Università degli Studi di Cagliari

DOTTORATO DI RICERCA

in Ingegneria Elettronica e Informatica

Ciclo XXVI

TITOLO TESI

Graph methods in Multi Agent Systems coordination and Social Network Analysis.

Settore scientifico disciplinare di afferenza

ING-INF/04 Automatica

Presentata da:	Daniele Rosa
Coordinatore Dottorato:	Prof. Fabio Roli
Tutor:	Prof. Alessandro Giua

Esame finale anno accademico 2012-2013

REGIONE AUTONOMA DELLA SARDEGNA

Ph.D. in Electronic and Computer Engineering
Dept. of Electrical and Electronic Engineering
University of Cagliari

Graph methods in Multi Agent Systems coordination and Social Network Analysis.

Daniele Rosa

Advisor: Prof. Alessandro Giua.
Curriculum: ING-INF/04 Automatica

XXVI Cycle
March 2013

Daniele Rosa gratefully acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1.)

Abstract

In this thesis several results on two main topics are collected: the coordination of networked multi agents systems and the diffusion of innovation of social networks. The results are organized in two parts, each one related with one of the two main topics. The common aspect of all the presented problems is the following: all the system are represented by graphs.

Two are the main contributions of the first part.

- A formation control strategy, based on gossip, which leads a set of autonomous vehicles to converge to a desired spatial disposition in absence of a common reference frame. If the vehicles have common direction, we prove that the proposed algorithm is robust against noise on displacement measurement.
- The formalization of the Heterogeneous Multi Vehicle Routing Problem, which can be described as follows: given an heterogeneous set of mobile robots, and a set of task to be served randomly displaced in a 2D environment, find the optimal task assignment to minimize the service cost. We firstly characterize the optimal centralized solution, and then we propose two distributed algorithms, based on gossip, which lead the system to a sub-optimal solutions and are significantly computationally more efficient than the optimal one.

The contributions of the second part are the following.

- Adopting the Linear Threshold Model, we propose an algorithm based on linear programming which computes the maximal cohesive subset of a network. Moreover, we define two problem of

interest in Social Networks analysis and characterize the optimal solution: the Influence Maximization Problem in Finite Time and the diffusion of innovation over a target set.

- We characterize the novel Non Progressive Linear Threshold Model, which extends the classical Linear Threshold Model. We formalize the model and we give a characterization of the network dynamics in terms of cohesive and persistent sets.

Acknowledgements

At the end of these three years, I want to thank all the people that have shared the important moments of this experience.

First, I would like to thank all the people with whom I have collaborated, that helped and supported me in my scientific work: my advisor Prof. Alessandro Giua, Prof. Carla Seatzu, Mauro Franceschelli and Prof. Francesco Bullo.

A special thank goes to my Lab-friends: Marco, Stefano, Mehran and Alessandro. Thank you guys and..."PRIMO!!".

A huge thank to my family, for their encouragement during all these years.

Finally, I thank Romina...you have been always newt to me.

And thank God, another step is done...

Contents

Introduction

Introduction to the thesis

This thesis collects several results on two main topics: the coordination of networked multi agents systems and the diffusion of innovation of social networks. Both topics have been widely studied in literature in recent years and in different fields, since it is evident in nature the enormous power of the collectivity respect to a single individual: the more a group of individuals is organized, the more it grows up and generate well-being to each member. Moreover, it has always been evident that many target can be better reached by a coordinated group of people than a single individual, and in some cases cooperation is necessary. At the same time, there are some phenomena in which some individuals (or group of them) have a greater influence in the community than others. Thus, in the last two decades, researchers of different fields have been attracted by such concepts: sociology, biology, informatics, electronics, artificial intelligence and control theory.

In this manuscript we address different problems characterized by some common aspects:

- all the considered the systems are sets of simple autonomous systems (agents or individuals), which are connected together by a network;
- in each system the behaviour of each agent is influenced by the behaviour of interconnected agents;
- all the described systems can be represented using graphs, thus all the mathematical results of this thesis are based on graph theory.

The thesis is organized in two parts, each one focused on one of the two main topics.

Part 1: Coordination of multi-agent systems through consensus

In the first part we focus on the coordination of multi vehicle systems. Given a set of autonomous vehicle, which can exchange information through a communication network, we propose several solutions of problems which were largely studied in literature in the recent years. All the results presented in this part are based on distributed consensus algorithm: agents exchange information according to a common protocol in order to reach an agreement on a certain quantity of interest. In particular, most of the proposed solutions are based on gossip algorithms, which are characterized by the following:

- the communication scheme involve only a couple of agents at each step;
- the communication steps between couple of agents are asynchronous.

The contribution of the first part are the following.

(1) *A formation control strategy.* We propose a novel decentralized coordination strategy, based on gossip, that allows a dynamic multi-agent system, in absence of a common reference frame, to estimate a common orientation and achieve arbitrary spatial formations with respect to the estimated frame. We assume that the agents are mobile point-mass vehicles that do not have access to absolute positions (GPS). To the best of our knowledge this strategy extends the state of art since it simultaneously solves two problem which are commonly considered separately:

   - the achievement of an agreement on a common reference frame in absence of it;
   - the achievement of a desired spatial disposition.

   The method is robust against measurement noise of odometry or inertial navigation.

(2) *Distributed solutions for the heterogeneous multi-vehicle routing Problem.* We focus on problems of MTSP (Multi Travelling Salesman Problem), and problems of MVRP (Multi Vehicle Routing Problem). Given a network,

characterized by a set of nodes and a set of connections between them, the problem of MTSP is to optimally assign nodes, which have to be visited, to the different vehicles, in order to minimize the sum of the costs of the paths. The problem of MVRP represents an extension of the MTSP in which other variables are taken into account such as the capacity of vehicles or costs assigned to the nodes. We extend the state of art since we consider the case where a set of heterogeneous tasks arbitrarily distributed in a plane has to be serviced by a set of mobile robots, each with a given movement speed and task execution speed. Our goal is to minimize the maximum execution time of robots. We propose two distributed algorithms based on gossip communication: the first algorithm is based on a local exact optimization and the second is based on a local approximate greedy heuristic.

Part 2: Diffusion of innovation in Social Networks

In the second part we focus on the diffusion of innovation in social networks. Whit the expression social network we identify a group of people which are connected together by some types of relationship: friendship, love, business. In particular we focus on the study of the mechanism which convince people to adopt an idea or an innovation, and how the behaviour of each individual is influenced by the behaviour of the connected individuals or groups. Following the trend of the control community, we study mechanisms of innovation spread in Social Networks in order to forecast, optimize, control some diffusion behaviours. Our reference mathematical model is the so called *linear threshold model*, and the contribution of this thesis are the following.

(1) *Analysis and control of the diffusion of innovation in the Linear Threshold Model.* We adopt the classical linear threshold model, which is characterized as follows:

- at each individual is assigned a threshold value, which is a value in $[0, 1]$;
- a node adopts the innovation as soon as the ratio of its neighbours who have already adopted it is above its threshold value;

- the innovation is incepted in the network by a seed set of individuals.

According to this model, we firstly present an integer programming problem and an iterative algorithm based on linear programming which take as input the set of innovators and compute the maximal cohesive set of the complement of the seed set. The output of these algorithms can be used to compute the set of final adopters in the network. We extend the state of art by proposing a way to compute the maximal cohesive set in a given social network, which was just defined so far, to the best of our knowledge.

Then we introduce and formalize with integer programming two problems. The "influence maximization in finite time problem (IMFT)" is that of finding a seed set of r individuals that maximizes the spread of innovation in the network in k steps. This problem represent an extension of the classical influence maximization problem, which considers an infinite time horizon.

The second one is that of finding a seed set of whose cardinality is minimal which diffuses the innovation to a desired set of individual in k steps.

(2) *A novel non-progressive instance of the linear threshold model.* The classical linear threshold model has a progressive nature, i.e., an individual can adopt the innovation if it hasn't adopted yet, but once adopted it cannot abandon it. We extend the classical model by proposing a novel model in which each individual in the social network is influenced by the behaviour of its neighbours, and at each steps it decides either to adopt, abandon or maintain the innovation by following a threshold mechanism.

We assume that the innovation is incepted in the network by a seed set of individuals which are assumed to maintain the innovation independently of the state of their neighbours for a finite time. We identify all the possible evolution of the network under the proposed model, and we describe in details the evolution of the system in terms of two particular type of subgroups, namely *Cohesive* and *Persistent* sets.

Part I

Coordination of Multi-Agent Systems

Chapter 1

Using consensus to coordinate multi-agent systems: introduction and literature overview.

Multi Agent Systems (MAS) are a class of systems characterized by a set of entities , agents, which interact in a shared environment to achieve a common target. Such systems have attracted the attention of many researchers from different fields in the last decades: economy, sociology , philosophy, and , of course, computer science and automation.

In the control theory community the term *agent* identify an autonomous system, with a simple dynamic, which interact with the environment where it operates and takes autonomous decision to reach a given target. A Networked Control System (NCS) is a system composed by a set of agents which exchange information through a communication network, and take decisions influenced by neighbours to reach a common target. These system presents many advantages with respect to isolated systems.

- In a MAS agents can execute in parallel sub-tasks of a single complex task: that reduces the total execution time and the computational coasts.
- The absence of a single decision center makes the system more reliable and robust to failures.
- The implementation of a set of simple agents which cooperates to solve a

problem can be less expensive than a complex centralized system.

Recently, in literature this concepts have been applied to problem such as:

- coordination of autonomous vehicles;
- environmental monitoring;
- localization systems;
- coordination of mobile robots.

Typical methods related with MAS are based on *distributed consensus algorithms*: agents exchange local information to reach an agreement on a certain quantity of interests. These algorithms have been applied to problems such as rendez-vous, flocking or intrusion detection. When the state of the agents converge to the average of their initial states we refer to it as *average consensus*.

In the next chapters we apply consensus algorithms to two different problems.

In Chapter 2 we present a novel formation control strategy, based on consensus, which leads a set of autonomous vehicles to converge to a desired formation in absence of a common reference frame. In Chapter 3 we use gossip algorithms to solve a particular instance of the Multi Vehicle Routing Problem.

All the presented approaches are based on a special type of consensus algorithms, , namely *gossip algorithms*. Gossip algorithms are characterized by an asynchronous pairwise communication scheme: at each step only two agents exchange information independently of the rest of the agents.

In the next sections we introduce the two studied problems in details.

1.1 Formation control for multi vehicle systems

Multi-agent systems consisting in a network of autonomous vehicles benefit greatly from the global positioning system (GPS) in that it allows to close feedback control loops on estimated positions in a global reference frame common to every vehicle, enabling several control tasks such as surveillance, patrolling, formation control or search and rescue missions to be performed. Unfortunately such a powerful tool may not always be exploited for several reasons: for instance the GPS signal is unreliable for indoor/underwater environments, during adverse

atmospheric conditions or in locations close to transmission power lines and is vulnerable to jamming attacks. Furthermore, if the desired scale of relative distances between the vehicles is of the order of meters, the accuracy provided by the GPS system might not be enough. The problem of how to coordinate a network of agents in absence of absolute position information has thus received great attention from the control theory community (1, 2, 3). Furthermore, it is usually assumed that the full network topology is not known by the agents and that only local point-to-point communication or sensing are available to model sensors with limited capabilities. In (4) a theoretical framework and a method to achieve flocking in a multi-agent system is proposed based on the famous three rules of flocking by Reynolds (5) and on local interaction rules based on virtual potentials that allow the achievement of flocking as global emergent behaviour. In (6, 7, 8, 9) the consensus problem, i.e., the problem of how to make the state of a set of agents converge toward a common value, was presented regarding also the application of multi-agent coordination. In particular control strategies based on consensus algorithms were described in these papers as a fundamental tool to achieve synchronization of velocities, directions or the attainment of constant relative distances between the agents.

In our approach we assume that each agent estimates relative positions with its neighbours in its own local reference frame centered on it. A similar assumption was made in (10), where a Nyquist criterion to determine the effect of the topology of a multi-agent system performing formation control was proposed; in this case the agents were assumed to have a common coordinate system but not a common origin. Furthermore we firstly assume that each agent has an onboard compass, which allows all the local frames to have the same orientation. Then we remove this assumption.

Many formation control strategies are based on Leader-based approaches (11, 12), which require the network of vehicles to properly follow one or more leaders, possibly controlled by a pilot, satisfying eventually some constraints. Also some formation control strategies in the literature take advantage from the presence of leaders exploiting network properties such as graph rigidity (13).

In Chapter 2 we design a coordination strategy for point-mass agents in which leaders are not required, and the desired formation is expressed with coordinates centred at the estimated common reference point. We also show that the proposed

strategy, based on an *overcompensation* of the agents' displacement, is robust against measurement noise. The concept of overcompensation is presented in the following sections.

In (14) a decentralized algorithm to make a network of agents agree on the location of the network centroid in absence of common reference frames was presented; the algorithm is based on gossip (only random asynchronous pairwise communications) and assumes static agents displaced in a 3-d space. In (15) a decentralized algorithm based on gossip to make a network of agents agree on a common reference point and frame was proposed, assuming static agents in a 2-d plane. Our approach differs from (14, 15) in that we consider dynamic agents that move while the the estimation process is executed, we assume that all the agents local reference frames are oriented in the same direction and that noise is affecting the relative position measurements. Furthermore, the proposed approach is used to implement formation control.

Summarizing, the following are the main contributions of Chapter2.

- A novel local interaction protocol that achieves robust estimation of the network centroid robust to parameter uncertainties.
- A method to achieve provably robust formation control with respect to parameter uncertainties in the agents' dynamics.
- An extended method to achieve robust formation control with formations of arbitrary shape by performing agreement on a common reference frame. We provide simulations to corroborate the description of this extended method.

1.2 The Heterogeneous Multi Vehicles Routing Problem

The travelling salesman problem (TSP) is a well known topic of research and can be stated as follows: find the Hamiltonian cycle of minimum weight to visit all the nodes in a given graph. Instructive surveys can be found in (16, 17, 18). This problem has received great attention for both its theoretical implications and its several practical applications. The Vehicle Routing Problem (VRP) is a generalization of the TSP and was firstly introduced in (19): given a fleet of n

vehicles and a set of locations to be visited, the vehicle routing problem consists of finding n tours to visit all locations in minimum time.

Several extensions of the TSP and the VRP have been proposed to better suit practical applications by introducing several additional constraints and objectives such as a variable number of vehicles, a finite load capacity, a cost associated to each node which represents the demand of the costumer, service time windows and several more. Numerous extensions are well summarized in (20, 21, 22). Finally, several extensions explore a dynamic setting in which multiple vehicles serve a dynamic number of tasks as discussed in (23).

Multi-vehicle routing problems have a combinatorial nature, as all the possible tours must be explored to find the optimal configuration. Exact algorithmic formulations are based, for example, on Integer Linear Programming (ILP) as described in (22, 24). General ILP solvers are characterized by an exponential computational complexity, thus in the last decades many approximate algorithms have been proposed which are characterized by a lower computational complexity. Examples of heuristics and approximate algorithms are presented in (21, 25, 26, 27, 28, 29).

We are interested in an instance of the VRP, called the Heterogeneous Multi Vehicle Routing Problem (HMVRP), with the following properties: the number n of vehicles is given a priori, a set $\mathcal{K}$ is given containing k tasks arbitrarily distributed in a plane, to each task is assigned a servicing cost, each vehicle is characterized by a movement speed and a task execution speed.

It has been shown in (30) that when comparing the length of the optimal tour of one vehicle that visits all tasks locations with the multiple vehicle case, the maximum length of the tours for the multiple vehicle case is proportional to the tour length of the single vehicle case and proportionally inverse to the number of vehicles. Both upper and lower bounds with such scaling were given.

In Chapter 3 we extend the result in (30) by considering execution times instead of tour lengths to account for vehicles of different speeds, tasks with arbitrary execution costs and vehicles with different task execution speeds. We provide upper and lower bounds to the optimal solution as function of the single vehicle optimal tour length to put in evidence how the performance is affected by the number of vehicles.

We propose two distributed and asynchronous algorithms for the HMVRP:

the first one is based on the iterative optimization of the local task assignment between pairs of vehicles (31), the second one is based on local task exchange of assigned tasks, one by one, between couples of vehicles (32). For both algorithms we provide deterministic bounds to their performance. The proposed approaches to the HMVRP are distributed algorithms easy to implement in a networked system and have favorable computational complexity with respect to the ratio k/n between the number of tasks and vehicles instead of k as in the centralized approach.

Note that the considered problem can also be seen as a particular instance of a min/max VRP problem whose main feature is the heterogeneity of the speed and the tasks execution speed of the vehicles. Related works on the min/max VRP problem include (33, 34, 35).

Summarizing, the following are the main contributions of Chapter3.

- We formalize the centralized problem in terms of a mixed integer linear programming (MILP) problem and extend the bounds in (30) for the multi TSP to the HMVRP.
- We propose a first distributed algorithm, based on gossip communication and on the solution of local MILP, to solve the HMVRP and characterize some of its properties.
- We propose a second distributed algorithm to solve HMVRP, based on gossip communication and on local task exchanges, characterized by a low computational complexity.
- We provide simulations that show that the proposed algorithms attain a constant factor approximation of the optimal solution with respect to the number of vehicles. A detailed comparison among the performances of the two proposed decentralized solutions is also presented.

Chapter 2

Formation Control Strategy

This chapter is organized as follow. In Section 2.1 we present the considered system and the set of assumptions adopted. In Section 2.2 we propose a formation control strategy which is characterize by a parallel application of two different decentralized algorithms: a local displacement control rule which move each agent toward a target point and a consensus algorithm which allows agents to reach an agreement on a common reference frame. The concept of overcompensation is here presented. In Section 2.2.4 the robustness of the proposed strategy is investigated and an optimal choice of the algorithm parameters is discussed.

2.1 Preliminaries

Let a network of agents be described by a time-varying undirected graph $\mathcal{G}(t) = \{V, \mathcal{E}(t)\}$, where $V = \{1, \ldots, n\}$ is the set of nodes (agents), $E \subseteq \{V \times V\}$ is the set of edges e_{ij} representing point-to-point bidirectional communication channels available to the agents, $\mathcal{E}(t) : \mathbb{R}^+ \to E$ is the set of edges being active at time t. Given a time interval T, the joint graph $\mathcal{G}([t, t+T))$ is the union of graphs $\mathcal{G}(t)$ in the time interval $[t, t+T)$ defined as $\mathcal{G}([t, t+T)) = \{V, \mathcal{E}([t, t+T)))\}$, where

$$\mathcal{E}([t, t+T)) = \mathcal{E}(t) \bigcup \mathcal{E}(t+1) \bigcup \ldots \bigcup \mathcal{E}(t+T)$$

A node $u \in V$ is said to be reachable from $v \in V$ if there exists a path in the graph from v to u. Node $u \in V$ is said to be a *center node* if it is reachable from any node in V. In a connected undirected graph all the nodes are center nodes.

A node $u \in V$ is said to be *aperiodic* if the greatest common divisor of all the possible path length from u to u is 1.

The state of each agent i is characterized by its absolute position x_i, an estimation of the origin of the common reference frame $s_i \in \mathbb{R}^2$ and an angle θ_i which represents the orientation of the x-axis of the local reference frame with respect to the x-axis of the global reference frame.

Let $\mathcal{N}_i(t) = \{j : e_{ij}(t) \in \mathcal{E}(t)\}$ be the set of agents that send and receive information to agent i at time t, these agents are called *neighbors* of agent i. We define the *degree* of agent i as $\delta_i(t) = |\mathcal{N}_i(t)|$ where $|\mathcal{N}_i(t)|$ denotes the cardinality of set $\mathcal{N}_i(t)$. The elements of the Laplacian matrix $\mathcal{L}$ of graph $\mathcal{G}(t)$ are defined as

$$l_{ij} = \begin{cases} -1, & if\ (i,j) \in \mathcal{E}(t) \\ \delta_i(t). & if\ i = j \\ 0 & \text{otherwise} \end{cases}$$

Given a generic square matrix $M_{n\times n}$, the *associated graph* $\mathcal{G}_M = \{V_M, E_M\}$ is composed as follow:

- $\mathcal{G}_M$ has n nodes, with index $i \in [1, n]$, so $V_M = \{1, \ldots, n\}$;
- $\mathcal{G}_M$ has an edge e_{ij} if the entry $m_{ij} \in M$ is nonzero, so $E_M = \{(i,j)|m_{i,j} \neq 0\}$

If M has non zero diagonal entry m_{ii}, than node $i \in \mathcal{G}_M$ has a self loop. If M is symmetric then $\mathcal{G}_M$ is an undirected graph. For a time-varying square matrix $M(t)$ the associated graph is denoted as $\mathcal{G}_M(t) = \{V_M, E_M(t)\}$.

A square matrix A is *stochastic* if its elements are non-negative and the row sums equals one. A stochastic matrix said to be *ergodic* if $rank\left(\lim_{k\to\infty} A^k\right) = 1$. An ergodic matrix A is *SIA* (stochastic, indecomposable and aperiodic) if

$$\lim_{k\to\infty} A^k = \mathbf{1}_n \pi^T,$$

where π is the left eigenvector of A corresponding to the unitary eigenvalue and $\mathbf{1}_n$ is the n-element vector of ones. Given two matrices $A_{(m\times n)}$ and $B_{(p\times q)}$, the *Kronecker product* is denoted as $A \otimes B_{(mp\times nq)}$.

In our discussion we consider the following working **assumptions**: i. Agents are modelled by discrete time single integrators; ii. Neighboring agents communicate with bidirectional channels and sense relative positions in a 2-D plane; iii.

Each agent owns a local coordinate system that moves rigidly with it and do not know the coordinate system of others.

2.1.1 Coordinate systems

A 2-d reference frame $\Sigma' = (o', \theta')$ is an orthogonal coordinate system characterized by an origin $o' \in \mathbb{R}^2$ and orientation of the x-axis $\theta' \in [0, 2\pi)$ respect to a global coordinate system Σ defined by $o = (0, 0)$ and $\theta = 0$. We deal with three kinds of coordinate systems, which are showed in Fig. 2.1.

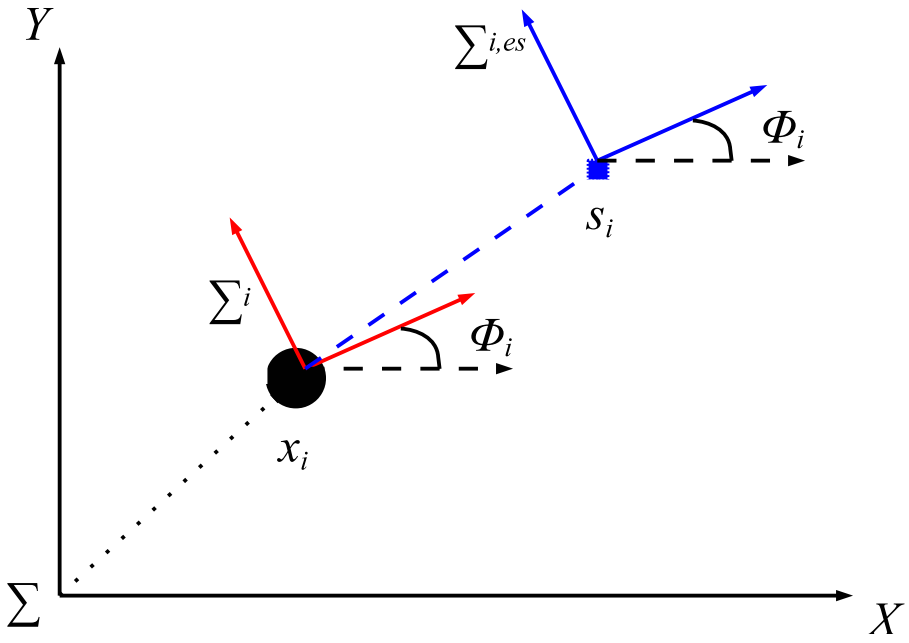


Figure 2.1: Coordinate systems.

- *Global coordinate system:* is the reference frame used to describe the system from the point of view of an external observer. We denote it with Σ, and the current position of agent i specified in Σ is $x_i \in \mathbb{R}^2$.
- *Local coordinate system:* each agent owns a local reference frame centered on it. The local coordinate system of agent i is denoted with $\Sigma^i = (x_i, \theta_i)$,

where x_i is the position of agent i in Σ and θ_i is the angle between the x-axis of Σ and the x-axis of Σ^i. We denote the position of a generic point j with respect to Σ^i as x_j^i. Therefore, the position of j is

$$x_j = R_i x_j^i + x_i$$

where

$$R_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$$

is a rotation matrix associated to the angle θ_i.

- *Estimated coordinate system:* each agent keeps a local estimation of the common reference frame. With respect to Σ the estimated common reference frame by agent i is denoted with $\Sigma^{i,es} = (s_i, \theta_i)$, where s_i is the estimated reference center and θ_i is the estimated angle between the x-axis of the common reference frame and the x-axis of Σ. Note that the orientation of the local estimated reference frame is the same as the orientation od Σ^i. We denote the position of a generic point j with respect to $\Sigma^{i,es}$ as $x_j^{i,es}$. The position of agent j in frame Σ^i is: $x_j^i = x_j^{i,es} + s_i^i$.

2.2 Formation control strategy

In this section we present a decentralized control strategy which allows a network of mobile agents in a 2-D space to reach an agreement on a common reference frame and simultaneously converge to a desired formation. Here we assume that all the agents have a compass on board, which allows them have a common reference direction. In particular, we assume that $\forall i \in \mathcal{V}$, $\theta_i = 0$.

The state of i-th agent is characterized by a position $x_i \in \mathbb{R}^2$ and a variable $s_i \in \mathbb{R}^2$ which represents the estimated center of the common reference frame. When referring to the state of the agent in its own reference frame Σ^i we denote its current estimation as $s_i^i \in \mathbb{R}^2$.

Our strategy involves three local state update rules:

- A rule to update the position of the agents;
- A rule to achieve agreement on a common reference point;

2.2.1 Position update rule

Each agent is modeled by discrete time single integrator dynamics

$$x_i(t+1) = x_i(t) + qu_i(t), \tag{2.1}$$

where $x_i \in \mathbb{R}^2$ is the agent position, $u_i \in \mathbb{R}^2$ is the control action representing a displacement and $q \in \mathbb{R}^+$ is a gain. Each agent has to reach a constant target position $D_i \in \mathbb{R}^2$ with respect to its estimated common reference frame. The target position $d_i^i(t)$ with respect to Σ_i at time t can be computed as

$$d_i^i(t) = s_i^i(t) + D_i.$$

In the common reference frame Σ the target position of agent i is

$$d_i(t) = x_i(t) + d_i^i(t) = x_i(t) + (s_i^i(t) + D_i). \tag{2.2}$$

Therefore, each agent drives itself toward its target position $d_i^i(t)$ with the following state update

$$x_i(t+1) - x_i(t) = q\,(d_i(t) - x_i(t)) \tag{2.3}$$

with respect to Σ. By replacing equation (2.2) in (2.3) we find the following position update rule:

$$x_i(t+1) = (1-q)x_i(t) + q(s_i(t) + D_i) \tag{2.4}$$

The reference frame of agent i thus moving rigidly with it, displace its current estimation of the common reference point. Therefore, the agent attempts to *compensate* this displacement by updating its estimation of the position of the common reference point as follows In other words, because the agents' local frame is centered on x_i and moves rigidly with it, each agent i needs to update s_i^i, and consequently d_i^i.

$$s_i^i(t+1) = s_i^i(t) - q\left(s_i^i(t) + D_i\right) \tag{2.5}$$

which, with respect to reference frame Σ, keeps the absolute position of the estimated point constant in time

$$s_i(t+1) = s_i(t).$$

To implement these updates, however, a perfect knowledge of parameter q is required which corresponds to an exact measurement of the movement or actuators with perfect precision.

