A network of agents in which agents with a diverse set of resources or capabilities interact and coordinate with each other to accomplish various tasks constitutes a heterogeneous cooperative network. In this paper, we investigate heterogeneity in terms of resources allocated to agents within the network. The objective is to distribute resources in such a way that every agent in the network should be able to utilize all these resources through local interactions. In particular, we formulate a graph coloring problem in which each node is assigned a subset of labels from a labeling set, and a graph is considered to be completely heterogeneous whenever all the labels in the labeling set are available in the closed neighborhood of every node. The total number of different resources that can be accommodated within a system under this setting depends on the underlying graph structure of the network. This paper provides an analysis of the assignment of multiple resources to nodes and the effect of these assignments on the overall heterogeneity of the network.

1. Introduction

Heterogeneity has emerged as an important aspect of multiagent and cooperative networks in which agents with assorted functionalities and capabilities interact with each other to accomplish complex tasks. Agents with various capabilities and properties when integrated together in the framework of heterogeneous networks, become specialized to achieve sub-goals which may not be accomplished by a team of homogeneous agents efficiently. Several applications of such heterogeneous systems have been studied in the literature. For instance, it is shown in [1] that the reliability and lifetime of a sensor network can be increased by introducing nodes in the network that are different in terms of power consumption and communication capabilities. In [2], it is shown that heterogeneity can be exploited to reduce the number of sensors required in a sensor network without compromising on
the coverage and the broadcast reach of the network. Heterogeneity among agents is explored to make a decentralized system more stable and efficient in [3]. Several other applications are also reported in other domains including multirobot systems (e.g., [4, 5, 6, 8]), and wireless sensor networks (e.g., [9, 10]) to name a few.

One of the challenges in heterogeneous networks is to optimally distribute agents having different capabilities and resources within a network. A part of the challenge is to articulate the notion of heterogeneity in order to guide the analysis and design of heterogeneous networks in a systematic way. Heterogeneity in cooperative networks can be understood along a number of dimensions. We can broadly classify the studies in this area into two categories; one that quantify heterogeneity from agents’ perspectives, including functional or behavioral dissimilarities among individual agents (e.g., [11, 12, 13, 14]), and the relative number of non-homogeneous agents (e.g., [15, 16, 17]); second that measures heterogeneity in terms of the underlying graph structure of the network with an aim to quantify degree distribution among the nodes while treating all the nodes similar\(^1\) (e.g., [18, 19, 20]). Despite their merits, all these studies address one of the two attributes of a cooperative network at a time, either agents’ classification or the topological properties of the underlying graph structure. However, to obtain a holistic overview of heterogeneity, a unified framework incorporating agents’ classification as well as network topology is needed.

In this paper, we extend our initial work in [7, 8], and investigate a resource assignment problem over a graph in the context of heterogeneous cooperative networks. The objective is to provide a framework for a network topology based characterization of heterogeneity in cooperative networks while distinguishing among agents. We investigate interactions among various types of agents, which are distributed throughout a heterogeneous network, by formulating a graph coloring problem. A network is modeled by a graph, in which vertices represent agents and edges abstract interactions among agents. Vertices corresponding to agents of a similar type are assigned a distinct color. An agent interacts locally with other types of agents to perform some task. The goal is to maximally exploit heterogeneous resources available within the network through these local interactions. Using the graph coloring formulation, we examine an inter-relationship between network topology and distribution of agents with various capabilities in heterogeneous networks. This framework also provides a way to characterize various network topologies in terms of their capabilities to incorporate heterogeneous entities under constraints on interactions among various types

\(^1\)I.e., there is no distinction among the nodes to account for their functional or behavioral differences.
of agents. The role and significance of individual agents and interactions on
the overall heterogeneity of the network is also explored.

The paper is organized as follows: In Section 2, a graph coloring based
model of heterogeneous networks is given. Using this model, distribution of
resources among nodes within a network is analyzed in Section 3. In Sec-
tion 4, the issue of maximum number of resources that can be incorporated
within a network under the constraints of the model is addressed. A re-
source assignment problem in $R$-disk proximity graphs, which is a widely
used inter-connection model in multiagent, multirobot, and sensor networks
is investigated in Section 5. The paper is concluded in Section 6 along with
some future directions of the work.

2. Graph coloring based model of heterogeneous networks

Graph theoretic tools are frequently applied to model and analyze various
cooperative networks including sensor networks, multiagent, and multirobot
networks. In this paper, a network is modeled by a graph $G(V, E)$ in which
the vertex set $V$ represents agents and the edges in the edge set $E$ corre-
spond to interactions among agents. In the case of heterogeneous networks
in which agents may be different from each other in terms of their resources
or capabilities (for instance, sensing, actuation, dynamics, capabilities, re-
sources, hardware, or software etc.), heterogeneity is modeled by associating
a unique color (or label) with each resource type available in the network.
Moreover, all of the vertices in an underlying graph of the network are as-
signed colors (or labels) in accordance with the resources contained by the
respective agents. A vertex may have multiple labels if the correspon-
ding agent has multiple types of resources. In heterogeneous cooperative net-
works, agents interact and utilize each others’ resources to accomplish tasks
such as surveillance, coverage, data analysis, and computation to name a few.
The availability of resources of different types in the local neighborhood of
an agent determines the agent’s overall capability to perform various tasks.
Correlation between network topology and distribution of agents with vari-
ous types of resources can be studied by casting a graph coloring formulation
of the problem in which vertices are assigned labels in accordance with the
resources contained by the agents.

