13

ARTICLE/PHOTOCOPY

University of North Carolina at Chapel Hill
Interlibrary Lending - (NOC)

ILLiad TN: 1081765

Lending String: *NOC,KUK,VA@,MFM,IXA

Patron: martinez, linda

Journal Title: International journal of robotics & automation.

Volume: 26 **Issue:** 1
Month/Year: 2011
Pages: 13-25

Article Author:

Article Title: bezzo; swingler; ferrari; a disjunctive programming approach for motion planning of mobile router networks

Imprint: Anaheim, Calif. ; Calgary, Alta. ; ACTA

ILL Number: 89564339

PROBLEMS: Contact ILL office
Email: nocill@email.unc.edu
Fax: 919-962-4451
Phone: 919-962-0077

Call #: v.26 (2011)

Location: Science Library Annex Serials AVAILABLE

In Process Date: 20120405

Maxcost: $50.00IFM
Billing: Exempt

Copyright: CCG

Odyssey

Borrower: NDD

Shipping Address:
PERKINS LIBRARY- DOC. DEL.
DUKE UNIVERSITY
BOX 90183
035 RESEARCH DRIVE
DURHAM,NC 27708-0183
illrequests@duke.edu
Fax: 919-660-5964
Ariel: 152.3.116.118
Odyssey :152.3.237.225

NOTICE: This material may be protected by Copyright Law (Title 17 U.S. Code).

International Journal of Robotics and Automation, Vol. 26, No. 1, 2011

A DISJUNCTIVE PROGRAMMING APPROACH FOR MOTION PLANNING OF MOBILE ROUTER NETWORKS

Nicola Bezzo,* Rafael Fierro,* Ashleigh Swingler,** and Silvia Ferrari**

Abstract

In this paper we develop a framework based on disjunctive programming for motion planning of robotic networks. Although the methodology presented in this paper can be applied to general motion planning problems we focus on coordinating a team of mobile routers to maintain connectivity between a fixed base station and a mobile user within a walled environment. This connectivity management problem is decomposed into three steps: (i) a feasible line-of-sight path between the base station and the mobile user is computed; (ii) the number of required routers and their goal locations are determined; and (iii) the motion planning with obstacle and inter-vehicle collision avoidance problem is solved. To illustrate the flexibility of the proposed approach we also formulate a novel motion planning algorithm for a team of mobile robots as a disjunctive program. Cell decomposition is used to take into account the size and orientation of the robots. In both cases, connectivity and motion planning, the mixed-integer optimization problems are solved using CPLEX. Moreover, the proposed approach can easily accommodate input and other constraints and mission objectives. Simulation results show the applicability of the proposed strategy.

Key Words

Motion planning, communications, mobile routers, disjunctive programming

1. Introduction

Multi-robot and sensor networks are increasingly being considered as a means of performing complex functions within dynamic environments, including such applications as homeland security, search and rescue operations, disaster relief operations, multi-targeting/multi-platform battlefield groups, intelligent highway/vehicle systems, and wireless surveillance networks. A fundamental challenge of multi-robot coordination is how to deploy a group of robots to carry out sensing (i.e., mobile sensors) and communication (i.e., mobile routers) tasks. Although significant progress has been made in the last decade, many issues still need to be addressed to make robotic networks commonplace. For instance, how to deal with realistic communications and sensing limitations of the mobile robots is still not well understood.

A mobile network that is deployed in a cluttered environment can experience uncertainty in communication, navigation and sensing. The objects in the environment (such as buildings) will attenuate, reflect, and refract the transmitted waves, thus degrading the performance of wireless communication. On one hand, mobile robots have to plan their paths such that collision with obstacles and other robots are avoided while minimizing an appropriate cost function. On the other hand, mobile robots should maintain certain connectivity constraints such that the coordination task can be accomplished. If robots are to provide a wireless communication infrastructure, then the motion planning problem needs to incorporate wireless communication constraints [1, 2]. This will create a multi-objective optimization problem in which optimum motion planning decisions considering only navigation may not be the best for communication. How to address this fundamental problem in a systematic way is the objective of this paper.

To be more specific, the goal of this paper is to provide a general framework based on disjunctive programming (DP) [3, 4] for the motion planning of cooperative robotic networks. We illustrate the applicability of the proposed methodology via two motion planning problems. First, we consider a mobile router network where the goal of the network is to maintain connectivity between a mobile user and a fixed base station while avoiding obstacles and inter-agent collisions. We assume that the trajectory of the user can be estimated and that the base station, the user, and the robots have limited connectivity coverage. Therefore, to establish a communication bridge between the user and stationary base station we need to form a chain of robots. The number of robotic routers necessary every time may vary depending on the position of the user. Using Mixed-Integer Linear Programming (MILP), it is possible to make

* MARHES Laboratory, School of Electrical & Computer Engineering, University of New Mexico, Albuquerque, NM 87131-0001, USA; e-mail: {nicbezzo, rfierro}@ece.unm.edu
** Laboratory for Intelligent Systems and Controls, Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC 27708-0300, USA; e-mail: ashleigh.swingler@gmail.com, sferrari@duke.edu
Recommended by Prof. Z. Qu
(10.2316/Journal.206.2011.1.206-3405)

the robots navigate safely in a walled environment. In the second motion planning problem, cell decomposition is used to take into account the size and orientation of the robots navigating within an environment populated by generally nonconvex obstacles. In both problems the mixed-integer programs are solved using CPLEX [5].

1.1 Related Work

Multi-robot coordination problems have witnessed intensive research activity in recent years. Most of the research relating sensor motion planning has focused on the effects that the uncertainty in the geometric models of the environment has on the motion strategies of the robot [6]. Hence, considerable progress has been made on planning strategies based on only partial or nondeterministic knowledge of the workspace [7].

Obstacle-avoidance motion planners have been effectively investigated in [8] to plan the path of mobile sensors for the detection and classification of stationary targets in an obstacle-populated environment. Coverage control for mobile sensors has been treated in [9, 10] using Voronoi diagrams to achieve uniform sensing performance over an area-of-interest. Moreover, in [11] we derive a decentralized coordination algorithm that allows a team of mobile sensors to navigate a region containing nonconvex obstacles and take measurements in areas with the highest probability of having *good* information first.

Recently, target tracking problems have been the focus of attention of the robotic network community. In our previous work [12], we developed an approach based on geometric optimization to deploy a mobile sensor network for the purpose of detecting and capturing mobile targets that are sensed intermittently. In this previous work, communication constraints were not considered. Most recently, the authors in [13] formulated simultaneously the problems of target coverage and network connectivity as linear-matrix-inequalities (LMIs).

It is well-known that communication plays a key role in the overall performance of cooperative mobile networks. Communication between mobile agents can be degraded due to distance-dependent path-loss, shadowing or fading. Most work on motion planning and control, however, does not consider the communication uncertainties introduced in realistic environments. For instance, it is common to assume either ideal links or links that are perfect within a certain radius of the node. The effects of noise, quantization, packet loss, and fading on wireless control of mobile sensors networks are studied in [14, 15]. In [14] the authors introduce communication-aware motion planning using an information-gain strategy. In this approach, each node can predict the information gained through its communications by online learning of link quality measures such as received signal-to-noise ratio (SNR).

Connectivity problems and their variants have been investigated by many researchers. A distributed connectivity control based on hybrid systems is proposed in [16]. Closely related to the connectivity problem considered in this paper is the work presented in [17–19]. In [17] unmanned aerial vehicles (UAVs) are used as communication relays for surveillance missions. It is assumed that a surveying UAV needs free line-of-sight (LOS) to a base station. A dual ascent algorithm and a modification of the Bellman–Ford algorithm are developed to generate communication relays. In [18] a team of robots are supposed to maintain a communication link between an exploring robot and a base robot. In this approach, the quality of connectivity is measured using the second eigenvalue of the weighted Laplacian representing the network topology while maintaining k-connectivity of the network. Finally, the authors in [19] develop motion planning algorithms for robotic routers to maintain connectivity between a base station and both a cooperative and non-cooperative user.

The remainder of the paper is organized as follows. Definitions and mathematical preliminaries are given in Section 2. A mobile router problem is considered in Section 3, where the emphasis is given to connectivity constraints and their representation with disjunctive programming techniques. Section 4 formulates a motion planning algorithm for a team of mobile robots as a disjunctive program, where cell decomposition is used to take into account the size and orientation of the robots and nonconvex obstacles in the environment. Conclusions are finally drawn in Section 5.

