Shortest Paths in Graphs of Convex Sets

Tobia Marcucci, Jack Umenberger, Pablo A. Parrilo, and Russ Tedrake

***Abstract*—Given a graph, the shortest-path problem requires finding a sequence of edges with minimum cumulative length that connects a source to a target vertex. We consider a generalization of this classical problem in which the position of each vertex in the graph is a continuous decision variable, constrained to lie in a corresponding convex set. The length of an edge is then defined as a convex function of the positions of the vertices it connects.**

Problems of this form arise naturally in road networks, robot navigation, and even optimal control of hybrid dynamical systems. The price for such a wide applicability is the complexity of this problem, which is easily seen to be NP-hard. Our main contribution is a strong mixed-integer convex formulation based on perspective functions. This formulation has a very tight convex relaxation and allows to efficiently find globally-optimal paths in large graphs and in high-dimensional spaces.

I. Introduction

The Shortest-Path Problem (SPP) is one of the most deeply-studied problems in combinatorial optimization. In its single-source single-target version, this problem asks to find a path of minimum length connecting two prescribed vertices of a graph, with the length of a path being defined as the sum of the length of its edges. Typically, the edge lengths are fixed scalars given as problem data and the assumptions made on their value can have a dramatic impact on the problem complexity [1, Chapters 6–8].

In this paper we consider a generalization of the SPP in which the edge lengths do not have fixed values but are convex functions of continuous variables representing the position of the vertices (see Figure 1). More precisely, we consider a graph in which each vertex is paired with a convex set. The spatial position of a vertex is a continuous decision variable constrained to lie in the associated convex set. The length of an edge is a generic convex function (e.g. the Euclidean distance) of the position of the vertices it connects. As a consequence, when looking for a path of minimum length, we now have the extra degree of freedom of optimizing the position of the vertices visited by the path.

In the past decades many variants of the SPP with generalized path length have been studied (see Section I-A). However, to the best of our knowledge, this specific generalization has not been analyzed before, even though problems with this structure emerge naturally in many areas. Taking for example the canonical scenario in which we seek a shortest path between two cities in a road network, we can imagine the size of the cities in the map to be nonnegligible with respect to the length of the roads. Modeling this as an ordinary SPP with one vertex per city might be very conservative, whereas constructing a purely discrete problem with a precise model of the road network might be too expensive. A natural approach is then to let the transit point in a city be a continuous variable and solve the resulting mixed-integer problem.

The authors are with the Massachusetts Institute of Technology, department of Electrical Engineering and Computer Science. E-mail: {tobiam,umnbrgr,parrilo,russt}@mit.edu.

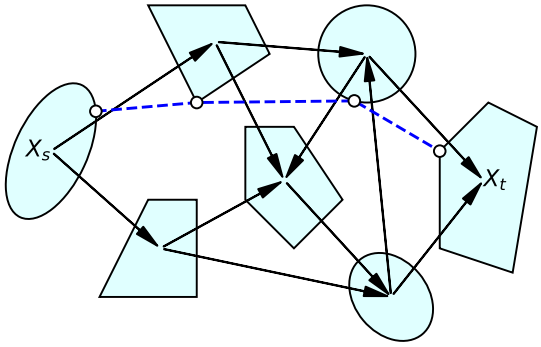


Fig. 1: Example of an SPP in a graph of convex sets. The dashed blue line is the shortest path from the source set X_s to the target set X_t. The position of each vertex along the path (white circles) is allowed to move within the corresponding convex set. Transitions are allowed only between sets connected by an edge of the graph (black arrows).

Another application is robot motion planning, in which case each vertex in our graph is paired with a convex region of space that does not intersect with obstacles (see Section VIII). Problems such as footstep planning for humanoid robots on uneven terrain or collision-free motion planning of autonomous vehicles under dynamic constraints are frequently tackled via mixed-integer programming [2], [3], [4], [5], [6], [7], [8]. However, our problem formulation is substantially different as we do not parameterize the discrete state of the system with binary variables, but instead we use binaries to select transitions between regions of the state space. This different parameterization leads to slightly larger but, in our experience, much stronger Mixed-Integer Convex Programs (MICPs). A similar abstraction allows to reformulate optimal control problems for hybrid dynamical systems [9] as SPPs (see Section VIII-C).

The proposed generalization of the SPP is easily seen to be NP-hard, therefore we do not expect to find an exact polynomial-time algorithm for its solution. In this paper we relax the requirement of polynomial-time solvability, and we present a strong mixed-integer convex formulation of the problem that can be effectively solved to global optimality. We derive this formulation using perspective functions: a tool from convex analysis which in the recent years has seen a multitude of applications in mixed-integer programming [10], [11], [12], [13] and also in optimal control [14], [15]. Numerical results show that the convex relaxation of our formulation is generally very tight, even when working in high-dimensional spaces and with large graphs. Furthermore, in our computational

experience, the proposed MICP outperforms state-of-the-art mixed-integer formulation for optimal control [15]. We also highlight that this MICP shares the same structure as the Linear-Programming (LP) formulation of the classical SPP: this feature might allow us to leverage a massive body of works to devise efficient approximation algorithms (see, e.g., [16]). Preliminary results suggest that the convex relaxation of this MICP might be well suited, e.g., for the development of rounding strategies.