Throughout this paper, a graph $G(V, E)$ with a vertex set $V$ and an
edge set $E$, is a simple undirected graph. An edge between nodes $v_i$ and $v_j$
is denoted by $v_i \sim v_j$. The open neighborhood of a vertex $v \in V(G)$, denoted
by $N(v)$, is the set of vertices adjacent to $v$. Its closed neighborhood, denoted
by $N[v]$, is $N(v) \cup \{v\}$. The degree of a vertex $v$, $\text{deg}(v)$, is the cardinality
of $\mathcal{N}(v)$. The minimum degree of a graph, $\delta(G)$, is $\min\{\deg(v) \mid v \in V\}$ and the maximum degree of a graph, $\Delta(G)$, is $\max\{\deg(v) \mid v \in V\}$. The terms color and label are used interchangeably.

Let $r = \{1, 2, \cdots, r\}$ be a set of labels representing $r$ different types of resources (or capabilities) available within a heterogeneous network. Furthermore, vertices are assigned labels according to the map, $f: V \rightarrow 2^r$.

Here, $2^r$ is the set of all subsets of $r$. $f(v)$ is a subset of resources contained by agent $v$, which interacts and utilizes the resources of its neighbors to perform some task. Thus, the heterogeneity of an agent $v$ within the network depends on the resources (or capabilities) contained by $v$ and its neighbors, and is defined as

$$H(v) = \bigcup_{u \in \mathcal{N}(v)} f(u)$$

Moreover, an agent is maximally heterogeneous within the network whenever $H(v) = r$, as $v$ can exploit all different functionalities and resources available within the network by interacting with its neighbors. Thus, from the network topological viewpoint, a completely heterogeneous graph is defined as

**Definition 2.1.** A graph $G(V, E)$ in which every $v \in V$ is assigned a subset of labels $f(v)$ from the set $r = \{1, 2, \cdots, r\}$, is completely heterogeneous with $r$ labels if

$$H(v) = r, \quad \forall v \in V$$

Note that in a completely heterogeneous network as defined above, every agent is capable of exploiting a complete set of resources and functionality available within the network to perform various tasks by working in conjunction with its neighbors.

### 2.1. Examples

Consider an industrial location in which some manufacturing process depends on environmental conditions, including temperature ($t$), light ($\ell$), humidity ($h$), and air pressure ($p$). A specific environmental condition that depends on all of the above parameters is needed to be maintained to get a desired yield. Let this condition be denoted by $\omega(t, \ell, h, p)$. Sensors for each of the above parameters $t, l, h$ and $p$ are mounted at various data collection points, which are inter-connected with each other and exchange data.
Figure 1: $G_1$ is a completely heterogeneous graph with four labels \{t, h, l, p\}. In $G_2$, label $p$ is missing from the closed neighborhoods of the circled nodes. Thus, $G_2$ is not completely heterogeneous.

The environmental condition $\omega(t, \ell, h, p)$ is computed at every such data collection point. The distribution of sensors with assorted sensing capabilities constitutes a heterogeneous network. It is further assumed that owing to some constraints (e.g., hardware, power, economical etc.), only a subset of sensors can be mounted at each data collection point. Since all four parameters are needed for the computation of $\omega(t, \ell, h, p)$, sensors need to be distributed in such a way that all of the four types of sensors are available in the closed neighborhood of every data collection point. In other words, underlying graph of the network needs to be completely heterogeneous with the set of labels \{t, l, h, p\} as shown in Fig. 1.

As another example, consider a society of some ‘species’ in which each member of the society has been assigned a specific role. Some members are food providers, some are shelter providers, while others hold the task of providing security to the members they interact with. In such a society, every member depends on other members to ensure the availability of all the facilities. For instance, a food providing member must interact with a shelter provider and a security provider for shelter and security respectively. This kind of cooperation constitutes a heterogeneous network in which availability of all the resources to each member of the society is possible if the underlying graph of the network is completely heterogeneous with three distinct labels.

2.2. Major issues related to the notion of completely heterogeneous graph

In the context of heterogeneous cooperative networks, the notion of completely heterogeneous graph has three major aspects. First, given a colored
Figure 2: $G_1$ can not be made completely heterogeneous with five labels by assigning at most two distinct labels to each node. In the case of $G_2$, although it is possible to assign two colors to each vertex and obtain a completely heterogeneous graph with five colors, under the given labeling $v_6$ is missing label 2 in its closed neighborhood. In $G_3$, each vertex has a complete set of five labels in its closed neighborhood.

In this section, distribution of resources or capabilities in heterogeneous cooperative networks is analyzed using the model introduced in Section 2. Various tasks performed by individual agents in such networks depend on the resources available locally to agents. Thus, information regarding missing
resources in the closed neighborhoods of agents along with the interactions needed to make these resources available is crucial. In this section, these issues are addressed.

Given a graph with \( n \) nodes, in which every node \( v_i \) is assigned a subset of colors (labels) \( f(v_i) \) from the set of colors \( r = \{1, 2, \cdots, r\} \). We define a color matrix, denoted by \( C \), as a binary matrix with dimensions \( n \times r \) as follows:

\[
C_{ij} = \begin{cases} 
1 & \text{if } j \in f(v_i), \text{ where } f(v_i) \subseteq r \\
0 & \text{otherwise.} 
\end{cases}
\]

In (2), \( f(v_i) \) indicates the colors assigned to the vertex \( v_i \). The column index of \( C \) indicates the color, thus \( C_{ij} = 1 \) means that color \( j \) has been assigned to the vertex \( v_i \).

Using the color matrix \( C \), and the adjacency matrix \( A \) of the graph, another integer matrix of dimensions \( n \times r \) is defined and named as the color distribution matrix as follows:

\[
\Phi = AC,
\]

where \( \mathcal{A} = (A + I) \). Here, \( I \) is the identity matrix of dimensions \( n \times n \).

The color distribution matrix gives information regarding the distribution of various colors within the network. In fact, it describes the exact number of different colors available in the closed neighborhood of any node.