2. Preliminaries

2.1 Disjunctive Programming

Disjunctive Programming (DP) has been introduced in [20] and lately extended in [21] where the authors use this tool as an alternative to mixed-integer programming (MIP) to solve large optimization problems. This technique has the powerful advantage of representing constraints as a conjunction $\bigwedge =$ AND of n clauses with each clause being a disjunction $\bigvee =$ OR of m_i inequalities [3]. A disjunctive program becomes

$$\begin{aligned} &\min f(x) \\ &\text{subject to} \bigwedge_{i=1,\ldots,n} \left(\bigvee_{j=1,\ldots,m_i} C_{ij}(x) \leq 0 \right), \end{aligned} \tag{1}$$

where x is a vector of decision variables and $f(x)$ is the cost function to be minimized. DP has been introduced as an alternative to mixed-integer programming for representing discrete/continuous optimization problems. In this paper, we combine the disjunctive model for the constraints with the use of MIP to have a more robust system, which is easier to understand and solve. In fact while the MIP model is based on algebraic equations and inequalities, the DP model allows a combination of algebraic and logical constraints facilitating the representation of discrete decisions. This technique will prove useful when including communication constraints and taking into account the geometry and rotation of both the robots and obstacles (not necessarily convex) in the environment.

2.2 Mixed-Integer Linear Programming

In this section we present the well-known theory of mixed-integer programming focusing in particular on the linear case (MILP). This section is based on previous work [22] and summarizes the main aspect of this technique.

Many of the practical multi-vehicle path planning problems can be modelled with continuous and discrete variables and using linear or quadratic constraints [23]. Discrete optimization shows its power and flexibility when using decision variables and indicator variables. Decision variables are discrete variables that can take only the value 1 or 0, and are usually related to the logical TRUE and FALSE. Indicator variables, like the decision variables, can take only the value 1 or 0, and they are usually used to indicate the state of certain continuous variables. These variables are especially useful when dealing with relation indicators such as $\leq, \geq, \neq$, *etc.*

MILP is a tool used to incorporate logical constraints in the problem formulation. These constraints that use logical operators such as AND, OR, and NOT can be combined to model complex logical statements. For this purpose the well-known **big M** technique is utilized. This technique allows to connect indicator variables to continuous variables. The basic building block for this technique is the translation of the *implication*

$$x > 0 \quad \rightarrow \quad \delta = 1, \tag{2}$$

that means $x > 0$ implies $\delta = 1$. The implication in (2) can be converted into the following mixed-integer linear constraint

$$x - M\delta \leq 0, \tag{3}$$

where M is a large positive number.

Unfortunately the MILP approach is a *NP-hard* problem meaning the solution time grows exponentially with the size of the problem.

2.3 Path Loss Model for Communication

Mobile radio propagation is an important field that has attracted researchers all around the world for several years, although only recently has an enormous attention been given to wireless communication in robotics. Communication can be seen as another sensing behaviour of mobile robots; therefore we need to build a model that takes into account this important characteristic. In this section a discussion about wireless communication is presented followed by a simple but realistic model for propagation.

The transmission between transmitter (Tx) and receiver (Rx) is dependent on the environment we are analyzing and can be a simple line-of-sight path or obstructed by obstacles such as people, furniture, walls, buildings, and mountains. Also, not only noise, interference, scattering and other channel impediments affect the quality of the transmitted signal but these impediments change in time due to the movements of people and dynamics of the environment. Therefore due to this random nature, radio channels are very difficult to analyze and modelling relies on statistical procedures based on specific measurements. The signal propagated from a transmitter to a receiver can be decomposed into three separate and well-known characteristics: *path loss*, *shadowing*, and *multipath* [24]. Path loss is caused by dissipation of the radiated power from the transmitter and by effects of the propagation channel. Shadowing is caused by obstacles between the transmitter and the receiver that cause reflection, scattering, absorption and attenuation of the signal propagated. Finally, the constructive and destructive addition of multipath components creates rapid fluctuations of the received signal strength over short periods of time. In this paper, we focus mainly on the path loss model since it is the most predominant characteristic in connectivity problems.

In free space, power received by a receiving antenna situated at a distance d from a transmitting antenna is given by the Friis formula [24]

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2}, \tag{4}$$

where P_t is the transmitted power, $P_r(d)$ is the received power which is a function of the separation Tx–Rx, G_t is the transmitter gain, G_r is the receiver gain, and λ is the wavelength.

A more complicated model is the two-ray model in which the received signal consists of two components: the line-of-sight component that is the transmitted signal through free space and the component that is reflected off the ground. The received signal power is approximately given by

$$P_r = P_t G_t G_r \left[\frac{h_t h_r}{d^2}\right]^2, \tag{5}$$

where h_t and h_r are the height of the transmitting antenna and of the receiving antenna, respectively. It is interesting to notice that for large distances ($d \gg \sqrt{h_t h_r}$) the received power falls off inversely with the fourth of the power of d or in dB at a rate of $-40\,\mathrm{dB/decade}$ while for shorter distances we have a behaviour close to the Friis model with power falling off at $-20\,\mathrm{dB/decade}$ [24]. The rapid falloff with distance is due to the fact that the signal components only combine destructively and, therefore, are out of phase of approximately π.

Several other models are available in the literature, most of them are empirical and come from specific experiments for intended communication systems or allocations such as the Okumura model, the Hata model and many more. For a detailed discussion, the reader is referred to [24].

In general a simplified path loss model can be built to capture the essence of signal propagation. The following equation

$$P_r = P_t K \left[\frac{d_0}{d}\right]^\gamma \tag{6}$$

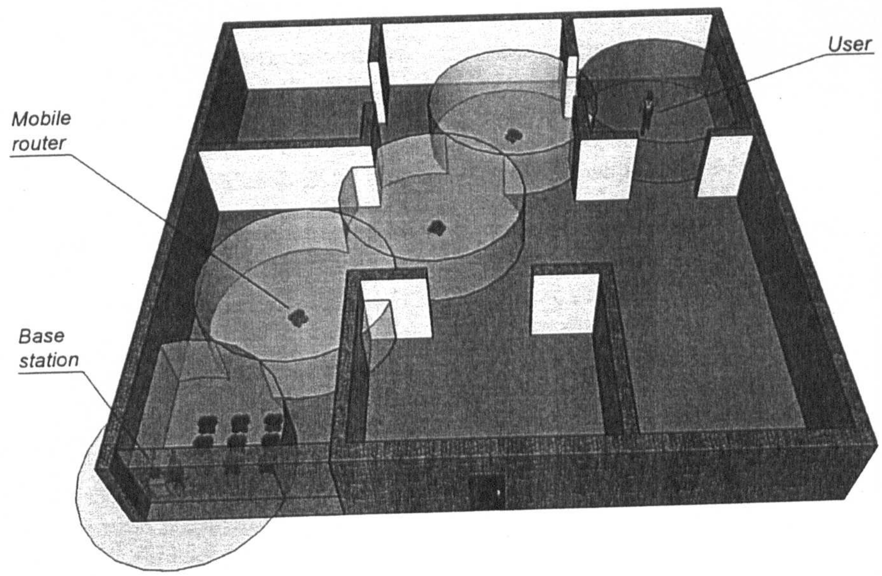


Figure 1. A virtual environment with mobile robot routers.

represents a generalized approximation of a real channel. Here K is a constant that depends on the antenna characteristics and channel attenuation, d_0 is a reference distance, and γ is the path loss exponent. Typical values of γ are in a range 1.6–3.5 in an office building and 3.7–6.5 in an urban environment [24]. It is important to note in (6) that the generalized path loss model is a function of the distance Tx–Rx. This last consideration is used later in Section 3.2.2 when building the communication constraints for the mobile router connectivity problem under investigation.

3. Coordination of Mobile Routers *via* Disjunctive Programming

The problem considered here is the control of a team of robots trying to maintain connectivity between a base station b and a user $\mathcal{T}$ moving in a known environment, as shown in Figs. 1 and 2. The environment is populated by walled obstacles that deteriorate the quality of the communication. Therefore, the mobile routers have to move in positions that guarantee a connectivity link between the base station and the user while avoiding obstacles and collisions. This section describes the model used for the routers and obstacles, and the task the routers need to accomplish.

