A Hybrid Framework for Resource Allocation among Multiple Agents Moving on Discrete Environments

Jorge L. Piovesan, Chaouki T. Abdallah, and Herbert G. Tanner

Abstract

We consider in this paper the problem of controlling a multi-agent system whose agents move across discrete locations. The agents attempt to extract resources from the environment in order to maximize their completion of tasks, while the environment, which may vary as the system evolves, distributes its resources according to the agents requests. The environment is modeled as a network with discrete nodes. Our ultimate goal is to design the dynamical policies that rule the behavior of agents and nodes such that the usage of resources in the network is optimized. In order to solve this problem, we propose a hybrid model to describe both agents and nodes. Several components of this model are design variables that may be obtained analytically. We then formulate an equivalent optimization problem that may be decomposed into two hierarchical optimization problems: An integer optimization problem that considers the distribution of agents among the nodes of the network, and a convex optimization problem within each node that corresponds to the distribution of resources of each node among its resident agents. We show that the optimization problem within each node is a special case of the formulation that models congestion control algorithms in the Internet. We then use the available results to solve the continuous portion of the proposed hybrid description for agents and nodes. Moreover, we show that the resulting continuous dynamics are globally asymptotically stable, with their equilibrium point coinciding with the solution of the optimization problem. As a consequence, the proposed continuous dynamics yield an interconnected system that is stable on each possible configuration of agents and nodes, i.e., on each possible combination of discrete states in their hybrid model. This completes the design of the continuous dynamics of the proposed model, while the design of the discrete dynamics is relegated to future work.

Index Terms

Multi-agent system, discrete environment, resource allocation, hybrid dynamical model.

Jorge Piovesan (corresponding author), and Chaouki Abdallah are with the Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque NM 87131 {jlpiovesan,chaouki}@ece.unm.edu, phone: 1-505-2770298, fax: 1-505-2771439. Herbert Tanner is with the Department of Mechanical Engineering, The University of New Mexico, Albuquerque NM 87131 tanner@unm.edu, phone: 1-505-2771493.

Jorge Piovesan and Chaouki Abdallah are supported by NSF award CNS 0626380 under the FIND initiative.

Hybrid Framework for Resource Allocation among Multiple Agents Moving on Discrete Environments

I. Introduction

Advances in computation and communication technologies provide interesting possibilities for substituting complex and expensive single devices, with more cost-effective distributed systems of multiple simple devices. The advantage of these distributed systems, better known as multi-agent systems, is that they can perform complicated tasks by generating a group behavior through the coordination of the agent's actions, usually using only local policies and information that is available through limited communication and sensing. This results in greater efficiency and robustness of the system, compared to that of a single device. However, the control of multiple entities with common goals raises new challenges which include but are not limited to the design of local policies that enable a stable (and maybe optimal) group behavior, the reliable information sharing under communication constraints, and consensus among agents with potentially different measurements.

There exists extensive literature on multi-agent systems varying by application and objective. One major thrust of the research on cooperative control is that of safe group navigation. Different approaches have been proposed to address this problem. They include, but are not limited to formation control [10], [22], [24], [28], [40], [43], flocking and swarming algorithms [12], [15], [20], [38], [39], and platooning [18], [37]. Another frequently discussed problem is that of positioning agents on a given environment. Results in this direction include facility location via distributed optimization [9], perimeter tracking [6], formations using implicit functions [5], and coordination using Internet-like protocols [34]. Researchers have also studied general consensus problems [23], [26], [27], [32], [33], sensor network applications [13], [25], and distributed task assignment using load balancing schemes [11].

The problem we address in this paper is the following: We consider a set of heterogeneous agents whose goal is to optimize a utility function via the utilization of resources available in the environment. The environment is composed of discrete locations connected by paths used by the agents to locate resources at such locations. Different locations may have different types and amounts of available resources, and each location (a node) allocates its resources according to requests from its resident agents. The agents request resources according to their particular tasks, which are encoded on their utility functions. The resources are allowed to vary in discrete form and according to environment related events. The agents are therefore capable of two types of decisions: Requesting more (or less) of a resource from a location they already occupy, and moving from one location to another in order to obtain better resources. The ultimate goal of the cooperative system is to optimize the aggregate of the agent's utility functions using only local policies i.e., control decisions at the agent level, such that the usage of the environment resources is globally optimized by the multi-agent system.

The original motivation for the problem described above is related to the design of future communication networks [29]. The communications related problem is due to the need of a smarter Internet [7] in order to deal with challenges that communication networks are facing due to their size explosion. An architecture that addresses this problem abstracts the functions of the network from the physical network [16]. This is done using software agents that implement the different functions of the network (routing, DNS resolution, storage, etc.) and viewing the hardware nodes as resource providers to be used by the agents for the completion of their tasks. The agents are then allowed to move autonomously among the nodes of the network in search of nodes that increase the efficiency and effectiveness of their task completion. A more detailed discussion of the optimization problem related to the communication network design is found in [29].

The problem we address in this paper has an important difference from prior literature on multi-agent control systems [5], [6], [9]–[13], [15], [18], [20], [22]–[28], [32]–[34], [37]–[40], [43]. We consider agents moving among discrete locations while the cited results consider agents moving in a continuous space or with continuous dynamics. We therefore model the environment as a graph where the nodes represent the discrete locations and the edges represent the paths that the agents use to obtain information about other nodes and to move between locations. Moreover, in order to capture the complete behavior of this interconnected system, we model the agents and nodes as hybrid systems. The general hybrid model we use allows us to capture both the continuous evolution of the resource allocation tasks (node and agent dynamics), and the discrete events related to the changes on resource availability in the locations (node dynamics) and the movement of agents among the locations in the environment (agent dynamics). Since the final goal is to optimize the usage of the network resources, we formulate an optimization problem, which turns out to be a mixed integer nonlinear optimization problem. We then obtain an equivalent hierarchical optimization problem composed of a (global) network integer optimization problem at the higher level of the hierarchy, and several decentralized (one for each node in the network) convex optimization problems. We then observe that the convex optimization problem within each node is a special case of the optimization problem used to model various Internet congestion control algorithms [17], [19], [36]. This allows us to perform several simplifications in the design of the agents and nodes dynamics. First, the continuous dynamics are designed separately from the discrete dynamics. Moreover, the continuous dynamics for both the agents and nodes are designed based on the dynamic model for the Internet congestion control algorithms [1], [19], [36], [41]. This implies that there is a globally asymptotically stable (continuous) equilibrium point for each possible (discrete) agent distribution in the network, and that these equilibria coincide with the solution of the optimization problem for that particular distribution. It is important to note that the results in this paper provide a complete design for the continuous dynamics of agents and nodes, but leave open the design of the discrete transition rules. In Section VII we discuss our ongoing and future research work related to this issue.

The optimization problem formulated in this paper may be solved using model-based techniques, such as combinatorial optimization [8], or mixed integer programming [14] approaches. These techniques, however, present a major problem in terms of computational complexity, since they are NP-complete in the number of integer variables ([35] Chapter 18) which in our case may become very large. Instead we propose a hierarchical approach exploiting the structure of the problem, which allows us to decentralize the continuous decision variables, leaving only the discrete ones for centralized optimization. Moreover in [29] we proposed the use of randomized algorithms for the centralized optimization in order to avoid the computational complexity issues of model-based techniques. While this hierarchical solution is similar in spirit to the dual decomposition approach proposed in [42], the discrete component of our problem makes it different from that handled by dual decomposition [42]. In fact, the dual decomposition approach which only considers continuous decision variables.

The rest of the paper is organized as follows: Section II presents our working assumptions, the general hybrid model for agents and nodes and the multi-agent system objective design. Section III outlines the hierarchical optimization approach that solves the problem stated, while Section IV presents the solution for the convex optimization problem within each node. Section V presents the design of the continuous dynamics of agents and nodes, Section VI shows a simulation example of the designed portion of the model, and Section VII summarizes our results.

II. Hybrid Model and Design Objective

A. Preliminary concepts

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a *graph* with *nodes* indexed by $\mathcal{V} = \{1, 2, ..., N_v\}$ and *edges* $\mathcal{E} = \{(v, w) : v, w \in \mathcal{V}, v \neq w, \text{ and } v \text{ connected to } w\}$. We call the graph *undirected* if $(v, w) \in \mathcal{E}$ whenever $(w, v) \in \mathcal{E}$. A graph is *connected* if there is a path between any pair of nodes in the graph, where a *path* from v to w is a sequence of different nodes starting at v and ending at w such that consecutive nodes are connected. We call neighborhood of v to the set $\mathcal{N}_v = \{w \in \mathcal{V} : (v, w) \in \mathcal{E}\}$.

Let a *Controlled Hybrid Dynamical System (CHDS)* [4] be a tuple $\mathbf{H} = [Q, \mathbf{\Sigma}, \mathbf{G}^A, \mathbf{Z}^A, \mathbf{S}, \mathbf{G}^C, \mathbf{Z}^C]$ where:

- Q is the set of discrete states.
- $\mathbf{\Sigma} = \{\Sigma_q\}_{q\in Q}$ is the collection of dynamical systems with $\Sigma_q = (X_q, \mathbb{R}^+, f_q, U_q)$, where X_q is the continuous state space, $\mathbb{R}^+$ is the time set, f_q is the continuous dynamics, and U_q is the set of continuous controls .
- $\mathbf{S} = \{S_q^A\}_{q\in Q} \bigcup \{S_q^C\}_{q\in Q}$ is the set of discrete transition labels. $s^A \in S_q^A$ determines the next discrete mode for an autonomous transition, and $s^C \in S_q^C$ determines the next mode for a controlled transition.
- $\mathbf{G}^A = \{G_q^A\}_{q\in Q}$, where $G_q^A : S_q^A \to X_q$, is a guard condition for an autonomous jump for each $q \in Q$.
- $\mathbf{G}^C = \{G_q^C\}_{q\in Q}$, where $G_q^C : S_q^C \to X_q$, is a guard condition for a controlled jump for each $q \in Q$.
- $\mathbf{Z}^A = \{Z_q^A\}_{q\in Q}$ where $Z_q^A : G_q^A \times S_q^A \to \{X_p\}_{p\in Q}$ is the autonomous transition map.
- $\mathbf{Z}^C = \{Z_q^C\}_{q\in Q}$, where $Z_q^C : G_q^C \times S_q^C \to \{X_p\}_{p\in Q}$ is the controlled transition map.

Finally, $H = \left(\bigcup_{q\in Q} X_q\right) \times Q$ is the hybrid state space of $\mathbf{H}$. Note that $\mathbf{S}$ may include the no transition element $\{id\}$.

The difference between autonomous transitions and controlled transitions is that the autonomous transitions S^A take place as soon as the continuous state reaches an autonomous transition guard G^A i.e., the transitions are forced, while controlled transitions only take place if a transition event S^C occurs while the continuous state is in a controlled transition guard G^C. This means that even if the continuous state is in a controlled transition guard the transition will not take place until an event enabling the transition occurs.

B. System's description

Consider the following scenario. A set of agents is moving on an environment composed of discrete locations. Each location (node) has different types and amounts of resources that may be allocated to the agents, while the agents use such resources for the completion of different tasks. The agents are greedy entities competing for the resources in the network, which means that each agent attempts to maximize its usage of resources considering only its own benefit. Agents are capable of requesting resources from the node that hosts them, as well as migrating to different nodes in the network seeking resources to complete their task. Task satisfaction is measured using a utility function that provides a real value as a function of the resources that the agent uses.