**Lemma 3.1.** \( \Phi_{ij} \) is the number of nodes with the color \( j \) in the closed neighborhood of node \( v_i \).

**Proof.** The entries in the \( i^{th} \) row matrix \( \mathcal{A} \), denoted by \( \mathcal{A}_i \), are 1 only for the vertices in \( \mathcal{N}[v_i] \) and 0 otherwise. The entries in the \( j^{th} \) column of \( C \), denoted by \( C_j \), are 1 only for the vertices with the color \( j \) and 0 otherwise. Thus, \( \mathcal{A}_iC_j \) is the number of vertices that have color \( j \) in the closed neighborhood of vertex \( v_i \).

The color distribution matrix turns out to be a useful object in characterizing the distribution of colors to vertices in a graph. For instance, it allows to determine the extra edges required to transform a given labeling of a graph into a completely heterogeneous one. In fact, \( \Phi_{ij} = 0 \) means that \( v_i \) is missing color \( j \) in its closed neighborhood. Thus, an extra edge is needed to connect \( v_i \) to some \( v_u \) with a color \( j \). Upper and lower bounds on the number of extra edges required to get a completely heterogeneous graph with \( r \) labels from a given coloring of \( G \) are presented in the following result.
Theorem 3.2. Let \( s \) be the maximum number of labels assigned to any vertex in a graph \( G \). The number of extra edges \( E \), needed to get a completely heterogeneous graph with \( r \) labels from a given coloring of \( G \) is
\[
\left\lceil \frac{z(\Phi)}{2s} \right\rceil \leq E \leq z(\Phi),
\]
where \( z(\Phi) \) is the number of 0’s in the color distribution matrix \( \Phi \).

Proof. Let \( v_i \sim v_j \) be an extra edge connecting vertex \( v_i \) with colors \( \kappa_1, \ldots, \kappa_s \), to vertex \( v_j \) with colors \( \tau_1, \tau_2, \ldots, \tau_s \). Since every vertex can have at most \( s \) distinct colors, \( v_i \sim v_j \) can add at most \( s \) missing colors in \( \mathcal{N}[v_i] \) and also at most \( s \) missing colors in \( \mathcal{N}[v_j] \). This is possible whenever \( v_i \) is missing colors \( \tau_1, \tau_2, \ldots, \tau_s \) in \( \mathcal{N}[v_i] \) given by \( \Phi_{i\tau} = 0 \), \( \forall \tau \in \{\tau_1, \ldots, \tau_s\} \), and \( v_j \) is missing \( \kappa_1, \kappa_2, \ldots, \kappa_s \) in \( \mathcal{N}[v_j] \), given by \( \Phi_{j\kappa}, \forall \kappa \in \{\kappa_1, \ldots, \kappa_s\} \).

In this case, \( v_i \sim v_j \) edge will change \( 2s \) zero entries in the \( \Phi \) matrix to ones. In any other case, i.e., \( v_i \) has at least one of the \( \tau_1, \tau_2, \ldots, \tau_s \) colors in its closed neighborhood or \( v_j \) has at least one of the \( \kappa_1, \kappa_2, \ldots, \kappa_s \) colors in \( \mathcal{N}[v_j] \), the number of zeros in \( \Phi \) that will be converted to 1 will be less than \( 2s \). Thus, \( \left\lceil \frac{z(\Phi)}{2s} \right\rceil \leq E \).

The upper bound is straightforward as \( \Phi_{i\tau} = 0 \) means that \( v_i \) is missing a color \( \tau \) in \( \mathcal{N}[v_i] \), and the color \( \tau \) can always be made available in \( \mathcal{N}[v_i] \) by the addition of a single edge \( v_i \sim v_j \), where \( v_j \) is any vertex with the color \( \tau \).

As an illustration, consider \( G \) shown in Fig. 3. Every node has at most two labels from the set of five labels given by \( \{1, 2, 3, 4, 5\} \). The corresponding \( C \) and \( \Phi \) matrices are,

\[
C = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0
\end{pmatrix}, \quad \Phi = \begin{pmatrix}
1 & 1 & 1 & 1 & 2 \\
2 & 2 & 1 & 1 & 2 \\
1 & 1 & 2 & 2 & 2 \\
1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 1
\end{pmatrix}.
\]

Since \( \Phi_{43} = \Phi_{51} = 0 \), \( v_4 \) is missing label 3 in \( \mathcal{N}[v_4] \) and \( v_5 \) is missing label 1 in its closed neighborhood. By adding \( \mathcal{E} \) number of edges, where \( 1 \leq \mathcal{E} \leq 2 \) (by Theorem 3.2), \( G \) can be transformed into a completely heterogeneous graph. Note that by adding a single edge, \( v_4 \sim v_5 \), a completely heterogeneous graph with five labels is obtained.
Characterizing heterogeneity in cooperative networks

3.1. Redundant edges

In dynamic networks, some edges may be lost. These edge deletions may take away the availability of certain resources in the neighborhood of an agent. Thus, we need to characterize edges whose deletion is not critical in the sense that their removal will preserve the number of resources available in the neighborhood of any agent. Let us define the deficiency of a node $v$ as the number of colors from the coloring set $\{1, 2, \cdots, r\}$ that are missing in $N[v_i]$. Similarly, deficiency of a network is the sum of all the node deficiencies. Based on this notion, we define a redundant edge to be the one whose deletion does not increase the deficiency of the network.

$\Phi_{ij} = 1$ means that $v_i$ has only one neighbor with the color $j$. Thus, an edge between $v_i$ and the $j$ colored node is not redundant. Similarly, $\Phi_{ij} > 1$ implies that $v_i$ has multiple nodes with the color $j$ in $N[v_i]$. As a result, there may be a redundant edge between $v_i$ and some of its neighbors.