3.1 Problem Formulation

There is an increasing interest in deploying autonomous robotic agents to create a reconfigurable communication infrastructure. The basic idea is to exploit the mobility of the robotic routers to maintain a communication link with a user $\mathcal{T}$. This problem is motivated by the DARPA LANdroids program [25]. The goal is to develop software tools that enable the following required capabilities:

- Self-Configuration.
- Self-Optimization.
- Self-Healing.
- *Tethering*: As users move through a LANdroid covered region, the network itself should adapt and stretch to keep them covered with communications whenever possible. When it is not possible to keep the user covered, the network should recognize this and advise the user to drop another router to extend the range [25].
- Intelligent Power Management.

In this paper we address in some degree all the above capabilities but focus on *tethering*. Specifically, the mobile router problem is formulated as a target assignment problem in which a user and a fixed base station are targets that need to be covered by mobile routers most of the time while maintaining connectivity of the network and avoiding collisions. Fig. 1 depicts a typical scenario where a mobile router network made of Pioneer-3AT robots maintains a connection between a human user and a base station (lower left corner in Fig. 1).

The objective of every robotic router in the network is to maintain connectivity between a fixed base station, located at (x_b, y_b), and a target moving along a known trajectory $(x_\mathcal{T}, g(x_\mathcal{T}, t)) \in \mathcal{W}$, with known initial conditions $(x_\mathcal{T}(0), y_\mathcal{T}(0))$, by navigating a two-dimensional Euclidian workspace denoted by $\mathcal{W} \subset \mathbb{R}^2$. The ith-router is denoted by $\mathcal{A}_i$ and assumed to be a point mass. The router's configuration $\mathbf{q}_i$ specifies the position of a moving Cartesian frame $\mathcal{F}_{\mathcal{A}_i}$, embedded in $\mathcal{A}_i$, with respect to a fixed Cartesian frame $\mathcal{F}_\mathcal{W}$.

3.1.1 Robot's Model

The dynamics of the ith robotic router can be approximated using the following model

$$\begin{aligned} \dot{\mathbf{q}}_i &= \mathbf{v}_i, \quad i \in I_\mathcal{P} = [1, \ldots, N] \\ \dot{\mathbf{v}}_i &= \mathbf{u}_i, \end{aligned} \tag{7}$$

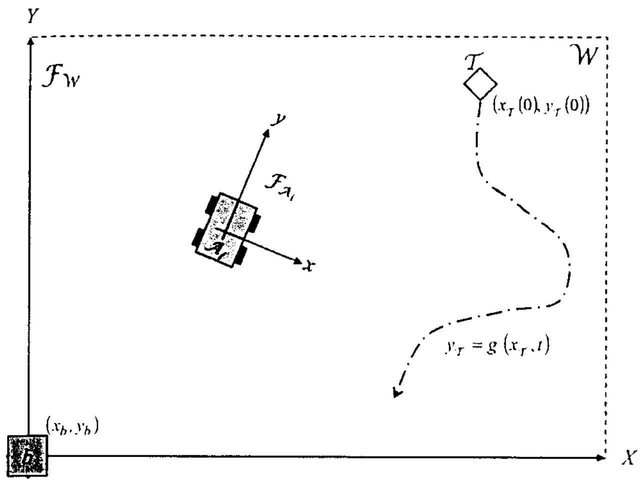


Figure 2. Base station, mobile router and user position within the area of interest.

where $\mathbf{q}_i = [x_i \quad y_i]^T \in \mathbb{R}^2$ is the position vector of $\mathcal{F}_{\mathcal{A}_i}$ relative to $\mathcal{F}_{\mathcal{W}}$, $\mathbf{v}_i \in \mathbb{R}^2$, and $\mathbf{u}_i \in \mathbb{R}^2$ denote the velocity and acceleration (control input), respectively, for each router $i \in I_{\mathcal{P}}$. The router's workspace, $\mathcal{W}$, is populated with N_o fixed polygonal obstacles $\{\mathcal{O}_1, \ldots, \mathcal{O}_{N_o}\}$, whose geometries and positions are assumed known *a priori*.

Since the disjunctive programs are solved in discrete-time, the model in (7) is discretized with sampling time ΔT. The discretized model is given by

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k), \tag{8}$$

where

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \Delta T & 0 \\ 0 & \Delta T \end{pmatrix}, \tag{9}$$

and $\mathbf{x} = [\mathbf{q}^T \quad \mathbf{v}^T]^T$, thus

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix}, \quad \mathbf{u}(k) = \begin{bmatrix} u_x(k) \\ u_y(k) \end{bmatrix}. \tag{10}$$

3.1.2 Environment

The robots move in an environment cluttered with obstacles. For the scenario we are investigating in this article these obstacles are walls that can be described by sets of linear inequalities of the form [22, 26]

$$\mathcal{O}X \leq r, \quad X = \begin{pmatrix} x \\ y \end{pmatrix}, \tag{11}$$

where $\mathcal{O}$ is an $R \times 2$ matrix, and R is the number of linear constraints needed to define an obstacle. With this representation it is possible to describe a wide range of different situations like the modelling of linear nonconvex obstacles by composing convex sets. All the obstacles must be enlarged to consider the actual dimensions of the robots and because we are considering a discretized model. In fact obstacle avoidance will be guaranteed only at the sampling time, so it would be possible to have a trajectory that is collision free at the sampling time but that would end up in a collision during the time between the samplings. This situation can occur when the robot is close to the corner of the obstacle or when the obstacle is relatively small and at the sampling time the robot is positioned right in one of the edge of the obstacle. The enlargement depends on the maximum distance D_{pp} that a robot can travel between samplings. D_{pp} is function of the maximum velocity of the robots, $V_{\max}$, and of the sampling time, ΔT, in particular $D_{pp} = V_{\max}\Delta T$. Once the obstacle is enlarged, the following implication needs to be true

$$\begin{aligned} &\forall x_1, x_2 \in \partial\mathcal{O}_e \quad \text{such that } \|x_1 - x_2\| \leq D_{pp}, \\ &x = \lambda x_1 + (1-\lambda)x_2, \quad \lambda \in [0,1], \quad x \notin \mathcal{O} \end{aligned} \tag{12}$$

where $\mathcal{O}_e$ is the convex set describing the enlarged obstacle, $\partial\mathcal{O}_e$ is its boundary, while $\mathcal{O}$ is the convex set describing the original obstacle [26].

3.2 Constraint Modelling using MILP and DP

In this section it is shown how using mixed-integer linear programming makes it possible to model some common constraints in coordination control like, for example, obstacle or collision avoidance. These constraints are used in the optimization problem. Also, the communication constraints are presented using a disjunctive programming technique.

3.2.1 Motion Planning with Obstacle and Collision Avoidance

The obstacle avoidance constraint makes the optimization problem nonconvex, since the feasible space of the solutions is nonconvex. Consequently, traditional convex optimization techniques cannot be used to solve the optimization problem. Using MILP instead, this nonconvex problem can be modelled and solved. As stated before, all the obstacles in the environment will be described by (11). For each row of the matrix $\mathcal{O}$, it is possible to define the following implication

$$\begin{aligned} o_{p1}x_j(k) + o_{p2}y_j(k) \leq r_p \Rightarrow \omega_{pj}^k = 1, \\ \forall p = 1, \ldots, R;\ \forall j = 1, \ldots, N_R;\ \forall k = 1, \ldots, T \end{aligned} \tag{13}$$

where N_R denotes the number of routers used to maintain a communication chain and T is the control/time horizon. These implications will drive the auxiliary variable ω_{pj}^k to one if the pth inequality defining the obstacle is satisfied by the jth robot at the kth sampling time. All the points inside the obstacle satisfy all the R inequalities in (11). To ensure obstacle avoidance at least one of the R inequalities defining the obstacle needs to be violated and so it is necessary to add the following constraint

$$\sum_{p=1}^{R} \omega_{pj}^{k} \leq R-1, \qquad \forall j = 1,\ldots,N_R;\ \forall k = 1,\ldots,T. \tag{14}$$

Collision avoidance among teammates can be seen as a special case of obstacle avoidance. We can assume that every robot has a safety zone around it that nobody can enter. In this way, the other team members can be thought as moving obstacles that need to be avoided. For the sake of simplicity, it is considered a square safety zone around the robot. The equations that model collision avoidance are

$$\begin{gathered} C_A \begin{pmatrix} x_i(k) - x_j(k) \\ y_i(k) - y_j(k) \end{pmatrix} \leq \mathbf{s}_d, \\ \sum_{p=1}^{4} \zeta_{pij}(k) \leq 3, \\ \forall i = 1,\ldots,N_R; j = i+1,\ldots,N_R; k = 1,\ldots,T \end{gathered} \tag{15}$$

where the 2×4 matrix C_A and the vector $\mathbf{s}_d$ define a *safety zone* around the robot. ζ_{pij} is an auxiliary binary variable which is 1 if the pth inequality in the first equation in (15) is satisfied. Note that imposing these constraints is equivalent to require that two robots never get closer than $\mathbf{s}_d$ in at least one coordinate.