Each node distributes the resources among the agents it hosts according to the requests of these agents. The nodes may, however, suffer changes in their resource amounts. The paths (edges) that connect different locations are used by the agents to move between nodes and/or to obtain information about resources in nearby locations, may change over time. The final objective is to design the nodes' and agents' dynamics such that the usage of resources in the environment (network) is optimized with respect to the requirements of the agents. A pictorial representation of the problem is shown in Figure 1. We impose the following assumptions:

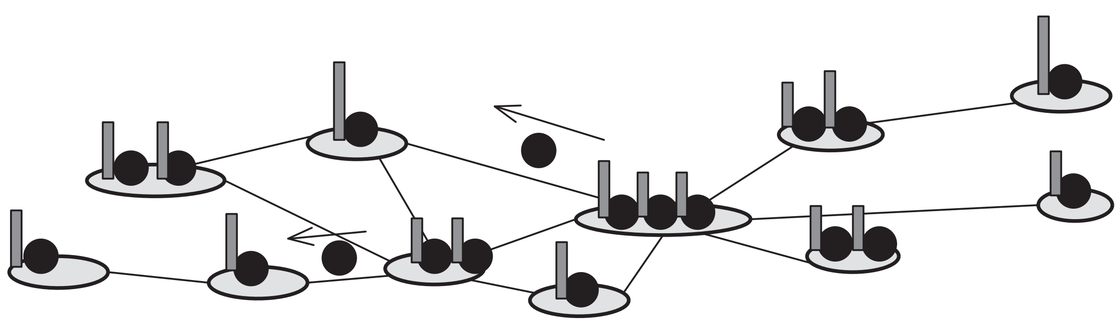

Fig. 1. Multi-agent system example: Each location in the network distributes its local resources among its residing agents. The locations are abstracted as nodes in a graph (gray ovals), the paths available for movement of agents and communication of states between different nodes are represented by edges in the graph. The agents are represented with black circles and the resources they use by gray bars. Agents move between nodes (identified with arrows on top) expecting better resources at a destination node

Assumption 1 (Network) *The network is an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the fixed total number of nodes is N_v.*

Assumption 2 (Resources) *There exist N_r types of resources in the network, where N_r is fixed. For each node $i \in \mathcal{V}$ the set of available amount of resources to be allocated among the agents located on node i is denoted by*

$R_i = \{r_{i,1}, r_{i,2}, ...r_{i,N_r}\}$, where $r_{i,j} \in \mathbb{R}$ is the amount of resource of type j available at node i. We assume that for all $i \in \mathcal{V}$ and all $j \in \mathcal{R} = \{1, 2, ..., N_r\}$, $r_{i,j}$ may vary on time, taking on values from a finite set $\Xi \subset \mathbb{R}$ according to the dynamics specified in Assumption 4.

Assumption 3 (Agents) *There is a fixed number of agents N_a, indexed by the set $\mathcal{A} = \{1, 2, ..., N_a\}$. The state of each agent $k \in \mathcal{A}$ consists of an ordered tuple $(x_{k,1}, x_{k,2}, ..., x_{k,N_r}, q_k)$, where $x_{k,j} \in \mathbb{R}$ represents the amount of resource of type j allocated to agent k, and $q_k \in \mathcal{V}$ denotes the location of agent k in the graph. Note that $0 \leq x_{k,j} < \infty$ for all $k \in \mathcal{A}$ and all $j \in \mathcal{R}$.*

If Assumption 1 is relaxed, the number of nodes may change over time and it may be possible to have asymmetric communication and agent movement capabilities between nodes. If Assumption 3 is relaxed, we may allow variations in the number of agents over time. Note however that Assumption 2 is strongly related to Assumption 1 because if $0 \in \Xi$, then $R_i = 0$ emulates the disappearance of node i from the network. The assumption that R_i varies over time may then be used to represent the appearance or disappearance of nodes in the network.

We believe that the existence of a fixed number of (non-negative but finite) resources' types in Assumption 2, and the description of the agents states in Assumption 3 are reasonable. Any practical problem that may be modeled under this framework could potentially generate a large set of resources' types to be allocated, but this set is still finite. Agent satisfaction depends upon their location in the network and the resources allocated to them, so the relevant information for the agents is contained in the state description introduced in Assumption 3.

We now extend the definition of neighborhoods from graph theory to consider agents and nodes neighborhoods in the network as follows:

Definition 1 (Neighborhoods) *Let the neighborhood of a node $i \in \mathcal{V}$ be $\mathcal{N}_i = \{w \in \mathcal{V} : (i, w) \in \mathcal{E}\} \bigcup \{k \in \mathcal{A} : q_k = i\}$ i.e., the neighborhood of a node is composed by the nodes that are connected to it by an edge in the network and the agents that are located inside the node (shown in Figure 2-top). Let the neighborhood of an agent $k \in \mathcal{A}$ be $\mathcal{N}_k = \{a \in \mathcal{A} : q_a = q_k\} \bigcup \{w \in \mathcal{V} : q_k = w \text{ or } (q_k, w) \in \mathcal{E}\}$, i.e., the neighborhood of an agents is composed by the agents that are located in the same node it is located, and the nodes that are connected through paths of length one to the node it occupies (shown in Figure 2-bottom).*

Note that the neighborhoods of agents $\mathcal{N}_k$ and nodes $\mathcal{N}_i$ are distinguished by the subindex i for the neighborhood of a node and k for the neighborhood of an agent. For the remainder of this paper, we will use this type of notation to distinguish nodes from agents where similar concepts are defined for both. Therefore a subindex k is used to indicate an agent description while a subindex i is used to indicate an node description. Similarly a subindex j is used to indicate a resource description.

Assumption 4 (Node dynamics) *Each node $i \in \mathcal{V}$ may be described as a Controlled Hybrid Dynamical System $\mathbf{H}_i$ that satisfies the following conditions:*

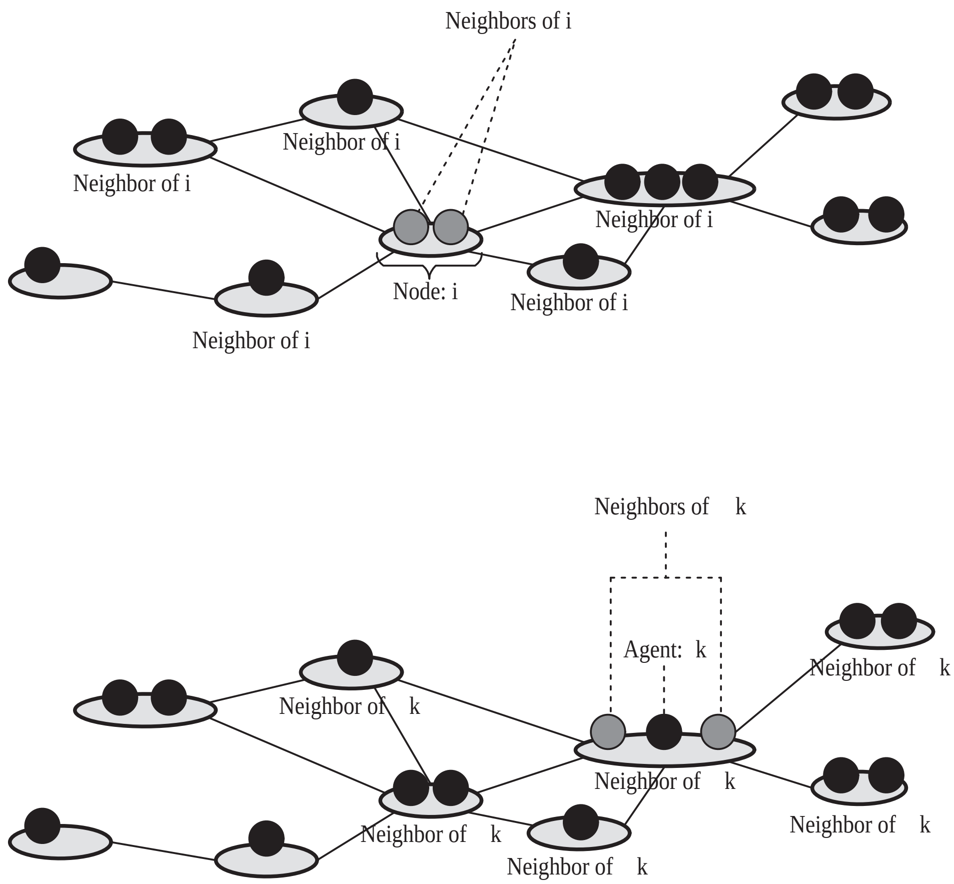

Fig. 2. Neighborhoods of nodes (top) and agents (agents). The neighborhood of a node i includes the nodes that are neighbors in the usual graph theoretic sense and the agents that are located in node i. The neighborhood of an agent k includes the agents that are located in the same node as k is, the node $i = q_k$ where agent k is located, and those agents located at nodes that are neighbors of node $i = q_k$.

- *There exists one* $q_i \in Q_i$ *for each* $(R_i, c_i) \in \Xi^{N_r} \times \mathcal{A}$, *where* c_i *represents the number of agents that occupy node* $i \in \mathcal{V}$. *Note that* Q_i *is guaranteed to be finite.*
- *The continuous dynamics* $\Sigma_{q,i}$ *are to be designed. The restrictions are:* $X_{q,i} = X_i$ *for all* $q_i \in Q_i$ *(the continuous state space is the same for all discrete modes), and* $U_{q,i}$ *is a function of the states of the agents located at node* i.
- *There are no autonomous discrete transitions i.e.,* $S^A_{q,i} = \emptyset$, $G^A_{q,i} = \emptyset$, $Z^A_{q,i} = \emptyset$ *for all* $q_i \in Q_i$. *Therefore, we drop the notation* $(\cdot)^C$ *for the controlled transitions.*
- *Every controlled transition label* $s_{q,i} \in S_{q,i}$ *is a function that maps the occurrence of an event in the node to the next discrete mode of the system:* $s_{q,i} : \mathbf{E}_v \to Q_i$ *for all* $q_i \in Q_i$, *where* $\mathbf{E}_v$ *is the set of possible node events.*
- $\mathbf{E}_v$ *contains two types of events: 1) Changes in the resources* R_i *for each node* i, *and 2) Changes in the number of agents* $k \in \mathcal{A}$ *contained in* $\mathcal{N}_i$ *for each node* i.

- *$G_{q,i} = X_i$ for all $q_i \in Q_i$ i.e., a discrete transition is always possible regardless of what the continuous state is.*
- *There are no restrictions for the controlled transition maps $Z_{q,i}$.*

Assumption 5 (Agent dynamics) *Each agent $k \in \mathcal{A}$ can be described as a Controlled Hybrid Dynamical System $\mathbf{H}_k$ that satisfies the following conditions:*

- *There exists one $q_k \in Q_k$ for each $i \in \mathcal{V}$. Note that Q_k is guaranteed to be finite.*
- *The continuous dynamics $\Sigma_{q,k}$ are to be designed. The restrictions are $X_{q,k} = X_k$ for all $q_k \in Q_k$ (the continuous state space is the same for all discrete modes), and $U_{q,k}$ is a function of the state of the node that agent k occupies. Note in Assumption 3 that the continuous part of the state of the agent $(x_{k,1}, x_{k,2}, ..., x_{k,N_r}) \in X_k$.*
- *There are no autonomous discrete transitions i.e., $S^A_{q,k} = \emptyset$, $G^A_{q,k} = \emptyset$, $Z^A_{q,k} = \emptyset$ for all $q_k \in Q_k$. Therefore, we drop the notation $(\cdot)^C$ for the controlled transitions.*
- *Every controlled transition label $s_{q,k} \in S_{q,k}$ is a function that maps the occurrence of an event in the agent to the next discrete mode of the system: $s_{q,k} : \mathbf{E}_a \rightarrow Q_k$ for all $q_k \in Q_k$, where $\mathbf{E}_a$ is the set of possible agent events.*
- *$\mathbf{E}_a$ is a set of logic valued functions (unspecified at this moment because of being part of the design parameters). If the output of $e_k \in \mathbf{E}_a$ is true, then an event is generated, otherwise no event is generated.*
- *$G_{q,k} = X_k$ for all $q_k \in Q_k$ i.e., a discrete transition is always possible regardless of what the continuous state is.*
- *There are no restrictions for the controlled transition maps $Z_{q,k}$.*

Assumptions 4 and 5 describe the dynamic behavior of the nodes and the agents. We believe they are as general as possible, since they are based on the general description of hybrid systems [4]. The *dynamics of a node* behave as follows: Given an initial hybrid condition - a discrete and a continuous state: (q_i, x_i) - the continuous state evolves according to the current continuous dynamics in force $(\Sigma_{q,i})$ until the occurrence of a discrete event, caused by a change in the resources of the node R_i or by the arrival (departure) of an agent to (from) the node. This event causes the system to change to a new discrete state q_i', where the evolution of the system continues according to the new continuous dynamics $\Sigma_{q',i}$.