**Theorem 3.3.** Let $v_i$ be a node with colors $\kappa_1, \kappa_2, \cdots, \kappa_s$, and $v_j$ be its neighbor with colors $\tau_1, \tau_2, \cdots, \tau_s$. An edge $v_i \sim v_j$ is redundant if and only if $\Phi_{i\tau_1}, \Phi_{i\tau_2}, \cdots, \Phi_{i\tau_s}$, and $\Phi_{j\kappa_1}, \Phi_{j\kappa_2}, \cdots, \Phi_{j\kappa_s}$ are all greater than one simultaneously.

**Proof.** Let $v_i \sim v_j$ be a redundant edge, then by definition $v_i$ has at least two neighbors for each of the colors $\tau_1, \tau_2, \cdots, \tau_s$ in $N[v_i]$. In other words, $\Phi_{i\tau_1}, \Phi_{i\tau_2}, \cdots, \Phi_{i\tau_s}$ are all greater than 1. Similarly, for $v_j$, the redundancy of a $v_i \sim v_j$ edge implies that for each of the colors, $\kappa_1, \kappa_2, \cdots, \kappa_s$, vertex $v_j$ has at least two neighbors in $N[v_j]$, implying that $\Phi_{j\kappa_1}, \Phi_{j\kappa_2}, \cdots, \Phi_{j\kappa_s}$ are all greater than 1.
Figure 4: Every vertex is assigned two distinct labels from the set \{1, 2, \ldots, 5\}. \(v_2 \sim v_3\) edge is redundant and removing this edge will not increase the deficiency of any node.

If \(v_i \sim v_j\) is not redundant, then at least one of the following is true. (a) there exists a \(\tau \in \{\tau_1, \tau_2, \ldots, \tau_s\}\), such that \(v_i\) has only \(v_j\) as a \(\tau\) colored vertex in \(N[v_i]\), i.e., \(\Phi_{i\tau} = 1\) for some \(\tau \in \{\tau_1, \tau_2, \ldots, \tau_s\}\). (b) there exists a \(\kappa \in \{\kappa_1, \kappa_2, \ldots, \kappa_s\}\), such that \(v_j\) has only \(v_i\) as a \(\kappa\) colored vertex in \(N[v_j]\), i.e., \(\Phi_{j\kappa} = 1\) for some \(\kappa \in \{\kappa_1, \kappa_2, \ldots, \kappa_s\}\).

In both cases, \(\Phi_{i\tau_1}, \Phi_{i\tau_2}, \ldots, \Phi_{i\tau_s}\) and \(\Phi_{j\kappa_1}, \Phi_{j\kappa_2}, \ldots, \Phi_{j\kappa_s}\) are not all greater than 1 simultaneously, proving the required result.

As an example, consider the graph in Fig. 4. Note that \(v_2\) has labels 4 and 5, while \(v_3\) has labels 2 and 5. In the color distribution matrix, \(\Phi_{22}, \Phi_{25}, \Phi_{34}, \text{ and } \Phi_{35}\) are all greater than 1. By Lemma 3.3, \(v_2 \sim v_3\) edge is redundant and its deletion is not increasing the deficiency of any node in the network.

### 3.2. Most deficient color and the effect of node deletion

The **most deficient color** in the network is the one that is missing from the closed neighborhood of the maximum number of vertices in \(G\). The \(j^{th}\) column of \(\Phi\) tells about the availability of the color \(j\) in the closed neighborhood of all the vertices in \(G\). By Lemma 3.1, \(\Phi_{ij} = 0\) means \(v_i\) does not have a color \(j\) in \(N[v_i]\). Thus, the column index of \(\Phi\) with the **maximum number of zeros** will be the most deficient color in the given labeling of \(G\).

The deletion of a vertex from a graph may increase the deficiency of the remaining vertices. If vertex \(v_i\) is the only vertex with the color \(\kappa\) in the closed neighborhood of vertex \(v_j\), deleting \(v_i\) will make \(v_j\) deficient in \(\kappa\). However, if \(v_j\) has multiple vertices with the color \(\kappa\) in \(N[v_j]\), which is indicated by \(\Phi_{j\kappa} \geq 2\), removal of \(v_i\) will not increase the deficiency of \(v_j\) for the color \(\kappa\). Using this observation, we can write a matrix \(U\) in which \(U_{ik}\)
Figure 5: A graph $G$ along with the labeling of its vertices from the set \{1, 2, \cdots, 5\}. Removal of $v_1$ makes $v_2$ deficient in color 3 and $v_3$ deficient in color 1. Thus, deletion of $v_1$ will increase the deficiency of the network by two, which is also indicated by the row sum of the first row of $U$.

is the number of vertices that will become deficient in the color $\kappa$ upon the deletion of $v_i$ from the graph $G$. If $C$ is the color matrix, and $\Phi$ be the color distribution matrix, then

$$U_{i\kappa} = \begin{cases} \left| \{v_j : v_j \in \mathcal{N}(v_i), \text{ and } \Phi_{j\kappa} = 1\} \right|, & \text{if } C_{i\kappa} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

$U$ is an integer matrix with dimensions $n \times r$, where $n$ is the total number of vertices in $G$, and $r$ is the total number of colors in the labeling of $G$. The $i^{th}$ row sum of $U$ indicates the increase in the deficiency of the network as a result of the deletion of $v_i$. If we define a critical node as the one whose removal from the network maximizes the increase in the deficiency of the remaining network, then the row index of $U$ corresponding to the maximum row sum indicates the most critical vertex.