A. Related generalizations of the shortest-path problem

Many generalizations of the classical SPP have been studied in the last seventy years. A taxonomy of the literature prior to 1984 can be found in [17], according to which our problem would be classified as an SPP with "generalized path length." Within this class of problems, it is worth mentioning [18] where the length of a sequence of K edges is defined recursively as a function of the final edge and the length of the first $K-1$ edges. This problem admits an efficient solution algorithm, provided that some extra assumptions (sufficient for the dynamic-programming principle to hold) are made. Knuth extended Dijkstra's algorithm [19] to the context of formal language theory in [20]. An improved algorithm for this SPP has been presented in [21], while [22] expanded the class of edge costs that these methods can handle. When applied to graphs, this more abstract setup yields a generalization of the notion of length of a path similar to the one from [18].

The closest family of problems to the one we consider here falls under the name of Euclidean SPP (see, e.g., the recent book [23]). This problem requires finding a continuous path that connects two points and does not collide with a collection of polygonal obstacles. In two dimensions, the shortest path is a polygonal line whose corners are vertices of the obstacles. By constructing a visibility graph, the problem is then reduced to a discrete graph search which is solvable in polynomial time [24], [25]. In three dimensions or more this strategy breaks; in fact, the problem becomes NP-hard [26, Theorem 2.3.2]. An approximation algorithm for the three-dimensional case has been proposed [27]. Efficient algorithms for the multidimensional case based on a grid-discretization of the space have been presented in [28], [29]. Exact-geometry algorithms for problems of this nature have been recently discussed in [30].

An evident difference between the Euclidean SPP and the problem we analyze in this paper is that the first requires the identification of a continuous path. In our case, we only require a finite set of points to lie in appropriate regions of space, without any conditions on the path connecting them. Another difference of fundamental importance concerns the notion of length we employ. While Euclidean shortest-path algorithms strongly exploit the metric structure of the underlying space, the notion of length we use here is much weaker: a convex function with extended-real values. As an example, this allows us to define the distance between two points as the energy consumed by a dynamical system to move between them, with the length being infinite in case the motion is infeasible (see Section VIII). Finally, the graph structure beneath our problem allows to prevent undesired transitions between regions of free space, increasing the flexibility of our framework.

To conclude, we mention the touring-polygon problem which, in its unconstrained version, requires finding the shortest path between two points that visits a set of polygons in a given order [31], [23, Chapter 10]. Except for special cases, e.g. convex polygonal regions, this problem is NP-hard [31, Theorem 6]. Similar in spirit are some classical problems in computational geometry: the safari [32], the zoo-keeper [33], and the watchman route [34] problems.

II. Problem Definition

We start this section with a formal definition of the generalized SPP we study in this paper. Then, in Section II-A, we use a simple reduction argument to prove the NP-hardness of this problem.

Let $G := (V, E)$ be a directed graph with vertices V and edges E. We assume each edge $(i, j) \in E$ to connect distinct vertices $i \neq j$. For each vertex $i \in V$, we have a compact convex set $X_i \subset \mathbb{R}^d$ and a point x_i contained in it. The length of an edge (i, j) is determined by the location of the points x_i and x_j via the expression $\ell_{ij}(x_i, x_j)$, where ℓ_{ij} is a nonnegative closed convex function with extended-real values that attains finite value in at least one point. Even though we will refer to ℓ_{ij} as the "edge length," we underline that this function needs not to be a valid metric, and axioms such as symmetry or the triangle inequality are not required to hold. A path P from a source vertex $s \in V$ to a target $t \in V - \{s\}$ is a sequence of distinct vertices $(i_k)_{k=0}^{K}$ such that $i_0 = s$, $i_K = t$, and $(i_k, i_{k+1}) \in E$ for all $k = 0, \dots, K-1$. We denote the set of edges traversed by the path P as $E_P := \{(i_k, i_{k+1})\}_{k=0}^{K-1}$. The length of the path P is then defined as

$$\ell_P(x_P) := \sum_{(i,j)\in E_P} \ell_{ij}(x_i, x_j),$$

where $x_P := (x_{i_0}, \dots, x_{i_K})$.

Among the paths P from s to t, we seek one of minimum length and, in doing that, we are allowed to optimize the location of the points x_i. Our problem can be compactly described as

$$\min_{P} \min_{x_P \in X_P} \ell_P(x_P), \tag{1}$$

where $X_P := X_{i_0} \times \cdots \times X_{i_K}$.

Remark 1. While we explicitly rule out self-edges of the form (i, i) for all $i \in V$, a self-transition from a region X_i to itself can be easily allowed. We define an auxiliary vertex i' associated with the set $X_{i'} = X_i$, and we add to E the edge (i, i') together with a copy (i', j) of any edge $(i, j) \in E$. Of course, this generalizes to the case in which we allow a finite number of self-transitions within a region.

Problem (1) generalizes the classical single-source single-target SPP in that the length of the edges is not know a priori, but it depends on the continuous variables x_i. The classical problem formulation is recovered as a special case when the sets X_i are singletons $\{\theta_i \in \mathbb{R}^d\}$.