Since measurements may be affected by disturbance and actuators subjected to malfunctioning, we introduce a different state update rule, which we prove is robust against uncertainties in the parameter q of any agent. We call this state update as *overcompensation* because it effectively moves the current estimation further away than necessary, as follows:

$$s_i^i(t+1) = s_i^i(t) - k\left(s_i^i(t) + D_i\right) \tag{2.6}$$

Equation (2.6) represents a *overcompensation* of agent displacement based on parameter k, which controls how much the agents compensate their displacement.

Using equation (2.4) and equation (2.6) in terms of $s_i(t)$, we can express the general update rule as follow:

$$\begin{cases} x_i(t+1) = x_i(t) + q((s_i(t) + D_i) - x_i(t)) \\ s_i(t+1) = s_i(t) - k((s_i(t) + D_i) - x_i(t)) + q((s_i(t) + D_i) - x_i(t)) \end{cases} \tag{2.7}$$

We can set $h = k - q$ and rewrite equation (2.7) as follows:

$$\begin{cases} x_i(t+1) = x_i(t) + q((s_i(t) + D_i) - x_i(t)) \\ s_i(t+1) = s_i(t) - h((s_i(t) + D_i) - x_i(t)) \end{cases} \tag{2.8}$$

$$\begin{cases} x_i(t+1) = (1-q)x_i(t) + qs_i(t) + qD_i \\ s_i(t+1) = (h)x_i(t) + (1-h)s_i(t) + (-h)D_i \end{cases} \tag{2.9}$$

Note that:

- if $h = -q$ $(k = 0)$ the distance vector $d_i(t) - x_i(t)$ is constant, thus there is no compensation;
- if $-q < h < 0$ $(0 < k < q)$, $d_i(t)$ translate in the same direction of $x_i(t)$ and $|d_i(t+1) - x_i(t+1)| < |d_i(t) - x_i(t)|$, thus there is only a partial compensation;
- if $h = 0$ $(k = q)$ the target position $d_i(t)$ is constant, thus the compensation is perfect;
- if $h > 0$, $(k > q)$ $d_i(t)$ moves toward $x_i(t)$, thus an *overcompensation* is made.

2.2.2 Consensus on the network centroid

Each agent has a local estimate $s_i^i(t)$ which considers as the center of a common estimated frame. By exchanging this local information with neighbours, agents are able to reach an agreement on a common reference center, which means that:

$$\forall i, j \in V, \quad \lim_{t \to \infty} \|s_i(t) - s_j(t)\| = 0$$

At each time step agent i receives the value s_j^j from each agent $j \in \mathcal{N}_i(t)$. In Figure 2.2 it is shown how agent i is able to determine the correct value s_j^i of agent j with respect to Σ^i by only knowing x_j^i and the received value s_j^j. The

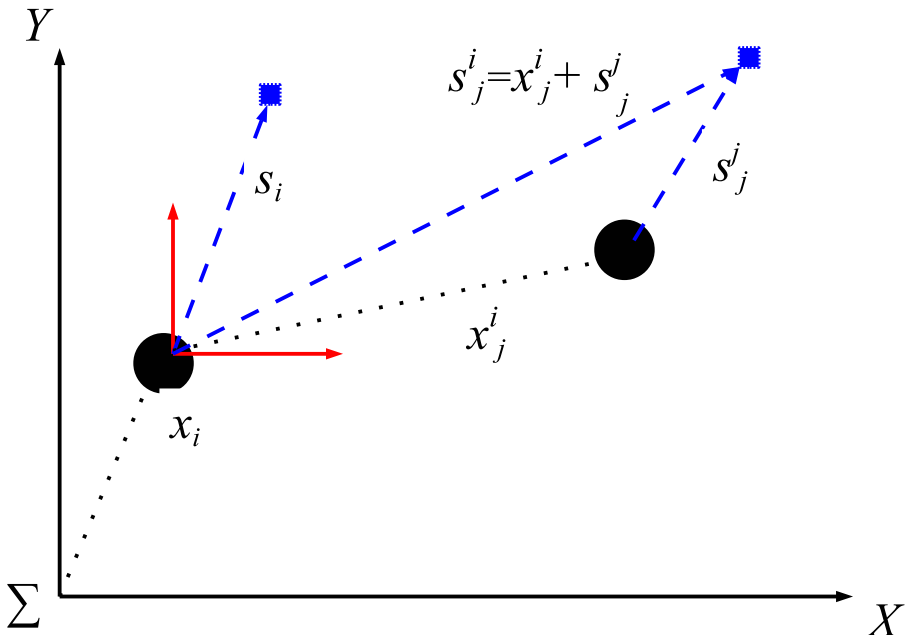


Figure 2.2: Information exchange between agent i and j.

update rule for the local estimate is:

$$s_i^i(t+1) = s_i^i(t) + \varepsilon \sum_{j \in \mathcal{N}_i(t)} (s_j^j(t) + x_j^i(t) - s_i^i(t)) \tag{2.10}$$

with $0 < \varepsilon \leq |\mathcal{N}_i(t)|$. The same rule could be written with respect to Σ:

$$s_i(t+1) = s_i(t) + \varepsilon \sum_{j \in \mathcal{N}_i(t)} l_{ij}(s_j(t) - s_i(t)) \tag{2.11}$$

With respect to Σ the overall estimate update rule could be expressed as follow:

$$s(t+1) = (P(t) \otimes I_{2\times 2})s(t) \tag{2.12}$$

where $P(t) \in \mathcal{P}$ is a time-varying matrix which depends on network topology at time t and ε, and $\mathcal{P}$ is the set of all possible matrices representing the system update defined in (2.11). Due to the update rule definition all matrices $P(t) \in \mathcal{P}$ are stochastic. Note that equation (2.12) can represent both deterministic synchronous consensus algorithms and randomized gossip algorithms. At each t, algorithm (2.12) can be represented by the associated graph $\mathcal{G}_P(t)$. If $\forall t > 0$ there exists a $T > 0$ such that $\mathcal{G}_P([t, t+T))$ is connected, than $\lim_{t\to\infty} s_1(t) = \ldots = \lim_{t\to\infty} s_n(t)$, where $\mathcal{G}_P([t, t+T))$ is the union of graphs $\mathcal{G}_P(t)$ in the time interval $[t, t+T)$ (7)(8).

2.2.3 Formation control strategy

Let us define column vectors $x(t) = \{x_1(t), \ldots, x_n(t)\}$, $s(t) = \{s_1(t), \ldots, s_n(t)\}$ and $D = \{D_1, \ldots, D_n\}$. Note that D represents the desired formation respect to a common center. By summing the contributions of equations (2.8) and (2.12) the overall formation control strategy could be expressed as follow:

$$\begin{bmatrix} x(t+1) \\ s(t+1) \end{bmatrix} = (M(t) \otimes I_{2\times 2}) \begin{bmatrix} x(t) \\ s(t) \end{bmatrix} + \begin{bmatrix} qD \\ -hD \end{bmatrix} \tag{2.13}$$

where

$$M(t) = \left[\begin{array}{c|c} (1-q)I_{n\times n} & qI_{n\times n} \\ \hline hI_{n\times n} & (P(t) - hI_{n\times n}) \end{array}\right] \tag{2.14}$$

For all t, $M(t) \in \mathcal{M}$, where $\mathcal{M}$ is the set of all possible matrices of type (2.14) corresponding to different $P(t) \in \mathcal{P}$. A given formation is considered to be achieved if

- $x(t) = s(t) + D$;
- $\forall i, j \in V, \quad \|s_i(t) - s_j(t)\| = 0$

Lemma 2.2.1 *Consider system* (2.13). *If*

$$\lim_{t\to\infty} (M(1)M(2)\ldots M(t) \otimes I_{2\times 2}) \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} = c\mathbf{1}_{2n} \tag{2.15}$$

then

- $\lim_{t\to\infty} x(t) = s(t) + D$,
- $\forall i, j \in V, \quad \lim_{t\to\infty} \|s_i(t) - s_j(t)\| = 0$.

Thus the desired formation is asymptotically achieved.

Proof: Condition (2.15) implies that system (2.13) is stable. At the equilibrium $x(t+1) = x(t)$ and $s(t+1) = s(t)$. From the first equation of (2.13) we find:

$$(1-q)Ix(t) + qIs(t) + qID = x(t)$$

$$x(t) = s(t) + D$$

By substituting in the second equation:

$$Ps(t) = (I - \varepsilon L)s(t) = s(t)$$

which implies $s(t) = c\mathbf{1}$, where $c \in \mathbb{R}$ is a constant. ■

Convergence of the proposed strategy toward the desired formation can thus be addressed by studying the stability of the following linear time-varying system

$$\begin{bmatrix} x(t+1) \\ s(t+1) \end{bmatrix} = (M(t) \otimes I_{2\times 2}) \begin{bmatrix} x(t) \\ s(t) \end{bmatrix} \tag{2.16}$$

2.2.3.1 Case I: static topology

If the network topology is static and connected, than $M(t) = M$, $\forall t$.

Lemma 2.2.2 *(Lin,(36)) A stochastic matrix M is SIA if and only if the associated graph $\mathcal{G}_M$ has a centre node which is aperiodic.* ■

Now we are able to prove the following result.

Theorem 2.2.1 *Consider a network of agents with a static connected topology. Given system (2.16) with $M(t) = M$, if*

$$0 \leq h \leq 1 - \varepsilon\delta_{max} \tag{2.17}$$

where $\delta_{max} = max\{\delta_1, \cdots, \delta_n\}$ represents the maximum degree for the network, then

$$\lim_{t\to\infty} \begin{bmatrix} x(t) \\ s(t) \end{bmatrix} = c\mathbf{1}_{2n},$$

where $c \in \mathbb{R}$ is a constant.

Proof If condition (2.17) holds $\mathbf{M}$ is stochastic as all entries are non negative and row sums are equal to 1. Now we have to prove that $\mathbf{M}$ is SIA. We can represent system (2.16) using a undirected graph $\mathcal{G}_{\mathbf{M}}$ associated to matrix $\mathbf{M}$. In this graph each agent i is represented by two nodes:

- one associated to the agent position x_i, that we call position node;
- one associated to the agent estimate s_i, that we call estimate node.

For each agent the two associated nodes are connected together by a bidirectional edge, as the position update depends on the position estimate and vice versa. The connections between agents depend on matrix $P - hI$. In particular, given a couple of agents (i, j) there exists an edge between their estimation nodes if the p_{ij} entry of P is non zero. As the network is connected and undirected by assumption, the graph $\mathcal{G}_{\mathbf{M}}$ is connected as well and each node is a center node. More, as all diagonal entries in $(1 - q)I$ are nonzero, each position node in the associated graph has a self loop, so $\mathcal{G}_{\mathbf{M}}$ is aperiodic. It follows from Lemma 2.2.2 that matrix $\mathbf{M}$ is SIA, so

$$\lim_{t\to\infty} \mathbf{M}^t \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} = c\mathbf{1}_{2n}$$

where c is a constant. ■

2.2.3.2 Case II: time-varying topology.

In order to prove the robustness of (2.16) we need first to present some preliminary notions.

Lemma 2.2.3 *(Jadbabaie et al.,(8)) Let $\{M_1, M_2, \ldots, M_m\}$ be a set of stochastic matrices of the same order such that the joint graph $\{\mathcal{G}(M_1) \bigcup \mathcal{G}(M_2) \bigcup \ldots \bigcup \mathcal{G}(M_m)\}$ is connected. Then the matrix product $M_1 M_2 \ldots M_m$ is ergodic.* ■

Lemma 2.2.4 *(Wolfowitz,(37)) Let* $\{M_1, M_2, \ldots, M_m\}$ *be a set of ergodic matrices with the property that for each sequence* $M_{i_1}, M_{i_2}, \ldots, M_{i_j}$ *of positive length* j *the matrix product* $M_{i_1} M_{i_2} \ldots M_{i_j}$ *is ergodic. Then for each infinite sequence* $M_{i_1}, M_{i_2}, \ldots$ *there exists a row vector* $\mathbf{c}$ *such that* $\lim_{j\to\infty} M_{i_1} M_{i_2} \ldots M_{i_j} = \mathbf{1c}$. ■

Now we can state the following theorem

Theorem 2.2.2 *Consider a network of agents with time-varying topology described by (2.16). Let us assume that* $\forall t > 0$ *there exists a* $T > 0$ *such that* $\mathcal{G}_P([t, t+T))$ *is connected. The following condition is sufficient for the system to converge to the desired formation:*

$$0 \leq h \leq 1 - \varepsilon\delta_{max} \tag{2.18}$$

Proof Let $\mathcal{M}_c$ be the set of all possible product matrices in $\mathcal{M}$ of length T such that the joint graph $\mathcal{G}_P([t, t+T))$ is connected. In the theorem we assume that for each time interval $[t, t+T)$ the matrix

$$M(t)M(t+1)\ldots M(t+T) \in \mathcal{M}_c$$

Thus we can represent the evolution of the system as a product of matrices $M_c(t) \in \mathcal{M}_c$. If condition (2.18) holds, then all matrices $M(t) \in \mathcal{M}$ are stochastic as showed in the proof of Theorem 2.2.1, and it follows from Lemma 2.2.3 that all matrices $M_c(t) \in \mathcal{M}_c$ are ergodic as well as all products in $\mathcal{M}_c$. Finally it follows from Lemma 2.2.4 that:

$$\lim_{t\to\infty} (M_c(1)M_c(2)\ldots M_c(t) \otimes I_{2\times 2}) \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} = c\mathbf{1}_{2n}$$

■

2.2.4 Characterization of the robustness of the approach

The proposed coordination strategy described in section 2.2 can be affected by errors due to the odometry or inertial navigation system. In particular the desired displacement that the generic agent $x_i(t)$ should achieve within one sample of time is as follows

$$x_i(t+1) = x_i(t) - q_i(t)(x_i(t) - s_i(t)). \tag{2.19}$$

where the time-varying parameter $q_i(t) = q + \Delta_i(t)$ models a random error in the position update at time t.

Thus, the proposed local interaction rule becomes

$$\begin{cases} x_i(t+1) = & (1-q_i(t))x_i(t) + q_i(t)s_i(t) \\ s_i(t+1) = & h(t)(x_i(t) - s_i(t)) \\ & +(s_i(t) + \varepsilon \sum_{j \in \mathcal{N}_i} l_{ij}(s_j(t) - s_i(t))) \end{cases} \tag{2.20}$$

where $h_i(t) = h - \Delta_i(t)$.

Let $Q(t)$ and $H(t)$ be $n \times n$ diagonal matrices where $Q_{ii} = q_i(t)$ and $H_{ii}(t) = h_i(t)$. The global system dynamics are thus described by

$$\begin{bmatrix} x(t+1) \\ s(t+1) \end{bmatrix} = (M_\Delta(t) \otimes I_{2\times 2}) \begin{bmatrix} x(t) \\ s(t) \end{bmatrix} \tag{2.21}$$

$$M_\Delta(t) = \left[\begin{array}{c|c} I - Q(t) & Q(t) \\ \hline H(t) & P(t) - H(t) \end{array}\right] \tag{2.22}$$

For all t, $M_\Delta(t) \in \mathcal{M}_\Delta$, where $\mathcal{M}_\Delta$ is a infinite set of matrices $M_\Delta(t)$ characterized by different values of $q(t)$, $h(t)$ and $P(t)$. Now we characterize the robustness of the proposed strategy with respect to measurement noise.

Theorem 2.2.3 *Consider a system as in eq. (2.21). Let us assume that $\forall t > 0$ there exists a $T > 0$ such that $\mathcal{G}_P([t, t+T))$ is connected . If the measurement noise $\Delta_i(t)$ is bounded by*

$$h + \varepsilon\delta_{max} - 1 \leq \Delta_i(t) \leq min\{h, (1-q)\}, \quad \forall i, t \tag{2.23}$$

then

$$\lim_{t\to\infty} \begin{bmatrix} x(t) \\ s(t) \end{bmatrix} = c\mathbf{1}_{2n}$$

where c is a constant.

Proof The diagonal entries of the matrices $I - Q(t)$ and $P(t) - H(t)$ are

$$[I - Q(t)]_{ii} = 1 - q - \Delta_i(t)$$

$$[P(t) - H(t)]_{ii} = 1 - \varepsilon\delta_i - h + \Delta_i(t)$$

We can assume that $q > \Delta_i(t)$. If condition (2.23) hold, then all matrices in $\mathcal{M}_\Delta$ are stochastic, because all entries are non negative and row sums equal to one. Thus, the proof follows as in theorem 2.2.2. ■

Note that $\Delta(t)$ could be positive or negative.

We now discuss what is the best parameter choice to achieve maximum robustness. Given a fixed value of q, the optimum value of h is the one which maximizes the following objective function:

$$\max_h \{\min\{h, (1-q), |h + \varepsilon\delta_{max} - 1|\}\}$$

By substitution it holds

- If $\dfrac{1-\varepsilon\delta_{max}}{2} \leq (1-q)$ the optimum value of h is $h = \frac{1-\delta_{max}}{2}$ thus the bound (2.23) becomes symmetric

$$-\frac{1-\varepsilon\delta_{max}}{2} \leq \Delta_i(t) \leq \frac{1-\varepsilon\delta_{max}}{2}, \quad \forall i, t$$

- If $\dfrac{1-\varepsilon\delta_{max}}{2} > (1-q)$ the optimum value of h is $h = (1-q)$. It holds

$$\varepsilon\delta_{max} - q \leq \Delta_i(t) \leq 1 - q, \quad \forall i, t$$

■

2.2.5 Convergence speed

We now characterize the convergence speed of the proposed strategy in the time-invariant case $M(t) = M$ and $P(t) = P$. Let Λ_M be the set of the $2n$ eigenvalues of M. As M is SIA, $\lambda = 1$ is a simple eigenvalue of Λ_M, and all other eigenvalues have module less than 1. The convergence speed of (2.16) depends on the second biggest module eigenvalue $\lambda_2 \in \Lambda$, which is called *algebraic connectivity*. By knowing the eigenvalues of P, Λ_M can be determined.

Theorem 2.2.4 *Let M be a 2×2 block matrix as in eq. (2.16). Let $\Lambda_P = \{\lambda_{p1}, \lambda_{p2}, \ldots, \lambda_{pn}\}$ be the set of the n eigenvalues of P. The $2n$ eigenvalues of M are function of the eigenvalues of P as follows:*

$$\begin{aligned} \lambda_{m_{i1,2}} &= \frac{(\lambda_p + 1 - h - q)}{2} \\ &\pm \frac{\sqrt{(\lambda_p + 1 - h - q)^2 - 4((1-q)\lambda_p - h)}}{2} \end{aligned} \tag{2.24}$$

Where $\lambda_{m_{i1,2}} \in \Lambda_M$ are the two eigenvalues corresponding to $\lambda_{pi} \in \Lambda_P$

Proof Following the work in (38) on how to compute the determinant of 2×2 block matrices as function of the blocks, we compute the eigenvalues of M solving $det\,(M - \lambda_m I) = 0$.

Since $(1-q)hI = h(1-q)I$

$$det\,(M - \lambda_m I) = det\,((1-q-\lambda I)(P - hI - \lambda I) - hqI)\,,$$

by some manipulations

$$det\left((\lambda^2 I - \lambda(P - hI - qI + I) + (1-q)P - hI)\right) = 0,$$

putting $(1-q-\lambda)$ in evidence:

$$(1-q-\lambda)^n det((\frac{\lambda^2 I - \lambda(1-h-q)I - hI)}{1-q-\lambda} + P))) = 0$$

for $(1-q-\lambda) \neq 0$,

$$det((\frac{\lambda^2 I - \lambda(1-h-q)I - hI)}{1-q-\lambda} + P))) = 0.$$

Now, let $\lambda_p = -\frac{\lambda^2 - \lambda(1-h-q)-h)}{1-q-\lambda}$. Since λ_p is the solution of $det(\lambda_p - P) = 0$, the eigenvalues of M as function of the eigenvalues of P are, after trivial manipulations, the solutions of

$$\lambda^2 - \lambda(1-h-q+\lambda_p) + (1-q)\lambda_p - h = 0$$

whose solutions are (2.24). ■

2.3 Formation control strategy in absence of a common reference frame

In the previous parts we have assumed that all the agents have a compass on board, which allows them to maintain a common orientation of the local reference frame. In this section we remove this assumption, thus each agent i belongs to a local reference frame $\Sigma_i = \{x_i, \theta_i\}$ centered on it, where x_i is he position of

the agent i and θ_i is the orientation of the x-axes with respect to the x-axes of the global reference frame. Under this new assumption the state of each agent i is described by the three state variables $\{x_i, s_i, \theta_i\}$. We modify the formation control strategy proposed in section 2.2 which is not suitable anymore to correctly control the system, by introducing an algorithm which leads the agent to reach a common reference direction. The new formation control strategy is characterized by:

(1) a rule to achieve agreement on a common reference direction;

(2) a rule to update the position of the agents;

(3) a rule to achieve agreement on a common reference point.

All the results in this section are presented with respect of the global reference frame Σ, and we assume that the agents are able to exchange local information. An interesting method which allows the agent to exchange local estimates of points and directions in absence of a common reference frame is presented in (14), thus we can assume that the agents exchange information by using it. Under this assumption, we don't need to modify the consensus algorithm on the network centroid, while the position update rule needs to take into account the variability of the target point due to the variability of the orientation of the orientation of the local reference frame.

This section is organized as follows: in the first part we characterize rule (1), then we characterize rule (2), by modifying the rule presented in section 2.2, and we point out the dependence of these rules from (1). Finally we describe the global formation control strategy.

2.3.1 Achieving consensus on a common reference direction

In order to lead the agents to reach a consensus on a common reference direction, we use Algorithm 1, originally proposed in (39), which allows the system to reach a global synchronization on a common heading. Algorithm 1 is based on a Gossip communication scheme: at each t a couple of nodes (i, j) such that $(i, j) \in \mathcal{E}(t)$ is randomly selected, and the selected nodes synchronize the orientation of their local reference frame by averaging on the shortest path arch between them. In

Algorithm 1 Gossip Algorithm for undirected graphs((39))

(i) At time t arch $(i,j) \in \mathcal{E}(t)$ is randomly selected.

(ii) Agents$i$ and j update the orientation of their local reference frame as follows:

- if $\max\{\theta_i(t), \theta_j(t)\} - \min\{\theta_i(t), \theta_j(t)\} \leq \frac{\pi}{2}$

$$\theta_i(t+1) = \theta_i(t+1) = \frac{\theta_i(t) + \theta_j(t)}{2}$$

- if $\max\{\theta_i(t), \theta_j(t)\} - \min\{\theta_i(t), \theta_j(t)\} > \frac{\pi}{2}$

$$\theta_i(t+1) = \theta_i(t+1) = \frac{\theta_i(t) + \theta_j(t)}{2} + \frac{\pi}{2}$$

- For each $a \in \mathcal{V}$ such that $a \neq i$ and $a \neq j$:

$$\theta_a(t+1) = \theta_a(t)$$

(39) a convergence analysis of Algorithm 1 is also provided: applying Algorithm 1 the set of agents globally asymptotically synchronize with probability 1.

2.3.2 Position update rule

The position update rule proposed in section 2.2 doesn't consider the orientation of the local reference frame $\theta_i(t)$ for each agent i, which may change among the time according to Algorithm 1. For each $i \in \mathcal{V}$, the estimated target point $d_i(t)$ and of the estimated common reference center $s_i(t)$ in global coordinates, at time t, depend on $\theta_i(t)$ as follows:

$$s_i(t) = x_i(t) + R_i(\theta_i(t))s_i^i(t) \tag{2.25}$$

and

$$d_i(t) = x_i(t) + R_i(\theta_i(t))(s_i^i(t) + D_i) = s_i(t) + R_i(\theta_i(t))D_i \tag{2.26}$$

where

$$R_i(\theta_i(t)) = \begin{bmatrix} \cos(\theta_i(t)) & -\sin(\theta_i(t)) \\ \sin(\theta_i(t)) & \cos(\theta_i(t)) \end{bmatrix}.$$

Following the same steps discussed in section 2.2.1, and introducing equations (2.25) and (2.26), we obtain the following position update rule:

$$\begin{cases} x_i(t+1) = (1-q)x_i(t) + qs_i(t) + qR_i(\theta_i(t))D_i \\ s_i(t+1) = (h)x_i(t) + (1-h)s_i(t) + (-h)R_i(\theta_i(t))D_i \end{cases} \tag{2.27}$$

2.3.3 Formation control strategy

Let us define now the column vector $D(\theta)$ as follows:

$$D(\theta) = \begin{bmatrix} R_1(\theta_1(t))D_1 \\ \vdots \\ R_n(\theta_n(t))D_n \end{bmatrix}$$

as the vector of the target point, which depend on the orientations of the local frames. The new formation control strategy can be expressed as follows:

$$\begin{bmatrix} x(t+1) \\ s(t+1) \end{bmatrix} = (M(t) \otimes I_{2\times 2}) \begin{bmatrix} x(t) \\ s(t) \end{bmatrix} + \begin{bmatrix} qD(\theta) \\ -hD(\theta) \end{bmatrix} \tag{2.28}$$

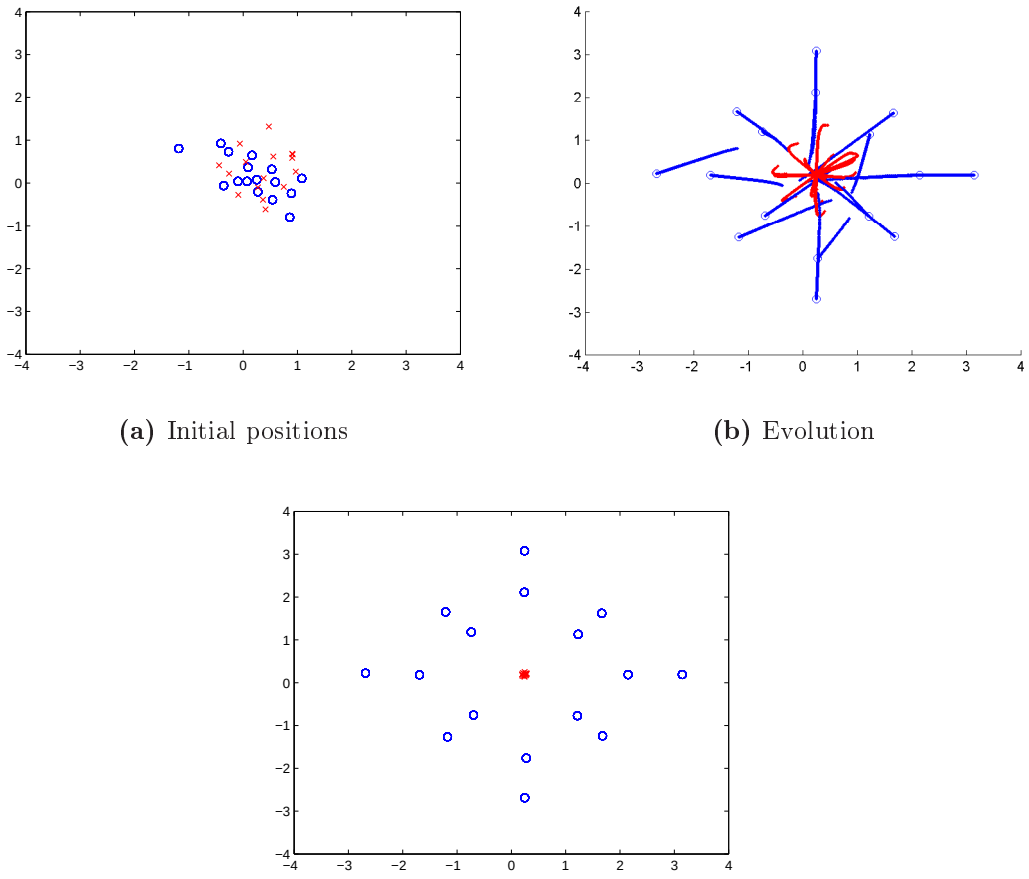
According to the assumptions made in this section, a given formation is considered to be achieved if

- $\forall i, j \in \mathcal{V}, \quad \theta_i(t) = \theta_j(t)$
- $x(t) = s(t) + D;$
- $\forall i, j \in \mathcal{V}, \quad \|s_i(t) - s_j(t)\| = 0$

The convergence of the agents to the desired formation depends on the convergence of Algorithm 1: a given formation cannot be achieved until all the local frames converge to a common orientation. In section 2.4 we provide a set of simulations which are useful to understand the behaviour of the system under the assumption made in this section.

