Fraction of travelers vs. time for the switching and chain models.

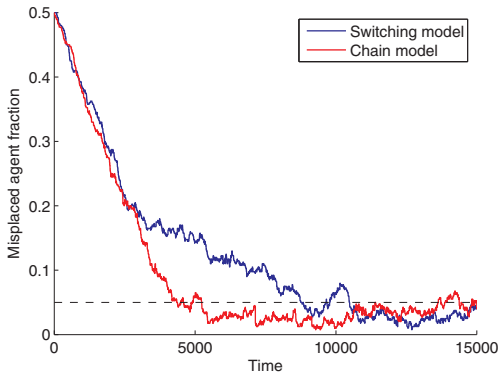


Fig. 1.12. Fraction of misplaced agents vs. time for the switching and chain models.

optimization of the transition rates can be posed as a convex optimization problem. Given an initial configuration of the swarm, Metropolis optimization can be used to obtain the optimal transition rates for the desired final configuration. Hence, the multi-site linear model can be analyzed with some very powerful tools: a well developed theoretical framework,[39] closed-form solutions of the corresponding ODEs, and an efficient macro-discrete simulation method that easily scales up to tens of thousands of agents.

In the quorum model, we endowed agents with the ability to choose between two sets of transition rates depending on whether the current site is above or below a certain threshold occupancy. As discussed earlier, the

exact motivations for these quorum strategies in ants are not clearly understood; however, 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 in our simulations, which indicate that the quorum strategy, used with a non–optimal choice for $\mathbf{K}$, consistently outperforms the linear 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. The quorum model is an example of a biologically inspired heuristic that provides a large benefit under limited sensing capability. We used our simulation framework to quantify the gain in efficiency, allowing for a future cost-benefit analysis when the cost of the necessary sensing can be properly defined.

Lastly, in a step toward more real–world applications, the final contribution is a method that can account for realistic distributions of travel times without leaving the linear ODE framework. We achieve this by augmenting the network of sites with specifically constructed sets of virtual sites that represent the progress of agents along the paths connecting the physical sites. This approach relies on two important points: in practice, travel times are highly variable due to factors such as collision avoidance and errors in localization; and the travel time over a chain of D_{ij} sites which are connected through Poisson transition rules converges to a gamma distribution whose relative standard deviation is proportional to $D_{ij}^{-1/2}$. Thus, the mean and standard deviation of an experimentally derived distribution of travel times can be matched by a properly chosen linear chain of sites. Our results illustrate the possibility of building larger linear ODE models that provide good approximations of variable agent travel times. The predictive value of such models is of course dependent on how well the distribution of the travel times is characterized. As our results show, the additional insight allows for better design of stochastic transition rate systems.

1.9. Conclusions

We have presented a bio–inspired approach to the (re)distribution of a swarm of robots among a set of available sites. Our methodology is built on top of our baseline strategy, which models the swarm as a continuum via a system of deterministic linear ordinary differential equations. We extended our linear model to a hybrid system in which agents switch between maximum transfer rates and constant transition rates dictated by the model.

We also considered the effects of travel time delays and proposed a methodology to synthesize optimal transition rates given gamma-distributed travel times. Although we model our system as a continuum, our methodology enables us to synthesize decentralized controllers that can be implemented at the agent level with little to no explicit wireless communication.

In physical systems, inaccuracies in navigation are common due to noisy sensors and actuators. However, these inaccuracies can be easily captured via our delay differential equation model. As discussed in Section 1.4.2, this model is in fact an abstraction of a more realistic model in which the delays are represented as random variables that follow a distribution which allows the time–delayed model to be converted into an equivalent linear model with no delays. This equivalence can be exploited as long as the inaccuracies introduced by the physical system can be mapped to stochastic time delays. Thus, the effects of crowding, localization errors, collision avoidance, and quorum estimation can be readily incorporated into our framework if we are able to model the distribution of the resulting delays. This is a topic of great interest for future work.

While stochastic controllers seem appealing because they involve significantly less communication, sensing, and planning compared to deterministic approaches, the purposeful introduction of stochasticity into an engineered system raises a number of questions. One issue is the development of simulation tools for the assessment of the performance of a mesoscopic swarm. In this work, we developed a mathematical framework, using methods borrowed from chemistry, to bridge the gap between the agent-based description (necessary for the implementation and testing of individual controllers) and the top-level description which is traditionally continuous. Our long term goal is to investigate the utility and limitations of this approach when used either by itself or in conjunction with more traditional models of swarm behavior synthesis.