For what concerns the edge length ℓ_{ij}, we can, for example, define it as the Euclidean distance between x_i and x_j:

$$\ell_{ij}(x_i, x_j) := \|x_j - x_i\|_2. \tag{2}$$

In this case the optimal location of the points x_P will define a polygonal line connecting x_s to x_t via a path as straight as possible (perfectly straight if $(s,t) \in E$). Conversely, letting the edge length be the squared 2-norm

$$\ell_{ij}(x_i, x_j) := \|x_j - x_i\|_2^2, \tag{3}$$

straight paths might be suboptimal if they require long steps $x_{i_{k+1}} - x_{i_k}$. By allowing ℓ_{ij} to take infinite values, we can enforce hard constraints that couple the position of the points x_i and x_j (see Section IV-G). This can be further generalized to model the scenario in which the points x_i describe the time evolution a dynamical system, allowing optimal control problems to be stated in the form (1) (see Section VIII).

A. Problem complexity

We have seen that problem (1) is a strict generalization of the classical SPP; more precisely, because of the assumptions $\ell_{ij} \geq 0$, the SPP with nonnegative edge lengths. This famous problem can be solved in polynomial-time using, e.g., Dijkstra's algorithm [19], or more efficient implementations of it [35]. The same cannot be expected for (1) since, as shown below, this problem is NP-hard.

Recall that an s-t path is called Hamiltonian if it visits each vertex in V, and a graph is said to be Hamiltonian if it contains such a path. The directed Hamiltonian-Path Problem (HPP) asks if a directed graph G is Hamiltonian. As an example, the graph in Figure 1 is not Hamiltonian.

Lemma 1. *The directed HPP is reducible to problem* (1) *in polynomial time.*

Proof. We construct an instance of (1) which shares the same graph G as the given HPP. We let the source $X_s := \{\theta_s \in \mathbb{R}^d\}$ and target $X_t := \{\theta_t \neq \theta_s\}$ sets be single points, while we let all the other regions X_i be large enough to contain the line segment Γ connecting θ_s and θ_t. We define the edge length as in (3).[1] At the optimum, the points x_P are arranged on the segment Γ and have equal Euclidean distance

$$\gamma := \|x_{i_{k+1}} - x_{i_k}\|_2 = \|\theta_t - \theta_s\|_2 / K,$$

for all $k = 0, \dots, K-1$. The cost associated with this arrangement is $K\gamma^2 = \|\theta_t - \theta_s\|_2^2 / K$. Since this value decreases with the number K of edges traversed by the optimal path P, whenever possible, K will attain the maximum value $|V| - 1$. We conclude that the graph G is Hamiltonian if and only if the solution P of this instance of problem (1) is Hamiltonian. Synthesizing this instance, as well as verifying if $K = |V| - 1$, takes polynomial time. □

Theorem 1. *Problem* (1) *is NP-hard.*

Proof. This follows from the directed HPP being NP-complete [36], [1, Theorem 8.11]. □

[1] Any strictly convex function of $x_j - x_i$ would be equally valid.

Remark 2. The hardness of problem (1) is not related to the dimension d of the space in which the sets X_i live. In fact, the reduction argument in Lemma 1 holds for any $d \geq 1$.

To sum up, problem (1) encompasses two classical problems in combinatorial optimization: the SPP with nonnegative edge lengths and the HPP. The first is solvable in polynomial time and is recovered as the sets X_i become zero dimensional, the second is NP-complete and associated with sets X_i of large-enough volume. We then expect the volume of these sets to play a crucial role in determining whether a good solution for problem (1) can be found quickly.

III. Mixed-Integer Convex Formulation

We take two steps to reformulate problem (1) as an MICP. First we extend the LP formulation of the classical SPP to the generalized setting described in Section II: this step involves the use of perspective functions and yields a bilinear program, i.e., a program whose nonconvexity comes exclusively from bilinear equality constraints. Successively we convert this bilinear program into an MICP. All the necessary background on the classical SPP and on perspective functions is introduced. For several special cases and examples of the upcoming reformulation we refer the reader to Section IV.

In the following, we let $O_i := \{j : (i,j) \in E\}$ and $I_i := \{j : (j,i) \in E\}$ be the set of vertices connected to $i \in V$ by outgoing and incoming edges, respectively. For two vertices i and j, we let $\delta_{ij} := 1$ if $i = j$ and $\delta_{ij} := 0$ if $i \neq j$.

A. Bilinear formulation

In the classical SPP the edge lengths are nonnegative finite constants c_{ij}, and the problem is typically formulated as

$$\min \quad \sum_{(i,j)\in E} c_{ij}\varphi_{ij} \tag{4a}$$

$$\text{s.t.} \quad \sum_{j\in O_i} \varphi_{ij} - \sum_{j\in I_i} \varphi_{ji} = \delta_{si} - \delta_{ti}, \quad \forall i \in V, \tag{4b}$$

$$\varphi_{ij} \geq 0, \qquad \forall (i,j) \in E. \tag{4c}$$

In this LP the shortest path is parameterized by the decision variables φ_{ij}, whose role is to take unit value if and only if the edge (i,j) is traversed by the shortest path. To this end, it is not necessary to explicitly enforce the integrality constraint $\varphi_{ij} \in \{0,1\}$: all the basic feasible solutions of 4 can be shown to have binary value and the addition of this constraint would not affect the optimal value of the problem. The multiplication of c_{ij} by φ_{ij} in (4a) ensures that only the edges along the shortest path contribute to the cost. Interpreting φ_{ij} as the units of flow traversing the edge (i,j), constraint (4b) is a standard conservation of flow: one unit of flow is injected from the source and ejected from the target, while the incoming and outgoing flows coincide for all the other vertices. For a binary solution, this guarantees that the set of edges for which $\varphi_{ij} = 1$ actually connects the source to the target. That these edges identify a valid path (i.e., a sequence of distinct vertices) is ensured only at optimality: since c_{ij} is nonnegative, visiting the same vertex more than once we would incur a nonnegative cost without making any progress towards the target.