3.2.2 Short Range Coverage Connectivity Constraints

Communication models can be very complicated especially when dealing with wireless technologies. As mentioned in Section 3.2, in this paper we consider the path loss model that gives us a good approximation of the real behaviour of wireless propagation. Following (6) we see that the received signal decreases with the distance. Therefore signal strength and distance are closely related. Generally a wireless antenna has omnidirectional propagation and limited range, that is, we can think of the coverage of the wireless router as a pattern centered at the position of the antenna (Fig. 3). For simplicity we consider a pattern with a circular shape. While we stand inside the circle we have reception from the transmitting antenna but when we are out of this pattern we cannot establish communication, that is

$$S = \begin{cases} 1 & \text{if } d \leq \delta \\ 0 & \text{if } d > \delta \end{cases}, \tag{16}$$

where d is the distance between the transmitting antenna and the receiving antenna and δ is the maximum range radius of the transmitting antenna. To maintain connectivity between base station and user, the constraints are

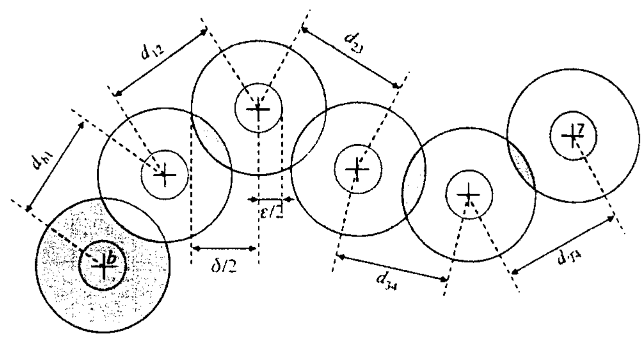


Figure 3. Geometric representation of a mobile router chain.

stated as a conjunction of disjunctions following the disjunctive programming technique introduced in Section 2. These constraints and the formulation of the problem are as follows:

$$\min_{d} \left\{ d_{lb} + d_{mT} + \left(\sum_{i=1}^{N_R} \sum_{j=1}^{N_R} d_{ij} \right) \right\} \quad \text{with } j \neq i \tag{17}$$

subject to:

$$\bigvee_{l=1,\ldots,N_R} \left\{ (\epsilon \leq d_{lb} \leq \delta) \bigwedge \left[\bigvee_{n=1,\ldots,N_R} (\epsilon \leq d_{ln} \leq \delta) \right] \right\}$$
$$\text{with } n \neq l,$$

$$\bigvee_{m=1,\ldots,N_R} \left\{ (\epsilon \leq d_{mT} \leq \delta) \bigwedge \left[\bigvee_{n=1,\ldots,N_R} (\epsilon \leq d_{mn} \leq \delta) \right] \right\}$$
$$\text{with } n \neq m,$$

$$\bigwedge_{i=1,\ldots,N_R} \left\{ \bigvee_{j=1,\ldots,N_R} (\epsilon \leq d_{ij} \leq \delta) \bigwedge \left[\bigvee_{k=1,\ldots,N_R} (\epsilon \leq d_{ik} \leq \delta) \right] \right\}$$
$$\text{with } i \neq m \neq l, j \neq i, k \neq j \neq i,$$

where $d_{ln}, d_{mn}, d_{ij}, d_{ik}$ represent the distance robot/robot, d_{lb} is the distance robot/base station and d_{mT} is the distance between the robot and the user (i.e., target). δ is the upper connectivity threshold and represents the dimension of the coverage circle around the antenna. ϵ is the lower threshold that takes into account the dimensions of the robots such that the robots do not get to close to each other or with the user and base station as depicted in Fig. 3.

With the first constraint we guarantee that one robot is connected to the base station. The second constraint in the same way guarantees that one robot is close enough to the user to allow connection. Finally, the last constraint guarantees that between the base station and the user the robots form a connectivity chain. A logical AND is interposed between the constraints because all constraints must be satisfied together. We take into account all the constraints only when $N_R \geq 3$. If $N_R = 2$ the third constraint is not necessary, if $N_R = 1$ just the first constraint is used with $n = T$ and finally when $N_R = 0$ we just need to make sure that the distance base station/user is bounded by the two thresholds ϵ and δ. Note also that in this case N_R is not the total number of robots but just the number of robots necessary to maintain connectivity. We could use all the robots each time but this solution would be sub-optimal with waste of energy whereas it is not necessary to have all the robots interconnected and possible interferences in the communication. Therefore each time the position of the user is sampled, the optimal configuration of robots necessary to maintain connectivity is determined.

To be more specific N_R is computed in the following way. A line-of-sight path is calculated between the base station and the user position by running a MIP that returns the optimal path avoiding obstacles. Knowing the threshold δ and the length of the line-of-sight ℓ, N_R is given simply by

$$N_R = \left(\frac{\ell}{\delta}\right) - 1. \tag{18}$$

From this point, following the constraint described above, it is straightforward to find the positions P ($N_P = N_R$) where the robots have to go in order to maintain connectivity. The assignment robot/position is discussed in the following section. For details about the algorithm see Section 3.3.

3.2.3 Network Formation

Network Formation is about finding the matching robot/position that minimizes the total energy spent from the whole team while maintaining connectivity. Network formation can be seen as the well-known target assignment problem. However, in this paper we prefer to refer to positions instead of targets, because the robots are moving to some spots in the environment to guarantee connection rather than trying to reach some targets. To model this problem the introduction of a set of binary decision variables is necessary. We define a variable τ_{ij} that is one if the jth robot (among the total N) is assigned to the ith position. Since there is not the same number of robots and positions, we need to assign the optimal robot to each position. The equations

$$\sum_{j=1}^{N_P} \tau_{jl} = 1, \quad l = 1, \ldots, N \tag{19}$$

$$\sum_{l=1}^{N} \left(\sum_{j=1}^{N_P} \tau_{jl} \right) = N_P \tag{20}$$

model the assignment robot/position constraint. It is now necessary to relate the decision variables to the continuous variables describing the state of the robots. More formally, the following implications need to be translated into a mixed-integer constraint:

$$\begin{aligned} \tau_{jl} &= 1 \Rightarrow x_j(T) - x_l = 0, \\ \tau_{jl} &= 1 \Rightarrow y_j(T) - y_l = 0, \\ \forall j &= 1, \ldots, N_P; l = 1, \ldots, N. \end{aligned} \tag{21}$$

The implications state that if the lth robot is assigned to the jth position ($\tau_{jl} = 1$), then the difference between the coordinates of the l-th robot at the end of the control horizon T and the coordinates of the jth position needs to be zero. This kind of implication can easily be translated into a mixed-integer constraint [27]. When the assignment robot position is terminated, N_R robots, among the total N available, are chosen and driven to the N_P positions.

3.2.4 Optimization Algorithm for Obstacle and Collision Avoidance

MIP is used here to drive the robots to the positions while avoiding obstacles and collisions. This problem is solved by defining a global cost function that includes all the constraints in a global flat optimization problem. In this way we find the global optimum but the complexity of the problem grows exponentially with the number of robots and the intricacy of the environment [26]. The global optimization problem can be formulated as follows:

$$\min \sum_{k=1}^{T} \sum_{j=1}^{N_R} [z_{x_j}(k) + z_{y_j}(k)], \tag{22}$$

where $z_{x_j|y_j}$ is a continuous variable constrained by

$$\begin{aligned} z_{x_j}(k) &\geq u_{x_j}(k), \\ z_{x_j}(k) &\geq -u_{x_j}(k). \end{aligned} \tag{23}$$

$z_{x_j|y_j}$ model the absolute value of the input $u_{x_j|y_j}(k)$.