As an example, consider $G$ shown in Fig. 5. The $U$ matrix for $G$ is

$$U = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

$U_{13} = 1$, indicates that a single vertex will become deficient in color ‘3’ upon the removal of $v_1$ from $G$. Note that both $v_2$ and $v_3$ are critical vertices here as both second and third rows have the maximum row sum.
4. Heterogeneity in terms of the maximum number of resources available within the network

In this section, we address the second issue highlighted in Section 2.2, which is related to the maximum number of resources’ types that can be incorporated within the system under the constraint that every node \( v \) can find every resource type in \( \mathcal{N}[v] \). If \( r \) different types of resources are available within the network and each agent is equipped with at most \( s \) of these resources, then the maximum value of \( r \), denoted by \( r^* \), such that the underlying graph of the network can be made completely heterogeneous with \( r^* \) labels is a crucial attribute of heterogeneous cooperative networks. In fact, in a completely heterogeneous network with \( r^* \) different types of resources, every agent can find all \( r^* \) types of resources in its closed neighborhood to accomplish various tasks. Thus, a higher value of \( r^* \) implies that more types of resources can be made available to agents in a completely heterogeneous network. As a result, agents can perform tasks of higher complexity. It is to be noted that, for a given graph \( G \) and a bound on the number of resources an agent can have, i.e., \( |f(v)| \leq s \), if \( r > r^* \), then it is impossible to distribute resources among the nodes to get a completely heterogeneous graph with \( r \) labels. We utilize the notion of domination in graphs to address this issue. A dominating set is a fundamental object in the field of domination in graphs (see [21] for details).

**Definition 4.1.** *(Dominating Set)* A set \( D \) is a dominating set if for each \( v \in V \), either \( v \in D \), or \( v \) is adjacent to some \( u \in D \).

Note that if \( D \) is a dominating set, then \( \bigcup_{u \in D} \mathcal{N}[u] = V \). The cardinality of the smallest dominating set in a graph is known as the domination number of the graph. A graph can have multiple disjoint\(^2\) dominating sets. A related and important concept is that of the domatic number.

**Definition 4.2.** *(Domatic Number)* The maximum number of disjoint dominating sets in a graph is the domatic number of the graph.

If \( D_i \) is a dominating set of a graph \( G \) and all the vertices in \( D_i \) are assigned a particular label \( i \), then \( i \) is available in the closed neighborhood of every vertex in \( G \). In other words, if \( f(u) = i, \forall u \in D_i \), then \( i \in \mathcal{H}(v), \forall v \in V \). Furthermore, if the domatic number of a graph \( G \) is \( \gamma \) and every vertex is assigned a single label, then by the definition of domatic number, \( G \) can be completely heterogeneous with at most \( \gamma \) labels, i.e., \( r^* = \gamma \). The notion of so-called \((r, s)\)-configuration [22] defined below is also helpful in this context.

\(^2\)The intersection of distinct dominating sets is empty.
Definition 4.3. \((r,s)\)-Configuration of a Graph) Let \(r = \{1, 2, \cdots, r\}\) be a set of labels (colors). A function
\[
f : V \longrightarrow [r]_s,
\]
where \([r]_s\) is a collection of all \(s\)-subsets of \(r\), is called an \((r,s)\)-configuration of a graph \(G\), whenever \(\bigcup_{u \in N[v]} f(u) = r, \forall v \in V\).

Thus, the maximum value of \(r\) in an \((r,1)\)-configuration of a graph is the domatic number of the graph. Moreover, if \(\gamma\) is the domatic number of a graph, it is obvious that for \(s > 1\), there always exists an \((r,s)\)-configuration for \(r = s\gamma\). In other words, each vertex in the graph can always be labeled with at most \(s\) colors such that the overall graph is completely heterogeneous with \(s\gamma\) labels. However, there are graphs for which \((r,s)\)-configurations exist for \(r > s\gamma\). For example, cycle graphs \(C_n\) in which \(n\) is not a multiple of 3, have a domatic number of 2, but \((5,2)\)-configurations of such graphs exist [22]. Here, we present a sufficient condition for a graph to have an \((r,s)\)-configuration with \(r = s\gamma + \lfloor \frac{s}{2} \rfloor\). A labeling scheme to obtain such a configuration can also be derived using this condition.

We begin by defining some terms that will be used to prove Theorem 4.1, which is the main result of this section.

Definition 4.4. (Minimal Partition of \(G\)) Let \(G\) be a graph with domatic number \(\gamma\), and vertex set \(V\). A minimal partition of \(G\), denoted by \(\Pi\), is a partitioning of \(V\) into \(\gamma + 1\) disjoint sets such that,
\[
\Pi = D_1 \cup D_2 \cup \cdots D_\gamma \cup V_\Pi,
\]
where \(D_i\) is a minimal dominating set, \(\forall i \in \{1, 2, \cdots, \gamma\}\), and \(V_\Pi = V - (\bigcup_{i=1}^{\gamma} D_i)\) is the set of vertices that are not included in any minimal dominating set \(D_i\).

We term \(V_\Pi\) in (5) as the set of non-critical vertices with respect to the minimal partition \(\Pi\). Note that \(V_\Pi \cap (\bigcup_{i=1}^{\gamma} D_i) = \emptyset\).

Consider a minimal partition of \(G\), denoted by \(\Pi\). Let \(D_{\gamma+1}\) be a dominating set such that \(V_\Pi \subseteq D_{\gamma+1}\). Since \(\text{dom}(G) = \gamma\), and \(V_\Pi\) is not a dominating set, we have
\[
D_{\gamma+1} = V_\Pi \cup I_\Pi,
\]
where \(I_\Pi \subset (\bigcup_{i=1}^{\gamma} D_i)\). We call a set \(I_\Pi\) with the smallest cardinality, a set of common vertices with respect to the minimal partition \(\Pi\).