A natural extension of the LP formulation 4 to the SPP described in Section II yields the following nonconvex program:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in E} \ell_{ij}(x_i, x_j)\varphi_{ij} && && \text{(5a)} \\ \text{s.t.} \quad & x_i \in X_i, && \forall i \in V, && \text{(5b)} \\ & \text{constraints of problem (4).} && && \text{(5c)} \end{aligned}$$

Note that here both the flows φ_{ij} and the vertex locations x_i are decision variables, and the product of the edge length $\ell_{ij}(x_i, x_j)$ by φ_{ij} in (5a) makes this problem nonconvex in general. Note also that, for the regions X_i not visited by the shortest path, the position of the points x_i is a free decision variable.

As stated, problem (5) has a small flaw. When $\varphi_{ij} = 0$, we would always want the cost contribution of the edge (i, j) in (5a) to be zero. However, as opposed to the classical SPP, in our problem formulation the edge lengths ℓ_{ij} are allowed to take infinite value, and the product in (5a) is undefined if $\ell_{ij}(x_i, x_j) = \infty$ and $\varphi_{ij} = 0$. We momentarily put aside this technical issue and postpone its resolution to the next subsection.

As a second step towards our MICP formulation, we move the nonconvexity of problem (5) from the objective (5a) to the constraints by defining two auxiliary variables per edge:

$$y_{ij} := \varphi_{ij} x_i, \quad z_{ij} := \varphi_{ij} x_j. \tag{6}$$

Momentarily assuming $\varphi_{ij} > 0$, we divide and multiply the arguments of ℓ_{ij} by φ_{ij} to restate the objective (5a) as

$$\sum_{(i,j)\in E} \ell_{ij}(y_{ij}/\varphi_{ij}, z_{ij}/\varphi_{ij})\varphi_{ij}. \tag{7}$$

For $\varphi_{ij} > 0$, the (i, j)th addend in (7) is recognized to be the *perspective* of the function ℓ_{ij} [37, Section 3.2.6]. The perspective operation preserves convexity: the convexity of ℓ_{ij} is hence inherited by the addends in (7), which are jointly convex in $(\varphi_{ij}, y_{ij}, z_{ij})$. The nonconvexity of our problem is now concentrated in the bilinear equations (6), whose convexification is the object of Section III-C. Before that, let us show how we can formally handle the case in which some of the flows φ_{ij} are zero.

B. Zero flows in the bilinear formulation

When a flow variable φ_{ij} is zero, the objective (7) is undefined: this issue is easily fixed by extending the domain of the perspective operation. (The definition below might appear cumbersome for numerical calculations but, as shown in Appendix A and Section IV, in most common cases it yields simple expressions readily amenable to standard optimization solvers.)

Definition 1. Let $f : \mathbb{R}^n \to \mathbb{R}^m \cup \{\infty\}^m$ be a closed convex function, and let $\bar{x} \in \mathbb{R}^n$ be any point such that $f(\bar{x})$ is finite. We define the *perspective* of the function f as[2]

$$\tilde{f}(\lambda, x) := \begin{cases} \lambda f(x/\lambda) & \text{if } \lambda > 0 \\ \lim_{\tau\downarrow 0} \tau f(\bar{x} + x/\tau) & \text{if } \lambda = 0 \\ \infty & \text{if } \lambda < 0 \end{cases},$$

where the value of the limit operation can be shown to be independent of the point $\bar{x}$.

Importantly, for a scalar-valued function, this "extended" perspective operation still preserves convexity [38, Propositions IV.2.2.1 and IV.2.2.1].

With this tool at our disposal, we rephrase (5) as the bilinear program

$$\begin{aligned} \min \quad & \sum_{(i,j)\in E} \tilde{\ell}_{ij}(\varphi_{ij}, y_{ij}, z_{ij}) && && \text{(8a)} \\ \text{s.t.} \quad & y_{ij} = \varphi_{ij} x_i, && \forall (i,j) \in E, && \text{(8b)} \\ & z_{ij} = \varphi_{ij} x_j, && \forall (i,j) \in E, && \text{(8c)} \\ & x_i \in X_i, && \forall i \in V, && \text{(8d)} \\ & \sum_{j\in O_i} \varphi_{ij} - \sum_{j\in I_i} \varphi_{ji} = \delta_{si} - \delta_{ti}, && \forall i \in V, && \text{(8e)} \\ & \varphi_{ij} \geq 0, && \forall (i,j) \in E. && \text{(8f)} \end{aligned}$$

When all the flows φ_{ij} are strictly positive, the objective (8a) is equal to (7) by definition, and this bilinear program is exactly the problem we derived above. On the other hand, when a flow φ_{ij} is zero, problem (8) does not raise any issue: the (i, j)th addend in (8a) is well defined and correctly evaluates to zero, even when $\ell_{ij}(x_i, x_j) = \infty$. To see this, notice that $\varphi_{ij} = 0$ implies $y_{ij} = z_{ij} = 0$ by (8b) and (8c), and Definition 1 reads

$$\tilde{\ell}_{ij}(0, 0, 0) = \lim_{\tau\downarrow 0} \tau \ell_{ij}(\bar{x}_i + 0/\tau, \bar{x}_j + 0/\tau) = 0,$$

where $\bar{x}_i$ and $\bar{x}_j$ are such that $\ell_{ij}(\bar{x}_i, \bar{x}_j)$ is finite.