The *dynamics of an agent* behave in a similar fashion to those of a node, with the following caveats: The discrete states in the hybrid model of the agent represent the nodes in the network that may host the agent. Similarly, the discrete transitions represent the migration of an agent between two nodes. Thus the agent events $\mathbf{E}_a$, which are discrete valued functions, must be designed to allow the agent to choose the best node in the network as a function of its requirements. Finally note that the *interactions between nodes and agents* happen at both the continuous and discrete levels: 1) The continuous input of the nodes dynamics are functions of the continuous states of the agents, and *vice-versa*. 2) The discrete dynamics of the nodes are influenced by the movements of agents between nodes, while the discrete dynamics of the agents are influenced by the availability of resources in the nodes.

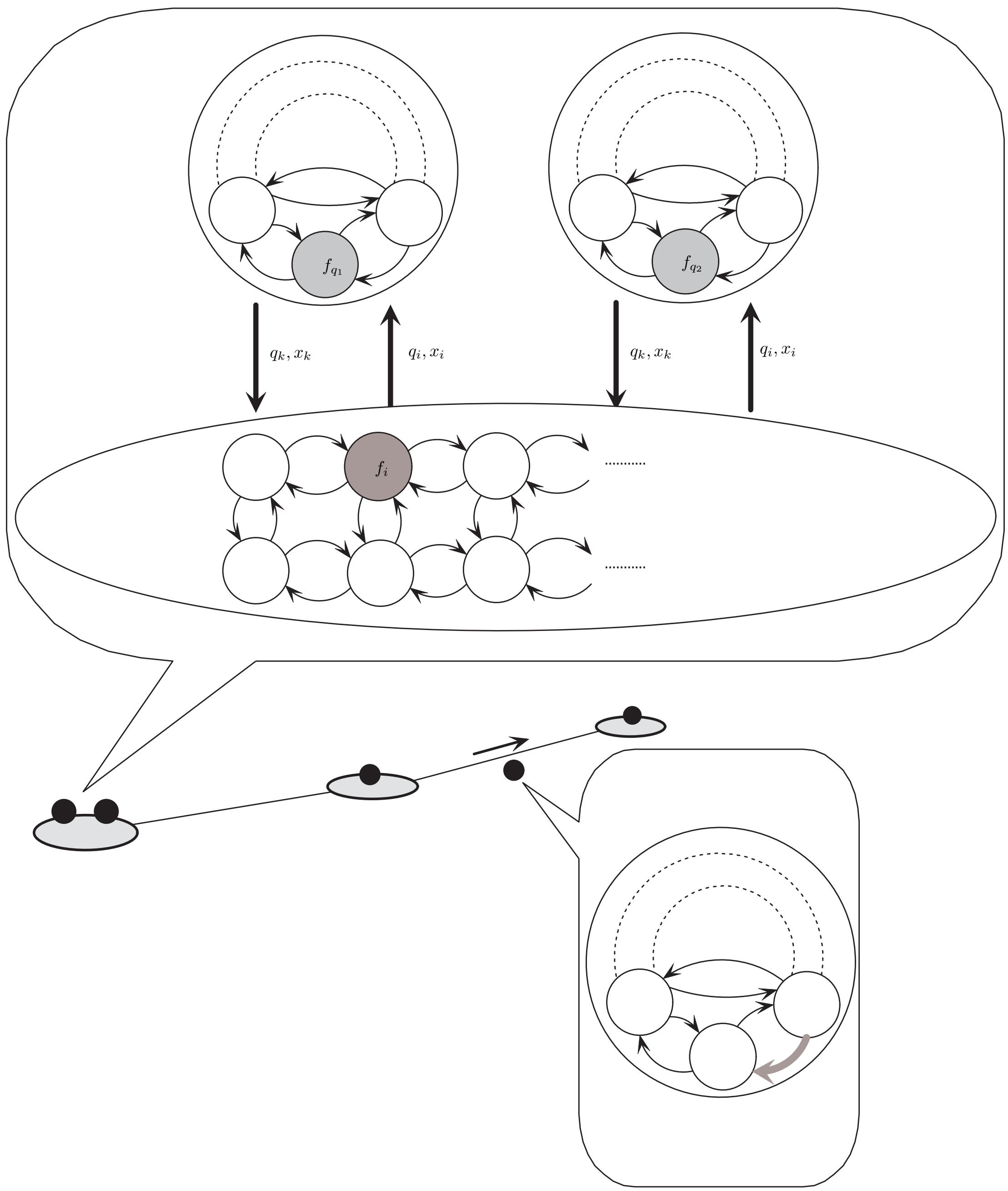

Fig. 3. Example of the dynamical behavior of agents and nodes. Agents are modeled as hybrid systems, which can be represented as hybrid automata. Each mode in an automaton corresponds to a possible location of an agent in the network (Agents on top). Each transition between modes represents a change of location made by an agent (agent at the bottom). The dynamics of the nodes are also modeled as hybrid systems. Each mode represents a number of agents residing at a node paired with the availability of resources that varies in discrete manner. The agents on top are located on a node, and therefore have a fixed discrete state, while the continuous dynamics of agents and the nodes that hosts them are interacting. The agent at the bottom is moving between nodes, so a discrete transition is happening.

C. Design objective

The objective is to design the nodes and agents dynamical equations such that the usage of the resources in the environment (network) is optimized with respect to the requirements of the agents. In order to express these requirements in a more formal way we assume:

Assumption 6 (Utility functions) *Each agent k has an expression of its utility function $U_k(x_k) : \mathbb{R}^{N_r} \to \mathbb{R}$ where $x_k = (x_{k,1}, x_{k,1}, ..., x_{k,N_r})^T$. The utility function is of the form:*

$$U_k(x_k) = \sum_{j=1}^{N_r} u_{k,j}(x_{k,j}) \tag{1}$$

where $u_{k,j}(x_{k,j}) : \mathbb{R} \to \mathbb{R}$ is assumed to be a strictly concave, non-decreasing, and differentiable function of $x_{k,j}$ for all $k \in \mathcal{A}$ and all $j \in \mathcal{R}$. Moreover, we assume that $u_{k,j}(x_{k,j}) \to -\infty$ as $x_{k,j} \to 0$

Note that this assumption is not very restrictive; the least information that each agent should have is its own utility function. Moreover, it is reasonable to assume that the more resources an agent obtains, the more benefit it achieves (strictly increasing utility function). The concavity and differentiability assumptions allow us to apply convex optimization techniques [3], [31] without restricting the problem solution (it is only necessary to look for the appropriate function that fits these constraints), and the requirement that $u_{k,j}(x_{k,j}) \to -\infty$ as $x_{k,j} \to 0$ allows us to avoid the possibility of any agent getting zero resources. Therefore, the design objective can be stated as:

Problem 1: Design the node and agent dynamics for all the components of the hybrid multiagent system described in Assumptions 1-6, such that it is asymptotically stable with equilibrium state (q_k, x_k) for all $k \in \mathcal{A}$, and the equilibrium maximizes the aggregate utility of all the agents in the network as given by $\sum_{k=1}^{N_a} U_k(x_k)$.

III. Equivalent Hierarchical Optimization Problem

In order to gain insight into the design problem we first analyze an optimization problem that is based on the network objective (maximization of $\sum_{k=1}^{N_a} U_k(x_k)$) and the constraints imposed by the system dynamics (Assumptions 1-6). Note that to formulate an optimization problem, we must consider a fixed configuration of the network as in the following result:

Lemma 1 *Given a fixed configuration of the network $\mathcal{G}$ (Fixed number of nodes, number of agents, amount of resources), the equilibrium state (q_k, x_k) for all $k \in \mathcal{A}$ that maximizes $\sum_{k=1}^{N_a} U_k(x_k)$ in **Problem 1**, is the solution, under the same configuration, to the following optimization problem:*

$$\max_{\substack{(x_1,..,x_k,..,x_{N_a}),\\ (q_1,..,q_k,..,q_{N_a})}} \sum_{k=1}^{N_a} U_k(x_k) \tag{2}$$

subject to

$$x_{k,j} \geq 0, \quad \text{for all } (k,j) \in \mathcal{A} \times \mathcal{R} \tag{3a}$$

$$q_k \in \mathcal{V}, \quad \text{for all } k \in \mathcal{A} \tag{3b}$$

$$\sum_{\{k:q_k=i\}} x_{k,j} \leq r_{i,j}, \quad \text{for all } (i,j) \in \mathcal{V} \times \mathcal{R} \tag{3c}$$

Proof: (2) follows from **Problem 1**. (3a) and (3b) are implied by Assumption 3. Finally (3c) follows from Assumption 2. ■

The optimization problem in Lemma 1 is a mixed integer-nonlinear programming problem, and as a consequence $\mathcal{NP}$-complete in the number of the discrete states ([35] Chapter 18), which in our case is given by the expression $N_d = N_v^{N_a}$ that grows exponentially with the number of nodes in the network. Therefore, the numerical solution of this problem becomes computationally intractable as the number of nodes in the network increases. We are not, however, interested in solving this problem directly. Instead, we would like to use the formulation in Lemma 1 to help us identify the desired characteristics of the dynamics of the nodes and the agents.

First, note that the resources of a node are allocated among the agents located in that node (Assumption 2), which means that the agents only have access to the resources of the nodes that hosts them, as implied by (3c). We show that this observation allows us to convert the mixed integer-nonlinear optimization problem into a hierarchical problem, with two subproblems: A convex optimization problem within each node in the network, and an integer optimization problem on the global behavior of the network.

Let V_i be the set of agents located at node i, i.e. $V_i = \{k \in \mathcal{A} : v_k = i\}$.