2.4 Simulation results

In this part we present the results of some simulations with two purpose: validate the analytical results obtained in sections 2.2 and 2.2.4, and introduce some conjectures about the behavior of the system in the scenario described in section 2.3 which do not belong to the studied cases.



(a) Initial positions

(b) Evolution

(c) Final formation

Figure 2.3: Example of formation

2.4.1 Agents with a common reference direction

In Fig 2.3 an example of achievement of a desired formation using formation control strategy (2.13) is presented. The system is composed by a set of agents with a common reference direction, that are initially randomly scattered in a 2-D space as in Fig 2.3a. They exchange local information through a gossip communication scheme, and for all of them $q = 0.1$ and $h = 0.05$. The agents reach the desired formation (a crux shape) by following the trajectories showed in Fig 2.3b. The red lines represent the trajectories of the estimated common reference centers, while the blue lines are the trajectories of the agent.

Our simulations have pointed out an interesting phenomena: using formation control strategy (2.13), in a system of agent with a common reference direction, the desired formation is reached for each value of h in $-q < h < 0$, i.e., for values of h that do not respect condition (2.18). In other words, a small compensation is enough for the system to converge to the desired formation.

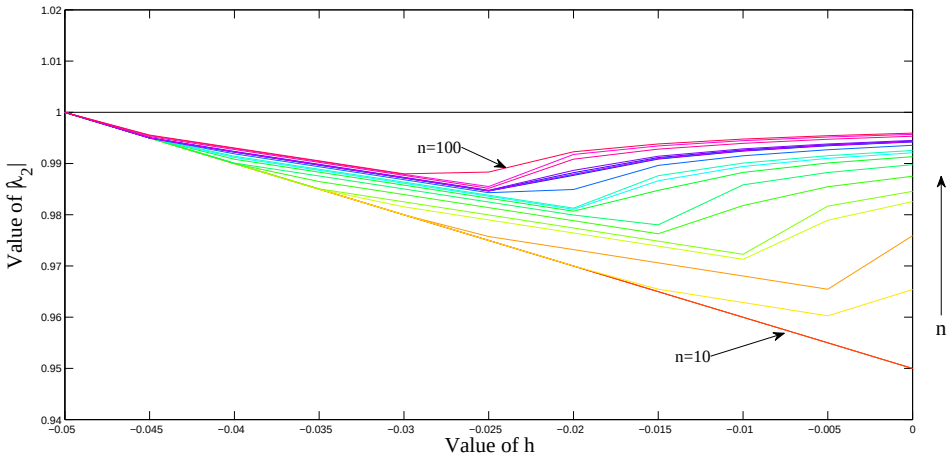


Figure 2.4: Value of $|\lambda_2|$ for $q = 0.15$, $-0.15 < h < 0$ and $n \in [10, 100]$

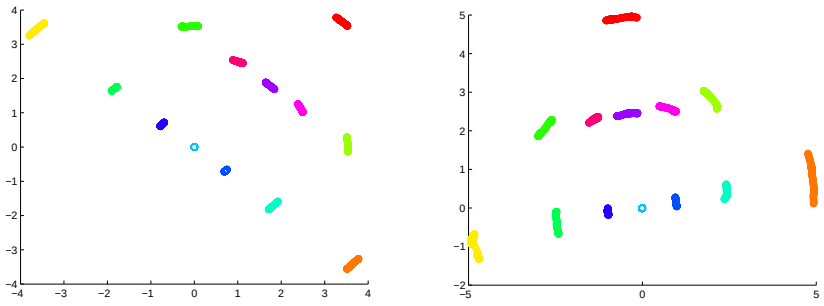
Fig 2.5 shows the the value of $|\lambda_2|$, i.e., the module of the second largest eigenvalue of the matrix $\mathcal{M}$, of a system of agents with $q = 0.15$, computed for $-q < h < 0$ and $n \in [10, 100]$. For all the simulations the topology of the network is connected and randomly generated. It can be observed that in case of no compensation, i.e., for $h = -q$, $|\lambda_2| = 1$, and the system is not stable, while for $-q < h < 0$ the second largest eigenvalue of $\mathcal{M}$ has a module smaller than one, and the system converge to the desired formation.

2.4.2 Agents in absence of common reference direction

Let us now consider the case of absence of common reference frame. In Section 2.3 we have characterized the algorithm which lead the agents to reach the target formation. We have supposed that the agents locally interact and exchange information using the method proposed in (14) which is based on the determination of the relative positions, i.e., relative distance and angles, and the correctness of the information exchange depends on the precision of this estimation. Here we

suppose that the relative localization is affected by an error, and we simulate the behavior of the system for different values of the error. The agents exchange information following a gossip communication scheme. The estimations of the relative distance d and the relative angle of view γ are affected by a uniformly distributed random error with a maximum amplitude $|\Delta d_m| = \alpha_d d$ and $|\Delta \gamma_m| = \alpha_\gamma \gamma$. In Fig. 2.5 is represented a system of 13 agents with a triangle-shape target formation. In Fig. 2.5a is $\alpha_d = \alpha_\gamma = 0.01$, while in Fig. 2.5b is $\alpha_d = \alpha_\gamma = 0.02$. It can be observed that each agent makes a random walk around its target position. The amplitude of the random walk grows as:

- α_d and α_γ grow;
- the distance of the target point from the estimated common reference center grows.



(a) $|\Delta d_m| = 0.01d$, $|\Delta \gamma_m| = 0.01\gamma$

(b) $|\Delta d_m| = 0.02d$, $|\Delta \gamma_m| = 0.02\gamma$

Figure 2.5: Example of formation

The same behavior can be observed in Fig 2.6, where the average amplitude of the random walk is reported for different values of α_d and α_γ. Each value is the average of 20 simulations, and for each simulation the initial positions and the local orientations of the agents were randomly generated. Moreover, Fig 2.6 show that in absence of errors in the relative localization, the system converge to the desired formation.

Target Point	$\lvert\Delta d_m\rvert = 0$ $\lvert\Delta\gamma_m\rvert = 0$	$\lvert\Delta d_m\rvert = 0.005$ $\lvert\Delta\gamma_m\rvert = 0.005\gamma$	$\lvert\Delta d_m\rvert = 0.01d$ $\lvert\Delta\gamma_m\rvert = 0.01\gamma$	$\lvert\Delta d_m\rvert = 0.015d$ $\lvert\Delta\gamma_m\rvert = 0.015\gamma$	$\lvert\Delta d_m\rvert = 0.02d$ $\lvert\Delta\gamma_m\rvert = 0.02\gamma$	$\lvert\Delta d_m\rvert = 0.025d$ $\lvert\Delta\gamma_m\rvert = 0.025\gamma$
$(0, 0)$	0	0.01	0.02	0.04	0.04	0.09
$(1, 0)$	0	0.10	0.15	0.25	0.35	0.45
$(-1, 0)$	0	0.09	0.18	0.28	0.30	0.44
$(-2.5, 0)$	0	0.21	0.46	0.71	0.69	1.21
$(2.5, 0)$	0	0.26	0.37	0.73	0.75	1.18
$(0, 2.5)$	0	0.23	0.43	0.74	0.85	1.06
$(1, 2.5)$	0	0.32	0.45	0.80	0.82	1.35
$(-1, 2.5)$	0	0.28	0.51	0.69	0.76	1.17
$(2.5, 2.5)$	0	0.29	0.58	1.05	1.02	1.62
$(-2.5, 2.5)$	0	0.30	0.60	1.07	1.24	1.51
$(0, 5)$	0	0.5	0.73	1.24	1.36	2.38
$(5, 0)$	0	0.58	0.9	1.44	1.51	2.14
$(-5, 0)$	0	0.49	0.89	1.44	1.40	2.01

Figure 2.6: Amplitude of the random walk for different values of α_d and α_γ

2.5 Conclusions

In this Chapter we firstly have proposed a novel coordination strategy, based on an overcompensation of agent displacement, to achieve an arbitrary formation in a multi-agent system. We have proved that our strategy is robust with respect to measurement noise of odometry or inertial navigation. Our strategy is characterized by a decentralized algorithm to achieve agreement on a common reference point and a consensus based strategy to provide cohesion in the network. The system achieves arbitrary formations by specifying positions in the estimated common reference frame on which the agents agree upon. Then we have extended our strategy to a multi vehicle system in absence of a common reference frame. Our future objective is to find analytical support for the extended strategy, whose performances are studied only through simulations so far.

Chapter 3

The Heterogeneous Multi Vehicle Routing Problem

This Chapter is structured as follows. In Section 3.1 the HMVRP is formalized. In Section 3.2 the HMVRP is discussed and solved via a centralized optimization based on MILP. In Section 3.3 a decentralized algorithm is proposed and characterized. In Section 3.4 an heuristic approach to solve the HMVRP is proposed, and in Section 3.5 simulations are shown to corroborate the analytical results presented in the previous sections. Finally, in Section 3.6 conclusions and future directions are discussed.

3.1 Problem statement

Consider a set $\mathcal{N}$ of n mobile robots scattered in a connected region $\mathcal{R}$ in a plane. Let $\mathcal{K}$ be a set of k tasks scattered in region $\mathcal{R}$, that should be assigned to robots to be executed.

Robots move at different speeds and have different execution speeds of tasks. Tasks have different costs. In particular, the following notation is used:

- v_r is the speed of robot R_r,
- w_r is the task execution speed of robot R_r,
- v_{min} (v_{max}) is the minimum (maximum) speed of robots,

- w_{min} (w_{max}) is the minimum (maximum) task execution speed of robots,
- c_i is the cost of the i-th task,
- c_{min} (c_{max}) is the minimum (maximum) cost of tasks.

Moreover, d_{max} is the maximum length of the shortest path between any two points in the region $\mathcal{R}$.

Robots are supposed to first coordinate themselves to decide upon their task assignment and then start to serve the tasks autonomously.

To use a notation that is standard in the literature, we assume that robots are initially positioned in depots and should go back to them after the execution of tasks. The set of depots is called $\mathcal{D}$ and the generic r-th depot is D_r.

Now, if $\mathcal{K}_r$ denotes the set of tasks assigned to robot R_r, our goal is to minimize the objective function:

$$J = \max_{r \in \mathcal{N}} J_r = \left(\frac{TSP(\mathcal{K}_r \cup \{D_r\})}{v_r} + \frac{\sum_{i \in \mathcal{K}_r} c_i}{w_r} \right) \tag{3.1}$$

where TSP ($\mathcal{K}_r \cup \{D_r\}$) is the minimum TSP tour length of robot R_r that, initially positioned in D_r, visits all tasks in $\mathcal{K}_r$ and go back to D_r.

In simple words we want to minimize the maximum execution time of the n robots that have to visit and execute all tasks assigned to them, guaranteeing that each task is executed by exactly one robot.

The above problem can be seen as a generalization of the classical multi-TSP problem. First, because we are also assuming that tasks should not only be visited by the robots, but should be processed by them. Secondly, because the optimization is carried out over an heterogeneous network due to the heterogeneity of the agents and the tasks. Similar problems have been recently addressed in the literature, see e.g. (30), but to the best of our knowledge, never under the assumption of heterogeneous agents and tasks.

Let us conclude this section with the introduction of some notation that will be used in the remaining of the chapter. Let $\mathcal{K}_r$ be the set of tasks assigned to robot R_r. We denote as $\tilde{\mathcal{K}}_r$ the ordered set with the same elements of $\mathcal{K}_r$, but whose ordering specifies the order in which tasks in $\mathcal{K}_r$ are visited by robot R_r. Therefore, sets $\tilde{\mathcal{K}}_r$ are the unknown variables of the optimization problem (3.1).

Finally, let $\tilde{\mathcal{K}} = \{\tilde{\mathcal{K}}_1, \ldots, \tilde{\mathcal{K}}_n\}$ be an ordered set of n ordered sets, that summarizes the generic solution of the considered tasks allocation problem. The set $\tilde{\mathcal{K}}$ is called *network state*.

3.2 Optimal centralized solution

In this section we first discuss a centralized strategy that leads to an optimal solution of the above task assignment problem. Such an approach is based on mixed linear integer programming (MILP). Then we provide a characterization of the optimal solution in terms of an upper and a lower bound on the optimal value of the objective function. This will be useful when evaluating the effectiveness of the decentralized approach proposed in the next section.

To represent all possible directed tours of n robots, let us define a complete directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where:

- $\mathcal{V} = \mathcal{N} \cup \mathcal{K}$ is the set of $n + k$ nodes;
- $\mathcal{E} = (\mathcal{N} \cup \mathcal{K}) \times (\mathcal{N} \cup \mathcal{K})$ is the set of $(n + k)^2$ edges representing directed paths from the depots in which robots are initially placed to tasks, and viz, and from tasks to tasks[1].

Moreover, we define the following binary variables that completely identify a task allocation and the order in which tasks are executed by robots. In simple words they completely identify a network state $\tilde{\mathcal{K}}$. Since we want to minimize the total execution times of robots, we always assume that distances among tasks, and among tasks and depots, are covered through straight lines.

- We assign n binary variables x_{ir} to each node $i \in \mathcal{V}$; here $r \in \mathcal{N}$: if $i \in \mathcal{N}$, $x_{ir} = 1$ means that robot R_r starts its tour from node i, while if $i \in \mathcal{K}$, $x_{ir} = 1$ means that task i is executed by robot R_r.
- We assign n binary variables y_{ijr} to each edge $(i, j) \in \mathcal{E}$; here $r \in \mathcal{N}$: $y_{ijr} = 1$ means that robot R_r goes directly from node i to node j in its path.

[1] In the sets $\mathcal{V}$ and $\mathcal{E}$ the generic r-th depot is identified via the r-th element in $\mathcal{N}$. This has been done for clearity of presentation as it will appear in the following.

Moreover, we introduce the following cost coefficients.

- We assign n costs $c_{ir} = c_i / w_r$ to each node $i \in \mathcal{K}$; here $r \in \mathcal{N}$: c_{ir} represents the execution time of task i by robot R_r with an execution speed of w_r.
- We assign n costs $d_{ijr} = l_{ij} / v_r$ to each edge $(i,j) \in \mathcal{E}$; here $r \in \mathcal{N}$: d_{ijr} represents the time spent by robot R_r to pass the length l_{ij} of edge (i,j) with speed v_r.

Proposition 3.2.1 *Let us consider the allocation problem formalized in Section 3.1. An optimal solution can be computed solving the following MILP problem:*

$$
\begin{cases}
J = \min \lambda & & \\
s.t. & & \\
\sum_{i \in \mathcal{K}} x_{ir} c_{ir} + \sum_{(i,j) \in \mathcal{E}} d_{ijr} y_{ijr} < \lambda, & \forall r \in \mathcal{N} & (a) \\
x_{rr} = 1, & \forall r \in \mathcal{N} & (b) \\
\sum_{r \in \mathcal{N}} x_{ir} = 1, & \forall i \in \mathcal{K} & (c) \\
\sum_{j \in \mathcal{V}} y_{jir} = \sum_{j \in \mathcal{V}} y_{ijr} = x_{ir}, & \forall i \in \mathcal{V},\ \forall r \in \mathcal{N} & (d) \\
\sum_{i \notin \mathcal{S}} \sum_{j \in \mathcal{S}} y_{ijr} \geq x_{qr} & \forall \mathcal{S} \subseteq \mathcal{K}, & \\
 & \forall q \in \mathcal{S},\ \forall r \in \mathcal{N} & (e) \\
\lambda \in \mathbb{R} & & (f) \\
x_{ir} \in \{0,1\} & \forall i \in \mathcal{V},\ \forall r \in \mathcal{N} & (g) \\
y_{ijr} \in \{0,1\} & \forall (i,j) \in \mathcal{E},\ \forall r \in \mathcal{N}. & (h)
\end{cases}
$$

Proof: The proof is carried out via a detailed explanation of all the constraints and the objective function.

— *Constraints (a)* and *objective function:* The left hand side term of (a) is equal to the total execution time of robot R_r. Thus, given the objective function, constraints (a) aim to minimize the maximum execution time of robots.

— *Constraints (b):* These constraints force each robot to move from its initial position (depot).

— *Constraints (c):* Each task i must be executed by exactly one robot.

— *Constraints (d):* If robot R_r executes task i, it must arrive at node i in some way and at the end of the execution has to leave it. The same holds if node i models a depot, i.e., $i \in \mathcal{N}$.

— *Constraints (e):* Each robot R_r has to make a single connected tour visiting all its tasks, so we have to exclude all the disjoint paths. In words constraint (e) relative to robot R_r, imposes that if robot R_r executes a task $i \in \mathcal{S} \subseteq \mathcal{K}$, there must be an edge passed by R_r to enter in $\mathcal{S}$. These constraints are named *Subtour Elimination Constraints* (SEC) and are typical of vehicle routing problems and TSP models (20). □

The number of unknowns in the MILP (3.2.1) is equal to

$$N = n(n+k)^2 + n(n+k) + 1 = \mathcal{O}(n^3 + nk^2 + n^2k).$$

The total number of constraints is $\mathcal{O}(n^2k+nk2^k)$. Indeed we have n constraints of type (a), n constraints of type (b), k constraints of type (c), $(n+k)n$ constraints of type (d), and $n\sum_{i=1}^{k} i\frac{k!}{(k-i)!i!} \leq nk2^k$ constraints of type (e).

The following two theorems provide a characterization of the optimal value of the performance index J^*.

Theorem 3.2.2 *The optimal solution J^* of the objective function (3.1) is upper bounded by*

$$J^* \leq C_{up} + D_{up} \tag{3.2}$$

where

$$C_{up} = \frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{min}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}\right), \tag{3.3}$$

$$D_{up} = 2\frac{d_{max}}{v_{min}} + \frac{c_{max}}{w_{min}}. \tag{3.4}$$

Proof: The proof is based on an heuristic that can be summarized in the following main steps.

- Generate an optimal tour that visits all tasks. Obviously, if an agent with speed v_{min} and execution speed w_{min} follows the tour and executes all tasks, its service time is equal to

$$\hat{J} = \left(\frac{TSP(\mathcal{K})}{v_{min}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}\right).$$

- Divide the tour in n consecutive sub-tours using the following rule. Take a robot (e.g. R_1) at random and make it follow the route of the optimal single vehicle tour at the previous item, starting from the position of an arbitrary task. Stop it as soon as its service time $\hat{J}_1$ satisfies the condition $\hat{J}_1 \geq \hat{J}/n$. Now, since the largest cost of tasks is equal to c_{max}, the smallest execution speed of robots is w_{min}, and the time taken to travel between tasks is continuous, it is

$$\hat{J}_1 \leq \frac{\hat{J}}{n} + \frac{c_{max}}{w_{min}}.$$

  Select at random a new robot (e.g. R_2) and put it at the end of the route of R_1 and repeat the same strategy, until all robots are considered. If there aren't enough tasks for the robots, simply consider null the service time for the remaining robots.

- Now, if d_{max} is the maximum length of the shortest path between any two points in the region $\mathcal{R}$, the execution time J_r of each robot R_r is such that $J_r \leq \hat{J}_r + 2d_{max}/v_{min}$. Indeed the total service time of each robot corresponds to the time it takes to complete its sub-tour along the route of the optimal single vehicle TSP, plus the time to go from its depot to its first task and go back to the depot. Therefore, it is

$$J_r \leq \frac{\hat{J}}{n} + \frac{c_{max}}{w_{min}} + 2\frac{d_{max}}{v_{min}}, \qquad \forall r \in \mathcal{N}.$$

Since the optimal solution J^* of the objective function (3.1) can only be smaller or equal than the solution resulting from the above heuristic, for sure it is

$$J^* \leq \max_{r \in \mathcal{N}} J_r \leq \frac{\hat{J}}{n} + \frac{c_{max}}{w_{min}} + 2\frac{d_{max}}{v_{min}} = C_{up} + D_{up}$$

thus proving the correctness of the upper bound. □

Theorem 3.2.3 *The optimal solution J^* of the objective function (3.1) is lower bounded by*

$$J^* \geq C_{lo} - D_{lo} \tag{3.5}$$

where

$$C_{lo} = \frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{max}} + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}}\right), \tag{3.6}$$

$$D_{lo} = \frac{d_{max}}{v_{min}}. \tag{3.7}$$

Proof: Let $S_{opt} = \sum_{r \in \mathcal{N}} J_r^*$ be the sum of all the service times corresponding to an optimal task assignment. Since, by definition $J^* = \max_{r \in \mathcal{N}} J_r^*$, obviously it is

$$J^* \geq \frac{S_{opt}}{n}. \tag{3.8}$$

Now, let S_{opt}^p be the sum of the contributions to J_r^*, with $r \in \mathcal{N}$, relative to the only time spent moving from one task to another one, or from/toward the depots, without including the time spent to execute tasks.

Obviously, it is

$$S_{opt} \geq S_{opt}^p + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}}. \tag{3.9}$$

Moreover, trivially generalizing the result in (30) to the case of heterogeneous robots, we have that

$$S_{opt}^p + \frac{TSP(\mathcal{D})}{v_{min}} \geq \frac{TSP(\mathcal{D} \cup \mathcal{K})}{v_{max}} \geq \frac{TSP(\mathcal{K})}{v_{max}} \tag{3.10}$$

or equivalently

$$S_{opt}^p \geq \frac{TSP(\mathcal{K})}{v_{max}} - \frac{TSP(\mathcal{D})}{v_{min}}. \tag{3.11}$$

By equations (3.9) and (3.11) it follows that

$$\begin{aligned} S_{opt} &\geq \frac{TSP(\mathcal{K})}{v_{max}} - \frac{TSP(\mathcal{D})}{v_{min}} + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}} \\ &\geq \frac{TSP(\mathcal{K})}{v_{max}} - n\frac{d_{max}}{v_{min}} + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}}. \end{aligned} \tag{3.12}$$

Finally, by equations (3.8) and (3.12), it is

$$\begin{aligned} J^* \geq \frac{S_{opt}}{n} &= \frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{max}} + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}}\right) - \frac{d_{max}}{v_{min}} \\ &= C_{lo} - D_{lo} \end{aligned} \tag{3.13}$$

thus proving the statement. □

3.3 Decentralized solution based on optimal local task assignment

In this section we first propose a decentralized approach to solve the task allocation problem in Section 3.1 that is based on gossip. Then, a comparison among the computational complexity of the proposed algorithm and the centralized algorithm is provided. Convergence properties of the gossip algorithm are discussed. Finally, some characterizations of the solution obtained via the decentralized approach are proposed.

3.3.1 MILP Gossip algorithm

The idea of the proposed decentralized algorithm is that robots locally balance their loads according to a gossip interaction rule, i.e., via pairwise communications, under the following main assumption:

(A1) All robots may interact with all the other robots.

Starting from an initial task assignment, e.g., assuming that robots have the same number of tasks, a couple of robots is selected at random. Selected robots optimally balance their load; a new couple of robots is selected and so on, until no better balancing among robots can be obtained. This can be summarized in Algorithm 1. The variable T_{max} denotes a maximum number of steps to be executed that is assumed to be large enough so that no further improvement of the objective function can be obtained.

3.3.2 Computational complexity of the local optimization

Let us now discuss the advantages in terms of computational complexity coming from local optimizations using Algorithm 2 with respect to a centralized optimization.

To this aim, let us first present some preliminary results. In particular, the following proposition ensures that when the number of iterations of Algorithm 2 increases, the optimal value of the objective function can never increase. Obviously this does not imply that an optimal solution is obtained.

Algorithm 2 MILP Gossip algorithm

(i) Tasks are initially assigned to robots so that each robot has either k/n or $k/n + 1$ tasks.

(ii) Let $t = 0$.

(iii) While $t \leq T_{max}$

(a) Choose at random two robots r and q. Let them solve the MILP (3.2.1) where $\mathcal{N} = \{r, q\}$ and $\mathcal{K} = \mathcal{K}_r \cup \mathcal{K}_q$.

(b) If the new task assignment leads to a smaller total execution time, then update the assignments of robots r and q accordingly,

else leave them unchanged.

(c) Let $t = t + 1$ and go back to Step 3.

(iv) All robots process their own set of tasks following the order specified by the optimal local solution.

Proposition 3.3.1 *Let $J_{gossip}(t)$ be the maximum execution time of robots computed after t iterations of Algorithm 2. For any $t \geq 0$, it is $J_{gossip}(t+1) \leq J_{gossip}(t)$.*

Proof: Let R_r and R_q be the two robots selected at time $t+1$. By Algorithm 2 this means that only the tasks allocation of such robots may change, while the load of all the other robots keeps unaltered. Now, since at Step 3.a of Algorithm 2 tasks are assigned to robots R_r and R_q so as to minimize the maximum execution time among them, this implies that the maximum execution time among R_r and R_q either decreases or it keeps unaltered at time $t+1$. Moreover, the maximum execution time among all robots may decrease at time $t+1$ if and only if either R_r or R_q, or both, are the robots to which it corresponds the maximum execution time among all robots at time t. Indeed with no loss of generality, we may assume that R_r is the "critical" robot at time t, i.e., the robot to which it corresponds the maximum execution time among all robots at time t. Three different cases may occur at time $t+1$, after the new tasks allocation. First, R_r may still be the robot with the maximum execution time, but in such a case for sure, its execution time cannot be larger than that at time t. Secondly, robot R_q may be at time $t+1$ the robot with the maximum execution time but for sure its execution time cannot be larger than that of robot R_r at time t. Finally, at time $t+1$, neither to R_r nor to R_q it corresponds the maximum execution time among robots. This implies that a third robot, e.g., R_p, has become the critical one at time $t+1$. In any case for sure its execution time is smaller than that of robot R_r at time t, since by assumption robot R_r was the critical robot at time t. □

Let us now provide an upper bound on the value of the maximum execution time of robots resulting from Algorithm 2 at a generic iteration t. To this aim, we first recall some deterministic upper bounds to the maximum length of the shortest path (SP) between a set $\mathcal{K}$ of k locations in a unit square area, that are due to (40) and (41), respectively:

$$SP(\mathcal{K}) \leq \sqrt{2}\sqrt{k} + 7/4, \tag{3.14}$$

and

$$SP(\mathcal{K}) \leq 0.984\sqrt{2}\sqrt{k} + 11. \tag{3.15}$$

To the best of our knowledge the above two upper bounds are the best actually proposed in the literature. Moreover, we cannot a priori say which of the

above bounds is the most strict one. Indeed the bound in (41) has a smaller multiplicative factor with respect to (40), but has a larger additive constant. In the following, we focus on upper bound (3.14), but obviously similar results can be repeated considering (3.15).