Before concluding that (8) is a valid formulation of our generalized SPP, we are left to verify that the integrality property of the LP formulation (4) is actually inherited by this bilinear program.

Proposition 1. *The addition of the integrality constraints $\varphi_{ij} \in \{0, 1\}$ for all $(i, j) \in E$ does not affect the optimal value of problem* (8)*.*

Proof. Let $\ell^\star$ be the optimal value of (8) and $\ell^\star_{\{0,1\}} \geq \ell^\star$ be its cost after the addition of the integrality constraints. If (8) is infeasible, we immediately have $\infty = \ell^\star = \ell^\star_{\{0,1\}}$. Hence, we assume $\ell^\star$ to be finite. We let $x^\star_i$ be the optimal location of the vertices obtained by solving (8). We fix $x_i = x^\star_i$ in (8), and let only the flows φ_{ij} be decision variables. The resulting problem is equivalent to the LP (4), provided that we define $c_{ij} := \ell_{ij}(x^\star_i, x^\star_j)$ if this value is finite and we remove the edge (i, j) from the graph G if $\ell_{ij}(x^\star_i, x^\star_j) = \infty$. The cost of this LP must equal $\ell^\star$, otherwise our solution of (8) would not be optimal.

[2]More precisely, the one defined is the closure of the perspective function of f; the perspective function is typically defined to be infinite for $\lambda = 0$ [38, Section IV.2.2]. Since in this paper we only work with the former, there is no risk of misunderstanding.

Furthermore, by the integrality property of (4), we can assume the optimal flows $\varphi^\star_{ij}$ of this LP to be binary. Paired with the variables $x^\star_i$, the flows $\varphi^\star_{ij}$ yield a feasible solution with cost $\ell^\star$ for problem (8) subject to the integrality constraints. This implies $\ell^\star \geq \ell^\star_{\{0,1\}}$ and, consequently, $\ell^\star_{\{0,1\}} = \ell^\star$. □

C. Convexification of the bilinear formulation

The bilinear equalities (8b) and (8c) make it prohibitive to solve problem (8) directly. Here we reformulate these equalities as mixed-integer convex constraints, allowing to effectively solve our generalized SPP (1) to global optimality via branch-and-bound techniques.

The plan is to replace (8b) and (8c) with a collection of convex constraints. The role of these convex constraints is twofold:

- When the flows φ_{ij} take binary values, they must ensure that the bilinear equations (8b) and (8c) are correctly enforced. This guarantees that the MICP we construct is actually a valid problem formulation.
- When the flows φ_{ij} are not integral, they must envelop the solution set of problem (8) (i.e. all the feasible points which we cannot guarantee to be suboptimal beforehand) as tightly as possible. This increases the efficiency of our formulation.

The family of convex constraints we derive below is sufficient to achieve the first of these two goals. Section V is devoted to the generation of additional convex constraints that increase the tightness of the proposed MICP.

The next lemma describes an algorithmic procedure to construct a convex envelope around the solution set of problem (8). In its statement, we let

$$\Phi_i := a + \sum_{j\in O_i} b_j\varphi_{ij} + \sum_{j\in I_i} c_j\varphi_{ji}.$$

be a generic scalar affine combination of the flows traversing vertex $i \in V$.

Lemma 2. *Any feasible solution of problem* (8) *that satisfies* $\Phi_i = 0$ *must also verify the linear equality constraint*

$$ax_i + \sum_{j\in O_i} b_j y_{ij} + \sum_{j\in I_i} c_j z_{ji} = 0. \tag{9}$$

Furthermore, any feasible solution of (8) *that satisfies* $\Phi_i \geq 0$ *must also verify the convex constraint*

$$ax_i + \sum_{j\in O_i} b_j y_{ij} + \sum_{j\in I_i} c_j z_{ji} \in \Phi_i X_i. \tag{10}$$

Proof. Expanding the left-hand side of the identity $\Phi_i x_i = \Phi_i x_i$, and using the bilinear constraints (8b) and (8c), we get the bilinear equality

$$ax_i + \sum_{j\in O_i} b_j y_{ij} + \sum_{j\in I_i} c_j z_{ji} = \Phi_i x_i.$$

When $\Phi_i = 0$, we obtain the linear equality (9). If $\Phi_i \geq 0$, the inclusion $x_i \in X_i$ from (8d) immediately implies (10). That (10) is a convex constraint (jointly in the flows and the "spatial" variables x_i, y_{ij}, and z_{ji}) follows from the fact that any constraint of the form $x \in \lambda S$ is jointly convex in λ and x, provided that S is convex and $\lambda \geq 0$. This can be verified by applying the definition of a convex set to $\{(\lambda, x) : \lambda \geq 0, x = \lambda y, y \in S\}$. □

Lemma 2 leverages the bilinearities (8b) and (8c) to translate any equality (or inequality) between the flows incident to a vertex into a corresponding "spatial" linear equality (9) (or convex constraint (10)). From every relation of the form $\Phi_i = 0$ or $\Phi_i \geq 0$ that is guaranteed to hold at optimality of (8), and which is not trivially implied by the ones we already used, we can then generate a spatial constraint that tightens our convex envelope around the solution set of problem (8).