Lemma 2 *Given a fixed possible distribution of agents* $\bar{D} = (\bar{q}_1, ..\bar{q}_k, ..\bar{q}_{N_a})$*, the solution of* (2)-(3) *is given by the solution for each* $(i,j) \in \mathcal{V} \times \mathcal{R}$ *of the concave optimization problem:*

$$\max_{\substack{(x_{\alpha,j}, x_{\beta,j}, \ldots, x_{\gamma,j}): \\ \alpha,\beta,\ldots,\gamma \in V_i}} \sum_{\{\kappa \in V_i\}} u_{\kappa,j}(x_{\kappa,j}) \tag{4}$$

subject to

$$x_{k,j} \geq 0, \quad \text{for all } k \in V_i \tag{5a}$$

$$\sum_{\{k \in V_i\}} x_{k,j} \leq r_{i,j} \tag{5b}$$

Proof: Assigning a fixed value $i \in \mathcal{V}$ to each q_k allows us to discard equation (3b), rewrite equation (2) as

$$\sum_{\{i \in \mathcal{V}\}} \left[\max_{\substack{(x_\alpha, x_\beta, \ldots, x_\gamma): \\ q_\alpha = i, q_\beta = i, .., q_\gamma = i}} \sum_{\{\kappa : q_\kappa = i\}} U_\kappa(x_\kappa) \right]$$

because agents at node i only have access to resources of node i. Equations (3a) and (3c) are also rewritten as a set of equations indexed by $\mathcal{V}$ that are independent of the choice of q_k, obtaining for each $i \in \mathcal{V}$:

$$\max_{\substack{(x_\alpha, x_\beta, \ldots, x_\gamma): \\ \alpha,\beta,\ldots,\gamma \in V_i}} \sum_{\{\kappa \in V_i\}} U_\kappa(x_\kappa)$$

subject to

$$x_{k,j} \geq 0, \quad \text{for all } (k,j) \in V_i \times \mathcal{R}$$

$$\sum_{\{k \in V_i\}} x_{k,j} \leq r_{i,j}, \quad \text{for all } j \in \mathcal{R}$$

but the objective equation can be rewritten using (1) (and reordering sums) as:

$$\max_{\substack{(x_\alpha, x_\beta, \ldots, x_\gamma): \\ \alpha, \beta, \ldots, \gamma \in V_i}} \sum_{j=1}^{N_r} \left(\sum_{\{k \in V_i\}} u_{k,j}(x_{k,j}) \right) \tag{7}$$

Since $u_{k,j}(x_{k,j})$ is a strictly concave function of $x_{k,j}$ for each $(k,j) \in \mathcal{A} \times \mathcal{R}$, the terms inside the parentheses of equation (7), and the complete utility function are all concave functions of their arguments. Similarly the constraint equations above are also concave in their arguments. These facts allows us to consider N_r independent concave optimization problems within each node (one for each resource) proving the claim. ■

Let $\mathbf{D} = \{\bar{D} = (\bar{q}_1, ..\bar{q}_k, ..\bar{q}_{N_a}) : \bar{D} \in \mathcal{V}^{N_a}\}$, be the set of all possible distributions of agents in the network.

Theorem 1 *The optimization problem* (2)-(3)*, is equivalent to the following hierarchical optimization problem:*

$$\max_{\bar{D} \in \mathbf{D}} \sum_{(i,j) \in \mathcal{V} \times \mathcal{R}} \mathbf{U}_{i,j}(\bar{D}) \tag{8}$$

where for each $(i,j) \in \mathcal{V} \times \mathcal{R}$*:*

$$\mathbf{U}_{i,j}(\bar{D}) = \max_{\substack{(x_{\alpha,j}, x_{\beta,j}, \ldots, x_{\gamma,j}): \\ \alpha, \beta, \ldots, \gamma \in V_i}} \sum_{\{\kappa \in V_i\}} u_{\kappa,j}(x_{\kappa,j}) \tag{9}$$

subject to

$$x_{k,j} \geq 0, \quad \text{for all } k \in V_i : (i,j) \in \mathcal{V} \times \mathcal{R} \tag{10a}$$

$$\sum_{\{k \in V_i\}} x_{k,j} \leq r_{i,j}, \quad \text{s.t. } (i,j) \in \mathcal{V} \times \mathcal{R} \tag{10b}$$

Proof: Equations (9)-(10) are identical to (4)-(5). Then by Lemma 2, (9)-(10) are solution for the optimization problem (2)-(3) if the distribution of agents is considered fixed. Therefore to obtain an equivalent description to problem (2)-(3), the agent location has to be added as a decision variable to the problem in Lemma 2. Note that because the agents are constrained to use resources from the node they occupy, the total benefit in the network $\sum_{k=1}^{N_a} U_k(x_k)$ is identical to the sum of the benefit that each node in the network generates through the agents it hosts i.e.

$$\sum_{k=1}^{N_a} U_k(x_k) = \sum_{(i,j) \in \mathcal{V} \times \mathcal{R}} \sum_{\{\kappa \in V_i\}} u_{\kappa,j}(x_{\kappa,j})$$

If both sides of this equations are maximized with respect to x and then with respect to q we obtain the equivalence between (2) and (8)-(9). ■

The solution of the problem in Theorem 1, then takes on the following conceptual form (as depicted in Figure 4): A centralized algorithm generates a set of possible distributions of agents in the network, and communicates this information to the nodes in the network, who solve the convex optimization problem (9)-(10) for each one of these possible configurations. As a result, they obtain a set of benefit values, one for each possible configuration, that are communicated back to the centralized algorithm which selects the configuration that yields the optimum performance for the complete network. While, this type of solution provides insight to the potential behavior of the final design of the system as shown in section V, it is however undesirable and may even be unfeasible because of its centralized nature (a feasible solution using a centralized randomized algorithm is discussed in [29]).

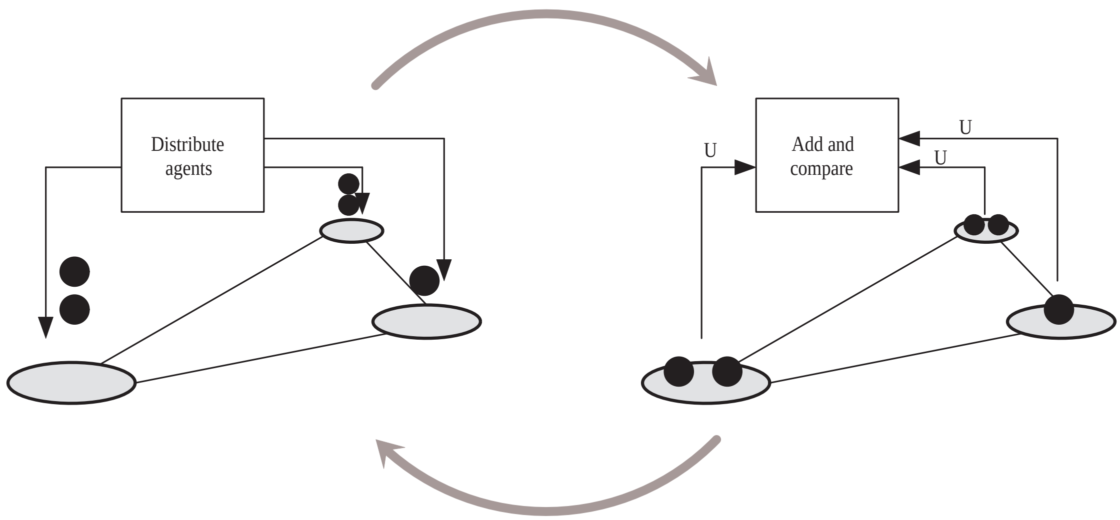

Fig. 4. Conceptual view of the hierarchical solution: A centralized algorithm distributes the agents (Left). The nodes then allocate the resources to the agents they host, compute the benefit, and send it back to the centralized algorithm that obtains the aggregate benefit in order to compare the possible distributions (Right).

IV. Solution for the Concave Optimization Problem within each Node

Consider the optimization problem in Lemma 2 (or equivalently the problem (9)-(10) in Theorem 1). For simplicity, we drop the notation that indicates the node and type of resource $(i, j) \in \mathcal{V} \times \mathcal{R}$. Thus we consider the problem of maximizing:

$$\mathcal{U} = \sum_{k=1}^{n_a} u_k(x_k), \tag{11}$$

subject to

$$x_k \geq 0, \quad \text{for all } k \in \{1, 2, ..., n_a\} \tag{12a}$$

$$\sum_{k=1}^{n_a} x_k \leq r \tag{12b}$$

where n_a (in general different from N_a) is the number of agents participating in this particular optimization task.

Lemma 3 *Let $\lambda_p \in \mathbb{R}$ for all $p \in \{1, 2, ..., n_a + 1\}$. The necessary and sufficient conditions for $(x_1^*, x_2^*, ..., x_{n_a}^*)$ to be the maximal solution of the problem in (11)-(12) are:*

$$\frac{du_p}{dx_p} + \lambda_{n_a+1} - \lambda_p = 0 \quad \forall p \in \{1, 2, ..., n_a\} \tag{13a}$$

$$\lambda_p x_p = 0 \quad \forall p \in \{1, 2, ..., n_a\} \tag{13b}$$

$$\lambda_{n_a+1} \left(\sum_{k=1}^{n_a} (x_k) - r \right) = 0 \tag{13c}$$

$$-x_p \leq 0 \quad \forall p \in \{1, 2, ..., n_a\} \tag{13d}$$

$$\sum_{k=1}^{n_a} (x_k) - r \leq 0 \tag{13e}$$

$$\lambda_p \leq 0 \quad \forall p \in \{1, 2, ..., n_a + 1\} \tag{13f}$$

Proof: Let $g_p = -x_p$ for all $p \in \{1, 2, ..., n_a\}$, and $g_{n_a+1} = \sum_{k=1}^{n_a} (x_k^*) - r$ which are the constraints (12) of the problem. Then, and using Lagrange multipliers, the Karush-Khun-Tucker conditions for optimality [31] become

$$\frac{\partial \mathcal{U}}{\partial x_p} + \sum_{j=1}^{n_a+1} \lambda_j \frac{\partial g_j}{\partial x_p} = 0 \qquad \forall p \in \{1, 2, ..., n_a\}$$

$$\lambda_p g_p = 0 \qquad \forall p \in \{1, 2, ..., n_a + 1\}$$

$$g_p \leq 0 \qquad \forall p \in \{1, 2, ..., n_a + 1\}$$

$$\lambda_p \leq 0 \qquad \forall p \in \{1, 2, ..., n_a + 1\}$$

Evaluating the derivatives we obtain $\frac{\partial \mathcal{U}}{\partial x_p} = \frac{du_p}{x_p}$ for all $p \in \{1, 2, ..., n_a\}$,

$$\frac{\partial g_j}{\partial x_p} = \begin{cases} 0 & j \neq p \\ -1 & j = p \end{cases}$$

for all $j, p \in \{1, 2, ..., n_a\}$, and $\frac{\partial g_{n_a+1}}{\partial x_p} = 1$ for all $p \in \{1, 2, ..., n_a\}$. Substituting these derivatives back into the previous equations we obtain equation (13). Since both the utility function (11) and the constraints (12) are strictly concave, equation (13) becomes a necessary and sufficient condition for the optimality of $(x_1^*, x_2^*, ..., x_{n_a}^*)$. ■

Lemma 4 *The solution $(x_1^*, x_2^*, .., x_{n_a}^*, \lambda_1^*, \lambda_2^*, ...\lambda_{n_a+1}^*)$ of equation (13) is given by $\lambda_p^* = 0$ for $p \in \{1, 2, ..., n_a\}$ and by*

$$\frac{du_p}{dx_p} + \lambda = 0 \quad \forall p \in \{1, 2, ..., n_a\} \tag{14a}$$

$$\sum_{k=1}^{n_a} (x_k) - r = 0 \tag{14b}$$

$$x_p > 0 \quad \forall p \in \{1, 2, ..., n_a\} \tag{14c}$$

$$\lambda \leq 0 \tag{14d}$$

for $(x_1^*, x_2^*, .., x_{n_a}^*, \lambda_{n_a+1}^*)$ *where* $\lambda = \lambda_{n_a+1}^*$.