The optimization problem is subject to:

- the dynamic equation of the system (8);
- the network formation constraints (19), (20), (21);
- the obstacles avoidance constraints (13), (14);
- the collision avoidance constraint (15);
- the bounds on the input:

$$|u_{x,y}| \leq U_{\max}; \tag{24}$$

- the bound on the maximum velocity:

$$|v_{x,y}| \leq V_{\max}; \tag{25}$$

- the equations that model the absolute value for $u_{x_j}(k)$ (23).

The above optimization problem will find the input $u_{x_j}(k)$ for all the robots and for all the time instants in the control horizon to reach the assigned positions and maintain in this way connectivity. The connectivity constraints do not enter into the MIP but are solved separately to make the robots move in position. In this way the computation complexity is reduced because the tasks are divided into two separate problems. The optimization problem is run over the whole set of continuous and binary variables and the complete set of constraints. Note that the time required to solve the optimization problem depends on the number of obstacles, robots and users, and therefore on the number of binary variables. More specifically it is easy to see that the number of binary variables is $O(T\,Nr(Nr+No))$ where T is the length of the control horizon, Nr is the number of robots and No is the number of obstacles.

The purpose of the hierarchical formulation presented in the next section is to explain the steps necessary for the algorithm to solve the problem under investigation.

3.3 DP Algorithm Description

All the algorithms described in the previous sections are parts of a higher level optimization. In other words, a more general optimization is run following Algorithm 1. This algorithm illustrates how the simulation scenario has been implemented and summarizes the chronology of the main events which occur during the simulation.

Algorithm 1 Tethering of Mobile Routers.

```
for all Target positions
  Compute the line-of-sight path ℓ from the base station
  to the target position
  Divide ℓ by the connectivity threshold and find the
  positions P in which the robots have to go for
  optimal connectivity
  forall Robots do
    forall Positions P do
      Compute the optimal path robot-position P
      avoiding obstacles and collisions
    end for
  end for
  for all Positions P do
    Select the shortest path robot-position to go
    Make the robots move to the assigned positions P
    avoiding obstacles and collision
  end for
end for
```

The first action taken during the simulation is the line-of-sight path computation. The main assumption here is that the path of the user is known from previous estimations or there are some other sensors in the network that take care of the tracking of the user in the environment. Also, since some obstacles can attenuate or block the communication depending on the material they are made from, their dimension and shape, we are considering a line-of-sight model. Therefore the first step consists of finding the line-of-sight path from the base station to the user that is to find a path to move from the base station to the user avoiding obstacles.

Following Algorithm 1, the connectivity constraints are taken into account and the thresholds discussed in Section 3.2.2 are used to determine the number of robots necessary to maintain the connectivity with the user. The positions P where the robots have to move are found following the algorithm in 3.2.2.

At this point the path planning algorithm takes place and the optimal path each robot has to compute to reach the position for connectivity is calculated. The optimal paths are determined and assigned to specific robots in such a way to minimize the total energy of the group.

Finally in the last step of the algorithm, once the assignment robot-position is ready, the MILP optimization problem (Section 3.2.4) consists of driving the robots to the positions while avoiding obstacles and collisions. Not all the robots need to be used each time, just some of them can be used. The other routers stay in the same positions ready to move if needed based on the next sampled position of the target.

These four steps are repeated in a loop until the target has reached a determined position and does not move. It is important to remark that the optimization procedure takes some time, therefore we are assuming that the user is moving slower than the robots. Faster is the computation and more positions of the user can be sampled.

3.4 Mobile Routers Simulation Results

All the algorithms and simulations have been coded in Matlab. To solve the MILP problem the well-known commercial solver CPLEX [5] has been used. CPLEX functions are called from Matlab through the TOMLAB interface [28].

We consider scenarios with obstacles shaped as to form rooms inside a building therefore nonconvex type obstacles and with a base station situated in a corner of the environment, eight robots located close to the base station and one user moving from one room to another.

Figure 4 reports successive snapshots of the user positions and the trajectories each time the robots have to complete to reach the optimal positions from the preview optimal positions. The overlapping circles in the figure represent the communication coverage for the base station, robots and user and as stated in the theory before this section, the combination base station/robots/user forms a connectivity chain with different shapes and length while the user is moving to the final position.

Important behaviours to remark are the following:

- the configuration can change from one step to another;
- there are not always the same amount of robots between user and base station;
- different robots are used during the simulation.

An example of these behaviours can be seen in Fig. 4, where the communication chain is made at first of four robots, then three in Fig. 4(d), then two in Fig. 4(f) and finally back to three in Fig. 4(g). Therefore, in this scenario some robots are never used and others are used only for certain intervals of time. Note also that the more steps the user can make, the more precise the simulation is, but as a drawback the optimization needs more time to complete the computation. Moreover, even if it is not shown specifically in this paper, collision does not take place during the simulation.

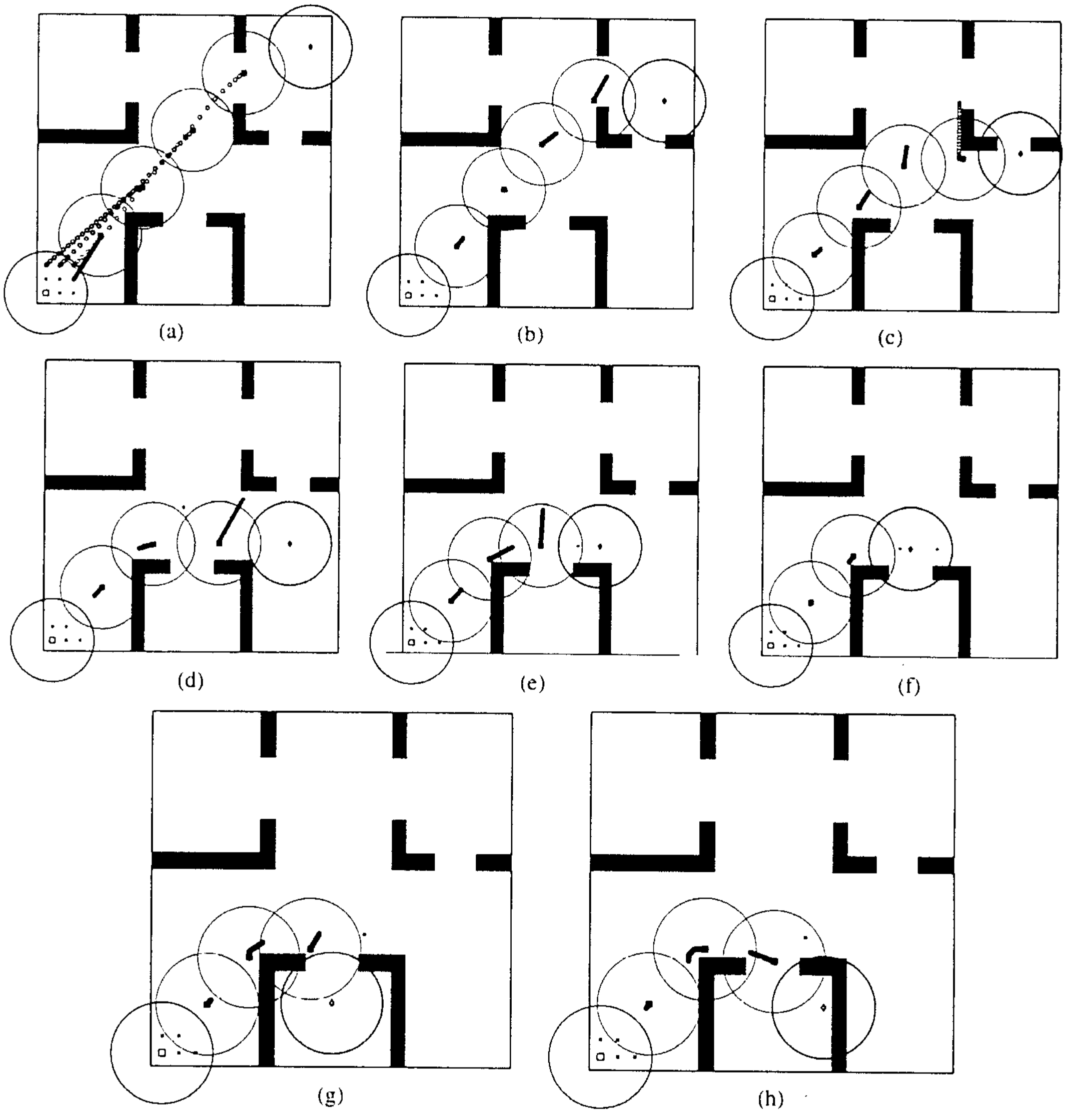


Figure 4. The sequence of steps the user follows from the starting point (top left) to the final point (bottom right).