We now apply Lemma 2 to derive a family of convex constraints sufficient for the bilinear constraints (8b) and (8c) to be correctly enforced in case of a binary flow. We start from the trivial affine combinations $\Phi_i := \varphi_{ij}$ and $\Phi_i := 1 - \varphi_{ij}$ for all $i \in V$ and $j \in O_i$, which are certainly nonnegative at optimality of problem (8). Applying (10) we get the convex constraints $y_{ij} \in \varphi_{ij} X_i$ and $x_i - y_{ij} \in (1 - \varphi_{ij})X_i$. For convenience, we define the set

$$\Lambda_i := \{(\varphi, x, y) : \varphi \in [0, 1], y \in \varphi X_i, x - y \in (1 - \varphi)X_i\}$$

and we rewrite the latter convex constraints as

$$(\varphi_{ij}, x_i, y_{ij}) \in \Lambda_i, \qquad \forall (i, j) \in E. \tag{11}$$

We will use the following property of the set Λ_i multiple times.

Proposition 2. *The projection of the set* Λ_i *onto the space of the* x *variables coincides with* X_i*.*

Proof. The inclusion $X_i \subseteq \mathrm{proj}_x \Lambda_i$ is verified intersecting Λ_i with the hyperplane $\varphi = 0$. For the reverse direction, assume $x \in \mathrm{proj}_x \Lambda_i$. Then there exist $\varphi \in [0, 1]$, y, and $x^0, x^1 \in X_i$ such that $y = \varphi x^1$ and $x - y = (1 - \varphi)x^0$. Summing these two equations we get $x = \varphi x^1 + (1 - \varphi)x^0 \in X_i$, where the inclusion follows from the convexity of X_i. □

We repeat the operation above inverting the roles of the vertices i and j, and considering i as a vertex incoming j. We let $\Phi_j := \varphi_{ij} \geq 0$ and $\Phi_j := 1 - \varphi_{ij} \geq 0$ for all $j \in V$ and $i \in I_j$. Using (10), these result in the convex constraints

$$(\varphi_{ij}, x_j, z_{ij}) \in \Lambda_j, \qquad \forall (i, j) \in E. \tag{12}$$

For a binary flow, the two convex constraints (11) and (12) are already sufficient to correctly enforce the bilinear equations (8b) and (8c). When $\varphi_{ij} = 0$, conditions (8b) and (8c) require the auxiliary variables y_{ij} and z_{ij} to collapse to zero, so that the (i, j)th cost addend vanishes. Correctly, condition (11) yields $y_{ij} = 0$ as well as the redundant condition $x_i \in X_i$. Similarly, from (12), we have $z_{ij} = 0$ and $x_j \in X_j$. When $\varphi_{ij} = 1$, the bilinear constraints require the variables y_{ij} and z_{ij} to match x_i and x_j, respectively, and ensure that (i, j)th cost addend takes the value $\ell_{ij}(x_i, x_j)$. Accordingly, from (11) and (12) we get $y_{ij} = x_i \in X_i$ and $z_{ij} = x_j \in X_j$.

Replacing the bilinear equations (8b) and (8c) with the convex constraints (11) and (12), we then obtain the following mixed-integer convex formulation of problem (1):

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E} \tilde{\ell}_{ij}(\varphi_{ij}, y_{ij}, z_{ij}) && && \text{(13a)}\\
\text{s.t.} \quad & (\varphi_{ij}, x_i, y_{ij}) \in \Lambda_i, && \forall (i,j) \in E, && \text{(13b)}\\
& (\varphi_{ij}, x_j, z_{ij}) \in \Lambda_j, && \forall (i,j) \in E, && \text{(13c)}\\
& \sum_{j\in O_i} \varphi_{ij} - \sum_{j\in I_i} \varphi_{ji} = \delta_{si} - \delta_{ti}, && \forall i \in V, && \text{(13d)}\\
& \varphi_{ij} \in \{0,1\}, && \forall (i,j) \in E. && \text{(13e)}
\end{aligned}$$

Notice that we dropped the inclusion (8d). This is allowed by Proposition 2: assuming, without loss of generality, that each vertex is incident to at least one edge, condition (8d) is implied by (13b) and (13c), even when the flows φ_{ij} are not integer.

D. Comments on the mixed-integer convex program

Overall, the size of problem (13) scales bilinearly with the number of vertices and edges and with the dimension d of the space in which the sets X_i live. More precisely, we have $|E|$ binary variables and $O(d(|V|+|E|))$ continuous variables. Assuming the number of constraints from (13b) and (13c) to be independent of d, the constraints are $O(|V| + |E|)$.

The convex relaxation of (13) is obtained simply by dropping condition (13e) from the problem formulation. Note that, in contrast to the bilinear program (8), we generally expect to have a nonzero relaxation gap between the optimal values of the MICP (13) and of its convex relaxation.

As for the classical LP formulation (4), the nonnegativity of ℓ_{ij} ensures that the shortest path identified by the MICP does not contain cycles. However, the same is not necessarily true for the convex relaxation of (13), whose optimal solution might involve cycles with nonzero flow. This is analyzed more in depth in Section V-B where we propose a set of additional constraints that partially counteract this phenomenon.