Proposition 3.3.2 *Let $J_{gossip}(t)$ be the maximum execution time of robots computed after t iterations of Algorithm 2, then $\forall\ t \geq 0$ it is*

$$J_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+2}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}}+\left(\frac{k}{n}+1\right)\frac{c_{max}}{w_{min}}.$$

Proof: By Algorithm 2 at time $t=0$ the maximum number of tasks that can be assigned to a robot is equal to $k/n+1$. Moreover, since each robot starts its path from its depot and has to come back to it, then by equation (3.14), for any $r \in \mathcal{N}$ it is

$$TSP(\mathcal{K}_r(0)\cup\{D_r\}) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+2}+\frac{7}{4}+\sqrt{2}\right)d_{max}. \tag{3.16}$$

Note that the additional term $\sqrt{2}$ between parenthesis comes from the fact that to form a Euclidean TSP tour from a path in a unit square it is sufficient to connect the start and end point to form a cycle, thus increasing the size of the path of at most $\sqrt{2}$ in the unit square. Moreover, d_{max} comes from the fact that in our problem statement depots and robots are not distributed in a square of unitary edge, but in a region $\mathcal{R}$ that is contained in a square of edge d_{max} being by definition d_{max} the maximum length of the shortest path between any two points in $\mathcal{R}$.

Finally, since by assumption $\sum_{i\in\mathcal{K}_r(0)} c_i \leq \left(\frac{k}{n}+1\right)c_{max}$, it follows that

$$J_{gossip}(0) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+2}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}}+\left(\frac{k}{n}+1\right)\frac{c_{max}}{w_{min}}$$

that proves the statement being by Proposition 3.3.1 $J_{gossip}(t) \leq J_{gossip}(0)$ for all $t \geq 0$. □

Let us now provide a proposition that characterizes the maximum number of tasks that are assigned to robots at a generic iteration t of Algorithm 2.

Proposition 3.3.3 *Let $K_{max}(t) = \max_{r\in\mathcal{N}} |\mathcal{K}_r(t)|$ be the maximum number of tasks that are assigned to robots at a generic iteration t of Algorithm 2. For any $t \geq 0$ it is:*

$$K_{max}(t) \leq \frac{w_{max}}{c_{min}} \left[\left(\sqrt{2}\sqrt{\frac{k}{n}+2} + \frac{7}{4} + \sqrt{2} \right) \frac{d_{max}}{v_{min}} + \left(\frac{k}{n} + 1 \right) \frac{c_{max}}{w_{min}} \right]. \tag{3.17}$$

Proof: By Proposition 3.3.2, for all $t \geq 0$, it holds

$$J_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+2} + \frac{7}{4} + \sqrt{2} \right) \frac{d_{max}}{v_{min}} + \left(\frac{k}{n} + 1 \right) \frac{c_{max}}{w_{min}}. \tag{3.18}$$

Now, it is

$$J_{gossip}(t) \geq \frac{K_{max}(t) c_{min}}{w_{max}} \tag{3.19}$$

since the execution time of $K_{max}(t)$ tasks is greater or equal than that we have if such tasks are at a null distance from the robot that has to process them, all tasks have a cost equal to c_{min} and the robot who process them has an execution speed equal to w_{max}. By equations (3.18) and (3.19) the statement of the proposition follows. □

An important remark needs to be done. The above proposition provides an upper bound on the maximum number of tasks that can be assigned to a robot at any iteration. For particular values of the parameters it may happen that the upper bound given by Proposition 3.3.3 is not significant because it is larger than k. However, this only occurs for very particular cases, while for most of the significant and general situations where the number of tasks is sufficiently large, robots and tasks are sufficiently distributed in $\mathcal{R}$ and their costs and speeds are in reasonable ratio, Proposition 3.3.3 enables us to conclude that

$$K_{max}(t) = \mathcal{O}(k/n).$$

Now, since local optimization considers two robots at a time, the number of tasks that are involved in a local optimization is surely smaller or equal than $2K_{max}(t)$. This means that the number of unknowns of the MILP that should be solved at the generic iteration t of Algorithm 2 is

$$N_{gossip} = \mathcal{O}(k^2/n^2)$$

rather than $N = \mathcal{O}(n^3+nk^2+n^2k)$ as in the centralized case. Moreover, the number of constraints is $\mathcal{O}(k2^{k/n}/n)$ rather than $\mathcal{O}(n^2k + nk2^k)$ as in the centralized case.

3.3.3 Finite time and almost sure convergence

We now introduce two definitions to formalize two important properties of gossip communication schemes, namely *deterministic persistence* and *stochastic persistence*. Similar definitions have been recently proposed in (42). As usual in this framework, we assume that the possible interactions among agents are modeled by an undirected graph $G = \{V, E\}$ where agents correspond to vertices, and an edge exists if and only if the interaction among the agents corresponding to the incidence nodes is possible. Obviously, assumption **(A1)** implies that in our case it is $E = V \times V$. At each iteration t of the gossip algorithm a different edge is selected. In the following we denote as $e(t)$ the edge selected at time t, while the set of edges selected in the time interval $[t_1, t_2]$ is denoted as $\bar{e}(t_1, t_2)$, i.e., we have

$$\bar{e}(t_1, t_2) = \bigcup_{t=t_1}^{t_2} e(t).$$

Definition 1 (Deterministic persistence)

A gossip communication scheme is said to be deterministically persistent *if* $\forall t \geq 0$ *there exists a finite* $T > 0$ *such that*

$$\forall e' \in E, \qquad Pr(e' \in \bar{e}(t, t+T)) = 1$$

or equivalently, $\bar{e}(t, t+T) = E$. ■

Deterministic persistence implies that, if we consider a finite but sufficiently large time interval, then for sure all arcs are selected at least once during such interval.

Definition 2 (Stochastic persistence)

A gossip communication scheme is said to be stochastically persistent *if* $\forall t \geq 0$ *there exists a finite* $T > 0$ *and a probability* $p \in (0, 1)$ *such that*

$$\forall e' \in E, \qquad Pr(e' \in \bar{e}(t, t+T)) \geq p$$

where $Pr(\cdot)$ *denotes a probability.* ■

In simple words, stochastic persistence implies that, if we consider a finite but sufficiently large time interval, then each edge has a probability greater or equal than a finite value p of being selected during such an interval.

Theorem 3.3.4 *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 2. If the gossip communication scheme satisfies the* deterministic persistence *property then, for every initial task assignment, there exists a network state $\tilde{\mathcal{K}}^*_{gossip}$ and a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{gossip}$, for all $t \geq T$.*

Proof: Let us present some preliminary comments.

— First, $\tilde{\mathcal{K}}^*_{gossip}$ is an invariant network state for the state evolution following Algorithm 2. This follows from Step 3.b of Algorithm 2.

— Secondly, if at a given time the network state is updated then the previous network state is no more visited during the algorithm evolution. This also follows from Step 3.b of Algorithm 2 and the monotonicity property expressed by Proposition 3.3.1.

— Thirdly, the number $N_{n,k}$ of admissible network states is finite since both the number of robots and the number of tasks are finite.

Now, with no loss of generality we assume that at the initial time $t = 0$ it is $\tilde{\mathcal{K}}_r \neq \tilde{\mathcal{K}}^*_{gossip,r}$ for all $r = 1, \ldots, n$, i.e., no robot is in its final assignment. If the communication scheme among agents is deterministically persistent, since the graph modeling the possible interactions among robots is fully connected and the number $N_{n,k}$ of admissible network states is finite, then for sure after some finite time T_0 the robot with the maximum cost in the final assignment reaches its final assignment. Let R_r be such a robot. By Step 3.b of Algorithm 2 this implies that the assignment of R_r is no more changed during the algorithm evolution, i.e., $\tilde{\mathcal{K}}_r(t) = \tilde{\mathcal{K}}^*_{gossip,r}$ for all $t \geq T_0$.

Analogously, after some further finite time T_1 the final assignment is reached by the robot with the second largest cost, and so on, until all robots have reached their final assignment. Since all T_i's are finite, this proves that the final network state $\tilde{\mathcal{K}}^*_{gossip}$ is reached in a finite time $T = \sum_{i=1}^{n} T_i$. □

Theorem 3.3.5 *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 2. If the gossip communication scheme satisfies the* stochastic

persistence *property, then, for every initial task assignment, there exists a network state* $\tilde{\mathcal{K}}^*_{gossip}$ *and almost surely a finite time* $T > 0$ *such that* $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{gossip}$ *for all* $t \geq T$*, i.e., the network state converges almost surely in finite time to* $\tilde{\mathcal{K}}^*_{gossip}$*.*

Proof: We prove this theorem following the same arguments an in (43). The proof is based on verifying the following three facts:

(i) $\tilde{\mathcal{K}}^*_{gossip}$ is an invariant network state for the state evolution following Algorithm 2;

(ii) $\tilde{\mathcal{K}}(t)$ is a Markov process on a finite number of states;

(iii) starting from any initial network state $\tilde{\mathcal{K}}(0)$, there is a positive probability for the network state to reach $\tilde{\mathcal{K}}^*_{gossip}$ in a finite number of steps.

Let us now check the above three properties in order.

— (i) As already discussed in Theorem 3.3.4, this follows from Step 3.b of Algorithm 2.

— (ii) As already discussed in the proof of Theorem 3.3.4, the number of admissible network states $N_{n,k}$ is finite, being finite both the number of robots and the number of tasks. Markovianity immediately follows from the fact that subsequent random selection of edges are independent.

— (iii) This issue can be proved using similar arguments as in Theorem 3.3.4 with the only difference that now the communication scheme is stochastically persistent, rather than deterministically persistent. This implies that for any initial network state $\tilde{\mathcal{K}}(0)$ there is a finite probability that after some finite time T_0 the robot with the maximum cost in the final assignment reaches its final assignment, that is no more changed during the algorithm evolution. The same holds for the robot with the second largest execution cost in the final assignment, and so, until the invariant network state $\tilde{\mathcal{K}}^*_{gossip}$ is reached. Since the number of possible states is finite, item (iii) holds. □

3.3.4 Performance characterization of the MILP algorithm

Algorithm 2 does not guarantee the convergence to an optimal solution. However, some results can be given to characterize its solution at the equilibrium, i.e.,

after a number of iterations that is sufficiently large so that no better balancing among robots may be obtained. In particular, the following theorem provides a characterization of the maximum distance among the processing times of robots that have locally balanced their loads.

Theorem 3.3.6 *Let $J^*_{gossip,r}$ and $J^*_{gossip,q}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Algorithm 2. It holds*

$$|J^*_{gossip,r} - J^*_{gossip,q}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}} \tag{3.20}$$

where d^{rq}_{max} is the maximum distance among tasks in $\mathcal{K}_r$ and tasks in $\mathcal{K}_q$, $v^{rq}_{min} = \min\{v_r, v_q\}$, and $w^{rq}_{min} = \min\{w_r, w_q\}$.

Proof: We prove the statement by contradiction, i.e., we assume that

$$|J^*_{gossip,r} - J^*_{gossip,q}| > K_{rq}.$$

With no loss of generality, we assume that it is $J^*_{gossip,r} > J^*_{gossip,q}$. Now, let z be the task in $\mathcal{K}_r$ whose distance with respect to tasks in $\mathcal{K}_q$ is minimum. Remove z from $\mathcal{K}_r$ and put it in $\mathcal{K}_q$. Let $\tilde{J}_r$ and $\tilde{J}_q$ be the resulting execution times of robots r and q, respectively. Obviously, we have

$$\tilde{J}_q \leq J^*_{gossip,q} + \frac{c_z}{w_q} + 2\frac{d^{rq}_{max}}{v_q} = J^*_{gossip,q} + K_{rq} \tag{3.21}$$

where the inequality follows from the fact that the optimal TSP of robot q is surely smaller than the path obtained by simply adding twice the path from the closest task in $\mathcal{K}_q$ to z. Now, by the contradictory assumption, we have

$$J^*_{gossip,r} > J^*_{gossip,q} + K_{rq} \tag{3.22}$$

thus (3.21) can be rewritten as

$$\tilde{J}_q < J^*_{gossip,r}. \tag{3.23}$$

As a consequence

$$\max\{\tilde{J}_q, \tilde{J}_r\} < \max\{J^*_{gossip,q}, J^*_{gossip,r}\}. \tag{3.24}$$

However, this contradicts the assumption that $J^*_{gossip,r}$ and $J^*_{gossip,q}$ are the time executions corresponding to an optimal task assignment, thus proving the statement. □

Corollary 3.3.7 *Let $J^*_{gossip,r}$ and $J^*_{gossip,q}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Algorithm 2. It holds*

$$|J^*_{gossip,r} - J^*_{gossip,q}| \leq D_{up} \tag{3.25}$$

where D_{up} is defined as in equation (3.4).

Let us now provide a theorem that gives an upper bound on the maximum execution time resulting from the application of Algorithm 2. First, we introduce the following Lemma necessary to the proof of Theorem 3.3.8.

Lemma 3.3.1 *Let $S_{gossip}(t)$ be the sum of all J_i's at iteration t of Algorithm 2. Then*

$$\forall t > 0, \quad S_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1} + \frac{7}{4} + \sqrt{2}\right)\frac{nd_{max}}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{3.26}$$

Proof: By definition $S_{gossip}(t) = \sum_{i=1}^{n} J_i(t)$. Since

$$J_i(t) = \frac{TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right)}{v_i} + \frac{\sum_{j\in\mathcal{K}_i} c_j}{w_i},$$

it is

$$S_{gossip}(t) = \sum_{i=1}^{n} \frac{TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right)}{v_i} + \sum_{i=1}^{n} \frac{\sum_{j\in\mathcal{K}_i} c_j}{w_i}.$$

By considering the worst case scenario in which each agent has speed $v_i = v_{min}$ and task execution speed $w_i = w_{min}$ for $i = 1, \ldots, n$, we have the following straightforward upper bound

$$S_{gossip}(t) \leq \sum_{i=1}^{n} \frac{TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right)}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{3.27}$$

To each robot $k_i(t) = |\mathcal{K}_i(t)|$ tasks are assigned at any given time. By exploiting the result by Few (40) and (41) given in eq. (3.14) and eq. (3.15), and taking into account that such results refer to a unit square area, the maximum tour length that each robot has to drive to visit all its assigned tasks is

$$TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right) \leq \left(\alpha\sqrt{k_i+1} + \beta\right) d_{\max} \tag{3.28}$$

where $\alpha, \beta \in \mathbb{R}$ are appropriate constants that depend on the considered bound. Thus, we may now write

$$S_{gossip}(t) \leq \frac{\alpha d_{\max}}{v_{min}} \sum_{i=1}^{n} \left(\sqrt{k_i(t)+1} \right) + \frac{n\beta d_{\max}}{v_{min}} + \frac{\sum_{j \in \mathcal{K}} c_j}{w_{min}}. \tag{3.29}$$

The only term in eq. (3.29) that is affected by the task assignment to the robots is $\sum_{i=1}^{n} \left(\sqrt{k_i(t)+1} \right)$. We now find the task assignment that maximizes the bound in eq. (3.29) by solving the following optimization problem:

$$\begin{cases} max \sum_{i=1}^{n} \left(\sqrt{k_i+1} \right) \\ s.t. \\ \sum_{i=1}^{n} k_i = k \\ k_i \geq 0 \qquad i = 1, \ldots, n \\ k_i \in \mathbb{N} \qquad i = 1, \ldots, n \end{cases} \tag{3.30}$$

Any solution to Problem (3.30) found by relaxing the constraint to have integer variables is an upper bound to the solution of the given problem. We therefore solve Problem (3.30) by relaxing the integer constraint using Lagrange multipliers:

$$f(k_1, \ldots, k_n, \lambda) = \sum_{i=1}^{n} \left(\sqrt{k_i+1} \right) + \lambda \left(\sum_{i=1}^{n} k_i - k \right) \tag{3.31}$$

By setting partial derivatives of the objective function (3.31) to zero we get

$$\begin{aligned} \frac{\partial f(k_1, \ldots, k_n, \lambda)}{\partial k_i} &= \frac{1}{2\sqrt{k_i+1}} + \lambda = 0 \qquad i = 1, \ldots, n \\ \frac{\partial f(k_1, \ldots, k_n, \lambda)}{\partial \lambda} &= \left(\sum_{i=1}^{n} k_i - k \right) = 0 \end{aligned} \tag{3.32}$$

Thus, for any $i, j \in \mathcal{N}$, it is

$$\frac{1}{2\sqrt{k_i+1}} = \frac{1}{2\sqrt{k_j+1}},$$

i.e., the maximum of function (3.31) is found for $k_i = \frac{k}{n}$ for all $i \in \mathcal{N}$. Therefore, an upper bound to the solution of Problem (3.30) is

$$\sum_{i=1}^{n} \left(\sqrt{k_i+1} \right) \leq \sum_{i=1}^{n} \left(\sqrt{\frac{k}{n}+1} \right) = n\sqrt{\frac{k}{n}+1}.$$

Finally, by substituting the solution of (3.31) into (3.29)

$$S_{gossip}(t) \leq \alpha n \left(\sqrt{\frac{k}{n} + 1} + \beta \right) \frac{d_{max}}{v_{min}} + \frac{\sum_{j \in \mathcal{K}} c_j}{w_{min}}. \tag{3.33}$$

If we consider the results by Few (3.14) we get

$$S_{gossip}(t) \leq \left(\sqrt{2} \sqrt{\frac{k}{n} + 1} + \frac{7}{4} + \sqrt{2} \right) \frac{n d_{max}}{v_{min}} + \frac{\sum_{j \in \mathcal{K}} c_j}{w_{min}}. \tag{3.34}$$

□

We are now ready to state one of the main results of this chapter.

Theorem 3.3.8 *The maximum execution time J^*_{gossip} resulting from the application of Algorithm 2 satisfies*

$$J^*_{gossip} \leq \left(\sqrt{2} \sqrt{\frac{k}{n} + 1} + \frac{7}{4} + \sqrt{2} \right) \frac{d_{max}}{v_{min}} + \frac{1}{n} \frac{\sum_{i \in \mathcal{K}} c_i}{w_{min}} + D_{up}. \tag{3.35}$$

Proof: Let $S_{gossip}(t)$ be the sum of all J_i's at iteration t of Algorithm 2. By Lemma 3.3.1 we have

$$S_{gossip}(t) \leq \left(\sqrt{2} \sqrt{\frac{k}{n} + 1} + \frac{7}{4} + \sqrt{2} \right) \frac{n d_{max}}{v_{min}} + \frac{\sum_{j \in \mathcal{K}} c_j}{w_{min}} \tag{3.36}$$

Let $J^*_{gossip,min}$ be the smallest execution time between the vehicles after the execution of Algorithm 2. Corollary 3.3.7 implies $J^*_{gossip,min} \geq J^*_{gossip} - D_{up}$. Moreover, $\forall t \geq 0$ it obviously is

$$J^*_{gossip,min}(t) \leq \frac{1}{n} S_{gossip}(t) \tag{3.37}$$

thus

$$\begin{aligned} J^*_{gossip} &\leq J^*_{gossip,min} + D_{up} \leq \frac{1}{n} S_{gossip}(t) + D_{up} \\ &\leq \left(\sqrt{2} \sqrt{\frac{k}{n} + 1} + \frac{7}{4} + \sqrt{2} \right) \frac{d_{max}}{v_{min}} + \frac{1}{n} \frac{\sum_{i \in \mathcal{K}} c_i}{w_{min}} + D_{up}. \end{aligned} \tag{3.38}$$

proving the statement. □

3.3.5 Asymptotic behavior

We now study what is the performance to expect from the proposed algorithm in the limit cases in which the ratio between tasks and robots goes to infinity. In particular we obtain the following result.

Proposition 3.3.9 *Let J^*_{gossip} be the maximum execution time resulting from the application of Algorithm 2 and let J^* be the optimal solution to the HMVR problem. Then*

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{c_{max}}{c_{min}} \frac{w_{max}}{w_{min}}. \tag{3.39}$$

Proof: By taking the ratio between the upper bound to J^*_{gossip} given in Theorem 3.3.8 and the lower bound of the optimal solutions to the HMVR problem J^* given in eq.(3.5) we get

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{\left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}} + \frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}} + D_{up}}{\frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{max}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}}\right) - D_{lo}}. \tag{3.40}$$

The term $\frac{1}{n}\frac{TSP(\mathcal{K})}{v_{max}}$, being at the denominator, can be lower bounded by zero. The term $\frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}$ at the numerator can be upper bounded by $\frac{k}{n}\frac{c_{max}}{w_{min}}$ while the equivalent term $\frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}}$ at the denominator can be lower bounded by $\frac{k}{n}\frac{c_{min}}{w_{max}}$. Therefore, we get

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{\left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}} + \frac{k}{n}\frac{c_{max}}{w_{min}} + D_{up}}{\frac{1}{n}\frac{TSP(\mathcal{K})}{v_{max}} + \frac{k}{n}\frac{c_{min}}{w_{max}} - D_{lo}}. \tag{3.41}$$

The term $\frac{k}{n}$ dominates both on the constants and on the term $\sqrt{\frac{k}{n}}$, thus we get

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{c_{max}}{c_{min}} \frac{w_{max}}{w_{min}}. \tag{3.42}$$

proving the statement. □

3.4 An heuristic gossip algorithm

In this section we present a new algorithm, called the Decentralized Heuristic Algorithm, and discuss its convergence properties and computational complexity in comparison with the algorithm in the previous section.

The robots update their states following Algorithm 3, while the task exchange rule is described in Algorithm 4. The basic idea is as follows. When two robots are selected at step 3.a of Algorithm 3, the two agents start to balance their execution time by the iterative execution of Algorithm 4. At each execution of Algorithm 4 only two scenarios are possible:

- the sets of assigned tasks of the two robots do not change;
- one task is given by the robot with the higher execution time to the other robot.

Note that the determination of the possible exchanges is made by the computation of the Approximated Euclidean TSP ($ATSP$), thus, unlike in the MILP gossip algorithm, this approach involves polynomial time algorithms. There exist a vast literature on polynomial time algorithms to compute approximations to the Euclidean TSP such that

$$ATSP \leq \alpha TSP$$

where TSP denotes the value of the optimal TSP and α represents the worst case ratio. In (44) some heuristics for the TSP problem are summarized. Many heuristics are based on the computation of the Minimum Spanning Tree (MST) among the nodes and guarantee a worst case ratio of $\alpha = 2$ with a running time of $\mathcal{O}(m^2)$, where m denotes the number of nodes to be visited. Another polynomial time heuristic based on MST which provides a value of $\alpha = 1.5$ is the Christofides algorithm described in (45), which is characterized by a running time of $\mathcal{O}(m^3)$.

We observe that the **STOP** of Algorithm 4 ensures that after the execution of Algorithm 4 it holds

$$\max\{J_r(t+1), J_q(t+1)\} \leq \max\{J_r(t), J_q(t)\}$$

whatever is the choice of the algorithm to compute the value of the $ATSP$.

As a final remark we note that conditions can be given on the gossip communication scheme which allow the robot to converge to stable task assignment in a

Algorithm 3 Decentralized Heuristic Algorithm

(i) Tasks are initially arbitrarily assigned to robots.

(ii) Let $t = 0$.

(iii) While $t \leq T_{max}$

  (a) Select two robot R_p and R_r at random.

  (b) Apply Algorithm 4 repeatedly on R_p and R_r until no more task exchanges are possible.

  (c) Let $t = t + 1$ and go back to Step 3.

(iv) All robots process their own set of tasks following the order specified by the local solution of an ATSP Algorithm.

finite time. In particular, the following two theorems can be given, whose proofs are omitted here because they follow the same lines of Theorems 3.3.4 and 3.3.5, respectively.

Theorem 3.4.1 *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 3. If the gossip communication scheme satisfies the* deterministic persistence *property then, for every initial task assignment, there exists a network state $\tilde{\mathcal{K}}^*_{heur}$ and a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{heur}$, for all $t \geq T$.*

Theorem 3.4.2 *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 3. If the gossip communication scheme satisfies the* stochastic persistence *property, then, for every initial task assignment, there exists a network state $\tilde{\mathcal{K}}^*_{heur}$ and almost surely a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{heur}$ for all $t \geq T$, i.e., the network state converges almost surely in finite time to $\tilde{\mathcal{K}}^*_{heur}$.*

3.4.1 Computational complexity of the local optimization

In this section we discuss about the advantages of the proposed heuristic in terms of computational complexity with respect to the MILP gossip algorithm.

Algorithm 4 Local Balancing between robots R_r and R_q

- **INPUT:** $\mathcal{K}_r(t)$ and $\mathcal{K}_q(t)$.
- **OUTPUT:** $\mathcal{K}_r(t+1)$ and $\mathcal{K}_q(t+1)$.
- **ASSUMPTION:** We assume, with no loss of generality, that $J_r(t) > J_q(t)$.
- **STEPS:**

(i) Let $\mathcal{K}_{ex} = \emptyset$, $\mathcal{K}_v = \mathcal{K}_r$ and $F = 0$.

(ii) **While** $F = 0$ **and** $\mathcal{K}_v \neq \emptyset$

- Select $i \in \mathcal{K}_v$ randomly.
- Let $\mathcal{K}_v = \mathcal{K}_v \setminus \{i\}$.
- Compute

$$J_{new} = \frac{ATSP(\mathcal{K}_q \cup \{i\})}{v_q} + \frac{\sum_{j \in (K_q \cup \{i\})} c_j}{w_q}.$$

- **If** $J_{new} < J_r(t)$
  (a) $\mathcal{K}_{ex} = \mathcal{K}_{ex} \cup \{i\}$.
  (b) $F = 1$.

End While.

- **STOP**:

- $\mathcal{K}_q(t+1) = \mathcal{K}_q(t) \cup \mathcal{K}_{ex}$ and $\mathcal{K}_r(t+1) = \mathcal{K}_r(t) \setminus \mathcal{K}_{ex}$.
- $$J_q(t+1) = \frac{ATSP(\mathcal{K}_q(t+1))}{v_q} + \frac{\sum_{j \in (\mathcal{K}_q(t+1))} c_j}{w_q},$$
$$J_r(t+1) = \min \left\{ J_r(t) - \frac{\sum_{i \in \mathcal{K}_{ex}} c_i}{w_r}, \frac{ATSP(\mathcal{K}_r(t+1))}{v_r} + \frac{\sum_{j \in (\mathcal{K}_r(t+1))} c_j}{w_r} \right\}.$$

Let us begin with the analysis of the computational complexity of the single task exchange rule in Algorithm 4. The following proposition characterizes the running time of Algorithm 4.

Proposition 3.4.3 *Assume to compute the ATSP using, at step 2 of Algorithm 4, an algorithm with a running time of $\mathcal{O}(k^p)$. The worst case running time of Algorithm 4 is $\mathcal{O}(k^{p+1})$.*

Proof: The maximum number of nodes assigned to a robot is k, thus at each iteration of the while loop of Algorithm 4 the running time of the algorithm to compute the ATSP is at maximum $\mathcal{O}(k^p)$. The while loop can be repeated at maximum k times, as there may be at maximum k tasks exchange. Thus the total running time of Algorithm 4 is $k \cdot \mathcal{O}(k^p) = \mathcal{O}(k^{p+1})$. □.

An important property of the proposed heuristic is presented in the following proposition.

Proposition 3.4.4 *Let $J_{heur}(t) = \max_{i \in \mathcal{N}} J_i(t)$ be the maximum execution time of robots at time t resulting from the execution of Algorithm 3. The following holds*

$$\forall t \in \mathbb{N}, \quad J_{heur}(t+1) \leq J_{heur}(t).$$

Proof: The proof directly follows from the update rules of Algorithm 4. Let R_r and R_q be the couple of robots selected by Algorithm 3 at time t with execution time respectively $J_r(t)$ and $J_q(t)$. Let R_{max} be the robot with the maximum execution time at time $t \geq 0$, so it is $J_{max}(t) = J_{heur}(t)$. Now, by Algorithm 4 is holds $\max\{J_r(t+1), J_q(t+1)\} \leq \max\{J_r(t), J_q(t)\}$, and only two cases may occur

- if $R_r, R_q \neq R_{max}$, $J_{heur}(t+1) = J_{heur}(t)$, i.e., the maximum execution time does not change;
- if either $R_r = R_{max}$ or $R_q = R_{max}$, $J_{heur}(t+1) \leq J_{heur}(t)$, i.e., the maximum execution time may be reduced.