The notions of minimal partition \(\Pi\), set of non-critical vertices with respect to \(\Pi\), and set of common vertices with respect to \(\Pi\) are shown in Fig. 6.
Figure 6: A cycle graph, $C_8$ having a domatic number $\gamma = 2$. A minimal partition $\Pi = D_1 \cup D_2 \cup V_{\Pi}$, where $D_1 = \{v_1, v_4, v_7\}$ and $D_2 = \{v_2, v_5, v_8\}$ are minimal dominating sets, and $V_{\Pi} = \{v_3, v_6\}$ is the set of non critical vertices with respect to $\Pi$. We can take another dominating set $D_3$ as $D_3 = V_{\Pi} \cup I_{\Pi}$, where $I_{\Pi} = \{v_8\}$ is a set of common vertices with respect to the minimal partition $\Pi$.

Theorem 4.1. Let $G$ be a graph with domatic number $\gamma$. Let $\Pi$ be a minimal partition of $G$ and $I_{\Pi}$ be a set of common vertices with respect to $\Pi$. If there exists another minimal partition of $G$, say $\tilde{\Pi} \neq \Pi$, such that $I_{\Pi} \subseteq V_{\tilde{\Pi}}$, then $G$ has an $(r,s)$-configuration with $r = s\gamma + \lfloor \frac{s}{2} \rfloor$. Here, $V_{\tilde{\Pi}}$ is a set of non-critical vertices with respect to $\tilde{\Pi}$.

Proof. Let $\Pi = \bigcup_{i=1}^{\gamma} D_i \cup V_{\Pi}$ in which $V_{\Pi}$ is the set of non-critical vertices with respect to the minimal partition $\Pi$. Let $D_{\gamma+1}$ be a dominating set with $D_{\gamma+1} = V_{\Pi} \cup I_{\Pi}$, in which $I_{\Pi}$ is a set of common vertices with respect to $\Pi$. Assign $\lfloor \frac{s}{2} \rfloor$ distinct labels to all the vertices in a dominating set $D_i$, for every $i \in \{1, 2, \ldots, \gamma + 1\}$. Under this labelling scheme, vertices in $I_{\Pi}$ will have $\lfloor 2\lfloor \frac{s}{2} \rfloor \rfloor$ distinct labels as they are included in two different dominating sets, including $D_{\gamma+1}$ and some other $D_i$ for $i \in \{1, 2, \ldots, \gamma\}$. Note that the vertices in $I_{\Pi}$ are the only ones with $\lfloor 2\lfloor \frac{s}{2} \rfloor \rfloor$ labels. Moreover, every $v \in V$ has a $\lceil \frac{s}{2} \rceil \lfloor \gamma + 1 \rfloor$ distinct labels in its closed neighborhood.

Consider another minimal partition of $G$, $\tilde{\Pi} = \bigcup_{i=1}^{\gamma} S_i \cup V_{\tilde{\Pi}}$, with $V_{\tilde{\Pi}}$ being the set of non-critical vertices with respect to $\tilde{\Pi}$, and each $S_i$ being a minimal dominating set. Let $I_{\Pi}$ be such that $I_{\Pi} \subseteq V_{\tilde{\Pi}}$. It means that every vertex in $V - V_{\Pi}$ has $\lfloor \frac{s}{2} \rfloor$ labels. Since $S_i \subseteq (V - V_{\Pi})$ for any $i \in \{1, 2, \ldots, \gamma\}$, every vertex $v \in S_i$ has $\lfloor \frac{s}{2} \rfloor$ labels. Furthermore, assign $\lceil \frac{s}{2} \rceil$ unique labels to each vertex in $S_i$, $\forall i$. Since each $S_i$ is a dominating set, every $v \in V$ has a set of $\lceil \frac{s}{2} \rceil \lfloor \gamma \rfloor$ unique labels in $\mathcal{N}[v]$. Moreover, since $\lfloor \frac{s}{2} \rfloor \lfloor \gamma + 1 \rfloor$ unique labels are already available in the closed neighborhood of every vertex, all

\[\lfloor \frac{s}{2} \rfloor \] labels assigned to the vertices of $D_i$ are different from the ones assigned to the vertices in $D_j$ where $i \neq j$.\]
Figure 7: (a) $\tilde{\Pi} = S_1 \cup S_2 \cup V_{\tilde{\Pi}}$, in which $S_1 = \{v_1, v_4, v_6\}$ and $S_2 = \{v_2, v_5, v_7\}$ are disjoint minimal dominating sets, and $V_{\tilde{\Pi}} = \{v_3, v_8\}$ is the set of non-critical vertices with respect to $\tilde{\Pi}$. (b) A $(5, 2)$-configuration of $C_8$.

vertices in $V$ have $\left\lfloor \frac{s}{2} \right\rfloor (\gamma + 1) + \left\lceil \frac{s}{2} \right\rceil \gamma = s\gamma + \left\lfloor \frac{s}{2} \right\rfloor$ distinct labels in their closed neighborhoods. Note that each vertex is assigned at most $s$ distinct labels. Thus, an $(r, s)$-configuration of $G$ with $r = s\gamma + \left\lfloor \frac{s}{2} \right\rfloor$ is obtained.

As an example, consider a $(5, 2)$-configuration of $C_8$. Domatic number of $C_8$ is 2, i.e., $\gamma = 2$. We consider two minimal partitions of $C_8$, denoted by $\Pi$ and $\tilde{\Pi}$, where $\Pi$ is shown in Fig. 6. For $\tilde{\Pi}$, we take $\tilde{\Pi} = S_1 \cup S_2 \cup V_{\tilde{\Pi}}$, as shown in Fig. 7. Since $I_{\tilde{\Pi}} \subseteq V_{\tilde{\Pi}}$, $(5, 2)$-configuration exists for $C_8$.