IV. Common Special Cases

We now discuss some special cases of the procedure presented in Section III. We analyze several common choices for the sets X_i and the edge-length functions ℓ_{ij}, providing explicit expressions for the various components of problem (13).

The following property of the perspective operation will allow to derive a functional description of the set Λ_i from a functional description of X_i.

Proposition 3. *Let the function f be defined as in Definition 1. Assume the set S to be bounded and to admit the functional description $S = \{x : f(x) \le 0\}$. For a nonnegative scalar λ, we have*

$$\lambda S = \{x : \tilde{f}(\lambda, x) \le 0\}.$$

Proof. We denote the set on the right-hand side as $\tilde{S}(\lambda)$. For $\lambda = 0$ we have $\lambda S = \{0\}$ and, using Definition 1, $\tilde{S}(0) = \{x : \lim_{\tau\downarrow 0} \tau f(\bar{x}+x/\tau) \le 0\}$. The latter set can be recognized to be the recession cone of S, which equals $\{0\}$ since S is bounded. For $\lambda > 0$, we have $\tilde{S}(\lambda) = \{x : \lambda f(x/\lambda) \le 0\} = \{x : f(x/\lambda) \le 0\} = \{\lambda y : f(y) \le 0\} = \lambda S$. □

A. When the sets X_i are singletons

We start by showing that when the sets X_i are singletons the MICP (13) is equivalent to the LP formulation (4) of the classical SPP. The valuable integrality property of this LP, for which all basic feasible flows φ_{ij} are binary, is hence inherited by our formulation in this limiting case.

Assuming $X_i := \{\theta_i\}$ for all $i \in V$, we have

$$\begin{aligned}
\Lambda_i &= \{(\varphi, x, y) : \varphi \in [0,1], y = \varphi\theta_i, x - y = (1-\varphi)\theta_i\}\\
&= \{(\varphi, x, y) : \varphi \in [0,1], y = \varphi\theta_i, x = \theta_i\}.
\end{aligned}$$

Therefore, constraints (13b) and (13c) can be solved out to get $x_i = \theta_i$ for all $i \in V$, as well as $y_{ij} = \varphi_{ij}\theta_i$ and $z_{ij} = \varphi_{ij}\theta_j$ for all $(i,j) \in E$. Using these values, the (i,j)th cost addend in (13a) becomes

$$\tilde{\ell}_{ij}(\varphi_{ij}, \varphi_{ij}\theta_i, \varphi_{ij}\theta_j) = \begin{cases} \varphi_{ij}\ell_{ij}(\theta_i, \theta_j) & \text{if } \varphi_{ij} > 0 \\ 0 & \text{if } \varphi_{ij} = 0 \end{cases}.$$

Assuming each constant edge length $c_{ij} := \ell_{ij}(\theta_i, \theta_j)$ to be finite, otherwise we can just remove the edge (i,j) from the problem, the latter equals $c_{ij}\varphi_{ij}$. Problem (13) is then reduced to the LP (4) with the additional integrality requirement (13e), which we know is redundant in this case.

B. When the sets X_i are polytopes

Assume the sets X_i are polytopes of the form $\{x : A_i x \le b_i\}$. To derive an explicit description of the set Λ_i, we use Proposition 3 and the expression of the perspective of an affine function from Appendix A-C. We obtain the polytope

$$\begin{aligned}
\Lambda_i = \{(\varphi, x, y) : \varphi \in [0,1], A_i y \le b_i\varphi,\\
A_i(x-y) \le b_i(1-\varphi)\}.
\end{aligned}$$

C. When the sets X_i are affine transformations of the unit ball

Let $X_i := \{x : \|A_i x + b_i\| \le 1\}$. Using Proposition (3) and the formula from Appendix A-D, we have

$$\begin{aligned}
\Lambda_i = \{(\varphi, x, y) : \varphi \in [0,1], \|A_i y + b_i\varphi\| \le \varphi,\\
\|A_i(x-y) + b_i(1-\varphi)\| \le (1-\varphi)\}.
\end{aligned}$$

For a p-norm with $p \in \{1, \infty\}$ the set Λ_i is a polytope, for $p = 2$ it is the intersection of two Second-Order Cones (SOCs).

D. When the edge length ℓ_{ij} is constant

Assume the edge-length function to be constant and finite as in the classical SPP: $\ell_{ij}(x_i, x_j) := c_{ij}$. Using the formula from Appendix A-A, the addends in (13a) become $c_{ij}\varphi_{ij}$, and the spatial variables x_i, y_{ij}, and z_{ij} in problem (13) become free decision variables. As in the case of singleton sets X_i from Section (IV-A), problem (13) simplifies to the LP formulation (4) of the classical SPP.

E. When the edge length ℓ_{ij} is positively homogeneous

Assume the edge length ℓ_{ij} to be positively homogeneous, i.e., $\ell_{ij}(ax_i, ax_j) = a\ell_{ij}(x_i, x_j)$ for all x_i, x_j, and $a \geq 0$. An example of such a function is $\ell_{ij}(x_i, x_j) := \|A_{ij}x_i + B_{ij}x_j\|$, from which (2) is recovered when the norm is the 2-norm and $B_{ij} := -A_{ij} := I$. As shown in Appendix A-B, in this special case the addends in (13a) verify

$$\tilde{\ell}_{ij}(\varphi_{ij}, y_{ij}, z_{ij}) = \ell_{ij}(y_{ij}, z_{ij}). \tag{14}$$

In case of a p-norm, using slack variables, this is implemented as a linear objective subject to linear constraints if $p \in \{1, \infty\}$ or subject to SOC constraints if $p = 2$.