□

A similar property was discussed for the MILP gossip algorithm as well: at each iteration of the local optimization rule the maximum execution time can not increase. Note that in the MILP gossip algorithm each local optimization requires to solve a MILP problem, which is an exponential time algorithm. Proposition 3.4.3 shows that the proposed heuristic is based on a local balance with a considerably smaller computational complexity than the MILP gossip algorithm.

We conclude this section with some considerations about the total number of local interactions required to reach a final task assignment. We conjecture that the expected number of iterations of Algorithm 3 required to converge are of the same order as the number of iterations required in the MILP gossip algorithm. Our conjecture is based on the following observations. The execution of Algorithm 4 leads to a different task assignment only if the maximum execution time among the involved robots can be decreased, otherwise the task assignment does not change. In the proposed framework if at time t the execution of Algorithm 4 leads to a decrement of the maximum execution time, the network state $\tilde{\mathcal{K}}(t)$ changes to a new one $\tilde{\mathcal{K}}(t+1)$. It follows from Proposition 3.4.4 that $\tilde{\mathcal{K}}(t)$ is no more visited during the algorithm evolution. This property holds for the MILP gossip algorithm as well. Starting from an initial network state $\tilde{\mathcal{K}}(0)$, in both decentralized solutions all the possible network states may be visited before to reach the equilibrium state. For that reason we can reasonably conjecture that the MILP gossip algorithm and Algorithm 3 have computational complexity of the same order in terms of total number of iterations. Our conjecture is supported also by the results of some simulations presented in the following.

3.4.2 Characterizations of the heuristic solution

In this section we focus on some properties of J^*_{heur}, i.e., the solution of Algorithm 3 at the equilibrium, when no better balancing among robots may be obtained. As the MILP gossip algorithm, Algorithm 3 does not guarantee the convergence to an optimal solution. Firstly we present a theorem that characterizes the maximum distance among the execution times of two robots that have locally balanced their loads. Then we provide an upper bound on the maximum execution time resulting from the application of Algorithm 3.

Theorem 3.4.5 *Let $J^*_{r,heur}$ and $J^*_{q,heur}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Step 2 of Algorithm 3. It holds*

$$|J^*_{r,heur} - J^*_{q,heur}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}} \tag{3.43}$$

where d^{rq}_{max} is the maximum distance among tasks in $\mathcal{K}_r$ and tasks in $\mathcal{K}_q$, $v^{rq}_{min} = \min\{v_r, v_q\}$, and $w^{rq}_{min} = \min\{w_r, w_q\}$.

Proof: Let R_r and R_q be a couple of robots selected in Algorithm 3 at time t with execution time respectively $J_r(t)$ and $J_q(t)$ after t iterations. By step 2 of Algorithm 3 robots R_r and R_q exchange tasks one by one until no more exchanges are possible. Assume, without lack of generality, that at time t it holds $J_r(t) > J_q(t)$. Now, let us assume to exchange one task from R_r to R_q. Surely the execution time of R_r decreases, thus $J_r(t+1) \leq J_r(t)$. On the contrary, the execution time of robot R_q increases but the resulting value is such that:

$$J_q(t+1) \leq J_q(t) + \frac{c^{rq}_{max}}{w_q} + 2\frac{d^{rq}_{max}}{v_q}.$$

Thus, by exchanging one task a reduction of the maximum execution time is guaranteed if

$$J_q(t) + \frac{c^{rq}_{max}}{w_q} + 2\frac{d^{rq}_{max}}{v_q} \leq J_r(t).$$

In other words, if

$$J_r(t) - J_q(t) \geq \frac{c_{max}}{w_q} + 2\frac{d^{rq}_{max}}{v_q}$$

then there exists at east task that can be exchanged such that

$$\max\{J_q(t+1), J_r(t+1)\} < \max\{J_q(t), J_r(t)\}.$$

Since the number of possible task assignments is finite and at each iteration of Algorithm 4 the local maximum may be decreased due to a task exchange, some of these configurations are never visited again. Thus we have that in finite time

$$|J^*_{r,heur} - J^*_{q,heur}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}}$$

□

By Theorem 3.4.5 and the fact that each robot interacts with any other sufficiently often, a significant result follows.

Corollary 3.4.6 *Let $J^*_{r,heur}$ and $J^*_{q,heur}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Algorithm 3. It holds*

$$|J^*_{r,heur} - J^*_{q,heur}| \leq D_{up} \tag{3.44}$$

where

$$D_{up} = 2\frac{d_{max}}{v_{min}} + \frac{c_{max}}{w_{min}}.$$

□

Finally, the following result can be proved using the same arguments as in the proof of Theorem 3.3.8.

Theorem 3.4.7 *Let J^*_{heur} be the value of the objective function (3.1) resulting from the execution of Algorithm 3. It is*

$$J^*_{gossip} \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1} + \frac{7}{4} + \sqrt{2}\right)\frac{d_{max}}{v_{min}} + \frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}} + D_{up}. \tag{3.45}$$

where $D_{up} = 2\frac{d_{max}}{v_{min}} + \frac{c_{max}}{w_{min}}$.

Proof: Follows the same steps of Theorem 3.3.8. □

3.5 Numerical simulations

In this section we present some numerical results comparing the performance of the proposed heuristic and the performance of the MILP gossip algorithm. We first analyze the value of J^*_{heur} and J^*_{gossip} for different values of k and n, comparing them with the lower and upper bounds, given in eq. (3.2) and eq. (3.5), respectively. We then compare the convergence time of the two decentralized solutions either in terms of number of iterations required or in terms of absolute time.

In all the experiments robots and tasks are randomly scattered in a square box of side 5. Costs of tasks are integer values randomly generated with uniform distribution in the interval $[1, 5]$. Speeds v_i and w_i are real values randomly

generated with uniform distribution in $[1, 2]$. In both decentralized algorithms the edge selection is performed in a uniformly random way. The MILP problems are solved using the MATLAB optimization tool *glpk*, while the results related with Algorithm 3 are obtained using our own MATLAB script. The value of the $ATSP$ is computed by calculating a minimum spanning tree and adding shortcuts in the induced cycle, thus approximating the optimal TSP length by a factor of $\alpha = 2$.

In Fig.3.1 are reported the results of the comparison between the following values:

- the value of J^*_{heur}, obtained by the execution of Algorithm 3;
- the value of J^*_{gossip} obtained by the execution of Algorithm 2;
- the upper and lower bound of the centralized approach given respectively by (3.2) and (3.5).

For each couple (n, k) of n robots and k tasks, J^*_{heur}, J^*_{gossip} and the two bounds are the mean values of 10 experiments. Simulations show that the maximum service time obtained with the two approaches lies always between the upper and the lower bound of the centralized approach. Moreover, the performance of the two approaches are similar. It can be observed that Algorithm 2 leads to better results than Algorithm 3 when the ratio $\frac{k}{n}$ is high.

In Fig. 3.2, Fig. 3.3 and Fig. 3.4 the execution times of Algorithm 3 are compared with the execution times of Algorithm 2. In particular, Fig. 3.2 and Fig. 3.3 show the execution time respectively of the MILP gossip algorithm and Algorithm 3 in terms of number of iterations, while in Fig. 3.4 the comparison is made in terms of time in seconds spent by MATLAB to execute the Algorithms. The two figures confirm that the proposed framework has a computational complexity considerably lower than the MILP gossip algorithm.

The results in Fig. 3.2 and Fig. 3.3 confirm also the conjecture that we have discussed in the final part of Section 3.4.1: the execution time in terms of number of iterations are of the same order in Algorithm 3 and in the MILP gossip algorithm.

Finally we focus on the execution time of Algorithm 3 in seconds and in terms of number of cycles. Figure 3.5 shows the number of iterations while Figure 3.6

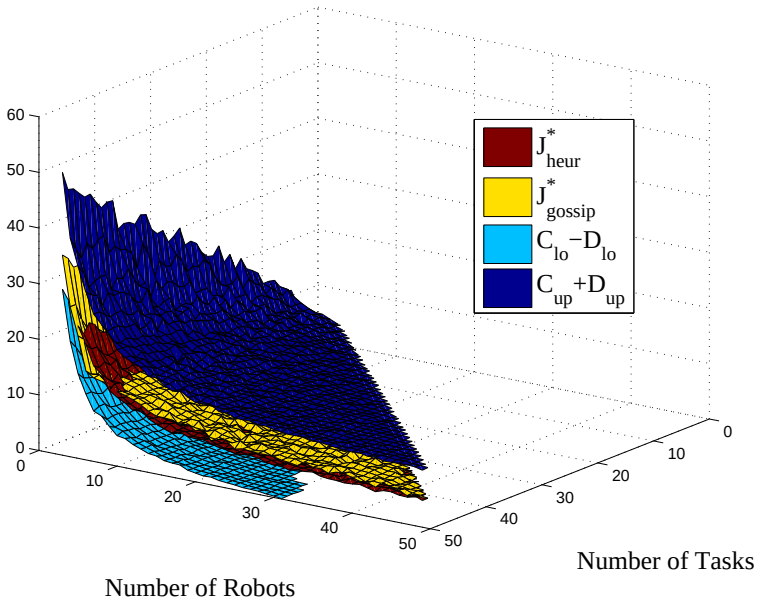


Figure 3.1: J^*_{heur}, J^*_{gossip} and the upper bound (3.2) and the lower bound (3.5) of the centralized solution.

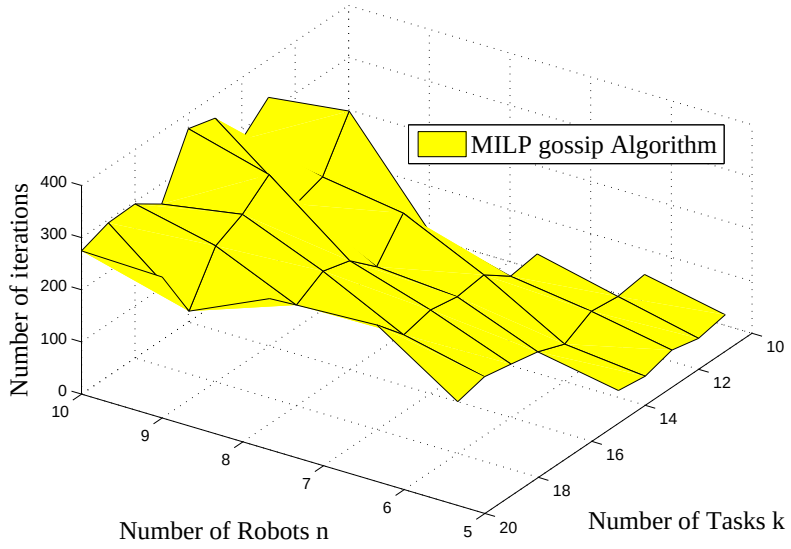


Figure 3.2: Number of iterations required to reach an equilibrium state with MILP gossip algorithm.

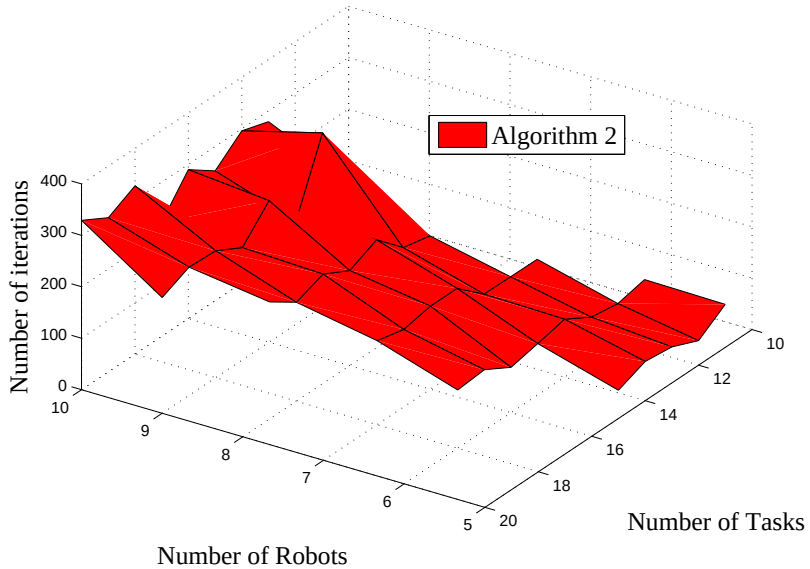


Figure 3.3: Number of iterations required to reach an equilibrium state with Algorithm 3.

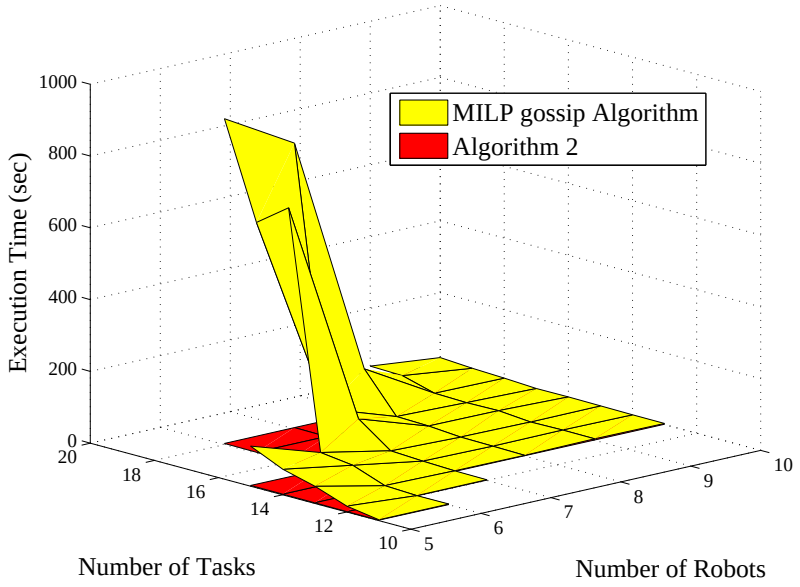


Figure 3.4: Execution time of MILP gossip algorithm and Algorithm 3.

shows the execution time in seconds for Algorithm 3 for different values of k in a system with $n = 10$ robots.

Figure 3.5 shows that the expected number of iterations of Algorithm 3 grows linearly with the number of tasks if the number of robots is kept constant. On the other hand, in Figure 3.6 is shown that the actual computational time is of the order of $\mathcal{O}(n^3)$ seconds. This is due to the fact that the complexity of the task exchange according to the heuristic grows linearly with the number of tasks for each iteration of Algorithm 3 thus accounting for at least a quadratic grow of computational time, the remaining difference can be accounted by the software implementation and execution in Matlab.

3.6 Conclusions and future work

In this chapter we proposed upper and lower bounds for the cost of the optimal solution to the HMVRP which considers vehicles with different movement and task execution speed and tasks with different servicing costs. We extended to

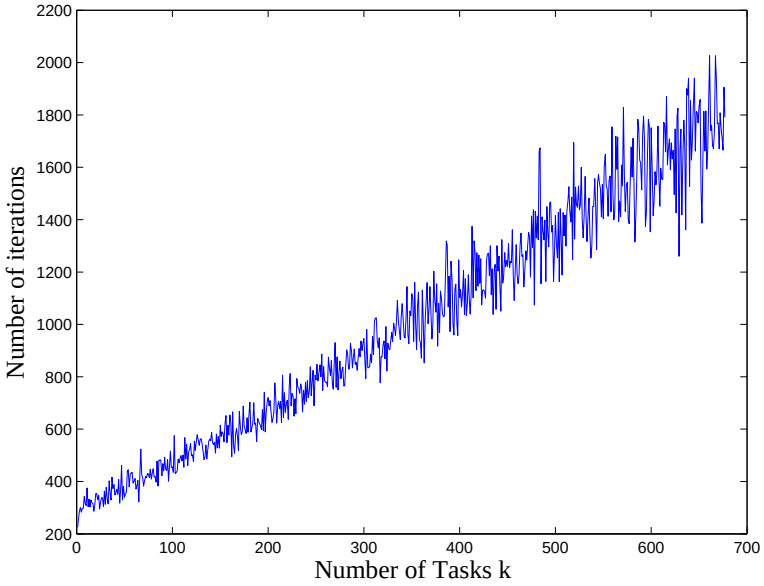


Figure 3.5: Number of iterations of Algorithm 3 for $n = 10$ and different values of k

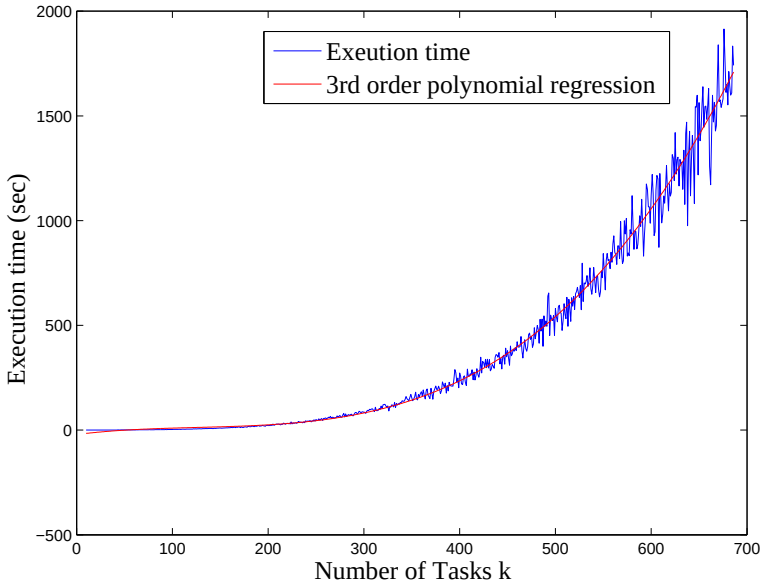


Figure 3.6: Execution time of Algorithm 3 for $n = 10$ and different values of k

our framework the bounds for the multi-vehicle routing problem in (30). Furthermore, we proposed two algorithms based on gossip to solve the HMVRP in a distributed fashion exploiting only pairwise task exchanges between vehicles. The first algorithm is based on local, asynchronous and pairwise optimizations to improve the local task assignment. The second one is an heuristic with linear complexity with respect to the number of tasks. The computational complexity of the first method scales with exponential complexity with respect to the ratio between the number of tasks and vehicles, improving with respect to a centralized optimization that scales exponentially with the number of tasks. The proposed algorithms have been characterized in terms of finite-time almost sure convergence and in terms of minimum guaranteed performance.

We validated through simulations that the proposed algorithms compute a solution that scales with the number of robots within a constant factor of approximation with respect to the optimal centralized solution.

As future work we plan to extend the framework to a dynamic case in which robots start to move and serve tasks while the decentralized optimization is being executed and new tasks appear in the region.

Part II

Graph methods for diffusion of innovation in social networks

Chapter 4

Mathematical models for the diffusion of innovation in social networks: Introduction and literature overview.

In the last decades many researchers from different fields have been interested in the study of how innovation spreads in social networks. What is the mechanism that convinces an individual to follow a new idea or to buy a new product? What is the best marketing strategy which a company should adopt to take a competitive advantage? How does viral marketing works? Many mathematical models have been proposed to give an answer to questions of this type.

Since the 40's, many mathematical models on the diffusion of innovation has been proposed ((46, 47)) such as: the Linear Threshold Model, the *Independent cascade model* ((48)) and epidemic models such as SIS and SIR ((49, 50)). All these models are based on the same concept: in a social network the behaviour of each individual is highly influenced by the behaviour of its neighbours.

Many of these models are based on the *threshold effect*: an individual adopts a behaviour if a certain ratio of its social contacts have already adopted it, differently from the epidemic models in which a node adopts a behaviour with a certain probability if at least one of its neighbours has adopted it. Threshold models are more suitable to describe social influence phenomena and individual behaviours,

while epidemic models are more used for mass behaviours. Examples of threshold models are presented in (51, 52, 53). The firsts examples of threshold approaches go back to the seventies ((51, 54)). Several aspects of the diffusion phenomena have been studied among he years, from the local interactions between neighbours ((55, 56)), to the analysis of groups behaviours ((53, 57, 58)), which is the aspect we focus on.

In the following chapters we deal with the *Linear Threshold model*, which was originally proposed in (51), and has been widely studied in recent years. As in most of the models appeared in the literature, the social network is represented by a graph in which each node represents an individual, and edges represent the relationships among individuals. In the original model a threshold value λ_i is assigned to each individual i, and all the neighbours of i have the same influence weight on it. An individual adopts the innovation as soon as the ratio of its neighbours who have already adopted it is above its threshold value. The origin of the previous rule is the following: many competitive games such an individual decision rule has been proved to be the best response to the actions of one's neighbours ((53, 57)). When a node adopts the innovation we say that it becomes *active*, otherwise is said to be *inactive*. It is implicitly assumed that a node can adopt the innovation, but once adopted, it cannot abandon it, i.e., a node can switch its state from inactive to active but cannot switch it from active to inactive. This model can be used to represent systems in which the adoption of a innovation is permanent and in the literature is called *progressive* ((59, 60)). For instance, the progressive Linear threshold model can be suitable to represent a group of people who want to buy a certain item: once an individual spends money to buy that, i.e., once it adopts the innovation, usually it cannot have the money back, thus we can say that the adoption of the innovation has a permanent nature.

In many cases, however, the progressive model is not suitable to correctly describe the spread of innovation, as habits may change: an individual who votes for a party for a period can decide to change its preference, a person who eats every day at the same restaurant can be persuaded to change of venue. Moreover, the influence pattern in real networks is usually time-varying, as the human connections are subjected to changes: friendships can become stronger or weaker due to the passing of time, new connections can be setted up and old connections

can be removed. All these changes in the network can influence the spread of the innovation, and in such systems an individual who has adopted the innovation can be persuaded to abandon it. Such types of mechanisms can be described using *non-progressive* models, in which each individual periodically updates its state by looking at its neighbours, deciding either to be active or inactive.

To the best of our knowledge, most of the model presented in literature are progressive ((61, 62, 63)), while the non-progressive diffusion of innovation has not received much attention ((64)).

In the following chapters of we deal both with the progressive and non progressive models: in particular in Chapter 5 and Chapter 6 we deal with the classical progressive Linear threshold model, while in Chapter 7 we present a novel non progressive instance of the line threshold model.

Our research has been focused on two main aspects:

- the role of cohesive subgroups in the spread of innovation in the network;
- how to influence the network.

The first aspect represent an analysis problem: we want to understand how a system behave starting from a certain initial state. The second aspect represents a control problem: we want to impose a specific state to the system in order to make it follow a desired behaviour.

The social cohesion is considered a key aspect to understand collective behaviours in social networks. Many definitions of cohesiveness and social subgroups have been proposed in literature, and good surveys can be found in (65, 66, 67). Here we study two particular types of cohesive subgroups, namely the cohesive and persistent sets, to characterize the system, since this two types of groups are strictly related to the adopting rules of the considered Threshold models. We can define cohesive sets in both progressive and non progressive models, while the persistent sets are important in the non-progressive model.

Chapters 5 and 6 collect the results discussed in (68), presented at the international conference Necsys 2013.

In Chapter 5 our analysis is inspired by the recent work (58), which extends an idea proposed in (57), and present a characterization of the spread of innovation in social networks, given a *seed set* – i.e., the set of initial adopters – based on groups cohesion. A group of individuals is said to be cohesive if none adopts

the innovation starting from any external seed set. Moreover, in (58) it was proven that, given a seed set, the final adopters set can be easily computed by knowing the maximal cohesive subset contained in the complement of the seed set. We firstly characterize with a Binary Programming Problem (BPP) the computation of the maximal cohesive set. This characterization is useful to model other problems in social network analysis such as the ones presented in the next sections. Secondly we propose an algorithm, based on the linear relaxation of the presented BPP, which takes as input a seed set and computes the maximal cohesive subset contained in the complement of the seed set.

In Chapter 6 we discuss the problem of *influence maximization*, which can be as follows: find a seed set of r individuals which maximizes the number of final adopters. This problem is NP-hard, as shown in (60), and many approximated and greedy algorithms have been proposed in literature ((60, 69, 70, 71, 72)). To the best of our knowledge the target of all the approaches proposed so far is the maximization of the number of final adopters. This represents a limitation, as in many realistic cases it would be required to maximize the spread of innovation on a network in a finite time horizon. For example, let's think about a company which proposes a new product, it has to chose the best possible advertising strategy to convince the maximum number of costumers to adopt its product before other similar products come to the market. In this chapter we introduce the *Influence Maximization in Finite Time Problem* with parameters r and k (IMFTP(r,k)), which represents a generalization of the classical influence maximization problem. The IMFTP(r,k) can be described as follows: find a seed set of r individuals which maximizes the set of adopters in k time steps. Choosing a value of k high enough the solution of the IMFTP(r,k) coincides with the solution of the classical influence maximization problem. In section 6.1 a BPP which solves the (IMFTP(r,k)) is proposed.

Chapter 7 collects the results discussed in (73), presented in Florence at the international conference CDC 2013. In that chapter we present a non-progressive instance of the linear threshold model which can be considered as a generalization of the model presented in (74). We assume that the innovation is incepted in the network by a seed set, and the seed nodes are supposed to maintain the innovation for a finite time - the seeding time -, after which they start to update their state by following the same rules adopted by all the other nodes in the network.

We characterize the system evolution in two different phases: during and after the seeding time. We show that during the seeding time the system behaves as in the progressive model in (74). The main contribution of our work is the analysis of the system evolution after the seeding time, which represent the main difference between our model and the previously ones presented in literature, as in this phase non-progressive mechanisms may occur. We use cohesive groups to characterize some conditions under which such mechanisms take place.

Chapter 5

Diffusion of innovation in the Progressive Linear Threshold Model

The chapter is organized as follows. In Section 5.1.1 we describe the representation of the network and the used model. In Section 5.2 we use binary and linear programming to compute the maximal cohesive set in a network.

5.1 Network representation and reference model

5.1.1 Network structure

We represent the network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. Each node $i \in \mathcal{V}$ represents an individual and an oriented edge $(i, j) \in \mathcal{E}$ denotes that node j is influenced by node i. For this reason in this manuscript we use the terms individual or node interchangeably. No selfloops, i.e., edges from one node to itself, are allowed. For each node i, let $\lambda_i \in [0, 1]$ denote its *threeshold value* and let $\mathcal{N}_i = \{j \mid (j, i) \in \mathcal{E}\}$ denote the set of its *in-neighbours*.

The topological information about the graph can be encoded in the *adjacency matrix* $A \in \{0, 1\}^{n \times n}$ which is defined as follows:

$$A(i, j) = \begin{cases} 1 & \text{if there is an edge from node } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

We define the *in-neighbours scaled adjacency matrix* $\hat{A} \in [0,1]^{n \times n}$ as follows:

$$\hat{A}(i,j) = \frac{A(i,j)}{|\mathcal{N}_j|}.$$

We denote with $\Lambda = diag([\lambda_1 \ \lambda_2 \ \dots \ \lambda_n])$ the diagonal matrix whose diagonal entries are the thresholds of the graph nodes.