Proof: Consider equation (13e), and note that since $x_p \geq 0$ for all $p \in \{1, 2, ..., n_a\}$ and that $u_p(x_p)$ is a strictly increasing function of x_p for all $p \in \{1, 2, ..., n_a\}$, any choice of $(x_1, x_2, ..., x_{n_a})$ for (13e) such that $\sum_{k=1}^{n_a}(x_k) - r < 0$ will be suboptimal. Thus equation (13e) must be modified as $\sum_{k=1}^{n_a}(x_k) - r = 0$. This conclusion automatically discards equation (13c) because it is trivially satisfied. Equation (13b) provides two choices for each p: $\lambda_p = 0$ or $x_p = 0$. The second choice however yields $u_p(x_p) = -\infty$, violating the maximization of the utility function $\mathcal{U}$. Thus $\lambda_p = 0$ for all $p \in \{1, 2, ..., n_a\}$. The same argument leads to the modification of equation (13d) to $-x_p < 0$ for all $p \in \{1, 2, ..., n_a\}$. As a consequence of these observations (13a) and (13f) are simplified. Conditions (13) are then modified to obtain (14). ■

Applying Lemmas 3 and 4 to equations (4) and (5), the main result of this section is stated as:

Theorem 2 *Given the available resource* $r_{i,j}$ *of type* $j \in \mathcal{R}$ *in the node* $i \in \mathcal{V}$*, the utility function* (4) *is maximized by* $(x_{\alpha,j}^*, x_{\beta,j}^*, ..., x_{\gamma,j}^*)$*, where* $\alpha, \beta, ...\gamma \in V_i$ *subject to* (5) *if and only if for each* $(i, j) \in \mathcal{V} \times \mathcal{R}$*,* $(x_{\alpha,j}^*, x_{\beta,j}^*, ..., x_{\gamma,j}^*)$ *satisfies:*

$$\left.\frac{du_{\kappa,j}}{dx_{\kappa,j}}\right|_{x_{\kappa,j}=x_{\kappa,j}^*} + \lambda_{i,j}^* = 0 \quad \forall \kappa \in V_i \tag{15a}$$

$$\sum_{\kappa \in V_i}(x_{\kappa,j}^*) - r_{i,j} = 0 \tag{15b}$$

$$x_{\kappa,j}^* > 0 \quad \forall \kappa \in V_i \tag{15c}$$

$$\lambda_{i,j}^* \leq 0 \tag{15d}$$

V. Design of the Continuous Dynamics of Agents and Nodes

Based on the results of Section III we now provide a precise description for the continuous dynamics for both agents and nodes. We start by recognizing that the hierarchical structure proposed in Theorem 1 allows us to design the continuous dynamics independently from the discrete dynamics. We say that an optimization problem $\mathcal{P}$ is *solved exactly* by a system $\mathbf{H}$, if $\mathbf{H}$ has a unique asymptotically stable equilibrium point h_e that solves $\mathcal{P}$.

Proposition 1 *In order to exactly solve the optimization problem stated in Lemma 2 using the hybrid model for nodes and agents given in Assumptions 4 and 5, only the continuous dynamics in such models need to be considered.*

Proof: Without loss of generality, consider a fixed amount of resources available on each node. This assumption is not restrictive because allowing variations in the resources according to Assumptions 2 and 4 does not change the logical arguments of the proof and only causes the size of the discrete state space to increase by a factor identical to the size of the set Ξ. Therefore, in order to simplify notation we restrict ourselves to the case where the resources remain fixed. From Assumption 4 note that there is a discrete state q_i in node's hybrid model for each possible number of agents residing in node i. So for any given choice of discrete states $(q_1, ..., q_k, ..., q_{N_a}) \in Q_1 \times ... \times Q_k \times ... \times Q_{N_a}$ in the agent's model, there is a discrete state $(q_1, ..., q_i, ..., q_{N_a}) \in Q_1 \times ... \times Q_i \times ... \times Q_{N_v}$

in each node's model that remains invariant as long as the discrete states of the agents remain fixed. Then the hybrid states of both nodes $h_i = (q_i, x_{q,i}),\ \forall i \in \mathcal{V}$ and agents $h_k = (q_k, x_{q,k}),\ \forall k \in \mathcal{V}$ have fixed discrete dynamics if the distribution of agents $\bar{D}$ remains fixed, which is true by assumption in Lemma 2. This implies that the continuous dynamics of the nodes $\mathbf{\Sigma_i}$ and the agents $\mathbf{\Sigma_k}$ interact without switching between discrete states as long as the distribution remains fixed. This implies that the optimization within each node must be solved by the corresponding continuous dynamics. ■

The reader may note that the optimization problem in Lemma 2 is a special case of a resource allocation problem considered in the literature, namely the dynamic modeling of congestion control algorithms on the Internet [1], [17], [19], [36], [41]. As a consequence, we use such results in the design of the continuous dynamics of agents and nodes in our problem, enabling a coordination algorithm that solves the optimization problem of interest. Specifically, following the treatment in [19], the optimization problem in Lemma 2 is a special case of equation (1 in [19]) when L has only one link. Therefore it is possible to apply the results in [1], [17], [19], [36], [41] to solve our problem.

According to [19] there are three types of dynamical systems capable of solving the optimization problem in Lemma 2: A primal algorithm, a dual algorithm, and a primal-dual algorithm. We choose the primal-dual approach for our problem because it is better suited for the hybrid models in Assumptions 4 and 5. It is important to note that the primal and the dual approaches may also be used. A discussion of these alternatives can be found in Section VII.

We now provide a description for a dynamical system that solves exactly the optimization problem in Lemma 2. This description is based on the primal-dual algorithm developed in [1], [41], and generalized in [19], [36]. Note that the state of the agents $x_{k,j}$ is similar to the state of the routes x_r in [19], and the resources $r_{i,j}$ are the analog of the link capacity c_l in [19]. We now let $p_{i,j}$ be the continuous state of the node $i \in \mathcal{V}$ and resource $j \in \mathcal{R}$, which is the analog to the price p_l in [19]. We note that the primal-dual description in [19] differs slightly from that in [36]. The following result uses the description in [36].

Lemma 5 *Given a fixed distribution of agents $\bar{D} \in \mathbf{D}$, the optimization problem in Lemma 2 is solved exactly for each $(i,j) \in \mathcal{V} \times \mathcal{R}$, by the following dynamical system:*

$$\dot{x}_{\alpha,j} = K_{\alpha,j}(x_{\alpha,j})\big(u'_{\alpha,j}(x_{\alpha,j}) - p_{i,j}\big) \tag{16a}$$

$$\dot{x}_{\beta,j} = K_{\beta,j}(x_{\beta,j})\big(u'_{\beta,j}(x_{\beta,j}) - p_{i,j}\big) \tag{16b}$$

$$\vdots \qquad \vdots$$

$$\dot{x}_{\gamma,j} = K_{\gamma,j}(x_{\gamma,j})\big(u'_{\gamma,j}(x_{\gamma,j}) - p_{i,j}\big) \tag{16c}$$

$$\dot{p}_{i,j} = \big[L_{i,j}(p_{i,j})(y_{i,j} - r_{i,j})\big]^+_{p_{i,j}} \tag{16d}$$

where $\alpha, \beta, ..., \gamma \in V_i$, $x_\kappa = (x_{\kappa,1}, x_{\kappa,2}, ..., x_{\kappa,N_r})$ is the continuous state of agent $\kappa \in V_i \subseteq \mathcal{A}$, $p_i = (p_{i,1}, p_{i,2}, ..., p_{i,N_r})$ is the continuous state of node $i \in \mathcal{V}$, $y_{i,j} = \sum_{\kappa \in V_i} x_{\kappa,j}$, $K_{\kappa,j}(x_{\kappa,j})$ is any nondecreasing, continuous function with $K_{\kappa,j}(x_{\kappa,j}) > 0$ for $x_{\kappa,j} > 0$ for all $\kappa \in V_i$, $L_{i,j}(p_{i,j})$ is a positive, nondecreasing continuous function,

$u'_{\alpha,j}(x_{\alpha,j}) = \frac{du_{\kappa,j}}{dx_{\kappa,j}}(x_{\alpha,j})$ *is the derivative of the utility function of agent* κ *and resource* j *with respect to its argument, and the notation*

$$\big[g(t)\big]_t^+ = \begin{cases} g(t), & t > 0, \\ \max(g(t), 0), & t = 0. \end{cases}$$

Proof: From Proposition 1 we know that the problem in Lemma 2 is solved using only continuous dynamics in agents and nodes. Then, given a pair $(i, j) \in \mathcal{V} \times \mathcal{R}$ we note that the optimization problem (4)-(5) is a special case of the problem (2.1)-(2.2) in [36] (or equivalently (1) in [19]). So following the discussion on the Primal-Dual Algorithm in [19], [36] it is clear that the optimization problem in Lemma 2 is exactly solved by (16). ■

Based on the previous result we can establish a detailed model for the continuous dynamics of the nodes and the agents. This dynamic description is guaranteed to solve the optimization problem in Lemma 2, or equivalently the problem (9)-(10) in Theorem 1. Therefore the continuous dynamics of this interconnected system will solve the optimization of the network resources *locally* leaving the global optimization to the discrete dynamics of the hybrid models.

Proposition 2 *For each* $q_i \in Q_i$ *and all* $i \in \mathcal{V}$ *the the continuous dynamical system has the following description:*

1) *The continuous state space* $X_{q,i} = P$ *where* $P = \{(p_1, p_2, ..., p_{N_r}) \in \mathbb{R}^{N_r}\}$.
2) *The continuous dynamics are given in a diagonal matrix:*

$$f_{q,i} = \begin{bmatrix} f_{q,i,1} & 0 & \dots & 0 \\ 0 & f_{q,i,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & f_{q,i,N_r} \end{bmatrix}$$

where

$$\dot{p}_{q,i,j} = f_{q,i,j} = \big[L_{q,i,j}(p_{q,i,j})\big(\big(\sum_{\mu \in U_{q,i}} \mu\big) - r_{q,i,j}\big)\big]^+_{p_{q,i,j}} \ , \quad \forall j \in \mathcal{R},$$

where $L_{q,i,j}(p_{q,i,j})$ *is a positive, nondecreasing continuous function.*

3) *The set of continuous inputs* $U_{q,i} = \bigcup_{j \in \mathcal{R}} U_{q,i,j}$ *where* $U_{q,i,j} = \{x_{q,\kappa,j} : \kappa \in V_i\}$.

Proof: Assumption 2 states that the number of types of resources in the network N_r is constant for all nodes $i \in \mathcal{V}$, and Lemma 5 implies there exists one state dimension for each type of resource in the network, then $X_{q,i} \subseteq \mathbb{R}^{N_r}$. Since there are no limitations for $X_{q,i}$ we make $X_{q,i} = \mathbb{R}^{N_r}$ for all $q_i \in Q_i$ and all $i \in \mathcal{V}$ in item 1).

For item 2) note that (16d) describes the dynamics of one resource being allocated inside each node. Therefore in order to completely describe the N_r resources available in each node one must consider N_r decoupled resource dynamics proving the claim.

Finally for item 3), the continuous control inputs for each node $i \in \mathcal{V}$ described by $y_{i,j}$ in (16d) are the states of all the agents located in that node i.e., $\{x_\kappa : \kappa \in V_i\}$, which implies the third item for all $q_i \in Q_i$ and all $i \in \mathcal{V}$. ■

Proposition 3 *For each $q_k \in Q_k$ the the continuous dynamical system has the following description:*

1) *The continuous state space $X_{q,k} = X$ where $X = \{x = (x_1, x_2, ..., x_{N_r}) \in \mathbb{R}^{N_r}; x_1 \geq 0, x_2 \geq 0, ..., x_{N_r} \geq 0\}$.*
2) *The continuous dynamics are given in a diagonal matrix:*

$$f_{q,k} = \begin{bmatrix} f_{q,k,1} & 0 & \dots & 0 \\ 0 & f_{q,k,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & f_{q,k,N_r} \end{bmatrix}$$

where

$$\dot{x}_{q,k,j} = f_{q,k,j} = K_{q,k,j}(x_{q,k,j})\big(u'_{q,k,j}(x_{q,k,j}) - \mu_{q,k,j}\big), \quad \forall j \in \mathcal{R},$$

where $K_{q,\kappa,j}(x_{q,\kappa,j})$ is any nondecreasing, continuous function with $K_{q,\kappa,j}(x_{\kappa,j}) > 0$ for $x_{q,\kappa,j} > 0$ for all $q_\kappa = i$ and $u'_{q,k,j}(x_{q,k,j}) = \frac{du_{q,k,j}}{dx_{q,k,j}}(x_{q,k,j})$.