4. Motion Planning by Cell Decomposition and Disjunctive Programming

In this section we switch our attention to a path planning problem using disjunctive programming. The method we present in this paper is novel compared to the approach described in [3] because by introducing the use of cell decomposition it allows the geometry of the robot to be taken into consideration, as well as polygonal obstacles that are not necessarily convex.

4.1 Robot Model

In this particular scenario the dynamics of the ith robot can be approximated using the nonholonomic unicycle model

$$[\dot{x}_i \quad \dot{y}_i \quad \dot{\theta}_i] = [v_i \cos\theta_i \quad v_i \sin\theta_i \quad \omega_i], \qquad (26)$$

where $\mathbf{q}_i = [x_i \quad y_i \quad \theta_i]^T \in \mathbb{R}^3$ and $\mathbf{p}_i = [x_i \quad y_i]^T \in \mathbb{R}^2$ is the position vector of $\mathcal{F}_{\mathcal{A}_i}$ relative to $\mathcal{F}_{\mathcal{W}}$.

Again, the workspace, $\mathcal{W}$, is populated with N_o fixed polygonal obstacles $\mathcal{O}_1, \ldots, \mathcal{O}_{N_o}$, whose geometries and positions are assumed known *a priori*.

4.2 Optimization Algorithm using Cell Decomposition and DP

Cell decomposition is a well-known obstacle avoidance method that decomposes the obstacle-free robot configuration space into a finite collection of non-overlapping convex polygons, known as cells, within which a robot path is easily generated. This method's advantage over other approaches, such as roadmap or potential field methods, is that, under proper assumptions, it is resolution complete. Exact cell decomposition, which has been applied to restricted classes of robot geometries, such as planar objects, three-dimensional convex polytopes and polyhedral objects [29–33], is guaranteed to find a free path in $\mathcal{W}$, whenever one exists, and otherwise to return failure. However, it is computationally intensive in high-dimensional configuration spaces (e.g., robot manipulators), and it does not typically allow for the incorporation of other motion constraints, such as, nonholonomic dynamics, or communication constraints, as required by the router problem formulated in Section 3.2.2. Also, it is not directly applicable to cooperative networks, in which the path of one robot is influenced by that of the other agents in the network. In this section, cell decomposition is combined with DP to overcome these limitations, while addressing nonconvex obstacles and rotations.

Let the configuration space $\mathcal{C}$ denotes the space of all possible robot configurations. A C-obstacle is a subset of $\mathcal{C}$ that causes collisions between the ith robot and at least one obstacle in $\mathcal{W}$, i.e., $\mathcal{CO}_j \equiv \{\mathbf{q}_i \in \mathcal{C} \mid \mathcal{A}_i(\mathbf{q}_i) \cap \mathcal{O}_j \neq \varnothing\}$, where $\mathcal{A}_i(\mathbf{q}_i)$ denotes the subset of $\mathcal{W}$ occupied by the platform geometry $\mathcal{A}_i$ when the robot is in the configuration $\mathbf{q}_i$ [32]. Then, the union $\bigcup_{j=1}^{N_o} \mathcal{CO}_j$ is the *C-obstacle region*, and the obstacle-free robot configuration space is defined as

$$\mathcal{C}_{free} \equiv \mathcal{C} \setminus \bigcup_{j=1}^{N_o} \mathcal{CO}_j = \left\{ \mathbf{q}_i \in \mathcal{C} | \mathcal{A}(\mathbf{q}_i) \cap \left(\bigcup_{j=1}^{N_o} \mathcal{O}_j \right) = \varnothing \right\}. \tag{27}$$

In classical cell decomposition, the union of the cells composing $\mathcal{C}_{free}$ is used to construct a connectivity graph representing the adjacency relationships between the cells. This graph is then searched for the shortest path between the cells containing the initial and final robot configurations.

The methodology presented in this paper relies on the computation of the C-obstacles corresponding to each obstacle in $\mathcal{W}$. This computation can be performed as explained in [32], by sliding the robot geometry $\mathcal{A}_i$ along the sides of each obstacle. When the robot is capable of rotating, the C-obstacles become three-dimensional polyhedra. To use DP all concave C-obstacle are decomposed into a finite collection of non-overlapping convex polygons (cells), resulting in a set of L convex polygons $\mathcal{K} = \{\kappa_1, \ldots, \kappa_L\}$. A line-sweeping algorithm can be utilized to compute $\mathcal{K}$, as illustrated in the sample workspace in Fig. 5.

Unlike classical cell decomposition, which decomposes $\mathcal{C}_{free}$, this approach decomposes the C-obstacle region to determine a consistent set of disjunctive inequalities. To avoid all obstacles in $\mathcal{W}$, the ith robot configuration must satisfy the following inequalities

$$\bigwedge_{\ell=1,\ldots,L} \bigvee_{l=1,\ldots,s} \mathbf{a}_{l\ell}^T \mathbf{q}_i(t) > b_{l\ell}, \tag{28}$$

where s is the number of sides of the ℓth cell, and $\mathbf{a}_{l\ell}^T$ and $b_{l\ell}$ are known constants defining the cell boundaries as described in [3]. This set of inequalities states that $\mathbf{q}_i(t)$ must lie outside of all cells in $\mathcal{K}$ at time t.

For a network of N robots, inter-collision avoidance is addressed by a set of inequalities that maintain a safety distance between robot configurations at all times. Let ρ_x and ρ_y denote the minimum safety distances in the x and y directions, respectively, which guarantee that any two robot geometries $\mathcal{A}_i$ and $\mathcal{A}_j$ will not collide. Then, for all $i = 1, \ldots, N$, and $j > i$, the clause

$$[x_i(t) - x_j(t)] > \rho_x \vee [x_j(t) - x_i(t)] > \rho_x \vee [y_i(t) - y_j(t)] > \rho_y \vee [y_j(t) - y_i(t)] > \rho_y \tag{29}$$

avoids collisions between any two robots i and j, at time t. The minimum-distance robots' trajectories between the set of initial and final configurations $\mathbf{q}_1(0), \ldots, \mathbf{q}_N(0)$ and $\mathbf{q}_1(t_f), \ldots, \mathbf{q}_N(t_f)$ is determined by minimizing the L_1-norm of the distance travelled by each robot defined as

$$J = \frac{1}{t_f} \lim_{t_f \to \infty} \int_0^{t_f} \sum_{i=1}^{N} \mathbf{r}^T |\mathbf{q}_i(t)| dt, \tag{30}$$

where $\mathbf{r}$ is a weighting vector that represents the desired tradeoff between translations and rotations. Using the L_1-norm in place of the traditional L_2-norm or Euclidian distance gives rise to a MILP in place of a mixed-integer quadratic program (MIQP), thus the solution can be easily obtained by available CPLEX and Matlab algorithms. The cost function in (30) can be easily modified to include the cost of control usage, or to minimize the travel time in place of distance [33].