5.1.2 Linear threshold model

Let us define ϕ_0 as the *seed set*, i.e., the set of nodes which have adopted the innovation at time $t = 0$. From the seed set the innovation spreads through the social network, and we denote as ϕ_t the set of nodes which adopt the innovation at time t. All the individuals that adopt the innovation during the time interval $[0, t]$ belong to the set $\Phi_t = \bigcup_{j=0}^{t} \phi_j$. In general, node i which has not adopted the innovation until time t, adopts the innovation at time $t+1$ – i.e., $i \in \phi_{t+1}$ – if the following holds:

$$\frac{|\Phi_t \bigcap \mathcal{N}_i|}{|\mathcal{N}_i|} = \frac{|\cup_{j=0}^{t} \phi_j \bigcap \mathcal{N}_i|}{|\mathcal{N}_i|} \geq \lambda_i \tag{5.1}$$

The innovation spreads in the network until no more individuals can adopt it, and we denote the set of final adopters as:

$$\Phi^* = \bigcup_{j=0}^{\infty} \phi_j.$$

Algorithm 5 describes the dynamic of the network and returns as output the set Φ^* computing at each step which nodes respect equation (5.1).

5.1.3 Other mathematical results

We associate to each set of nodes $X \subset \mathcal{V}$ a characteristic vector defined as follows.

Definition 3 *Given a set $X \subset \mathcal{V}$, the associated characteristic vector $\boldsymbol{x} \in \{0,1\}^n$ is such that $x_i = 1$ if $i \in X$ else $x_i = 0$.*

Algorithm 5 Computing Φ^*

INPUT: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A set $\phi_0 \subset V$.

OUTPUT: The set of final adopters Φ^*.

(i) Let $\Phi = \phi_0$, $\bar{\Phi} = \mathcal{V} \setminus \phi_0$, and $\Phi_{old} = \emptyset$.

(ii) Let $k = 0$.

(iii) **While** $\Phi \neq \Phi_{old}$

(a) $k = k + 1$.

(b) Let $\Phi_{old} = \Phi$.

(c) **For** $i \in \bar{\Phi}$

– **If** $\frac{|\Phi_{old} \bigcap \mathcal{N}_i|}{|\mathcal{N}_i|} \geq \lambda_i$, **then**:

1. $\Phi = \Phi \cup \{i\}$.
2. $\bar{\Phi} = \bar{\Phi} \setminus \{i\}$.

(d) end **while**.

(iv) Let $\Phi^* = \Phi$.

In the rest of the chapter we denote with $\boldsymbol{x}_i$ the characteristic vector of the set ϕ_i and with $\boldsymbol{w}_i$ the characteristic vector of the set Φ_i. According to the linear threshold model, for each couple of sets (ϕ_i, ϕ_j) such that $i, j \geq 0$ and $i \neq j$ it holds:

$$\phi_i \cap \phi_j = \emptyset$$

It follows that, $\forall t \in \mathbb{N}$:

$$\boldsymbol{w}_t = \boldsymbol{x}_0 + \boldsymbol{x}_1 + \ldots + \boldsymbol{x}_t \leq \mathbf{1}_n$$

The following definition formalizes the concept of cohesive set.

Definition 4 *A set $X \subset \mathcal{V}$ is called* cohesive *if for all $i \in X$ it holds*

$$\frac{|X \cap \mathcal{N}_i|}{|\mathcal{N}_i|} > 1 - \lambda_i. \tag{5.2}$$

In other world a set $X \subset \mathcal{V}$ is said to be cohesive if for each $i \in X$ the ratio of neighbours which do not belong to X is strictly smaller than λ_i. If X is a cohesive set it follows that if $\phi_0 \cap X = \emptyset$, then no individual in X can adopt the innovation. An important result of

Lemma 5.1.1 (Lemma 2 in (58)) *Let $\phi_0 \subset \mathcal{V}$ be the seed set of a network and let $\mathcal{M} \subset \mathcal{V} \setminus \phi_0$ be the maximal cohesive set of the complement of ϕ_0. The set of final adopters Φ^* is given by:*

$$\Phi^* = \mathcal{V} \setminus \mathcal{M}. \tag{5.3}$$

5.2 Computing maximal cohesive set

Lemma 5.2.1 shows that, given a network with seed set ϕ_0, the knowledge of the maximal cohesive set $\mathcal{M} \subset \mathcal{V} \setminus \phi_0$ permits an immediate computation of the set of final adopters Φ^*. In this section we propose an algorithm that computes the maximal cohesive subset of $\mathcal{V} \setminus \phi_0$ by solving some Linear Programming Problems (LPPs) . We first present a Binary Programming Problem (BPP), whose optimal solution is the characteristic vector of $\mathcal{M}$, then we prove that the LPP obtained by the relaxation of the BPP can be used to iteratively compute $\mathcal{M}$.

Lemma 5.2.1 *A set $X \subset \mathcal{V}$ is cohesive if and only if its characteristic vector $\boldsymbol{x}$ for all $i \in X$ satisfies*

$$\boldsymbol{x}^T \hat{A}(\cdot, i) \geq 1 - \bar{\lambda}_i$$

where

$$\bar{\lambda}_i = \begin{cases} \lambda_i - \dfrac{1}{|\mathcal{N}_i|} & \text{if } \lambda_i \cdot |\mathcal{N}_i| \in \mathbb{N} \\ \lambda_i & \text{if } \lambda_i \cdot |\mathcal{N}_i| \notin \mathbb{N} \end{cases}$$

Proof: Firstly we make the following obvious remark:

$$\boldsymbol{x}^T \hat{A}(\cdot, i) = \frac{\boldsymbol{x}^T A(\cdot, i)}{\mathbf{1}^T A(\cdot, i)} = \frac{|X \cap \mathcal{N}_i|}{|\mathcal{N}_i|}.$$

Then we observe that equation (5.2) can be rewritten as follows:

$$\frac{|X \cap \mathcal{N}_i|}{|\mathcal{N}_i|} > 1 - \lambda_i \quad \Leftrightarrow \quad |X \cap \mathcal{N}_i| > |\mathcal{N}_i| - \lambda_i \cdot |\mathcal{N}_i|. \tag{5.4}$$

Since the LHS of the last inequality of (4) is an integer, we consider two cases:

- if $\lambda_i \cdot |\mathcal{N}_i| \in \mathbb{N}$ the inequality can be rewritten as:

$$|X \cap \mathcal{N}_i| \geq |\mathcal{N}_i| - \lambda_i \cdot |\mathcal{N}_i| + 1;$$

- if $\lambda_i \cdot |\mathcal{N}_i| \notin \mathbb{N}$ the inequality can be rewritten as:

$$|X \cap \mathcal{N}_i| \geq |\mathcal{N}_i| - \lambda_i \cdot |\mathcal{N}_i|.$$

Dividing these inequalities by $|\mathcal{N}_i|$ the result follows immediately. □

According to the definition of $\bar{\lambda}_i$ introduced in Lemma 5.2.1 we define the diagonal matrix $\bar{\Lambda} = diag([\bar{\lambda}_1 \ \bar{\lambda}_2 \ \ldots \ \bar{\lambda}_n])$.

Now we are able to present the following BPP.

Proposition 5.2.1 *Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, let $\phi_0 \subset \mathcal{V}$ be a seed set with characteristic vector $\boldsymbol{y}$. The maximal cohesive set $\mathcal{M}$ contained in $V \setminus \phi_0$ has a characteristic vector $\boldsymbol{x}$ that is the solution of the following BPP:*

$$\begin{array}{ll} \max & \mathbf{1}^T \cdot \boldsymbol{x} \\ \left\{ \begin{array}{rcl} \boldsymbol{x} & \leq & \mathbf{1} - \boldsymbol{y} \\ \left[I - \bar{\Lambda} - \hat{A}^T \right] \cdot \boldsymbol{x} & \leq & \mathbf{0} \\ \boldsymbol{x} \in \{0,1\}^n & & \end{array} \right. \end{array} \tag{5.5}$$

Proof: Firstly, we observe that

$$\mathcal{M} \cap \phi_0 = \emptyset \qquad \Longleftrightarrow \qquad \boldsymbol{x} + \boldsymbol{y} \leq \mathbf{1},$$

which can be rewritten as the first constraint in (5.5).

Secondly, since $\mathcal{M}$ is a cohesive set, by Lemma 5.2.1 it holds

$$\begin{array}{c} \forall i \in \mathcal{M}, \quad \boldsymbol{x}^T \hat{A}(\cdot, i) \geq 1 - \bar{\lambda}_i \\ \Updownarrow \\ \forall i \in \mathcal{V}, \quad \boldsymbol{x}^T \hat{A}(\cdot, i) \geq (1 - \bar{\lambda}_i) x_i \\ \Updownarrow \\ \boldsymbol{x}^T \hat{A} \geq \boldsymbol{x}^T [I - \bar{\Lambda}] \end{array}$$

and this can be immediately rewritten as the second constraint in (5.5).

Finally, the cohesive set computed by BPP (5.5) is maximal because of the chosen objective function. □

Note that, as shown in a such a maximal cohesive set always exists – but may be the empty set – and is unique.

The main advantage of our characterization is that using characteristic vectors we can model several problems which are difficult to represents, such as the influence maximization problem presented in section 6.1. However, according to the previous proposition, computing a maximal cohesive set $\mathcal{M}$ requires solving a BPP, a task that may be computationally hard for large graphs. We will present in the following an alternative approach that requires solving a series of linear programming problems and is thus computationally viable.

First we consider a relaxed version of BPP (5.5) and characterize its solutions.

Proposition 5.2.2 *Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, let $\phi_0 \subset \mathcal{V}$ be a seed set with characteristic vector $\boldsymbol{y}$, and let $\mathcal{M}$ be the maximal cohesive set contained in $V \setminus \phi_0$. Consider the following LPP:*

$$\begin{array}{ll} \max & \mathbf{1}^T \cdot \boldsymbol{x} \\ \left\{ \begin{array}{rcll} \boldsymbol{x} & \leq & \mathbf{1} - \boldsymbol{y} & (a) \\ \left[I - \bar{\Lambda} - \hat{A}^T \right] \cdot \boldsymbol{x} & \leq & \mathbf{0} & (b) \\ \boldsymbol{x} & \geq & 0 & \end{array} \right. \end{array} \tag{5.6}$$

and let $\boldsymbol{x}^* \in [0,1]^n$ be an optimal solution of LPP (5.6).

(i) *For all* $i \in \mathcal{M}$, $x_i^* = 1$.

(ii) *If* $\boldsymbol{x}^* \in \{0,1\}^n$ *then* $\mathcal{M} = \{i \in V \mid x_i^* = 1\}$.

Proof: We prove separately the two statements.

(i) The first result can be proved by contradiction. Assume $\boldsymbol{x}$ is an optimal solution of (5.6) such that $Z = \{i \in \mathcal{M} \mid x_i < 1\}$ is not empty, and consider $\boldsymbol{x}'$ where $x'_i = x_i$ if $i \notin Z$ else $x'_i = 1$. We claim that $\boldsymbol{x}'$ satisfies the constraint set of (5.6).

In fact constraint (a) is trivially verified by $\boldsymbol{x}'$, since for all $i \in Z$ it holds $y_i = 0$.

Consider now constraints of the form (b). For all $i \in \mathcal{V} \setminus Z$ it holds

$$\boldsymbol{x}'^T \hat{A}(\cdot, i) \geq \boldsymbol{x}^T \hat{A}(\cdot, i) \geq (1 - \bar{\lambda}_i) x_i = (1 - \bar{\lambda}_i) x'_i$$

while for all $i \in Z \subseteq X$ it holds

$$\boldsymbol{x}'^T \hat{A}(\cdot, i) \geq \frac{|X \cap \mathcal{N}_i|}{|\mathcal{N}_i|} \geq 1 - \bar{\lambda}_i = (1 - \bar{\lambda}_i) x'_i$$

since $\mathcal{M}$ is a cohesive set. As shown in the proof of Proposition 5.2.1 these two results imply that $\boldsymbol{x}'$ satisfies constraints (b).

Finally, since $\mathbf{1}^T \cdot \boldsymbol{x}' > \mathbf{1}^T \cdot \boldsymbol{x}$, then $\boldsymbol{x}$ is not an optimal solution, which contradicts the assumption.

(ii) If $\boldsymbol{x}^* \in \{0,1\}^n$ then $\boldsymbol{x}^*$ is also the optimal solution of BPP (5.5) and thus it is the characteristic vector of set $\mathcal{M}$. □

We can finally write Algorithm 6 for the iterative computation of the maximal cohesive subset of the complement of the seed.

Some comments about the algorithm.

(1) Each time the LPP is solved, all nodes i with $x_i^{(k)} < 1$ do not belong to $\mathcal{M}$ (according to Proposition 5.2.2, part 1). Hence at step iii.(b) we can safely change the input of the LLP to $\boldsymbol{y}^{(k+1)}$ setting for these nodes $\boldsymbol{y}_i^{(k+1)} = 1$. Clearly the set $\mathcal{M}$ we want to determine is also the maximal cohesive set contained in $\mathcal{V} \setminus Y^{(k+1)}$, where $Y^{(k+1)}$ is the set whose characteristic vector is $\boldsymbol{y}^{(k+1)}$.

Algorithm 6 Computing Maximal Cohesive Set using LPP

INPUT: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with scaled adjacency matrix $\hat{A}$ and matrix $\bar{\Lambda}$. A set $\phi_0 \subset \mathcal{V}$ with characteristic vector $\boldsymbol{y} \in \{0,1\}^n$.

OUTPUT: The characteristic vector of the maximal cohesive set $\mathcal{M}$ contained in $\mathcal{V} \setminus \phi_0$.

(i) Let $k = 0$ and $\boldsymbol{y}^{(0)} = \boldsymbol{y}$.

(ii) Let $\boldsymbol{x}^{(k)} \in [0,1]^n$ be an optimal solution of the LPP

$$\begin{array}{ll} \max & \mathbf{1}^T \cdot \boldsymbol{x} \\ \left\{ \begin{array}{rcl} \boldsymbol{x} & \leq & \mathbf{1} - \boldsymbol{y}^{(0)} \\ \left[I - \bar{\Lambda} - \hat{A}^T \right] \cdot \boldsymbol{x} & \leq & \mathbf{0} \\ \boldsymbol{x} & \geq & 0 \end{array} \right. \end{array}$$

(iii) **While** $\boldsymbol{x}^{(k)} \notin \{0,1\}^n$

(a) Let $k = k + 1$.

(b) Let $\boldsymbol{y}^{(k)} = \left\lceil \mathbf{1} - \boldsymbol{x}^{(k-1)} \right\rceil$.

(c) Let $\boldsymbol{x}^{(k)} \in [0,1]^n$ be an optimal solution of the LPP

$$\begin{array}{ll} \max & \mathbf{1}^T \cdot \boldsymbol{x} \\ \left\{ \begin{array}{rcl} \boldsymbol{x} & \leq & \mathbf{1} - \boldsymbol{y}^{(k)} \\ \left[I - \bar{\Lambda} - \hat{A}^T \right] \cdot \boldsymbol{x} & \leq & \mathbf{0} \\ \boldsymbol{x} & \geq & 0 \end{array} \right. \end{array}$$

(iv) End **while**.

(v) $\boldsymbol{x}^{(k)}$ is the characteristic vector of $\mathcal{M}$.

(2) When the optimal solution of the LPP is a binary vector, we can be sure that it represents the characteristic vector of set $\mathcal{M}$ (according to Proposition 5.2.2, part 2).

The final result we present in this section concerns a bound on the number of steps the previous algorithm requires before halting.

Proposition 5.2.3 *Algorithm 5 and Algorithm 6 require a number $\bar{k}$ of repetitions of the while-loop where*

$$\bar{k} \leq n - |\phi_0| - |\mathcal{M}| + 1.$$

Proof: In Algorithm 5 at each execution of the *while*-loop it holds that the cardinality of Φ increases at least of 1. In Algorithm 6 one can immediately see that each time the *while*-loop is executed vector $\boldsymbol{y}$ increases in at least one component, and in both cases the maximal number of increments is equal to $n - |\phi_0| - |\mathcal{M}|$. □

Algorithm 6 provides an alternative way, with respect to Algorithm 5, to compute the set of final adopters that does not require to determine the evolution of the network. However, we cannot claim that Algorithm 6 is more efficient than Algorithm 5 at the light of Proposition 5.2.3. Algorithm 6 is based on the characterization of cohesive sets given in Proposition 5.2.1, and its interest consist in showing how a BPP for analysis of social network is amenable to a linear relaxation. We believe that other problems may exists which can be solved by using this type of approaches, and for that reason we have included this preliminary result.

Chapter 6

Influence Problems in the Progressive Linear Threshold Model

The Chapter is organized as follows. In section 6.1 we deal with the Influence Maximization problem. In section 6.2 another BPP model is proposed to solve the following problem: choose the minimum seed set which can diffuse the innovation over a target set in a finite time horizon. Finally, in the last section, we present some simulations and some numerical results related with the presented problem.

6.1 The Influence Maximization in Finite Time Problem (IMFTP).

The *influence maximization* represents one of the most attractive problems related with the diffusion of innovation in social networks. It can be summarized as follows: given a network described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, find a seed set $\phi_0 \subseteq \mathcal{V}$ of r innovators to maximize the diffusion of innovation, i.e., find a ϕ_0 such that $|\phi_0| = r$ and $|\Phi^*|$ is maximal.

The classical influence maximization problem presented above considers as quantity of interest the final number of adopters. Sometime it could be required to maximize the spread of innovation in a finite time horizon. The *Influence Maximization in Finite Time Problem* with parameters r and k (IMFTP(r, k)) can be formalized as follows: choose a seed set of r nodes to maximize the influence on the network in k time steps, i.e., find a ϕ_0 such that $|\phi_0| = r$ and

$|\Phi_k|$ is maximal. It's evident that the IMFTP(r,k) represents an extension of the classical influence maximization problem: choosing a value of k high enough the IMFTP(r,k) has the same solution as the classical problem.

As the number of possible subsets of r elements in a set of n is

$$\left(\begin{array}{c} n \\ r \end{array} \right) = \frac{n!}{r!(n-r)!}$$

the IMFTP(r,k) has a combinatorial nature. We characterize a solution to this problem using binary programming.

We first introduce the definition of *k-evolution* vector associated to a seed set ϕ_0.

Definition 5 (k-evolution vector) *Consider the diffusion of innovation in a net starting from a seed set ϕ_0 according to the linear threshold model presented in subsection 5.1.2. Given a positive integer k, let Φ_t be the set of nodes that adopt the innovation at time t (for $t = 0, 1, \ldots, k$) and let $\boldsymbol{w}_t$ be the characteristic vector of Φ_t. The vector $\boldsymbol{w}^T = [\boldsymbol{w}_0^T \ \boldsymbol{w}_1^T \ \ldots \ \boldsymbol{w}_k^T]$ is the k-evolution vector associated to ϕ_0.*

Lemma 6.1.1 *Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, let $\phi_0 \subset \mathcal{V}$ be a seed set, and at each time t let $\boldsymbol{x}_t$ and $\boldsymbol{w}_t$ be the characteristic vectors respectively of ϕ_t and Φ_t. The following property holds.*

$$\forall t \in \mathbb{N}, \qquad [\hat{A}^T + \Lambda]\boldsymbol{w}_t - \Lambda \boldsymbol{w}_{t+1} \geq \mathbf{0}_n \tag{6.1}$$

Proof: A node $i \in \mathcal{V}$ such that $i \notin \Phi_t$ adopt the innovation at time $t+1$, i.e., $i \in \phi_{t+1}$, if and only if

$$\boldsymbol{w}_t^T \hat{A}(:,i) \geq \lambda_i \tag{6.2}$$

Equation (6.2) follows from the following observation:

$$\boldsymbol{w}_t^T \hat{A}(:,i) = \frac{\boldsymbol{w}_t^T A(\cdot,i)}{\mathbf{1}^T A(\cdot,i)} = \frac{|\Phi_t \cap \mathcal{N}_i|}{|\mathcal{N}_i|}$$

It follows that: $\forall i \in \phi_{t+1}, \quad \hat{A}^T(\cdot,i)\boldsymbol{w}_t \geq \lambda_i$.

Thus:

$$\begin{gathered} \forall i \in \mathcal{V}, \quad \hat{A}^T(:,i)\boldsymbol{w}_t \geq \lambda_i \boldsymbol{x}_{t+1}(i) \\ \Updownarrow \\ \hat{A}^T \boldsymbol{w}_t - \Lambda \boldsymbol{x}_{t+1} \geq \mathbf{0}_n \end{gathered}$$

As $\boldsymbol{x}_{t+1} = \boldsymbol{w}_{t+1} - \boldsymbol{w}_t$ it follows:

$$\forall t \in \mathbb{N}, \qquad [\hat{A}^T + \Lambda]\boldsymbol{w}_t - \Lambda \boldsymbol{w}_{t+1} \geq \mathbf{0}_n$$

□

Given a seed set ϕ_0, all the components of the associated k-evolution vector respect equation (6.1). The k-evolution vector $\boldsymbol{w}$ associated to ϕ_0 is unique, and keeps all the information about the evolution of the innovation diffusion in k steps. There may exist however other vectors whose components satisfy equation (6.1) but do not represent the evolution of the innovation diffusion. We define these vectors as k-step vectors.

Definition 6 (k-step vector) *Let ϕ_0 be a seed set with characteristic vector $\hat{\boldsymbol{w}}_0$, and $\hat{\boldsymbol{w}}_0, \hat{\boldsymbol{w}}_1, \ldots, \hat{\boldsymbol{w}}_k$ be $k+1$ vectors of n elements. The vector $\hat{\boldsymbol{w}}^T = [\hat{\boldsymbol{w}}_0{}^T \ \hat{\boldsymbol{w}}_1^T \ \ldots \ \hat{\boldsymbol{w}}_k^T]$ is a k-step vector associated to ϕ_0 if $\forall i \in \{1, \ldots, k\}$ the component $\hat{\boldsymbol{w}}_i \in \{0,1\}^n$, and respects equation (6.1).*

Observe that, given a seed set ϕ_0 there could be several k-step vectors associated to it. Let us consider the network represented in Figure 1, and let $\lambda_1 = \lambda_2 = 0.49$ and $\lambda_3 = \lambda_4 = 0.60$. Let $\phi_0 = \{2\}$, whose characteristic vector is $\boldsymbol{x}_0 = [0100]^T$, then it is $\Phi_1 = \{1,2,3,4\}$ and $\boldsymbol{w}_1 = [1111]^T$. Thus, according to Lemma 6.1.1 and Definition 5, vector $\boldsymbol{w} = [01001111]^T$ is surely a possible 1-step vector associated to ϕ_0 and it is also its unique 1-evolution vector. However it is easy to verify that $\boldsymbol{w}$ is not the only possible 1-step vector associated to ϕ_0, but also $\hat{\boldsymbol{w}}' = [01000100]^T$ and $\hat{\boldsymbol{w}}'' = [01000101]^T$.

Lemma 6.1.2 *Let ϕ_0 be a seed set whose k-evolution vector is $\boldsymbol{w}$. For all possible k-step vectors $\hat{\boldsymbol{w}}'$ associated to ϕ_0 it holds:*

$$\boldsymbol{w}_k \geq \hat{\boldsymbol{w}}_k.$$

Proof: According to the linear threshold model, if an individual i can adopt the innovation at time $t \leq k$, then for each component $j \geq t$ of the k-evolution vector it holds $\boldsymbol{w}_j(i) = 1$, while in a k-step vector $\hat{\boldsymbol{w}}$ it can be $\hat{\boldsymbol{w}}_j(i) = 1$ or $\hat{\boldsymbol{w}}_j(i) = 0$, as in both cases equation (6.1) is respected. If an individual i can't adopt the innovation during the k steps, then for each component $j \geq k$ it must be $\boldsymbol{w}_j(i) = \hat{\boldsymbol{w}}_j(i) = 0$. Thus $\boldsymbol{w}_k \geq \hat{\boldsymbol{w}}_k$. □

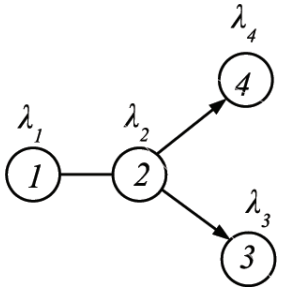


Figure 6.1: A network with $n = 4$ nodes.

Using the above definitions we propose now a BPP which solves the IMFTP(r, k). For a given network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the choice of the constraints guarantees that the optimal solution of the following BPP is a k-step vector associated to a seed set ϕ_0^* of r nodes, which maximize the spread of innovation in $\mathcal{G}$ in k steps. Moreover, we prove that the weights of the objective function guarantee that the optimal solution is the k-evolution vector associated to ϕ_0^*.

Proposition 6.1.1 *Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $|\mathcal{V}| = n$, consider the following BPP problem:*

$$\begin{array}{ll} \max & [\mathbf{1}_{nk}^T \quad (nk)\mathbf{1}_n^T] \cdot \boldsymbol{w} \\ \left\{ \begin{array}{ccccc} \mathbf{1}_n^T \boldsymbol{w}_0 & = & \boldsymbol{r} & & (a) \\ \forall i \in \{1, \ldots, k\}, & & & & \\ [\hat{A}^T + \Lambda]\boldsymbol{w}_{i-1} - \Lambda \boldsymbol{w}_i & \geq & \mathbf{0}_n & & (b) \\ \boldsymbol{w} & \in & \{0,1\}^{n(k+1)} & & (c) \end{array} \right. \end{array} \tag{6.3}$$

where $\boldsymbol{w}^T = [\boldsymbol{w}_0^T \ \boldsymbol{w}_1^T \ \ldots \ \boldsymbol{w}_k^T]$. Let $\boldsymbol{w}^*$ be an optimal solution of (6.3). Then:

- *$\boldsymbol{w}_0^*$ is the characteristic vector of the seed set ϕ_0^* which solve the IMFTP(r, k);*
- *$\boldsymbol{w}^*$ is the k-evolution vector of ϕ_0^*.*

Proof: From Definition 6 it follows that constraints (b) and (c) guarantee that each feasible solution of (6.3) must be a k-step vector associated to ϕ_0^*. We prove the properties above in two steps:

(i) firstly we prove that vector $\boldsymbol{w}_k^*$ is the characteristic vector of Φ_k^* starting from a seed set ϕ_0^*;

(ii) secondly we prove that $\boldsymbol{w}_0^*$ is the k-evolution vector of ϕ_0^*.

We analyse the two steps separately.

(i) We prove this statement by contradiction. Let the i-th component $\boldsymbol{w}_i^*$ of the optimal solution $\boldsymbol{w}^*$ be the characteristic vector of a set Θ_i. Let us suppose that $\Theta_k \neq \Phi_k$ starting from ϕ_0^*. As $\boldsymbol{w}^*$ is a k-step vector, by Lemma 6.1.2 it follows that $|\Phi_k| \geq |\Theta_k|$.

Let $|\Phi_k| = m \leq n$, than at maximum it can be $|\Theta_k| = m - 1$. For the characteristic vector $\boldsymbol{w}_k$ of Φ_k it holds:

$$(nk) \cdot \mathbf{1}_n^T \boldsymbol{w}_k = nkm$$

For the optimal solution Φ^* it can be at maximum:

$$|\phi_0^*| = |\Theta_1| = \ldots = |\Theta_k| = m - 1,$$

thus

$$\begin{aligned} [\mathbf{1}_{nk}^T \ (nk) \cdot \mathbf{1}_n^T]\boldsymbol{w}^* &\leq k(m-1) + nk(m-1) \\ &= nkm - nk + mk - k \end{aligned}$$

As $mk - nk$ is for sure a non-positive value, it follows that:

$$[\mathbf{1}_{nk}^T \ (nk) \cdot \mathbf{1}_n^T]\boldsymbol{w}^* < (nk) \cdot \mathbf{1}_n^T \boldsymbol{w}_k$$

thus Θ_k can't be the set whose characteristic vector is the k-th component of the optimal solution.