3) *The set of continuous inputs $U_{q,k} = \bigcup_{j \in \mathcal{R}} U_{q,k,j}$ where $U_{q,k,j}$ is the singleton $\{p_{q,i,j};\ k \in V_i\}$ i.e., $\mu_{q,k,j} = p_{q,i,j}$ for $k \in V_i$.*

Proof: Follows from similar arguments to the proof of Proposition 2. ■

We now prove that each possible distribution of agents forms a continuous dynamical system that is globally asymptotically stable with an equilibrium point that solves exactly the optimization problem in Lemma 2. This result is an extension of Lemma 5, but is important for our problem, because it guarantees that whatever the location of the agents in the nodes is, the system will achieve the local optimum solution for that particular choice of location. Therefore, each particular combination of discrete states of nodes and agents $(q_1, ..., q_i, ..., q_{N_a}) \in Q_1 \times ... \times Q_i \times ... \times Q_{N_v}$, and $(q_1, ..., q_k, ..., q_{N_a}) \in Q_1 \times ... \times Q_k \times ... \times Q_{N_a}$, (which we call interconnection) will have a globally asymptotically stable point. The state will only be perturbed from that equilibrium when a discrete event occurs on the agents or on the nodes, but will automatically go towards the equilibrium point of the new interconnection, and locally optimizes the resource distribution for this new interconnection.

Theorem 3 *A selection of discrete states of the agents $(q_1, ..., q_k, ..., q_{N_a}) \in Q_1 \times ... \times Q_k \times ... \times Q_{N_a}$, and of discrete states of the nodes $(q_1, ..., q_i, ..., q_{N_a}) \in Q_1 \times ... \times Q_i \times ... \times Q_{N_v}$, generates $N_v \times N_r$ interconnected systems indexed by $(i, j) \in \mathcal{V} \times \mathcal{R}$, where each of them is governed by the following continuous dynamics:*

$$\dot{x}_{q,\alpha,j} = K_{q,\alpha,j}(x_{q,\alpha,j})\big(u'_{q,\alpha,j}(x_{q,\alpha,j}) - p_{q,i,j}\big) \tag{17a}$$

$$\dot{x}_{q,\beta,j} = K_{q,\beta,j}(x_{q,\beta,j})\big(u'_{q,\beta,j}(x_{q,\beta,j}) - p_{q,i,j}\big) \tag{17b}$$

$$\vdots \qquad \qquad \vdots$$

$$\dot{x}_{q,\gamma,j} = K_{q,\gamma,j}(x_{q,\gamma,j})\big(u'_{q,\gamma,j}(x_{q,\gamma,j}) - p_{q,i,j}\big) \tag{17c}$$

$$\dot{p}_{q,i,j} = \Big[L_{q,i,j}(p_{q,i,j})\big((\sum_{\kappa \in V_i} x_{q,\kappa,j}) - r_{q,i,j}\big)\Big]^+_{p_{q,i,j}} \tag{17d}$$

where $\alpha, \beta, ..., \gamma \in V_i$.

Moreover, each interconnected system (indexed by $(i, j) \in \mathcal{V} \times \mathcal{R}$) is globally asymptotically stable with an equilibrium point that satisfies the conditions in Theorem 2 i.e., solves exactly the optimization problem in Lemma 2.

Proof: Given a selection of discrete states $(q_1, ..., q_k, ..., q_{N_a}) \in Q_1 \times ... \times Q_k \times ... \times Q_{N_a}$ made by the agents, the nodes, which have information about the resource availability, automatically jump to a discrete set of modes $(q_1, ..., q_i, ..., q_{N_a}) \in Q_1 \times ... \times Q_i \times ... \times Q_{N_v}$. The agents selection $(q_1, ..., q_k, ..., q_{N_a})$ imply that each one of these agents $k \in \mathcal{A}$ has located itself in a node identified by q_k. This implies that each node $i \in \mathcal{V}$ indexes an interconnected system composed of itself and the set of agents located on it $\{k \in \mathcal{A} :\in V_i\}$ obtaining N_v interconnected systems. However, since the the dynamics for both agents and nodes are decoupled in the resources (second item in Propositions 2 and 3) we can consider the system as formed by $N_v \times N_r$ interconnected systems, indexed by $(i, j) \in \mathcal{V} \times \mathcal{R}$. Then from Propositions 2 and 3 the interconnected system (i, j) is governed by the dynamics composed by $f_{q,i,j}$ and $f_{q,\alpha,j}, f_{q,\beta,j}, ..., f_{q,\gamma,j}$ for $\alpha, \beta, ..., \gamma \in V_i$, which is written as (17). Since this equation is identical to (16), except for the notation stressing the dependence on the discrete mode, Lemma 5 implies that equations (17) are globally asymptotically stable, and that they exactly solve the optimization problem in Lemma 2. ■

VI. A Simulation Example

A. Experiment set-up

In this section we provide a simulation example to clarify the concepts developed in the paper. We are interested in testing the validity of Theorem 3 and its relationship to the solution of the optimization problem (9)-(10) as given in Theorem 2.

We consider a set of ten agents ($N_a = 10$) and a graph composed of three nodes ($N_v = 3$). We assume for simplicity that the graph is completely connected and that there is only one type of resource available in the network ($N_r = 1$). The utility functions of the agents, in reference to Assumption 6, have the form:

$$U_k(\mathbf{x}_k) = w_k \ln(x_k) \tag{18}$$

for all $k \in \mathcal{A} = \{1, .., 10\}$, where we have dropped the dependence on the resource index for simplicity. The utility function (18) satisfies Assumption 6 as long as $w_k > 0$. Note that $w_k; k = 1, \cdots, 10$ are weighting factors for each agent, and are used to quantify the importance that the resource has for each agent (the greater the value of w_k the more important is the resource for agent k). In our example, we arbitrarily choose w_k as: $(w_1, w_2, ..., w_{10}) = (0.5, 0.6, 0.1, 0.3, 0.4, 0.9, 0.4, 0.3, 0.2, 0.1)$. Note that the particular choice of utility function for this test is commonly referred to as proportional fairness [1], [17], [19], [36], [41].

The dynamics of nodes and agents following Propositions 2 and 3, are described by:

$$f_{q,i} = \left[L_{q,i}(p_{q,i})\big((\sum_{\{k:q_k=i\}} x_{q,k}) - r_{q,i}\big)\right]^+_{p_{q,i}}, \quad \forall i \in \mathcal{V} = \{1,2,3\}, \tag{19a}$$

$$f_{q,k} = K_{q,k}(x_{q,k})\big(\frac{w_k}{x_{q,k}} - p_{q,i=q_k}\big), \qquad \forall k \in \mathcal{A} \tag{19b}$$

where $L_{q,i}(p_{q,i}) = \tanh(p_{q,i}) + 1,\ \forall q_i \in Q_i\ \forall i \in \mathcal{V}$, and $K_{q,k}(x_{q,k}) = 50x_{q,k}\ \forall q_k \in Q_k\ \forall k \in \mathcal{A}$, satisfying the conditions in Propositions 2 and 3.

The interconnected system is tested over the time interval $T = [0, 9]\ sec$. The agents start at $t = 0$ located as: $(q_1, q_2, ..., q_{10}) = (1, 3, 2, 3, 2, 1, 1, 1, 3, 2)$ with the continuous initial condition $(x_1(0), x_2(0), ..., x_{10}(0)) = (2, 1, 3, 2.2, 2, 1, 4.5, 8, 2, 2)$. The nodes start with the resource amounts $(r_1, r_2, r_3) = (2, 4, 3)$ and the continuous initial conditions $(p_1(0), p_2(0), p_3(0)) = (2, 3, 2)$. During the simulation, two events are generated to test different conditions on the interconnected system: At $t = 3$ agent 7 changes its location from $q_7 = 1$ to $q_7 = 2$, creating the new configuration $(q_1, q_2, ..., q_{10}) = (1, 3, 2, 3, 2, 1, 2, 1, 3, 2)$, and at $t = 6$ the resource at node 3 is changed from $r_3 = 3$ to $r_3 = 2$, so the new resource vector becomes $(r_1, r_2, r_3) = (2, 4, 2)$. Note from this simulation conditions that agents and nodes only visit a subset of the discrete modes in their model: Agents $1, 2, ..., 6, 8, 9, 10$ only visit the mode that corresponds to their location in the graph, which does not change during the test, while agent 7 starts at $MODE : q_7 = 1$ and at $t = 3$ changes to $MODE : q_7 = 2$. Node 1 which initially hosts agent 7 starts the simulation at $MODE : q_1 = (4\ agents, r = 2)$ and at $t = 3$ switches to $MODE : q_1 = (3\ agents, r = 2)$. Node 2, which is the final destination of agent 7, starts the simulation at $MODE : q_2 = (3\ agents, r = 4)$ and at $t = 3$ switches to $MODE : q_2 = (4\ agents, r = 4)$. Finally node 3 starts the simulation at $MODE : q_3 = (3\ agents, r = 3)$ and at $t = 6$ switches to $MODE : q_3 = (3\ agents, r = 2)$. These changes of modes have a direct effect on the continuous dynamics. A mode switch on an agent causes the term $p_{q,i=q_k}$ to change in (19b) (because the rest of the term are identical for all modes int his model). A switch on a node causes $r_{q,i}$ to change in (19a) is because of a change in the resource while causing $(\sum_{\{k:q_k=i\}} x_{q,k})$ to change in (19a) if the switch is caused by a change in the number of agents residing in the node.

B. Results

Given the initial conditions for the continuous states, the locations of the agents in the network, and the resources available at each node as explained in the previous subsection, the state of the system is expected to converge to an asymptotically stable equilibrium point that coincides with the solution of the optimization problem (9)-(10) with our particular choice of utility function (18). In order to obtain such equilibrium point we apply Theorem 2 to (18), obtaining, for each $i \in \mathcal{V}$:

$$x^*_k = \frac{r_i w_k}{\sum_{\{\kappa \in V_i\}} w_\kappa} \quad \forall k \in V_i \tag{20a}$$

$$\lambda^*_k = -\frac{1}{r_i} \sum_{\{\kappa \in V_i\}} w_\kappa \quad \forall k \in V_i \tag{20b}$$

Substituting the values for w_k and r_i given in previous subsection, and considering the three possible discrete configurations, one for each interval between the beginning of the simulation, the events, and the end of the simulation, we obtain the optimal solutions shown in Table I

Time Interval	$t \in I$	$t \in [0, 3)$	$t \in [3, 6)$	$t \in [6, 9]$
Agent Location	$(q_1, q_2, ..., q_{10})$	$(1, 3, 2, 3, 2, 1, 1, 1, 3, 2)$	$(1, 3, 2, 3, 2, 1, 2, 1, 3, 2)$	$(1, 3, 2, 3, 2, 1, 2, 1, 3, 2)$
Resource Availability	(r_1, r_2, r_3)	$(2, 4, 3)$	$(2, 4, 3)$	$(2, 4, 2)$
Optimal Solution	$(x_1^*, ..., x_5^*)$	$(0.47, 1.63, 0.66, 0.81, 2.66)$	$(0.58, 1.63, 0.40, 0.81, 1.60)$	$(0.58, 1.09, 0.40, 0.54, 1.60)$
	$(x_6^*, ..., x_{10}^*)$	$(0.85, 0.38, 0.28, 0.54, 0.66)$	$(1.05, 1.60, 0.35, 0.54, 0.40)$	$(1.05, 1.60, 0.35, 0.36, 0.40)$

TABLE I

OPTIMAL SOLUTION TO THE OPTIMIZATION PROBLEM (9)-(10) WITH UTILITY FUNCTION GIVEN BY (18) FOR EACH CONFIGURATION THAT THE INTERCONNECTED SYSTEM VISITS DURING THE SIMULATION.