Finally, the MILP representation of the obstacle and collision avoidance problem is obtained by discretizing the cost function and constraints with respect to time. Let $k = t/\Delta T$ represents the k-sampling time, where ΔT is the discretization interval, and the minimization of (30) is transformed into a finite-horizon problem by choosing $M = t_f/\Delta T$ as an arbitrary large integer, and by introducing a terminal cost $\phi[\mathbf{q}(M)]$. Then, the set of collision-free minimum-distance trajectories of the N robots, $P = \{\mathbf{q}_1(1), \ldots, \mathbf{q}_1(M-1), \ldots, \mathbf{q}_N(1), \ldots, \mathbf{q}_N(M-1)\}$, is obtained by solving the following MILP in P:

$$\min_P \left\{ \phi[\mathbf{q}(M)] + \sum_{i=1}^{N} \sum_{k=1}^{M} \mathbf{r}^T |\mathbf{q}_i(k)| \right\} \tag{31}$$

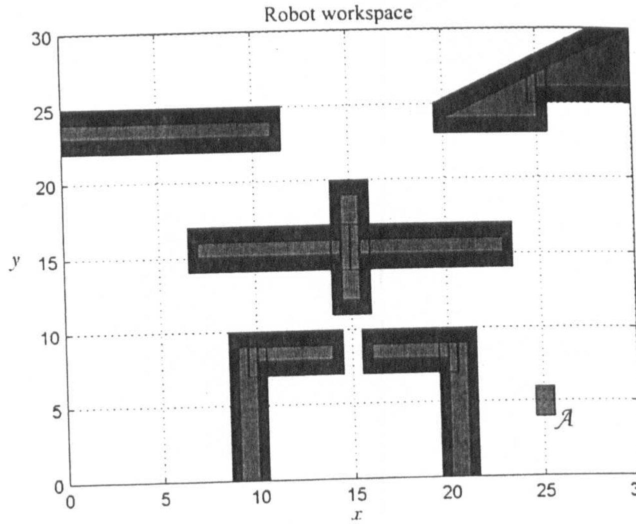


Figure 5. C-obstacle region decomposition for a robot with geometry $\mathcal{A}$.

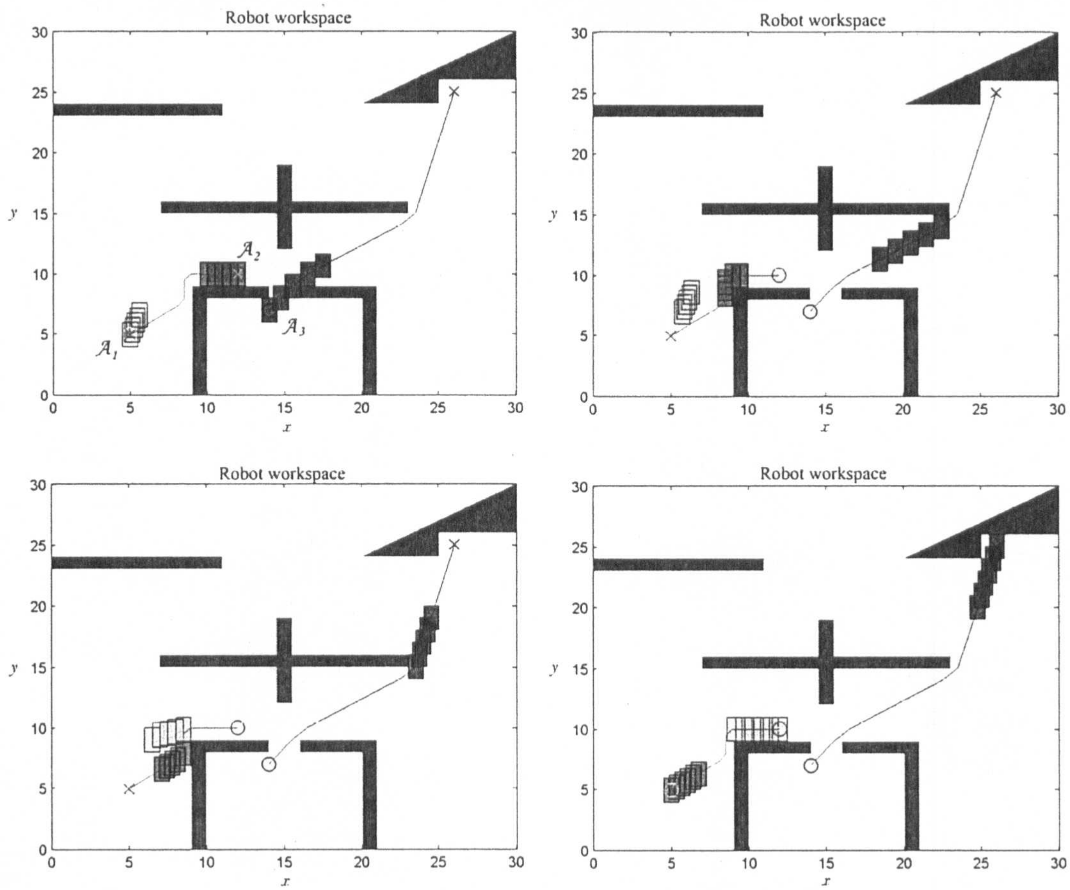


Figure 6. Sequence of steps for the minimum-distance trajectories of three robots computed *via* MILP.

subject to:

$$\mathbf{q}_i(k+1) = \mathbf{F}\mathbf{q}_i(k) + \mathbf{G}\mathbf{u}_i(k), \ \ \forall i, k,$$

$$\bigwedge_{\ell=1,\dots,L} \bigvee_{l=1,\dots,s} \mathbf{a}_{l\ell}^T \mathbf{q}_i(k) > b_{l\ell}, \ \ \forall i, k,$$

$$[x_i(k) - x_j(k)] > \rho_x \bigvee [x_j(k) - x_i(k)] > \rho_x, \ \ i, k, j > i,$$

$$[y_i(k) - y_j(k)] > \rho_y \bigvee [y_j(k) - y_i(k)] > \rho_y, \ \ \forall i, k, j > i$$

where, the Jacobian matrices $\mathbf{F}$ and $\mathbf{G}$ for the robot's dynamics in configuration space are obtained by a coordinate transformation from (8).

4.3 Motion Planning Simulation Results

The effectiveness of the MILP solution is illustrated by planning the trajectories of three rectangular robots in the obstacle-populated workspace of Fig. 5. As shown in Fig. 6, the robots' initial and final positions are purposely chosen such that the minimum-distance trajectories intersect in the workspace. However, by accounting for the robots' dynamics and positions in time, the MILP approach computes minimum-distance trajectories that avoid mutual collisions, as well as the obstacles. By this approach, non-holonomic dynamic constraints and multiple vehicles can be treated within a common trajectory planning framework that allows to simultaneously account for concave robot and obstacle geometries.

5. Conclusions

In this work we have presented a framework based on disjunctive programming for solving two different problems: a connectivity problem and a motion planning problem. In both scenarios not only have we presented a feasible solution, but have also allowed for new features to be taken into account. For the communication part we are able to maintain connectivity between a base station and a moving target by interposing a connectivity chain of robots between user and base station. This chain has the unique characteristic that it is able to change configuration and adapt depending on the situation, i.e., the number of robots necessary to form the chain changes and not always the same amount are used because not necessary, therefore giving a more optimal solution. For the motion planning part, thanks to the combined use of disjunctive programming and cell decomposition, the dimensions and the orientation of the robots are taken into account and hence a more realistic model is given allowing inter-collision avoidance in convex and nonconvex obstacles scenarios.

The main drawback of this technique is the time required for solving the optimizations. The dimensions of the environment, the number of robots, the number of obstacles together with all the constraints slow the simulations and these issues become more critical when trying to combine together the two problems solved in this paper. However future work will be centered on extending the proposed methodologies to three-dimensional scenarios with UAVs relays together with UGVs. Experimental deployment of a robotic router network using the MARHES multivehicle testbed [34] is also in our research agenda.

In conclusion, disjunctive programming in the context of robotics is a powerful tool that provides a good modelling representation of a class of real-world planning problems with optimal results, as outlined in this paper.

Acknowledgements

This work is supported by NSF grants ECCS CAREER #0811347 and IIS #0812338. The work of S. Ferrari is supported in part by the Office of Naval Research (Code 321) and by NSF grant ECS CAREER #0448906. We also thank Carlo Branca for useful discussions and providing his code for MILP target assignment.

References

[1] H. Hashemi, The indoor radio propagation channel, *Proceedings of the IEEE*, *81*(7), 1993, 943–968.

[2] D. Spanos & R. Murray, Motion planning with wireless network constraints, *Proceedings of the American Control Conference*, 2005, 87–92.

[3] L. Blackmore & B. Williams, Optimal manipulator path planning with obstacles using disjunctive programming, *Proceedings of the American Control Conference*, Minneapolis, MN, 2006, 3200-3202.

[4] S. Lee & I.E. Grossmann, New algorithms for nonlinear generalized disjunctive programming, *Computers and Chemical Engineering Journal*, *24*(9–10), 2000, 2125–2141.

[5] ILOG, CPLEX 11 Users Manual, 2009.