(ii) As the problem is a maximization, the value of the objective function is maximized when each individual adopts the innovation as soon as condition (5.1) is satisfied, hence each component $\boldsymbol{w}_i^*$ is the characteristic vector of Φ_k starting from the seed set ϕ_0^*. □

6.2 Diffusion of innovation over a target set

Another interesting problem in social network is the following: minimize the seed set ϕ_0 to diffuse the innovation over a target set of nodes $\Phi_d \subseteq \mathcal{V}$ in k time step. In this section we use the definitions of k-evolution vector and k-step vector to model a BPP which can be used to solve this problem.

Proposition 6.2.1 *Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $|\mathcal{V}| = n$, let $\boldsymbol{w}^T = [\boldsymbol{w}_0^T\ \boldsymbol{w}_1^T\ \dots\ \boldsymbol{w}_k^T]$ be a $n(k+1)$ vector and $\boldsymbol{x}_d$ be the characteristic vector of the target set $\Phi_d \subseteq \mathcal{V}$. Consider the following BPP:*

$$\begin{array}{l} \min \quad [\mathbf{1}_n^T\ \ \mathbf{0}_{nk}^T] \cdot \boldsymbol{w} \\ \left\{ \begin{array}{rcll} \boldsymbol{w}_k & \geq & \boldsymbol{x}_d & (a) \\ \forall i \in \{1, \dots, k\}, & & & \\ [\hat{A}^T + \Lambda]\boldsymbol{w}_{i-1} - \Lambda \boldsymbol{w}_i & \geq & \mathbf{0}_n & (b) \\ \\ \boldsymbol{w} & \in & \{0,1\}^{n(k+1)} & (c) \end{array} \right. \end{array} \tag{6.4}$$

Let $\boldsymbol{w}*$ be an optimal solution of (6.4). Then $\boldsymbol{w}_0^*$ is the characteristic vector of the minimum seed set which can diffuse the innovation over the target set Φ_d in k steps.

Proof: Constraints (b) and (c) guarantee that the optimal solution $\boldsymbol{w}^*$ is a k-step vector. Constraint (a) guarantees that, starting from a seed set ϕ_0^* with characteristic vector $\boldsymbol{w}_0^*$ a set $\Phi_k^* \supseteq \Phi_d$ can be reached in k steps. Moreover, as the problem is a minimization BPP, the seed set must be the minimum. □

Like (6.3), in BPP (6.4) the complexity grows as the number of steps k increases. The relaxed version of (6.4) can be used to compute a lower bound of its optimal solution.

Proposition 6.2.2 *Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $|\mathcal{V}| = n$, let $\boldsymbol{w}^T = [\boldsymbol{w}_0^T\ \boldsymbol{w}_1^T\ \dots\ \boldsymbol{w}_k^T]$ be a $n(k+1)$ vector and $\boldsymbol{x}_d$ be the characteristic vector of the target set $\Phi_d \subseteq \mathcal{V}$. Consider the following LPP:*

$$\begin{array}{l} \min \quad [\mathbf{1}_n^T\ \ \mathbf{0}_{nk}^T] \cdot \boldsymbol{w} \\ \left\{ \begin{array}{rcll} \boldsymbol{w}_k & \geq & \boldsymbol{x}_d & (a) \\ \forall i \in \{1, \dots, k\}, & & & \\ [\hat{A}^T + \Lambda]\boldsymbol{w}_{i-1} - \Lambda \boldsymbol{w}_i & \geq & \mathbf{0}_n & (b) \\ \\ \boldsymbol{w} & \geq & \mathbf{0}_{n(k+1)} & (c) \end{array} \right. \end{array} \tag{6.5}$$

Let $\boldsymbol{w}^*$ be an optimal solution of (6.5). The following properties hold:

(i) if $\boldsymbol{w}^* \in \{0,1\}^{n(k+1)}$ then $\boldsymbol{w}^*$ is also an optimal solution of (6.4);

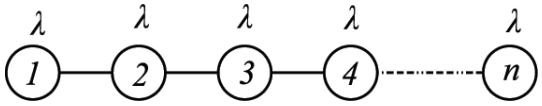


Figure 6.2: In this network if $\lambda \ll 0.5$ Algorithm 6 is more efficient than Algorithm 5.

(ii) $\lceil \mathbf{1}_n^T \boldsymbol{w}_0^* \rceil$ *is a lower bound on the cardinality of the minimum seed set which diffuses the innovation to the whole target set* Φ_d *in* k *steps.*

Proof: The two statements trivially follow by the definition of relaxed BPP. □

6.3 Numerical results

In this section we propose a small selection of the results obtained by the simulations of the proposed algorithms. Firstly we present a case in which Algorithm 6 is more efficient than Algorithm 5.

The network in Figure 2 represents a case in which Algorithm 6 can be more efficient than Algorithm 5 depending on the choice of λ.

Table 6.1

Algorithm 5				
ϕ_0	n	λ	Executed *while*-loops	Execution time (sec.)
{1}	1000	0.01	1000	6.6
{1}	1000	0.005	1000	6.6
{1}	1000	0.001	1000	6.6
Algorithm 6				
ϕ_0	n	λ	Executed *while*-loops	Execution time (sec.)
{1}	1000	0.01	100	5.7
{1}	1000	0.005	67	4.1
{1}	1000	0.001	30	1.9

Table 6.1 shows the results of the comparison of the two algorithms for different values of λ and $n = 1000$.

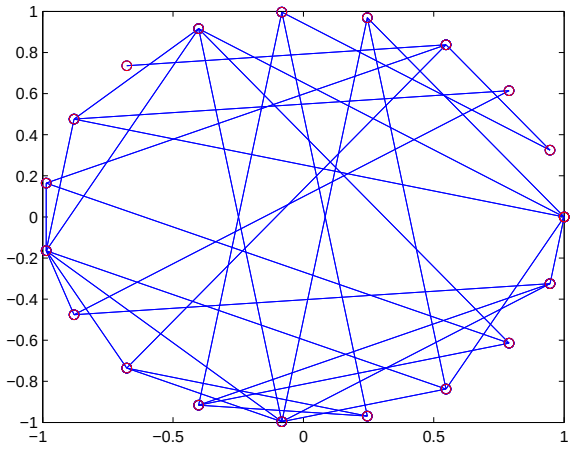


Figure 6.3: Network used to test BPP (6.3)

We have solved the IMFTP(r, k) in the network represented in Figure 6.3 using BPP (6.3) for different values of r and k. The values of λ are different at each node and have been randomly generated. The results of the experiment are plotted in Figure 6.4, in which the value of $|\Phi_k^*|$ is computed for different values of the parameters (r, k). As it was expected, if the value of k is fixed, the function $|\Phi_k^*|(r)$ is non-decreasing as well as the function $|\Phi_k^*|(k)$ if the value of r is fixed.

We have tested BPP (6.4) and LPP (6.5) in the network represented in Figure 6.5. In this case also the values of *lambda* are different at each node and have been randomly generated. The chosen target set is $\Phi_d = \mathcal{V}$. Figure 6.6 show the variation of $|\phi_0^*|$ computed with BPP (6.4) for different values of k, and the respective lower bound computed with LPP (6.5). As it was expected the function $|\phi_0^*|(k)$ is non-decreasing.

6.4 Conclusions

In this chapter we have discussed different aspects related to the diffusion of innovation in social networks. In the first part we have proposed a BPP characterization and an iterative algorithm based on LPP which compute the maximal cohesive subset of the complement of the seed set when the seed set is known. In

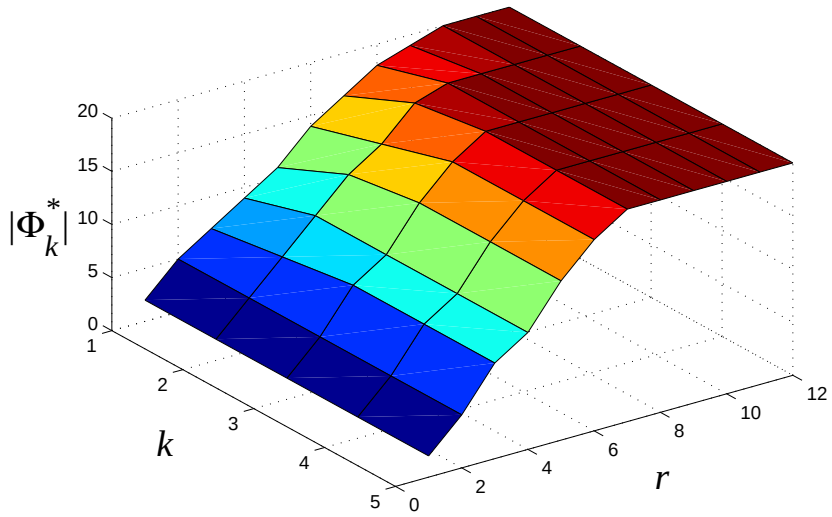


Figure 6.4: $|\Phi_k^*|$ obtained by BPP (6.3) for different values of k and r.

the second part a BPP model is presented that determines the set which maximizes the spread of innovation over the network in k steps.

This chapter presents a useful characterization of the Linear Threshold Model using vectors and matrices, and shows that there exist some problems which can be represented with BBPs and solved using their linear relaxations. We believe this preliminary approach can be applied to solve efficiently other problems of interest in social network analysis.

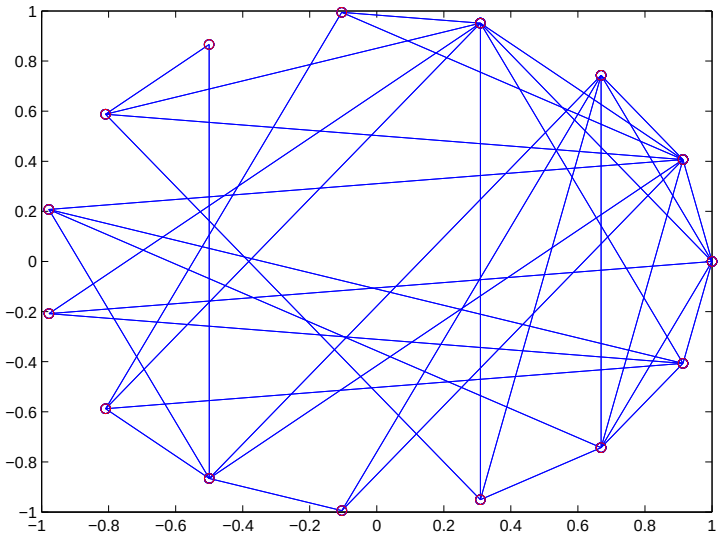


Figure 6.5: Network used to test BPPs (6.4) and (6.5).

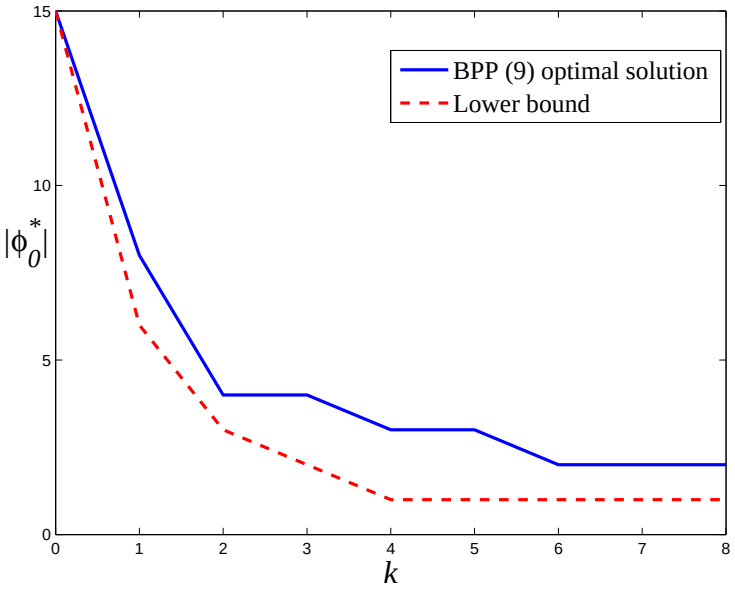


Figure 6.6: $|\phi_0^*|$ obtained by BPP (6.4) and its lower bound obtained by LPP (10) for different values of k

Chapter 7

A Non-Progressive instance of the Linear Threshold Model

The Chapter is organized as follows. In section 7.2 we introduce the *non-progressive linear threshold model*, formalizing the used notation, the main assumptions and the adopting conditions. In section 7.3 we define and characterize the *persistent sets* with respect of the presented model. Finally, in section 7.4 we analyse how the innovation spreads in a social network according to the non-progressive linear threshold model, and we confirm the analytical results through some numerical examples.

7.1 Background

Let us represent a social network with a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where each node $i \in \mathcal{V}$ represents an individual and each edge $(i, j) \in \mathcal{E}$ denotes that node i influences node j. We denote as $n = |\mathcal{V}|$ the number of individuals in the network. No self-loops are allowed, thus $(i, i) \notin \mathcal{E}$, $\forall i \in \mathcal{V}$. For all nodes $i \in \mathcal{V}$ we denote as $\mathcal{N}_i = \{j \mid (j, i) \in \mathcal{V}\}$ the set of the *in-neighbours*. A weight $w_{ij} \in [0, 1]$ is associated to each edge $(i, j) \in \mathcal{E}$ and denotes how much node i influences node j. We assume that for all $i \in \mathcal{V}$ it holds: $\sum_{j \in \mathcal{N}_i} w_{j,i} = 1$.

7.2 Non-Progressive Linear Threshold Model

In this section we introduce a non-progressive instance of the linear threshold model. Firstly we list the assumptions on which the model is based, then we define the update rule. For the rest of the chapter we refers to this model as the *non-progressive linear threshold model*.

7.2.1 System description

A threshold value $\lambda_i \in [0, 1]$ is associated to all nodes $i \in \mathcal{V}$. We assume that the independent variable time t belongs to $\mathbb{N}$. The innovation spreads in the network starting from a seed set ϕ_0, i.e., a set of individuals are active at time $t = 0$. We assume that all the nodes in ϕ_0 are active for a time interval $t \in [0, T_s]$, independently of the state of their neighbours, then for $t > T_s$ they update their state following the same rule as the rest of the nodes. We call T_s the *seeding time*.

We assume that:

- the topology of the network is static and all the connections and the influence weights are known;
- the thresholds $\lambda_i, \forall i \in \mathcal{V}$ are static and known;
- a node can be more influenced by some neighbours than others, thus for each node the weights of the in-edges may be different.

7.2.2 Update rule

Let Φ_t be the set of active nodes at time t. In the non-progressive linear threshold model the nodes update their states at time t according to the following equation:

$$\Phi_t = \begin{cases} \phi_0 & t = 0 \\ \phi_0 \bigcup \{i \mid \sum_{j \in (\mathcal{N}_i \cap \Phi_{t-1})} w_{ji} \geq \lambda_i\}, & t \in [1, T_s] \\ \{i \mid \sum_{j \in (\mathcal{N}_i \cap \Phi_{t-1})} w_{ji} \geq \lambda_i\}, & t > T_s \end{cases} \tag{7.1}$$

In other words, after the seeding time a node is active at time t if the sum of the weights of the in-edges coming from active neighbours at time $t-1$ is greater than or equal to its threshold. Differently from the progressive model, in which a

node maintains the innovation indefinitely once adopted, in the non-progressive model a node can switch its state from inactive to active and vice versa.

Additional notation that will be used in the rest of the chapter is the following.

- $\phi_t^+ = \Phi_t \setminus \Phi_{t-1}$, i.e., the set of nodes which become active at time t;
- $\phi_t^- = \Phi_{t-1} \setminus \Phi_t$, i.e., the set of nodes which become inactive at time t;
- $\Phi^* = \lim_{t \to +\infty} \Phi_t$ denotes if it exists, the set of final adopters.

Note that the set Φ^* does not always exist. The existence of this set will be discussed in section 7.4.

7.3 Cohesive and Persistent Sets

In this section we define two types of cohesive groups in the non-progressive linear threshold model, which are useful to analyse the spread of innovation in the network. We firstly adapt to our model the concept of *cohesive sets* as presented in (74). Then we introduce the idea of *persistent sets*, which describe a different type of coherence with respect to cohesive sets.

Definition 7 (Cohesive set ((74))) *A set X is cohesive if for all nodes $i \in X$ the sum of the weights of the in-edges coming from nodes which are not in X is lower than their threshold λ_i, i.e.:*

$$\forall i \in X, \quad \sum_{j \in (\mathcal{N}_i \cap X)} w_{ji} > 1 - \lambda_i. \tag{7.2}$$

An important property of a cohesive set, proved in (74), is that if none of the nodes within the set is active at time t, then none of them can become active for all $t' > t$. In Figure 7.1 the sets $\{1, 2, 3\}$ and $\{8, 9\}$ are cohesive, while $\{4, 5, 6, 7\}$ is not cohesive.

Definition 8 (Persistent set) *A set X is persistent if for all nodes $i \in X$ the sum of the weights of the in-edges coming from nodes within X is greater than or equal their threshold λ_i, i.e.:*

$$\forall i \in X, \quad \sum_{j \in (\mathcal{N}_i \cap X)} w_{ji} \geq \lambda_i. \tag{7.3}$$

The following theorem points out the reason why such type of sets are important in the non-progressive linear threshold model.

Theorem 7.3.1 *Let X be a persistent set. If at time t' all the nodes in X are active, then they remain active for all $t > t'$.*

Proof: If all nodes in X are active at time t', i.e., $X \subseteq \Phi_{t'}$, from (7.3) follows that

$$\forall i \in X, \quad \sum_{j \in (\mathcal{N}_i \cap \Phi_{t'})} w_{ji} \geq \sum_{j \in (\mathcal{N}_i \cap X)} w_{ji} \geq \lambda_i.$$

hence $X \subseteq \Phi_{t'+1}$. The result follows by recursion. □

Property 7.3.2 *Let X_1 and X_2 be two persistent sets. The set $X_1 \cup X_2$ is a persistent set as well.*

Proof: As X_1 is persistent, each node i in X_1 satisfies equation (7.3). As $X_1 \subseteq X_1 \cup X_2$ it holds for $k = 1, 2$:

$$i \in X_k, \quad \sum_{j \in (\mathcal{N}_i \cap (X_1 \cup X_2))} w_{ji} \geq \sum_{j \in (\mathcal{N}_i \cap X_k)} w_{ji} \geq \lambda_i.$$

Thus all the nodes in $X_1 \cup X_2$ satisfy equation (7.3), i.e., $X_1 \cup X_2$ is a persistent set. □

In Figure 7.1 the sets $\{1, 2, 3\}$ and $\{4, 5, 6, 7\}$ are persistent, while $\{3, 4\}$ is not persistent. We conclude this section by observing that a set can be both cohesive and persistent, e.g., the set $\{1, 2, 3\}$.

7.4 System's dynamic

The purpose of this section is to characterize how the innovation spreads in the network according to the non-progressive model. We analyse separately two different phases of the evolution in the network:

- during the seeding time, i.e. for $0 \leq t \leq T_s$;
- after the seeding time, i.e., for $t > T_s$.

We pay particular attention to the evolution of the innovation after the seeding time: which are the nodes that are able to hold their states active after T_s?

We use the following definitions to describe the evolution of the innovation in the network according to the presented model.

Definition 9 (Progressive evolution) *The diffusion of the innovation in the network is* progressive *(or* non-decreasing*) during a time interval* $[t_1, t_2]$ *if:*

$$\forall t \in [t_1, t_2], \qquad \phi_t^- = \emptyset.$$

In other words, for all $t \in [t_1, t_2]$ all active nodes $i \in \Phi_{t-1}$ remain active at time t. If $t_1 = t_2 = t'$, we said that the evolution is progressive in t' if $\phi_{t'}^- = \emptyset$.

Definition 10 (Non-progressive evolution) *The diffusion of the innovation in the network is* non-progressive *during a time interval* $[t_1, t_2]$ *if:*

$$\exists t \in [t_1, t_2], \qquad \phi_t^- \neq \emptyset.$$

In other words, during the time interval $t \in [t_1, t_2]$ there is at least a node which becomes inactive.

Definition 11 (Degressive evolution) *The diffusion of the innovation in the network is degressive (or* non-increasing*) during a time interval* $[t_1, t_2]$ *if:*

$$\forall t \in [t_1, t_2], \qquad \phi_t^+ = \emptyset.$$

Definition 12 (Periodic evolution) *The diffusion of the innovation in the network is periodic after time* t *if there exist a* $T > 0 \in \mathbb{N}$ *such that:*

$$\forall k \in \mathbb{N}, t' \geq t \qquad \Phi'_t = \Phi_{t'+kT}.$$

where T *is the period of the evolution.*

The definitions of progressive and degressive follow the usual definitions in literature. Note that an evolution can be both progressive and degressive if the set of active nodes is constant. In the following parts we prove analytically the following results:

(a) during the seeding time the system has a progressive evolution;

(b) after the seeding time the evolution of the system is progressive if Φ_{T_s} is persistent, otherwise is non-progressive;

(c) if T_s is sufficiently large (larger than a parameter T_d called diffusion time and introduced in the following section) two results holds: a) the set of final adopters Φ^* exists and is the maximal persistent set in Φ_{T_s}; b) if Φ_{T_s} is not persistent the system has a degressive evolution for $t > T_s$.

Examples of evolutions, including a case in which the system has a periodic evolution, are given in the final subsection.

7.4.1 Evolution during the seeding time: $0 \leq t \leq T_s$

In this part we prove that in the non-progressive model, according to the assumptions made so far, during the seeding time $[0, T_s]$ the system has a *progressive evolution*.

Theorem 7.4.1 *The evolution of a social network with seed set ϕ_0 and seeding time T_s is progressive in the time interval $[0, T_s]$.*

Proof: We prove the statement by induction on the time step t, assuming $T_s \geq 1$ (if $T_s = 0$ the result is trivial).

(base step) At time step $t = 1$, the evolution is progressive because by equation (7.1) $\Phi_0 = \phi_0 \subseteq \Phi_1$, hence $\phi_1^- = \emptyset$.

(inductive step) Assume that at time step $t-1$ (where $t \in [2, T_s]$) the evolution is progressive: we now show that the evolution is also progressive at time step t thus completing the proof.

Observe that the assumption $\phi_{t-1}^- = \emptyset$ implies $\Phi_{t-2} \subseteq \Phi_{t-1}$, hence for all $i \in \mathcal{V}$ holds:

$$\mathcal{N}_i \cap \Phi_{t-2} \subseteq \mathcal{N}_i \cap \Phi_{t-1}.$$

By (7.1) this implies that $\Phi_{t-1} \subseteq \Phi_t$, hence $\phi_t^- = \emptyset$. □

The previous analysis also points out that as long as the nodes of the seed set are active, no node in the network can become inactive, i.e., during the seed time a node, which is not in the seed set, adopts the innovation as soon as the sum of the weights of the in-edges coming from active nodes is greater than or equal its threshold value, and maintains it.

This behaviour is also typical of the progressive instance of the linear threshold model presented in (74). Differently from our model, the progressive in (74)

assumes that all in-edges at each node have the same weight, i.e., for all $i \in \mathcal{V}$ it holds:

$$w_{ji} = \frac{1}{|\mathcal{N}_i|}, \quad \forall j \in \mathcal{N}_i.$$

In the progressive model an inactive node i adopt the innovation at time t if at time $t-1$ it holds:

$$\sum_{j \in (\mathcal{N}_i \cap \Phi_{t-1})} w_{ji} = \frac{|\Phi_{t-1} \bigcap \mathcal{N}_i|}{|\mathcal{N}_i|} \geq \lambda_i \tag{7.4}$$

According to the previous equation, also in the progressive model a node adopts the innovation as soon as the sum of the weights of the in-edges coming from active nodes is above its threshold value, but differently from our non-progressive model an individual is assumed to never abandon the innovation once adopted. Thus we can claim that the non-progressive linear threshold model represents a generalization of the progressive model. In particular, the evolution of the progressive model corresponds to the evolution of the non-progressive model in case of $T_s \to \infty$.

We can exploit this similarity even further. We know from (74) that the progressive model reaches in a finite time a steady state where the set of active nodes remains constant and is:

$$\hat{\Phi}^* = \mathcal{V} - \mathcal{M}$$

, where $\mathcal{M}$ denotes the maximal cohesive set in the complement of the seed set.

Motivated by this, we define a parameter, the diffusion time, which will play an important role in the analysis of the evolution of the non-progressive model as will be shown in the following sections.

Definition 13 (Diffusion Time T_d) *For T_s sufficiently large the innovation spreads in the network until a time $T_d \leq T_s$ such that $\Phi_{T_d} = \Phi_{T_d+1} = \cdots = \Phi_{T_s}$. The parameter T_d is the* diffusion time *of the network.*

7.4.2 Evolution after the seeding time: $t > T_s$

At time $T_s + 1$ some nodes in the seed set may become inactive, as they may not satisfy equation (7.1). If that happens, at time $T_s + 2$ some active nodes

connected to the seed set may become inactive, etc. Such a tendency to abandon the innovation leads to a *non-progressive evolution*.

In this section we characterize the evolution of our model after the seeding time and also present some particular results that hold in the special case $T_s < T_d$.

Lemma 7.4.1 *Consider a social network with seeding time T_s. If there exists a time step $\bar{t} > T_s$ such that the evolution in $\bar{t}$ is progressive, then the evolution is also progressive for all $t > \bar{t}$.*

Proof: Observe that the assumption $\phi_{\bar{t}}^- = \emptyset$ implies $\Phi_{\bar{t}-1} \subseteq \Phi_{\bar{t}}$, hence for all $i \in \mathcal{V}$ holds

$$\mathcal{N}_i \cap \Phi_{\bar{t}-1} \subseteq \mathcal{N}_i \cap \Phi_{\bar{t}}.$$

By (7.1) this implies that $\Phi_{\bar{t}} \subseteq \Phi_{\bar{t}+1}$, hence $\phi_{\bar{t}+1}^- = \emptyset$. The result follows by recursion. □

The following theorem fixes the conditions under which the evolution of the system remains progressive for $t > T_s$.

Theorem 7.4.2 *Consider a social network with seed set ϕ_0 and seeding time T_s. The evolution of the network is progressive for all $t > 0$ if and only if Φ_{T_s} is persistent.*

Proof: We prove separately the if and only if parts.

(if) For $0 \leq t \leq T_s$ it has been shown in Theorem 7.4.1 that the network has a progressive evolution. If Φ_{T_s} is persistent, Theorem 7.3.1 implies that the evolution at time step $T_s + 1$ is progressive. From Lemma 7.4.1 one concludes that the evolution is also progressive for all time steps $t > T_s + 1$.

(only if) If Φ_{T_s} is not persistent, by Definition 8 there exists a node $i \in \Phi_{T_s}$ such that $\sum_{j \in (\mathcal{N}_i \cap \Phi_{T_s})} w_{ji} < \lambda_i$. By (7.1) if follows that node i becomes inactive at step $T_s + 1$, hence the network has a non-progressive evolution. □

The following corollary points out that to determine if the system has a progressive evolution after T_s it is sufficient to determine if all nodes in the seed set remain active at time $T_s + 1$.