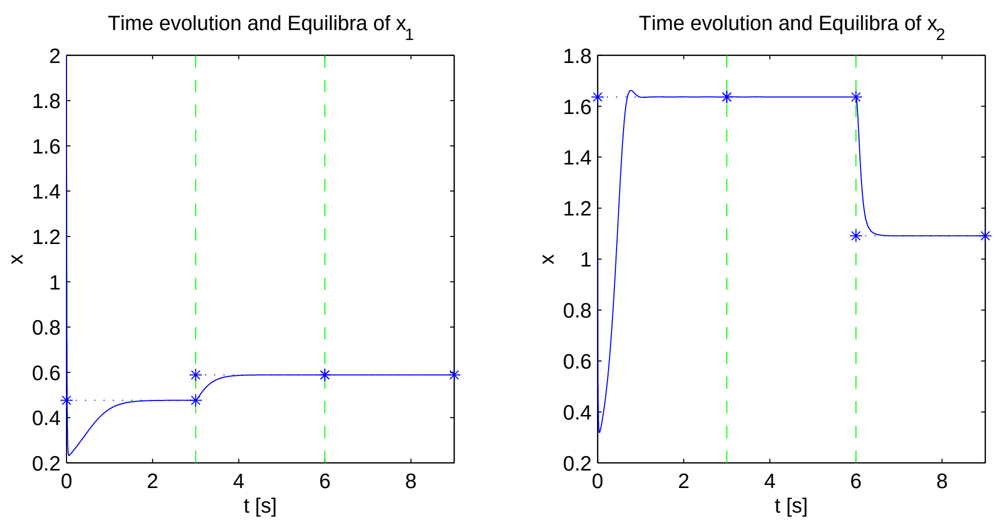

Fig. 5. Dynamic behavior and optimal stable equilibria of agents 1 (left) and 2 (right). The solid (blue) curves show the dynamic behavior of the states. The segmented (green) vertical lines indicate the occurrence of events that change the operating conditions of the system and are summarized in Table I. The dotted (blue) horizontal lines with $*$-marks at the extreme points indicate the optimal solution to the corresponding optimization problem for the system configuration during that time interval, which is found in Table I and is expected to coincide with the equilibrium point where the dynamics approach during such interval.

The results are summarized in Figures 5- 10. The plots in Figures 5-9, show the time evolution of the continuous states of the 10 agents (in order) involved in the test. The vertical segmented lines indicate the time of occurrence of the events that were mentioned in the previous subsection. The horizontal dotted lines with $*$-marks at the extreme points indicate the expected equilibrium points during for each interval between events given by the solution of the optimization problem (9)-(10) with utility function (18), which are shown in Table I. As seen from Figures 5-9,

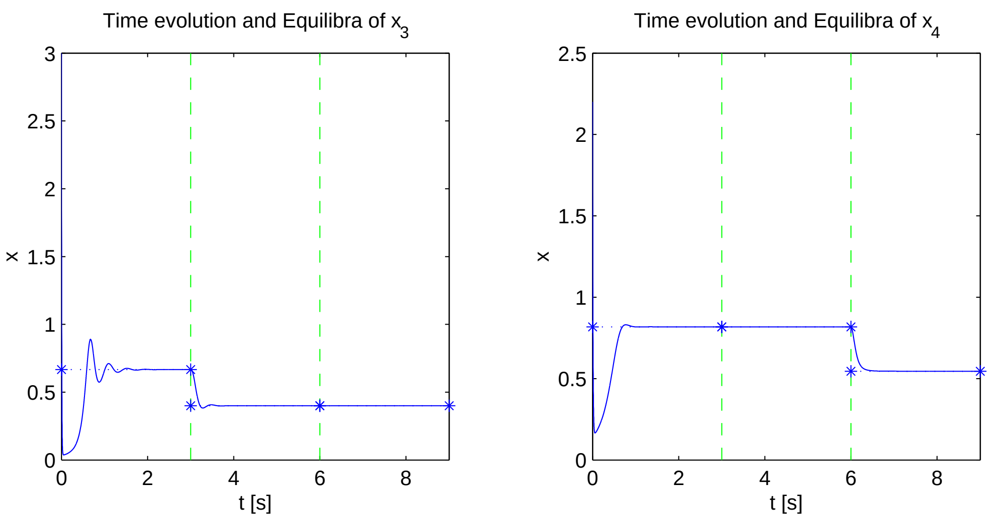

Fig. 6. Dynamic behavior and optimal stable equilibria of agents 3 (left) and 4 (right). The solid (blue) curves show the dynamic behavior of the states. The segmented (green) vertical lines indicate the occurrence of events that change the operating conditions of the system and are summarized in Table I. The dotted (blue) horizontal lines with $*$-marks at the extreme points indicate the optimal solution to the corresponding optimization problem for the system configuration during that time interval, which is found in Table I and is expected to coincide with the equilibrium point where the dynamics approach during such interval.

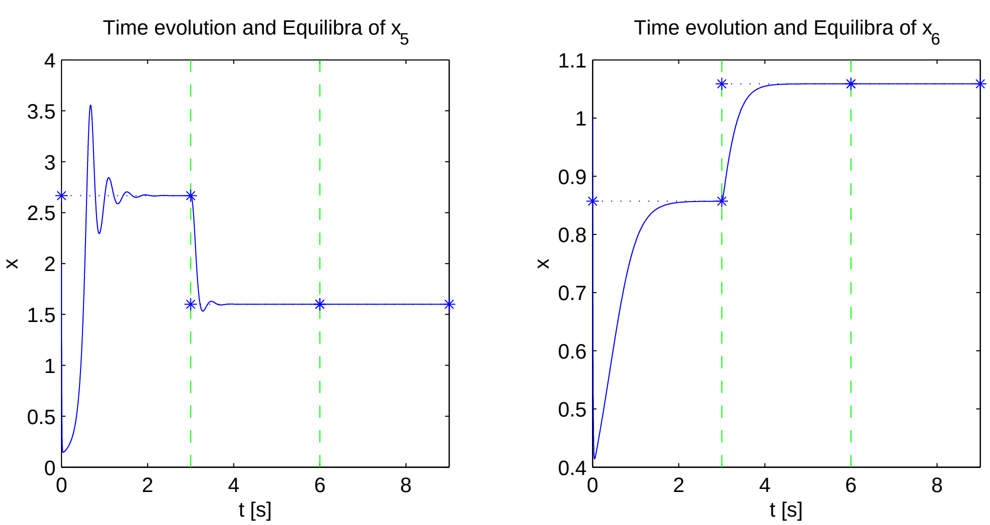

Fig. 7. Dynamic behavior and optimal stable equilibria of agents 5 (left) and 6 (right). The solid (blue) curves show the dynamic behavior of the states. The segmented (green) vertical lines indicate the occurrence of events that change the operating conditions of the system and are summarized in Table I. The dotted (blue) horizontal lines with $*$-marks at the extreme points indicate the optimal solution to the corresponding optimization problem for the system configuration during that time interval, which is found in Table I and is expected to coincide with the equilibrium point where the dynamics approach during such interval.

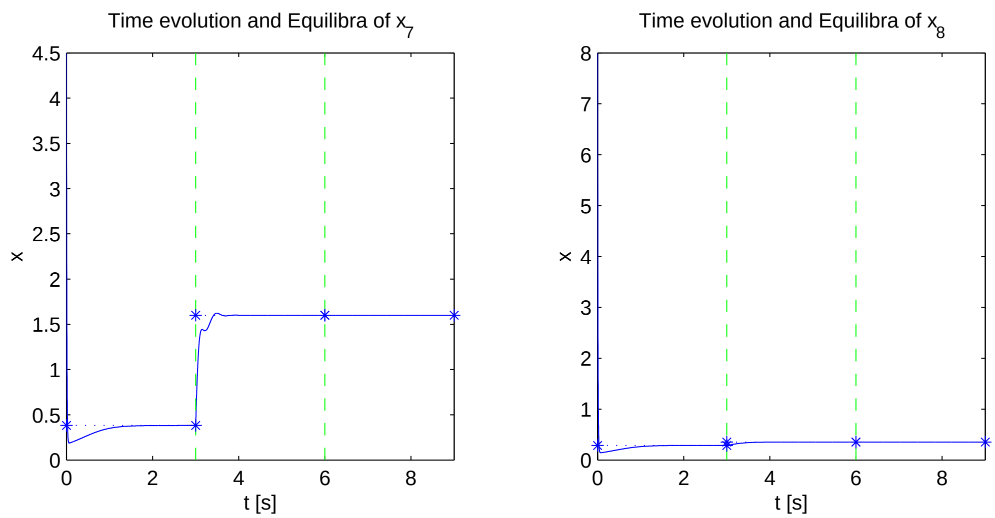

Fig. 8. Dynamic behavior and optimal stable equilibria of agents 7 (left) and 8 (right). The solid (blue) curves show the dynamic behavior of the states. The segmented (green) vertical lines indicate the occurrence of events that change the operating conditions of the system and are summarized in Table I. The dotted (blue) horizontal lines with $*$-marks at the extreme points indicate the optimal solution to the corresponding optimization problem for the system configuration during that time interval, which is found in Table I and is expected to coincide with the equilibrium point where the dynamics approach during such interval.

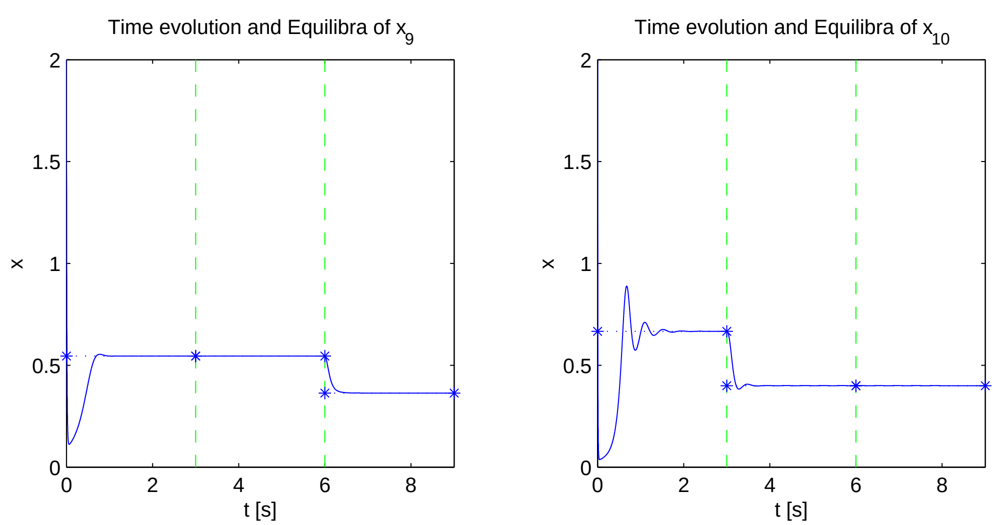

Fig. 9. Dynamic behavior and optimal stable equilibria of agents 9 (left) and 10 (right). The solid (blue) curves show the dynamic behavior of the states. The segmented (green) vertical lines indicate the occurrence of events that change the operating conditions of the system and are summarized in Table I. The dotted (blue) horizontal lines with $*$-marks at the extreme points indicate the optimal solution to the corresponding optimization problem for the system configuration during that time interval, which is found in Table I and is expected to coincide with the equilibrium point where the dynamics approach during such interval.