Acknowledgements

We gratefully acknowledge the comments and support from Professor Vijay Kumar from the University of Pennsylvania and the support of NSF grants CCR02-05336 and IIS-0427313, and ARO Grants W911NF-05-1-0219 and W911NF-04-1-0148.

References

1. G. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control*. (MIT Press, 2005).
2. N. F. Britton, N. R. Franks, S. C. Pratt, and T. D. Seeley, Deciding on a new home: how do honeybees agree?, *Proc R Soc Lond B.* **269**, (2002).
3. S. Pratt, E. B. Mallon, D. J. T. Sumpter, and N. R. Franks, Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant Leptothorax albipennis, *Behavioral Ecology and Sociobiology.* **52**, 117–127, (2002).
4. I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, Collective memory and spatial sorting in animal groups, *Journal of Theoretical Biology.* **218**, (2002).
5. J. K. Parrish, S. V. Viscido, and D. Grunbaum, Self-organized fish schools: An examination of emergent properties, *Biol. Bull.* **202** (June, 2002).
6. C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer Graphics (SIGGRAPH'87)*, pp. 25–34. ACM Press, (1987).
7. T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, Novel type of phase transition in a system of self-driven particles, *Physical Review Letters.* **75**(6), 1226–9, (1995).
8. A. Jadbabaie, J. Lin, and A. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Transactions on Automatic Control* (July. 2003).
9. H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. In *Transactions on Automatic Control*, (2007).
10. L. Lin and Z. Zheng. Combinatorial bids based multi-robot task allocation method. In *Proc. of the 2005 IEEE Int. Conference on Robotics and Automation (ICRA'05)*, (2005).
11. J. Guerrero and G. Oliver, Multi-robot task allocation strategies using auction-like mechanisms, *Artificial Research and Development in Frontiers in Artificial Intelligence and Applications.* (2003).
12. E. G. Jones, B. Browning, M. B. Dias, B. Argall, M. Veloso, and A. T. Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *International Conference on Robotics and Automation*, pp. 570 – 575 (May, 2006).
13. M. B. Dias, R. M. Zlot, N. Kalra, and A. T. Stentz, Market-based multirobot coordination: a survey and analysis, *Proceedings of the IEEE.* **94**(7), 1257 – 1270 (July, 2006).
14. D. Vail and M. Veloso. Multi-robot dynamic role assignment and coordination through shared potential fields. In eds. A. Schultz, L. Parker, and F. Schneider, *Multi-Robot Systems.* Kluwer, (2003).
15. B. P. Gerkey and M. J. Mataric, Sold!: Auction methods for multi-robot control, *IEEE Trans. on Robotics & Automation: Special Issue on Multirobot Systems.* **18**(5), 758–768 (Oct, 2002).
16. E. G. Jones, M. B. Dias, and A. Stentz. Learning-enhanced market-based