Corollary 7.4.1 *The evolution of a social network with seed set ϕ_0 and a seed time T_s is progressive for all $t > 0$ if and only if at time T_s+1 it holds: $\phi_0 \cap \phi_{T_s+1}^- = \emptyset$.*

Proof: Since $\phi_0 \cap \phi^-_{T_s+1} = \emptyset$ it holds

$$\phi_0 \subseteq \left\{ i \mid \sum_{j \in (\mathcal{N}_i \cap \Phi_{T_s})} w_{ji} \geq \lambda_i \right\}$$

hence

$$\begin{aligned} &\phi_0 \bigcup \left\{ i \mid \sum_{j \in (\mathcal{N}_i \cap \Phi_{T_s-1})} w_{ji} \geq \lambda_i \right\} \\ &\subseteq \left\{ i \mid \sum_{j \in (\mathcal{N}_i \cap \Phi_{T_s})} w_{ji} \geq \lambda_i \right\} \end{aligned}$$

and by (7.1) this implies that $\Phi_{T_s+1} \subseteq \Phi_{T_s}$. The result follows from Lemma 7.4.1. □

The following theorem points out a sufficient condition on the structure on the seed set under which the evolution of the system is progressive.

Theorem 7.4.3 *Consider a social network with seed set ϕ_0 and seeding time T_s. If ϕ_0 is persistent, the evolution of the network is progressive for all $t > 0$.*

Proof: To prove this statement is sufficient to prove that if ϕ_0 is persistent, then Φ_{T_s} is persistent as well. We can consider Φ_{T_s} as:

$$\Phi_{T_s} = \phi_0 + \phi_1^+ + \phi_2^+ + \dots \phi_{T_s}^+$$

. Since ϕ_0 is persistent, it holds:

$$\phi_0 \in \Phi_{T_s+1}.$$

Since all the nodes in ϕ_0 are active at time $T_s + 1$, it holds:

$$\phi_1^+ \in \Phi_{T_s+1}.$$

Using the same argument we can observe that:

$$\phi_2^+ \in \Phi_{T_s+1};\ \dots\ ; \phi_{T_s}^+ \in \Phi_{T_s+1}$$

. Thus it follows that:

$$\phi^-_{T_s+1} = \emptyset$$

and from Corollary 7.4.1 it follows that the evolution is progressive for $t > 0$. □

We now present some results that apply to the special case in which $T_s \geq T_d$. If this condition holds, the progressive evolution during the seeding time reaches a steady state and $\Phi_{T_d} = \Phi_{T_d+1} = \dots = \Phi_{T_s}$.

Next theorem points out which are the nodes that remain active for all $t > T_d$.

Theorem 7.4.4 *Let ϕ_0 be a seed set of a social network with a seed time T_s and diffusion time $T_d < T_s$. If $\Phi_{T_s} = \Phi_{T_d}$ is not persistent, then the system has a degressive evolution for $t > T_s$.*

Proof: The proof is based on verifying the following two facts.

(a) Firstly, we prove that if Φ_{T_s} is non-persistent, then $\phi^+_{T_s+1} = \emptyset$ and $\phi^-_{T_s+1} \neq \emptyset$. Observe that if $\Phi_{T_s} = \Phi_{T_d}$ is not persistent it follows from Theorem 7.4.3 that $\phi^-_{T_s+1} \neq \emptyset$. Moreover, as $T_s > T_d$, it holds that $\mathcal{V} - \Phi_{T_s} = \mathcal{M}$, where $\mathcal{M}$ is the maximal cohesive subset of the complement of the seed set. Thus no nodes can adopt the innovation at time $T_s + 1$, i.e., $\phi^+_{T_s+1} = \emptyset$.

(b) Secondly we prove that for all $t > T_s + 1$ it holds $\phi^+_t = \emptyset$. At time $T_s + 1$ it holds $\Phi_{T_s+1} \subseteq \Phi_{T_s}$, thus according to equation (7.1) it holds $\phi^+_{T_s+2} = \emptyset$. By the iteration of the same argument, for all $t > T_s + 1$ it is:
$\Phi_t \subseteq \Phi_{t-1} \quad \Leftrightarrow \quad \phi^+_{t+1} = \emptyset$ □

Theorem 7.4.5 *Let ϕ_0 be a seed set of a social network with seed time T_s and diffusion time $T_d < T_s$. The set Φ^* of active nodes for $t \to \infty$ is the maximal persistent set contained in Φ_{T_s} and is reached at time $T_f \leq T_s + |\Phi_{T_s}| - |\Phi^*|$.*

Proof: If the set of active nodes at step t is not persistent, there is at least one node in Φ_t that becomes inactive at step $t + 1$. This, since the evolution is degressive according to Theorem 7.4.4, the number of active nodes decreases at each step until the system reaches a persistent set of active nodes Φ^*, which is the maximal persistent set contained in Φ_{T_s}. The steady state is achieved from T_s in a number of steps which is at maximum $|\Phi_{T_s}| - |\Phi^*|$, thus:

$$T_f \leq T_s + |\Phi_{T_s}| - |\Phi^*|$$

. □

7.4.3 Some examples

In this section we consider social networks with seeding time T_s smaller than the diffusion time T_d because in this case several types of evolutions are possible, as opposed the networks with $T_s \geq T_d$ that we have shown can only admit degressive evolutions after the seeding time. We illustrate three different scenarios separately through examples.

Example 7.4.6 (Scenario 1: progressive evolution) *Consider the network in Fig. 7.1 with seed set* $\phi_0 = \{1, 2\}$ *and seeding time* $T_s = 2$*. The diffusion time for the considered network is* $T_d = 4$*. As it is shown in Fig. 7.2, the evolution of the system is progressive. According to Theorem 7.4.3 the progressive evolution can be predicted by observing that* $\Phi_{T_S} = \Phi_2$ *is a persistent set, as all the nodes that belong to it satisfy equation (7.3). The set of final adopters exists and is* $\Phi^* = \{1, 2, 3, 4, 5, 6, 7\}$.

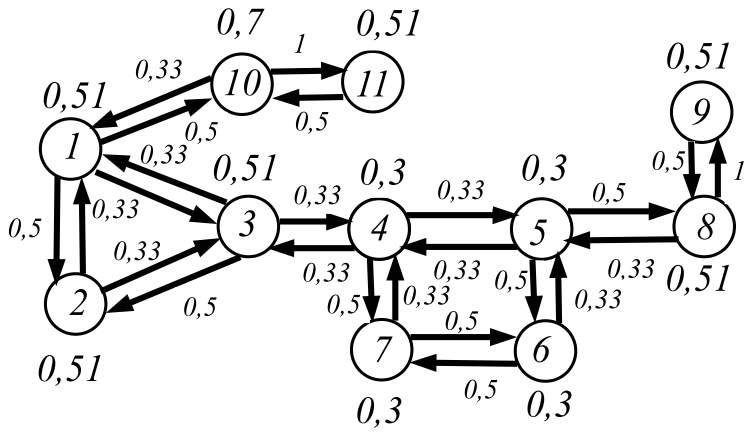


Figure 7.1: Network in scenario 1.

t	Φ_t	ϕ_t^+	ϕ_t^-
0	$\{1, 2\}$		
1	$\{1, 2, 3\}$	$\{3\}$	$\emptyset$
2	$\{1, 2, 3, 4\}$	$\{4\}$	$\emptyset$
3	$\{1, 2, 3, 4, 5, 7\}$	$\{5, 7\}$	$\emptyset$
4	$\{1, 2, 3, 4, 5, 6, 7\}$	$\{6\}$	$\emptyset$
5	$\{1, 2, 3, 4, 5, 6, 7\}$	$\emptyset$	$\emptyset$

Figure 7.2: Evolution in scenario 1.

Example 7.4.7 (Scenario 2: non-progressive evolution) *Consider the network in Fig. 7.3 with seed set* $\phi_0 = \{1, 3\}$ *and seeding time* $T_s = 1$*. The diffusion time for the considered network is* $T_d = 3$*. As it is shown in Fig. 7.4, the evolution of the system is non-progressive. The set of final adopters exists and is* $\Phi^* = \emptyset$*.*

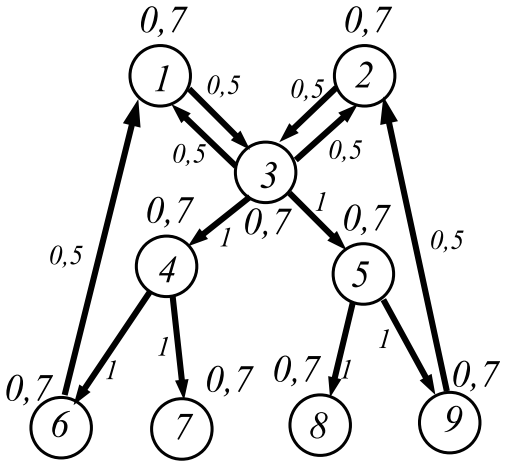


Figure 7.3: Network in scenario 2.

t	Φ_t	ϕ_t^+	ϕ_t^-
0	$\{1, 2\}$		
1	$\{1, 2, 3\}$	$\{3\}$	$\emptyset$
2	$\{3, 4, 5\}$	$\{4, 5\}$	$\{1, 2\}$
3	$\{4, 5, 6, 7, 8, 9\}$	$\{6, 7, 8, 9\}$	$\{3\}$
4	$\{6, 7, 8, 9\}$	$\emptyset$	$\{4, 5\}$
5	$\emptyset$	$\emptyset$	$\{6, 7, 8, 9\}$

Figure 7.4: Evolution in scenario 2.

The numerical results confirm the analytical result obtained in Theorem 7.4.3: as the set Φ_{T_s} *is non-persistent, the system has a non-progressive evolution.*

The next example represent a case in which the evolution of the system is periodic after T_s. This is a particular, but interesting, case of non-progressive evolutions

but so far we have not found any analytical characterization of this behavior.

Example 7.4.8 (Scenario 3: periodic evolution.) *Consider the network in Fig. 7.5 with seed set* $\phi_0 = \{1, 3\}$ *and seeding time* $T_s = 1$*. The diffusion time for the considered network is* $T_d = 2$*. As it is shown in Fig. 7.6, the evolution of the system is non-progressive after* T_s*, as the set* Φ_{T_s} *is non-persistent. Moreover, the system has a periodic evolution with period* $T = 2$ *from* $t = 2$*. In this case the set* Φ^* *cannot be defined.*

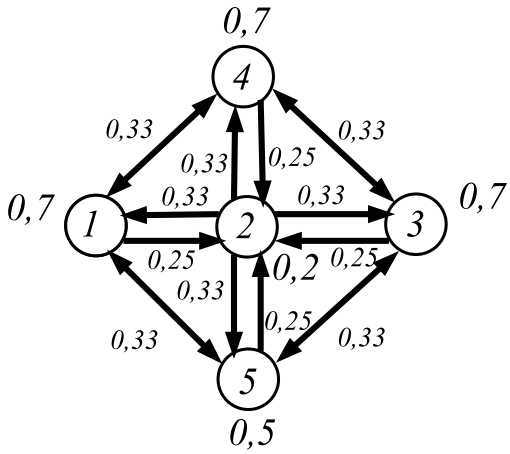


Figure 7.5: Network in scenario 3.

t	Φ_t		
0	$\{1, 3\}$	ϕ_t^+	ϕ_t^-
1	$\{1, 2, 3, 5\}$	$\{2, 5\}$	$\emptyset$
2	$\{2, 4, 5\}$	$\{4\}$	$\{1, 3\}$
3	$\{1, 2, 3\}$	$\{1, 3\}$	$\{4, 5\}$
4	$\{2, 4, 5\}$	$\{4, 5\}$	$\{1, 3\}$
5	$\{1, 2, 3\}$	$\{1, 3\}$	$\{4, 5\}$
6	$\{2, 4, 5\}$	$\{4, 5\}$	$\{1, 3\}$

Figure 7.6: Evolution in scenario 3.

7.5 Conclusions

In this chapter we have presented a non-progressive instance of the linear threshold model, in which the diffusion of the innovation starts from a seed set whose nodes are assumed to maintain the innovation for a finite time. We characterized analytically the conditions under which the system has a progressive, non-progressive and degressive evolution. This model represents a first step in the analysis of non-progressive mechanisms dealing with the linear threshold model. In our future works we want to extend the presented model by exploring other mechanisms which can lead the network to a non-progressive evolution, such as changes in the network topology or in the influence weights. Furthermore we also plan to characterize the set of final adopters when $T_s < T_d$ and to find some conditions on the graph structure to characterize the evolution on the network.

Chapter 8

Conclusions

In this thesis we have presented several algorithm, based on graph theory, on two main topics: the coordination of multi-agent systems through consensus and the diffusion of innovation in social networks.

Regarding the coordination of multi-agent systems, the following are the presented results.

- In Chapter 2 a formation control strategy for a set of autonomous vehicle in absence of a common reference frame, based on gossip, is proposed. If the agent have a common reference direction the algorithm is proved to be robust to noise on the displacement measurement. To the best of our knowledge this algorithm is a rare example in literature of formation control strategy in absence of a common reference frame, which is not characterized by a leader.

- In Chapter 3 we have proposed the Heterogeneous Multi Vehicle Routing Problem (HMVRP), which represent an extension of the classical Multi Vehicle Routing Problem. We have proposed upper and lower bounds for the cost of the optimal solution. Furthermore, we proposed two algorithm based on gossip to solve the HMVRP in a distributed fashion exploiting only pairwise task exchanges between vehicles, thus greatly reducing the computational complexity required to compute a solution. The proposed methods scales with exponential complexity with respect to the ratio between the number of tasks and vehicles instead of scaling with respect to the number of tasks. We believe that our framework can be extended to the

case of Dynamic MVRP, in which robots start to move and serve tasks while the decentralized optimization is being executed and new tasks appear in the region

Regarding the diffusion of the innovation in social networks, the following are the proposed results.

- In Chapter 5 we have adopted the classical Linear Threshold Model for the diffusion of innovation in Social Network. We firstly have proposed an algorithm, based on linear programming, which computes the maximal cohesive subset of a network. This algorithm can be used to compute the set of final adopters for a given seed set of nodes. Then we have characterized the optimal solutions of two problems: the Influence Maximization in Finite Time and the diffusion of innovation over a target set. The framework presented in this chapter represents a useful characterization of the Linear Threshold Model using vectors and matrices, and shows that there exist some problems which can be represented with BBPs and solved using their linear relaxations. We believe this preliminary approach can be applied to solve efficiently other problems of interest in social network analysis,. Another interesting orientation for future work is the study of heuristic approaches to the presented problems, since most of them have a combinatorial nature.

- In Chapter 7 we have defined and analysed a novel model, the Non Progressive Linear Threshold Model, which extends the classical model and, differently from it, is suitable to represent non progressive phenomena of innovation diffusion. We have characterized the evolution of the network in therms of Cohesive and Persistent sets. The analysis of innovation diffusion phenomena through the analysis of the cohesion in the network represent an actual and still open problem. We believe that this technique an be extended to other models which represent diffusion phenomena. Furthermore, the proposed model represents a first step in the analysis of non-progressive mechanisms dealing with the linear threshold model, which can be extended by exploring other phenomena which can lead the network to a non-progressive evolution, such as time-varying network topology or in the time varying edge weights.

List of papers by the author

Journal articles

(1) Franceschelli M., Seatzu C., Rosa D., and Bullo F., "Gossip algorithms for heterogeneous multi-vehicle routing problems." International Journal of Nonlinear Analysis: Hybrid Systems: Nov 2013

Conference Proceedings with acceptance based on submission of full paper and peer-review.

(1) Rosa D., Giua A., "A non progressive model of innovation diffusion in social networks" CDC conference 2013.

(2) Rosa D., Giua A., "On the spread of innovation in social networks. " Necsys conference 2013 (Germany, Sept. 2013).

(3) Rosa D., Franceschelli M., Seatzu C., and Bullo F., "An Heuristic for the initial task assignment in the heterogeneous multi-vehicle routing problems. ", NECSYS Conference 2012 (Santa Barbara, Sept 2012).

(4) Franceschelli M., Seatzu C., Rosa D., and Bullo F., "A gossip algorithm for heterogeneous multi-vehicle routing problems." ADHS conference 2012 (Ehindoven, June 2012).

(5) Rosa D., Franceschelli M., Giua A., "Robust Common Reference Estimation and formation control for Multi-Agent Systems" American Control Conference, ACC 2012 (Montreal, June 2012).

Appendix A

Appendix

A.1 Algebraic graph theory

A graph can be defined as $G = \{V, E\}$ where $V = \{1, \ldots, n\}$ is the set of n nodes or vertices, which in our thesis represent agents or individuals and $E \subseteq \{V \times V\}$ is the set of edges, which represents the existence of an interaction between any given couple of nodes. A graph can be directed (digraph) or undirected. A graph G is directed if to each edge (i, j) we associate a direction. We call *head* of the edge node i and *tail* node j, finally we say that edge (i, j), which sometime is referred as $e_{i,j}$ in short, goes from node j to node i.

A *loop* is an edge whose endpoints are the same. A *walk* $w_{i,j}$ from node i to node j in G is an alternate sequence of vertices and edges, for instance

$$w_{1,3} = v_1, e_{1,2}, v_2, e_{3,2}, v_3.$$

A *path* $p_{i,j}$ from node i to node j in G is an alternate sequence of vertices and edges, for instance

$$p_{1,3} = v_1, e_{1,2}, v_2, e_{2,3}, v_3.$$

In an undirected graph in which edges do not have a direction, a walk is equivalent to a path.

A graph is:

- *disconnected* if there exists two nodes i and j and there does not exist a walk from i to j;

- *weakly connected* if for any couple of nodes $i, j \in V$ there exists a walk between i and j;
- *quasi-strongly connected* if from each node $i \in V$ there exist a path to node w;
- *strongly connected* if there exists a path between each pair of nodes $i, j, \in V$.

If graph G is undirected, it can be only disconnected or connected.

Dynamic case

We define a time-varying graphs as $G(t) = \{V, \mathcal{E}(t)\}$ where $V = \{1, \ldots, n\}$ is the set of nodes and $\mathcal{E}(t) \subseteq \{V \times V\}$ is the time-varying set of edges that map each instant of time into a set of edges $\mathcal{E} : \mathbb{R} \longrightarrow E$. We define the union of graph $G_1 = \{V_1, E_1\}$ and $G_2 = \{V_2, E_2\}$ as the graph $G = G_1 \bigcup G_2 = \{V_1 \bigcup V_2, E_1 \bigcup E_2$ whose vertex and edge set is the union of those of G_1 and G_2. Given an interval of time $[t, t']$ we define the union graph $\mathcal{G}[t, t']$ over an interval of time as

A time-varying graph $G(t)$ is uniformly strongly connected if for any t there exists T in which $\mathcal{G}[t, t + T]$ is strongly connected.

Bibliography

[1] J. Yu, S. LaValle, and D. Liberzon, "Rendezvous without coordinates," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on.* IEEE, 2008, pp. 1803–1808. 9

[2] A. Tahbaz-Salehi and A. Jadbabaie, "Distributed coverage verification in sensor networks without location information," *Automatic Control, IEEE Transactions on*, vol. 55, no. 8, pp. 1837–1849, 2010. 9

[3] V. de Silva and R. Ghrist, "Coordinate-free coverage in sensor networks with controlled boundaries via homology," *The International Journal of Robotics Research*, vol. 25, no. 12, p. 1205, 2006. 9

[4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006. 9

[5] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques.* ACM, 1987, pp. 25–34. 9

[6] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005.* IEEE, 2005, pp. 1859–1864. 9

[7] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007. 9, 20

[8] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, 2003. 9, 20, 22

[9] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in fixed and switching networks," *Automatic Control, IEEE Transactions on*, vol. 52, no. 5, pp. 863–868, 2007. 9

[10] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1465–1476, 2004. 9

[11] T. Balch and R. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998. 9

[12] H. Tanner, G. Pappas, and V. Kumar, "Leader-to-formation stability," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 443–455, 2004. 9

[13] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 947–951, 2001. 9

[14] M. Franceschelli and A. Gasparri, "Decentralized centroid estimation for multi-agent systems in absence of any common reference frame," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 512–517. 10, 27, 31

[15] A. Gasparri and M. Franceschelli, "On Agreement Problems with Gossip Algorithms in absence of common reference frames," in *IEEE International Conference on Robotics and Automation, ICRAâĂŹ10*. 10

[16] G. Gutin and A. P. Punnen, *The Traveling Saleman Problem and Its Variations*, ser. Series in Combinatorial Optimization, Springer. Kluwer Academic Publishers, 2002, vol. 12. 10

[17] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992. 10

[18] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy-Kan, and D. B. Shmoys, *The Traveling Saleman Problem: A Guided Tour of Combinatorial Optimization*, ser. Wiley Series in Discrete Mathematics and Optimization. John Wiley and Sons, 1985. 10

[19] G. Dantzig and J. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959. 10

[20] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006. 11, 39

[21] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, 2007. 11

[22] P. Toth and D. Vigo, *The Vehicle Routing Problem*, ser. Monographs on Discrete Mathematics and Applications. SIAM, 2002, vol. 9. 11

[23] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith, "Dynamic vehicle routing for robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, 2011. 11

[24] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992. 11

[25] J. Cordeau, M. Gendreau, G. Laporte, J. Potvin, and F. Semet, "A guide to vehicle routing heuristics," *Journal of the Operational Research Society*, vol. 53, no. 5, pp. 512–522, 2002. 11

[26] M. Fisher and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, no. 2, pp. 109–124, 1981. 11

[27] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Management Science*, vol. 40, no. 10, pp. 1276–1290, 1994. 11

[28] G. Laporte, M. Gendreau, J. Potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *International Transactions in Operational Research*, vol. 7, no. 4-5, pp. 285–300, 2000. 11

[29] R. Nallusamy, K. Duraiswamy, R. Dhanalaksmi, and P. Parthiban, "Optimization of multiple vehicle routing problems using approximation algorithms," *Arxiv preprint arXiv:1001.4197*, 2010. 11

[30] J. Carlsson, D. Ge, A. Subramaniam, A. Wu, and Y. Ye, "Solving min-max multi-depot vehicle routing problem," *Lectures on Global Optimization*, vol. 55, pp. 31–46, 2009. 11, 12, 36, 41, 67

[31] M. Franceschelli, D. Rosa, C. Seatzu, and F. Bullo, "A gossip algorithm for heteroeneous multi-vehicle routing problems," in *4th IFAC Conf. on Analysis and Design of Hybrid Systems*, Eindhoven, Netherlands, Jun. 2012. 12

[32] D. Rosa, M. Franceschelli, C. Seatzu, and F. Bullo, "A gossip based heuristic algorithm for heteroeneous multi-vehicle routing problems," in *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Santa Barbara, California, USA, Sep. 2012. 12

[33] D. Applegate, W. Cook, S. Dash, and A. Rohe, *Solution of a min-max vehicle routing problem.* Forschungsinstitut für Diskrete Mathematik, Rheinische Friedrich-Wilhelms-Universität, 2001. 12

[34] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009. 12

[35] E. Arkin, R. Hassin, and A. Levin, "Approximations for minimum and min-max vehicle routing problems," *Journal of Algorithms*, vol. 59, no. 1, pp. 1–18, 2006. 12

[36] Z. Lin, "Coupled dynamic systems: From structure towards stability and stabilizability," Ph.D. dissertation, University of Toronto, 2006. 21

[37] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 15, pp. 733–736, 1963. 23

[38] J. Silvester, "Determinants of block matrices," *The Mathematical Gazette*, vol. 84, no. 501, pp. 460–467, 2000. 26

[39] A. Sarlette, S. Tuna, V. Blondel, and R. Sepulchre, "Global synchronization on the circle," in *Proceedings of the 17th IFAC World Congress*, 2008. 27, 28

[40] L. Few, "The shortest path and the shortest road through n points," *Mathematika*, vol. 2, no. 2, pp. 141–144, 1955. 44, 45, 51

[41] H. J. Karloff, "How long can a Euclidean traveling salesman tour be?" *SIAM Journal on Discrete Mathematics*, vol. 2, no. 1, pp. 91–99, 1989. 44, 45, 51

[42] F. Bullo, R. Carli, and P. Frasca, "Gossip coverage control for robotic networks: Dynamical systems on the space of partitions," *SIAM Journal on Control and Optimization*, vol. 50, no. 1, pp. 419–447, 2012. 47

[43] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008. 49

[44] C. Nilsson, "Heuristics for the traveling salesman problem," *Department of Computer Science, Linkoping University*, 2003. 55

[45] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem." DTIC Document, Tech. Rep., 1976. 55

[46] B. Ryan and N. Gross, *Acceptance and diffusion of hybrid corn seed in two Iowa communities*. Agricultural Experiment Station, Iowa State College of Agriculture and Mechanic Arts, 1950, vol. 372. 71

[47] R. Burt, "Innovation as a structural interest: Rethinking the impact of network position on innovation adoption," *Social Networks*, vol. 2, no. 4, pp. 327–355, 1980. 71

[48] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing letters*, vol. 12, no. 3, pp. 211–223, 2001. 71

[49] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM review*, vol. 42, no. 4, pp. 599–653, 2000. 71

[50] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Physical review letters*, vol. 86, no. 14, p. 3200, 2001. 71

[51] M. Granovetter, "Threshold models of collective behavior," *American journal of sociology*, pp. 1420–1443, 1978. 72

[52] T. Schelling, "C.(1978). micromotives and macrobehavior." 72

[53] J. Wortman, "Viral marketing and the diffusion of trends on social networks," 2008. 72

[54] T. Schelling, "Micromotives and macrobehavior." 1978. 72

[55] G. Ellison, "Learning, local interaction, and coordination," *Econometrica: Journal of the Econometric Society*, pp. 1047–1071, 1993. 72

[56] L. Blume *et al.*, "The statistical mechanics of strategic interaction," *Games and economic behavior*, vol. 5, no. 3, pp. 387–424, 1993. 72

[57] S. Morris, "Contagion," *The Review of Economic Studies*, vol. 67, no. 1, p. 57, 2000. 72, 73

[58] D. Acemoglu, A. Ozdaglar, and E. Yildiz, "Diffusion of innovations in social networks," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 2329–2334. 72, 73, 74, 80

[59] E. Mossel and S. Roch, "On the submodularity of influence in social networks," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 2007, pp. 128–134. 72

[60] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 137–146. 72, 74

[61] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009, pp. 807–816. 73

[62] D. Kempe, J. Kleinberg, and E. Tardos, "Influential nodes in a diffusion model for social networks," pp. 1127–1138, 2005. 73

[63] C. L. Chang and Y. D. Lyuu, "Spreading messages," *Theoretical Computer Science*, vol. 410, no. 27, pp. 2714–2724, 2009. 73

[64] M. Fazli, M. Ghodsi, J. Habibi, P. J. Khalilabadi, V. Mirrokni, and S. S. Sadeghabad, "On the non-progressive spread of influence through social networks," in *LATIN 2012: Theoretical Informatics.* Springer, 2012, pp. 315–326. 73

[65] S. Borgatti, M. Everett, and P. Shirey, "Ls sets, lambda sets and other cohesive subsets," *Social Networks*, vol. 12, no. 4, pp. 337–357, 1990. 73

[66] S. Wasserman and K. Faust, *Social network analysis: Methods and applications.* Cambridge university press, 1994, vol. 8. 73

[67] B. Balasundaram, S. Butenko, and I. V. Hicks, "Clique relaxations in social network analysis: The maximum k-plex problem," *Operations Research*, vol. 59, no. 1, pp. 133–142, 2011. 73

[68] D. Rosa and A. Giua, "On the spread of innovations in social networks," in *Necsys 2013, 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems, 2013.* 73

[69] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2001, pp. 57–66. 74

[70] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2002, pp. 61–70. 74

[71] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 88–97. 74

[72] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 1029–1038. 74

[73] D. Rosa and A. Giua, "A non progressive model for the diffusion of innovation in social networks," in *CDC 2013, Conference on Decision and Control, 2013*. 74

[74] D. Acemoglu, A. Ozdaglar, and E. Yildiz, "Diffusion of innovations in social networks," in *50th IEEE Conference on Decision and Control, 2011*. 74, 75, 99, 102, 103