5. Assignment of multiple resources in $R$-disk proximity graphs

In this section, a resource assignment problem based on the graph coloring formulation in Section 2 is investigated for the $R$-disk proximity graphs, which are frequently employed to model inter-connections among nodes in multiagent networks. In such a model, a disk of radius $R$, which represents interaction range of a node, is associated with every node $v$ that lies at the center of the disk. A node forms an edge with other nodes if and only if they exist within the $R$ radius disk of the node. Applications of this model include ad hoc communication networks, wireless sensor networks (e.g., see [25]), multiagent and multirobot systems (see e.g., [26]), and other broadcast networks with limited range transmitters and receivers.

Analysis of $(r, s)$-configurations of $R$-disk proximity graphs is of significance, particularly in the context of heterogeneous multiagent systems. Here, we show that under some mild conditions, $R$-disk graphs always have an $(r, s)$-configuration for $r = \left\lfloor \frac{5s}{2} \right\rfloor$, where $s$ is any positive integer. It is assumed that agents, which are equipped with multiple capabilities or resources, are lying in a plane, and interactions among them are modeled by the $R$-disk proximity graph model.
We start by translating the geometric property of such graphs into a graph-theoretic one by first defining the following special graphs. A graph $G$ is a complete bi-partite graph if there exists a partition of its vertex set, $V = X \cup Y$, such that an edge $u \sim v$ exists whenever $u \in X$ and $v \in Y$. If $|X| = x$ and $|Y| = y$, then we denote a complete bi-partite graph by $K_{x,y}$.

Examples of complete bi-partite graphs are shown in Fig. 8. We also define a double cycle graph, denoted by $C_4 \cdot C_4$, as the one obtained by identifying a vertex of $C_4$ with a vertex of another $C_4$, as shown in Fig. 8. Furthermore, a graph $G$ is said to be an $H$-free graph, if $H$ is not an induced subgraph of $G$.

It is shown in [23] that $K_{2,3}$ cannot be an $R$-disk graph. In the following Lemma, it is shown that $R$-disk graphs are always $K_{1,6}$-free.

**Lemma 5.1.** An $R$-disk proximity graph is $K_{1,6}$ free.

**Proof.** Let $G(V, E)$ be an $R$-disk proximity graph. Let $v \in V$, such that $\mathcal{N}(v) = \{v_1, v_2, \cdots, v_p\}$, where $p \geq 6$. We define $\theta_{(v_i,v_j)}$ to be the angle $v$ makes with $v_i$ and $v_j$. If $\|v_i, v_j\|$ is the euclidean distance between $v_i$ and $v_j$, then it is easy to see that $\|v_i, v_j\| > R$, whenever $\theta_{(v_i,v_j)} > 60^\circ$. Thus, $v_i, v_j \in \mathcal{N}(v)$ are non-adjacent if and only if $\theta_{(v_i,v_j)} > 60^\circ$. To have $K_{1,6}$ as an induced subgraph of $G$, there must be a subset of $\mathcal{N}(v)$ with six nodes, say $\{x_1, x_2, \cdots, x_6\} \subseteq \mathcal{N}(v)$, such that $\theta_{(x_i,x_j)} > 60^\circ$, $\forall x_i, x_j$. This will give $\sum_{i=1}^{5} \theta_{x_i x_{(i+1)}} + \theta_{x_6 x_1} > 360^\circ$, which is not possible. Thus, an $R$-disk graph is $K_{1,6}$-free. \qed

Since every $R$-disk graph is $K_{1,6}$-free, we can focus on the results regarding $(r,s)$-configurations of $K_{1,6}$-free graphs to study the resource assignment problem in $R$-disk graphs. A useful result regarding $(r,s)$-configurations of $K_{1,6}$-free graphs is stated in [24].
Theorem 5.2. [24] For any positive integer $s$, a connected $K_{1,6}$-free graph $G$ with a minimum degree of at least two has a $(5,2)$-configuration whenever $G$ is not isomorphic to $C_4, C_7, K_{2,3}$, or $C_4 \cdot C_4$.

The above result can be generalized for any positive integer $s$.

Corollary 5.3. For any positive integer $s$, a connected $K_{1,6}$-free graph $G$ with a minimum degree of at least two has an $(r,s)$-configuration with $r = \left\lfloor \frac{5s}{2} \right\rfloor$ whenever $G$ is not isomorphic to $C_4, C_7, K_{2,3}$, or $C_4 \cdot C_4$.

Proof. If $s$ is an even number, let $s = 2s'$. Obtain a $(5,2)$-configuration of $G$. Repeat this process $s'$ number of times assigning distinct labels to vertices in each step. At the end of $s'$ steps, at most $2s'$ labels are assigned to every $v$, i.e., $|f(v)| \leq 2s' = s$, and $5s' = \frac{5s}{2}$ distinct labels are available in the closed neighborhood of every $v$. Thus, a $(\frac{5s}{2}, s)$-configuration of $G$ is obtained.

If $s$ is an odd number, let $s - 1 = 2s'$, then $(\frac{5(s-1)}{2}, s - 1)$-configuration can be obtained as above. Furthermore, using the fact that every connected graph has a domination number of at least 2, it is possible to assign a single label to each vertex such that every vertex has at least two distinct labels in its closed neighborhood. Thus, for a given positive odd integer $s$, an $(r,s)$-configuration is possible with $r = \frac{5(s-1)}{2} + 2 = \frac{5s}{2} - \frac{1}{2} = \left\lfloor \frac{5s}{2} \right\rfloor$.

Now, using Theorem 5.2, Lemma 5.1, and the fact that an $R$-disk graph can never have a component isomorphic to $K_{2,3}$ graph, we state the following result regarding $(r,s)$-configurations of $R$-disk graphs.