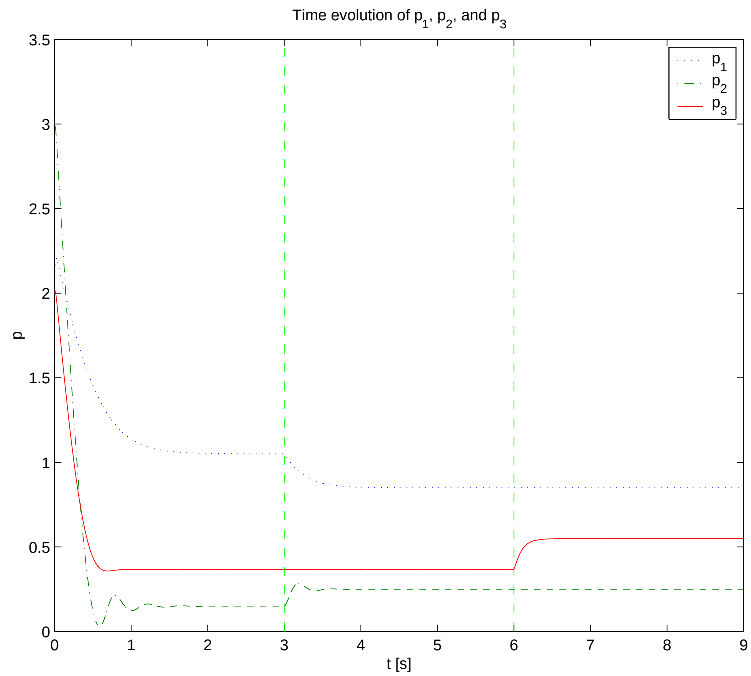

Fig. 10. Dynamic behavior of nodes 1 (dotted blue curve), 2 (segmented green curve), and 3 (solid red curve). The segmented (green) vertical lines indicate the occurrence of events that change the operating conditions of the system and are summarized in Table I.

the states of all the agents converge to a stable equilibrium point on each interval between events. This equilibria coincides with the optimal solution given in Table I.

To see the effects of the events on agents, observe for example the behavior of agents 1 (Figure 5 left), 6 (Figure 7 right), 7 (Figure 8 left), and 8 (Figure 8 right), which start the simulation at node 1. The states of these agents converge to an equilibrium point that coincides with that given in Table I column 1 before $t = 3$. Then at $t = 3$ agent 7 switches from node 1 to node 2. This releases some resources from node 1, and such resources may be allocated to the remaining agents in the node (1, 6, 8). As a consequence, the state (x_1, x_6, x_8) is no longer at an optimal configuration, and thus becomes unstable. The new configuration, however, has a new optimal stable point, given in Table I column 2, which is reached by agents state (x_1, x_6, x_8) after a transient period before $t = 6$.

A similar behavior is observed in the time evolution of the states of agents 3 (Figure 6 left), 5 (Figure 7 left), and 10 (Figure 9 right), which start the simulation at node 2. After having converged (before $t = 3$) to the optimal stable equilibrium point given in Table I column 1, agent 7 arrives to node 2 at $t = 3$. This generates a new system composed by the dynamics of node 2, and agents 3, 5, 7, and 10 that after a transient period (before $t = 6$) converge to the new equilibrium point given in Table I column 2. The effect of the movement of agent 7 can also be observed in the state of the nodes in Figure 10. Both states of nodes 1 and 2 reach a stable equilibrium point before $t = 3$, then the event changes the operating point of the systems at nodes 1 and 2 which is seen in the new transient period towards a new stable equilibrium.

Note that the event that moves agent 7 from node 1 to 2 does not affect the dynamics of node 3 (Figure 10) and of the agents residing on it i.e., agents 2 (Figure 5 right), 4 (Figure 6 right), and 9 (Figure 9 left).

The second event that changes the amount of resources at node 3 from $r_3 = 3$ to $r_3 = 2$ only affects the dynamics of the interconnected system at node 3 formed by the node's dynamics (Figure 10) and agents 2 (Figure 5 right), 4 (Figure 6 right), and 9 (Figure 9 left). In these figures, it can be observed that the agents and the node converge (before $t = 6$) to an equilibrium point given in Table I column 2, and after the event at $t = 6$ they switch their dynamics to approach a new equilibrium point given in Table I column 3.

To summarize, we observe as expected from Theorem 3, that each different configuration of agents locations and amounts of resources, with dynamics given in Propositions 2 and 3 have a stable equilibrium point which coincides with the solution to its corresponding optimization problem. We have also observed that agent-related events affect the dynamic behavior and optimal solution of the systems at both the origin and destination nodes, while node-related events only affect the condition at the local node. This happens in part because we have not included discrete transition rules in the agent's and node's hybrid models. We expect this to change when the design is complete.

VII. Conclusion and Future Work

The problem studied in this paper considers agents moving on a network of discrete locations. The agents need resources in order to perform some tasks, and such resources are provided by the environment. The agents' objective is to obtain the best possible resources from the network in order to maximize their satisfaction measured using a utility function. The objective of the multi-agent system, however, is to achieve a group behavior such that the utilization of the network resources is globally optimized.

The overall behavior of the system includes resource allocation, movement of agents between discrete locations, and a change of network conditions. Therefore, both agents and nodes need to be described using continuous (resource allocation) and discrete dynamics (agents movement and varying network conditions) that can be captured by a hybrid model [4], [21]. The hybrid model we propose is incomplete, and this paper outlines how to obtain the continuous dynamics only, leaving the discrete dynamics unspecified.

The continuous dynamics are designed using results borrowed from Internet congestion control algorithms [1], [17], [19], [36], [41]. This is done by obtaining an optimization problem that is equivalent to the multi-agent system overall objective, and then using the results in [19], [36] to obtain a precise dynamical description of the continuous dynamics of nodes and agents. This model forms an interconnected system for each possible configuration of agents and nodes that is globally asymptotically stable by design, and that optimizes the usage of resources locally within each node.

The discrete dynamics are a key factor to achieving global optimization of resource utilization in the network. The design of this part of the model is expected to take on several analytical steps. The first step which is already being pursued is to apply an abstraction procedure [2], [30] to the continuous dynamics of the system, in order to obtain a simplified, but still meaningful description of the dynamic behavior of the interconnected system. To obtain this abstract description note that since each possible configuration of the system generates a stable system, so it can be substituted by its unique stable equilibrium point given by the solution of the optimization problem (9)-(10) in Theorem 2. This may be done using a similar procedure to that discussed in [30]. Then, with a discrete description of the interconnected system available, we expect to be able to design simple discrete transition rules that achieve a global optimization of the utilization of resources in the network, or to obtain a suboptimal solution.

REFERENCES

[1] T. Alpcan and T. Başar. A utility-based congestion control scheme for internet-style networks with delay. In *Proc. of the IEEE Infocom*, volume 3, pages 2039–2048, San Francisco, CA, USA, April 2003.

[2] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[4] M. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, Jan. 1998.

[5] L. Chaimowicz, N. Michael, and V. Kumar. Controlling swarms of robots using interpolated implicit functions. pages 2498–2503, Barcelona, Spain, April 2005.

[6] J. Clarck and R. Fierro. Cooperative hybrid control of robotic sensors for perimeter detection and tracking. In *Proceedings of the American Control Conference*, pages 3500–3505, Portland, OR, USA, June 2005.

[7] D. Clark, C. Partridge, J. Ramming, and J. Wroclawski. A knowledge plane for the internet. In *In Proc. of the SIGCOMM'03*, pages 3–10, Karlsruhe, Germany, Aug. 2003. ACM 1-58113-735-4/03/0008.

[8] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley and Sons, 1997.

[9] J. Cortez and F. Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, 2005.

[10] M. Egerstedt and X. Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, Dec. 2001.

[11] J. Finke, K. Passino, and A. Sparks. Stable task load balancing strategies for cooperative control of networked autonomoues vehicles. *IEEE Transactions on Control Systems Technology*, 14(5):789–803, September 2006.

[12] V. Gazi and M. Passino. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–697, Apr. 2003.

[13] B. Ghosh, A. Polpitiya, and W. Wang. Bio-inspired networks of visual sensors, neurons and oscillators. *Proceedings of the IEEE*, 95(1):188–214, January 2007.

[14] I. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Journal on Optimization and Engineering*, 3(3):227–252, Sept. 2002.

[15] A. Jadbabaie, J. Lin, and S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

[16] H. Jerez, C. Abdallah, and J. Khoury. A mobile transient network architecture. Pre-print available at `http://hdl.handle.net/2118/hj_tran_06`, 2006.

[17] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[18] M. Khatir and E. Davison. Decentralized control of a large platoon of vehicles operating on a plane with steering dynamics. In *Proceedings of the American Control Conference*, pages 2159–2165, Portland, OR, USA, June 2005.

[19] S. Liu, T. Basar, and R. Srikant. Controlling the internet: A survey and some new results. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3048–3057, Maui Hawaii, USA, December 2003.

[20] Y. Liu, K. Passino, and M. Polycarpou. Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology. *IEEE Transactions on Automatic Control*, 48(1):76–95, Jan. 2003.

[21] J. Lygeros. Lecture notes on hybrid systems. Notes for an ENSIETA workshop, February–June 2004.

[22] J. Marshall, M. Broucke, and B. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963–1974, Nov. 2004.

[23] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, Feb. 2005.

[24] P. Ogren, M. Egerstedt, and X. Hu. A control lyapunov function approach to multiagent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, Oct. 2002.

[25] S. Oh, L. Schenato, P. Chen, and S. Sastry. Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments. *Proceedings of the IEEE*, 95(1):234–254, January 2007.

[26] R. Olfati-Saber. Consensus problems in networks of agents with switching topology and time delay systems. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, Sept. 2004.

[27] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

[28] A. Pant, P. Seiler, and K. Hedrick. Mesh stability of look-ahead interconnected systems. *IEEE Transactions on Automatic Control*, 47(2):403–407, Feb. 2002.

[29] J. Piovesan, C. Abdallah, H. Tanner, H. Jerez, and J. Khoury. Resource allocation for multi-agent problems in the design of future communication networks. UNM Technical Report EECE-TR-07-001, Dept. Electrical and Computer Engineering, Univeristy of New Mexico, Albuquerque, NM, 87131, April 2007. Available at `http://hdl.handle.net/1928/2973`.

[30] J. Piovesan, H. Tanner, and C. Abdallah. Discrete asymptotic astractions of hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 917–922, San Diego, CA, USA, December 2006.

[31] S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley and Sons, Inc., third edition, 1996.

[32] W. Ren, R. Beard, and E. Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the American Control Conference*, pages 1859–1864, Portland, OR, USA, June 8-10 2005.

[33] W. Ren, R. Beard, and E. Atkins. Information consensus in multi-vehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, April 2007.

[34] R. Sandoval-Rodriguez, C. Abdallah, P. Hokayem, E. Schamiloglu, and R. Byrne. Robust mobile robotic formation control using internet-like protocols. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5109–5112, Maui, Hawaii, USA, Dec 2003.

[35] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1998.

[36] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[37] S. Swaroop and J. Hedrick. String stability of interconected systems. *IEEE Transactions on Automatic Control*, 41(3):349–356, Mar. 1996.

[38] H. Tanner, A. Jadbabaie, and G. Pappas. Stable flocking of mobile agents, part I: Fixed topology. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2010–2015, Maui, Hawaii, USA, Dec 2003.

[39] H. Tanner, A. Jadbabaie, and G. Pappas. Stable flocking of mobile agents, part II: Dynamic topology. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2016–2021, Maui, Hawaii, USA, Dec 2003.

[40] H. Tanner, G. Pappas, and V. Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, June 2004.

[41] J. Wen and M. Arcak. A unifying passivity framework for network flow control. In *Proc. of the IEEE Infocom*, volume 2, pages 1156–1166, San Francisco, CA, USA, April 2003.

[42] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. on Communications*, 52(7):1136–1144, July 2004.

[43] B. Young, R. Beard, and J. Kelsey. A control scheme for improving multi-vehicle formation maneuvers. In *Proceedings of the American Control Conference*, volume 2, pages 704–709, Arlington, VA, USA, June 2001.