[6] S. Koenig, C. Tovey, & Y. Smirnov, Performance bounds for planning in unknown terrain, *Artificial Intelligence*, *147*, 2003, 253–279.

[7] J.C. Latombe, A. Lazanas, & S. Shekhar, Robot motion planning with uncertainty in control and sensing, *Artificial Intelligence*, *52*, 1991, 1–47.

[8] C. Cai & S. Ferrari, Information-driven sensor path planning by approximate cell decomposition, *IEEE Transactions on System, Man, and Cyberretics - Part B*, *39*(3), 2009.

[9] J. Cortés, S. Martínez, T. Karatas, & F. Bullo, Coverage control for mobile sensing networks, *IEEE Transactions on Robotics and Automation*, *20*(2), 2004, 243–255.

[10] M. Schwager, D. Rus, & J.-J. Slotine, Decentralized, adaptive coverage control for networked robots, *International Journal Robotic Research*, *28*(3), 2009, 357–375.

[11] A. Cortez, R. Fierro, & J. Wood, Prioritized sensor detection via dynamic voronoi-based navigation, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009, 5815–5820.

[12] S. Ferrari, R. Fierro, B. Perteet, C. Cai, & K. Baumgartner, A geometric optimization approach to detecting and intercepting dynamic targets using a mobile sensor network, *SIAM Journal on Control and Optimization*, *48*(1), 2009, 292–320, special Issue on Control and Optimization in Cooperative Networks.

[13] J. Derenick, J. Spletzer, & A. Hsieh, An optimal approach to collaborative target tracking with performance guarantees, *Journal of Intelligent and Robotic Systems*, *56*(1–2), 2009, 47–67.

[14] Y. Mostofi, Decentralized communication-aware motion planning in mobile networks: An information-gain approach, *Journal of Intelligent and Robotic Systems*, *56*(1–2), 2009, 233–256.

[15] E.W. Frew, Information-theoretic integration of sensing and communication for active robot networks, *Mobile Networks and Applications*, *14*(3), 2009, 267–280. [Online]. Available: http://dx.doi.org/10.1007/s11036-008-0103-z

[16] M.M. Zavlanos & G.J. Pappas, Distributed connectivity control of mobile networks, *IEEE Transaction on Robotics*, *24*(6), 2008, 1416–1428.

[17] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnstrom, & P.-R. Olsson, Positioning unmanned aerial vehicles as communication relays for surveillance tasks, *Proceedings of Robotics: Science and Systems*, Seattle, USA, 2009.

[18] E. Stump, A. Jadbabaie, & V. Kumar, Connectivity management in mobile robot teams, *IEEE International Conference on Robotics and Automation*, Pasadena, CA, 2008, 1525–1530.

[19] O. Tekdas, W. Yang, & V. Isler, Robotic routers: Algorithms and implementation, *International Journal of Robotics Research*, 2009, 110–126.

[20] E. Balas, Disjunctive programming and a hierarchy of relaxations for discrete optimization problems, *SIAM Journal on Algebraic and Discrete Methods*, *6*, 1985, 466.

[21] R. Raman & I. Grossmann, Modelling and computational techniques for logic based integer programming, *Computers and Chemical Engineering*, *18*(7), 1994, 563–578.

[22] C. Branca & R. Fierro, Hierarchical optimization strategies for deployment of mobile robots, *International Journal of Intelligent Control and Systems (IJICS), Special Issue on Swarm Robotics*, *11*(3), 2006, 141–153.

[23] A. Bemporad & M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, *35*(3), 1999, 407–427.

[24] A. Goldsmith, Wireless communications. (Cambridge University Press, Cambridge, UK, 2005).

[25] M. McClure, D.R. Corbett, & D.W. Gage, The DARPA LANdroids program, *Proceedings of SPIE, ser. Unmanned Systems Technology XI*, in G.R. Gerhart, D.W. Gage & C.M. Shoemaker (Eds.), vol. 7332, Orlando, FL, 2009, 0A–1–0A–10.

[26] C. Branca & R. Fierro, A Hierarchical Optimization Algorithm for Cooperative Vehicle Networks, in American Control Conference (ACC), 2006, 4225-4230.

[27] H.P. Williams, Model building in mathematical programming (New York: John Wiley & Sons, 1985).

[28] K. Holmström, A.O. Göran, & M.M. Edvall, USERS GUIDE FOR TOMLAB/CPLEX v11.2, 2009.

[29] J. Schwartz & M. Sharir, *On the 'Piano Movers' problem: II. General techniques for computing topological properties of real algebraic manifolds* (New York, NY: Academic, 1983).

[30] K. Kedem & M. Sharir, An efficient motion planning algorithm for convex polygonal object in 2-dimensional polygonal space, Courant Institute of Mathematical Science, New York, NY, Technical Report 253, 1986.

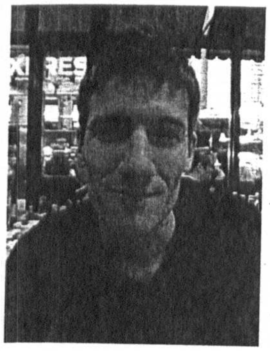
[31] F. Avnaim, J.D. Boissonnat, & B. Faverjon, A practical motion planning algorithm for polygonal objects amidst polygonal obstacles, INRIA, Sophia-Antipolis, France, Technical Report 890, 1988.

[32] J.C. Latombe, Robot motion planning (Springer Verlag, New York: Kluwer Academic Publishers, 1991).

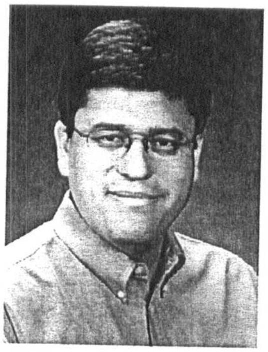
[33] T. Schouwenaars, B.D. Moor, E. Feron, & J. How, Mixed integer programming for multivehicle path planning, *Proceedings of European Control Conference, Porto, Portugal*, 2001, 2603–2608.

[34] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, & R. Fierro, Decentralized cooperative control: A multivehicle platform for research in networked embedded systems, *IEEE Control Systems Magazine*, *27*(3), 2007, 58–78.

Biographies



Nicola Bezzo received his B.S. degree in electrical engineering from Politecnico di Milano, Italy and the M.S. degree in electrical engineering with orientation automation from Oklahoma State University and Politecnico di Milano. He is currently pursuing a Ph.D. degree in control and robotics at University of New Mexico, Albuquerque working at MARHES Laboratory. His research interests include wireless communication applied to robotic systems, computer vision and autonomous aerial and ground vehicles. Nicola is also a licensed glider pilot.



Rafael Fierro received a M.Sc. degree in control engineering from the University of Bradford, England and a Ph.D. degree in electrical engineering from the University of Texas-Arlington. He is currently an associate professor in the Department of Electrical and Computer Engineering at University of New Mexico. Prior to joining UNM in August 2007, he held a postdoctoral appointment with the GRASP Lab at the University of Pennsylvania and a faculty position with the Department of Electrical and Computer Engineering at Oklahoma State University. His research interests include hybrid and embedded systems, optimization-based cooperative control and heterogeneous multivehicle coordination. He directs the Multi-Agent, Robotics, Hybrid and Embedded Systems (MARHES) Laboratory. Dr. Fierro was the recipient of a Fulbright Scholarship and a 2004 National Science Foundation CAREER Award. He is serving as Associate Editor for the IEEE Control Systems Magazine, IEEE Transactions on Automation Science and Engineering and Journal of Intelligent and Robotics Systems.

Ashleigh Swingler is a senior at Duke University completing a major in Mechanical Engineering and a certificate in Aerospace Engineering. Her main research interests include disjunctive programming and approximate cell decomposition applied to robotic path planning.

Silvia Ferrari is associate professor of mechanical engineering and materials science at Duke University, where she directs the Laboratory for Intelligent Systems and Controls (LISC). Her principal research interests include robust adaptive control of aircraft, learning and approximate dynamic programming and optimal control of mobile sensor networks. She received the B.S. degree from Embry-Riddle Aeronautical University and the M.A. and Ph.D. degrees from Princeton University. She is a senior member of the IEEE and a member of ASME, SPIE, and AIAA. She is the recipient of the ONR young investigator award (2004), the NSF CAREER award (2005) and the Presidential Early Career Award for Scientists and Engineers (PECASE) award (2006).