Theorem 5.4. For any positive integer $s$, an $R$-disk proximity graph $G$ with a minimum degree of at least 2 has an $(r,s)$-configuration with $r = \left\lfloor \frac{5s}{2} \right\rfloor$ whenever $G$ has no component isomorphic to $C_4, C_7$, or $C_4 \cdot C_4$.

5.1. Example

Consider a group of robots deployed in a planar region $D$ for the purpose of environment modeling. These robots interact and exchange information with each other, and this interaction is modeled by an $R$-disk proximity graph model. For environment modeling, five different parameters are considered, including temperature, relative humidity, air pressure, light, and soil texture. Every robot performs some spatio-temporal data processing using the data of the environment parameters, which is obtained from the sensors mounted on robots. If each sensor observes a specific environment parameter, five different types of sensors are needed.
One possible course of action is to install all five sensors on each robot so that robots can collect information of all parameters for further processing. However, this might not be a feasible approach for one or more of the following reasons: first, a large amount of power supplies will be needed to keep all five sensors operational on every robot, and power is always a limiting factor for a continuous operation of such networks over an extended period of time; second, a large number of sensors of each type will be required which may not be cost effective; third, mounting all sensors on a single robot may not be feasible from a hardware viewpoint in certain cases. Another approach is to install a subset of sensors on each robot and utilize the fact that robots exchange information with each other. However, this approach requires sensors to be distributed among robots in such a way that each robot can obtain the data of the missing parameters from its neighbor robots. In other words, sensors missing on a robot are installed on its neighbor robots. This set-up requires a much smaller number of sensors of each type for the overall operation of the system. If each robot is allowed to have at most two of the five sensors’ types, then sensors need to be installed in such a way that each robot can find a complete set of five distinct sensors in its closed neighborhood. However, it is possible if and only if the underlying $R$-disk graph of the network has a $(5,2)$-configuration. It is shown in Theorem 5.4 that every $R$-disk graph has a $(5,2)$-configuration under some conditions. Thus, it is possible to make a robot network completely heterogeneous with five labels under the restriction that each robot can have at most two labels. An example is illustrated in Fig. 9.

6. Discussion and conclusions

The notion of completely heterogeneous graph can be extended and generalized in different ways. In Section 2, a completely heterogeneous graph with a set of $r$ unique labels is defined as the one in which every node finds a complete set of $r$ labels in its closed neighborhood. There might be situations in which a complete set of resources might not be required in the closed neighborhood of all the nodes. This motivates to further extend the notion of a completely heterogeneous graph for more general scenarios. For instance, the notion of neighborhood can be extended to the $k$-neighborhood. If the distance between nodes $v$ and $u$, denoted by $d(u,v)$, is the length of the shortest path in a graph $G$, then the open $k$-neighborhood of node $v$ is the set $N_k(v) = \{u \in V : d(u,v) \leq k\}$. Likewise, the closed $k$-neighborhood of $v$ is $N_k[v] = N_k(v) \cup \{v\}$. We can define a completely $k$-heterogeneous graph with $r$ labels as the one in which every vertex finds a complete set of $r$ labels in its
Figure 9: A group of robots connected via $R$-disk proximity graph model. Each robot is assigned two labels from the set $\{1, 2, 3, 4, 5\}$ in such a way that a complete set of five labels is available in the closed neighborhood of every node.

closed $k$-neighborhood. The concept of completely $k$-heterogeneous graph with $r$ labels is particularly useful in situations in which $r$ is quite large, i.e., a large number of different types of resources exist within the network, and it might not be possible to ensure the availability of all these resources in the closed neighborhood of every node. In such situations, one can aim to distribute resources among nodes to get a completely $k$-heterogeneous graph with $r$ labels for a small value of $k$ to ensure that each node finds all $r$ resources within a small distance $k$ from it.

Analogous to the color distribution matrix introduced in Section 3 for the analysis purpose, we can define $k$-color distribution matrix for the case of completely $k$-heterogeneous graphs as

$$\Phi(k) = A_k C.$$  

Here, $A_k = A_k + I$ and $C$ is the color matrix. $A_k$ is an $n \times n$ matrix whose $ij^{th}$ entry is 1 whenever $d(v_i, v_j) \leq k$ and $i \neq j$.

It is to be observed that $ij^{th}$ entry of the $\Phi(k)$ matrix is the number of vertices with the color $j$ in the closed $k$-neighborhood of $v_i$. Thus, we can use the same approach as in Section 3 to analyze the distribution of labels when using the notion of $k$-neighborhood for $k > 1$.

In conclusion, we investigated heterogeneity in cooperative and multiagent systems from a network topology viewpoint in this paper. The notion of $(r, s)$-configuration of a graph was used to characterize a distribution of agents with multiple capabilities (or resources). In such a distribution, every agent could find all types of resources available in the network in its
closed neighborhood. The role of individual agents and interactions among them in attaining \((r, s)\)-configurations was also examined. The study not only analyzed the role of network topology in the context of heterogeneous multiagent systems, but also provided ways to design network structures in which agents equipped with various resources coordinate and compliment each others capabilities to accomplish complex tasks.

**Acknowledgment**

The authors would like to thank Robin Thomas of the Mathematics Department at Georgia Institute of Technology for helpful discussions.

**References**


Characterizing heterogeneity in cooperative networks


Waseem Abbas  
School of Electrical & Computer Engineering  
Georgia Institute of Technology  
Atlanta, GA 30332  
USA  
E-mail address: wabbas@gatech.edu

Magnus Egerstedt  
School of Electrical & Computer Engineering  
Georgia Institute of Technology  
Atlanta, GA 30332  
USA  
E-mail address: magnus@gatech.edu

Received October 16, 2013