task allocation for oversubscribed domains. In *Proceedings of the Conference on Intelligent Robot Systems (IROS'07)*, (2007).
17. M. B. Dias. *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. PhD thesis, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA (Jan, 2004).
18. M. Golfarelli, D. Maio, and S. Rizzi. Multi-agent path planning based on task-swap negotiation. In *Proc. 16th UK Planning and Scheduling SIG Workshop*, (1997).
19. N. Franks, S. C. Pratt, N. F. Britton, E. B. Mallon, and D. T. Sumpter, Information flow, opinion-polling and collective intelligence in house-hunting social insects, *Phil. Trans.: Biological Sciences*. (2002).
20. A. Martinoli, K. Easton, and W. Agassounon, Modeling of swarm robotic systems: a case study in collaborative distributed manipulation, *Int. J. of Robotics Research: Special Issue on Experimental Robotics*. **23**(4-5), 415–436, (2004).
21. W. Agassounon, A. Martinoli, and K. Easton, Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes, *Auton. Robots*. **17**(2-3), 163–192, (2004).
22. K. Lerman, C. V. Jones, A. Galstyan, and M. J. Mataric, Analysis of dynamic task allocation in multi-robot systems, *Int. J. of Robotics Research*. **25**(4), 225–242, (2006).
23. M. A. Hsieh, L. Chaimowicz, and V. Kumar, Decentralized controllers for shape generation with robotic swarms, *Accepted to Robotica*.
24. S. W. Pacala, D. M. Gordon, and H. C. J. Godfray, Effects of social group size on information transfer and task allocation, *Evolutionary Ecology*. **10** (2), 127–165 (March, 1996).
25. M. J. B. Krieger, J.-B. Billeter, and L. Keller, Ant–like task allocation and recruitment in cooperative robots, *Nature*. **406**, 992–995 (August, 2006).
26. T. H. Labella, M. Dorigo, and J.-L. Deneubourg, Division of labor in a group of robots inspired by ants' foraging behavior, *ACM Trans. Auton. Adapt. Syst.* **1**(1), 4–25, (2006). ISSN 1556-4665. doi: http://doi.acm.org/10.1145/1152934.1152936.
27. N. Kalra and A. Martinoli. A comparative study of market-based and threshold-based task allocation. In *Proceedings of the The 8th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Minneapolis/St. Paul, Minnesota, USA (July, 2006).
28. W. Agassounon and A. Martinoli. Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems. In *Proc. of the First Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems AAMAS-02*, pp. 1090–1097, (2002).
29. E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg. Adaptive task allocation inspired by a model of division of labor in social insects. In eds. D. Lundh, B. Olsson, and A. Narayanan, *Biocomputing and Emergent Computation*, pp. 36–45. World Scientific, (1997).
30. K. Lerman, A. Martinoli, and A. Galstyan. A review of probabilistic macroscopic models for swarm robotic systems. In *Swarm Robotics*, pp. 143–152,

(2004).
31. S. Berman, Á. Halász, V. Kumar, and S. Pratt. Algorithms for the analysis and synthesis of a bio-inspired swarm robotic system. In eds. E. Sahin, W. M. Spears, and A. F. T. Winfield, *Swarm Robotics*, vol. 4433, *LNCS*, pp. 56–70. Springer, (2006).
32. S. Berman, Á. Halász, V. Kumar, and S. Pratt. Bio-inspired group behaviors for the deployment of a swarm of robots to multiple destinations. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, pp. 2318–2323, (2007).
33. Á. Halász, M. A. Hsieh, S. Berman, and V. Kumar. Dynamic redistribution of a swarm of robots among multiple sites. In *Proc. of the Conf. on Intelligent Robot Systems (IROS)*, pp. 2320–2325, (2007).
34. S. C. Pratt, Quorum sensing by encounter rates in the ant Temnothorax albipennis, *Behavioral Ecology*. **16**(2), (2005).
35. B. Harris, *Theory of Probability*. (Addison-Wesley, Reading, MA, 1966).
36. N. MacDonald, *Time-lags in biological models*. vol. 27, *Lecture Notes in Biomathematics*, (Springer, Berlin, 1978).
37. S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar. Navigation-based optimization of stochastic deployment strategies for a robot swarm to multiple sites. Submitted to the 2008 Conf. on Decision and Control, Cancun, Mexico.
38. M. A. Hsieh, Á. Halász, S. Berman, and V. Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. Submitted to a Special Issue of the Swarm Intelligence Journal, 2008.
39. J. Sun, S. Boyd, L. Xiao, and P. Diaconis, The fastest mixing Markov process on a graph and a connection to the maximum variance unfolding problem, *SIAM Review*. **48**(4), (2006).
40. S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar. Optimization of stochastic strategies for redistributing a robot swarm among multiple sites. To be submitted to IEEE Transactions on Robotics.
41. D. P. Landau and K. Binder, *A guide to Monte-Carlo simulations in statistical physics*. (Cambridge University Press, 2000).
42. D. Gillespie, Stochastic simulation of chemical kinetics, *Annu. Rev. Phys. Chem.* **58**, 35–55, (2007).
43. M. A. Hsieh, S. Loizou, and V. Kumar. Stabilization of multiple robots on stable orbits via local sensing. In *Proc. of the Int'l Conf. on Robotics and Automation (ICRA) 2007*, Rome, Italy (April, 2007).
44. D. C. Conner, A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In *Proc. of 2003 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS 2003)*, vol. 4, pp. 3546–3551. IEEE (October, 2003).

Index