F. When the edge length ℓ_{ij} is a positive semidefinite quadratic form

Assume the edge length to be defined as $\ell_{ij}(x_i, x_j) := \|A_{ij}x_i + B_{ij}x_j\|_2^2$. Notice that the edge length (3) is recovered as a special case for $B_{ij} := -A_{ij} := I$. Following Appendix A-E, for $\varphi_{ij} > 0$ the addends in (13a) become

$$\tilde{\ell}_{ij}(\varphi_{ij}, y_{ij}, z_{ij}) = \|A_{ij}y_{ij} + B_{ij}z_{ij}\|_2^2/\varphi_{ij}, \tag{15}$$

while for $\varphi_{ij} = 0$ we have

$$\tilde{\ell}_{ij}(0, y_{ij}, z_{ij}) = \begin{cases} 0 & \text{if } A_{ij}y_{ij} + B_{ij}z_{ij} = 0 \\ \infty & \text{otherwise} \end{cases}. \tag{16}$$

The latter expressions can be modeled using a SOC constraint. We introduce a nonnegative slack variable l_{ij} per edge, we replace the objective (13a) with $\sum_{(i,j)\in E} l_{ij}$, and we add the rotated SOC constraint

$$\varphi_{ij}l_{ij} \geq \|A_{ij}y_{ij} + B_{ij}z_{ij}\|_2^2. \tag{17}$$

For $\varphi_{ij} > 0$, the slack l_{ij} is forced by the cost to coincide with the right-hand side of (15). For $\varphi_{ij} = 0$, l_{ij} is pushed to zero which, recalling that (13b) and (13c) imply $y_{ij} = z_{ij} = 0$ whenever $\varphi_{ij} = 0$, coincides with the right-hand side of (16). We conclude that (17) models the two expressions above correctly.

G. When the edge length ℓ_{ij} enforces a hard constraint

Imagine we want to couple the position of the two points x_i and x_j via a constraint of the form $(x_i, x_j) \in X_{ij}$, with X_{ij} closed and convex. One way to proceed is to define ℓ_{ij} such that $\ell_{ij}(x_i, x_j) = \infty$ if (x_i, x_j) does not belong to X_{ij}. Following Appendix A-F, we see that this results in the following extra constraint for problem (13):

$$(y_{ij}, z_{ij}) \in \varphi_{ij}X_{ij}. \tag{18}$$

When $\varphi_{ij} = 0$, constraints (13b) and (13c) imply $y_{ij} = z_{ij} = 0$, and condition (18) is trivially verified. When $\varphi_{ij} = 1$, from (13b) and (13c) we have $y_{ij} = x_i$ and $z_{ij} = x_j$, and (18) becomes $(x_i, x_j) \in X_{ij}$ as desired.

With the hope of increasing the tightness of the convex relaxation, one could think of preprocessing the MICP and remove the edge (i, j) from the graph if a transition from vertex i to vertex j is never feasible, i.e., if the intersection of $X_i \times X_j$ and X_{ij} is empty. This turns out to be unnecessary.

Proposition 4. *Assume that $(X_i \times X_j) \cap X_{ij} = \emptyset$, then any feasible solution of the convex relaxation of* (13)*, subject to* (18)*, is such that $\varphi_{ij} = 0$.*

Proof. Assume $\varphi_{ij} > 0$. By (13b) and (13c) we have $y_{ij}/\varphi_{ij} \in X_i$ and $z_{ij}/\varphi_{ij} \in X_j$, whereas (18) implies $(y_{ij}/\varphi_{ij}, z_{ij}/\varphi_{ij}) \in X_{ij}$. A contradiction. □

V. Formulation Strengthening

In Section III-C we have constructed the MICP formulation (13) of our generalized SPP by replacing the bilinear constraints (8b) and (8c) with their convex relaxations (13b) and (13c). Even though these convex constraints are sufficient to ensure the correctness of the MICP, they envelop the solution set of our problem very loosely, making our formulation weak. In this section we analyze two simple constraints that substantially increase the tightness of the convex relaxation of our MICP.

A. Spatial counterpart of the conservation-of-flow constraint

After the trivial inequalities $\varphi_{ij} \geq 0$ and $1 - \varphi_{ij} \geq 0$ exploited in Section III-C, the next natural candidate to take the role of the affine combination Φ_i in Lemma 2 is the difference between the outgoing and incoming flows of vertex $i \in V$. According to the conservation of flow (8e), this value is always zero at optimality of problem (8).

Applying (9), we obtain $d|V|$ linear equality constraints

$$\sum_{j\in O_i} y_{ij} - \sum_{j\in I_i} z_{ji} = \delta_{si}x_s - \delta_{ti}x_t, \qquad \forall i \in V. \tag{19}$$

At optimality of the MICP, these simplify to trivial identities but, for noninteger flows, they dramatically increase the strength of our formulation. A toy example that illustrates this claim is given in Section IX-A.

B. Degree constraints

The second group of constraints we analyze are degree constraints, i.e., constraints on the maximum amount of flow that can go though a vertex [39].

As discussed in Section III, an optimal solution of the MICP (13) identifies a sequence of edges that never visits the same vertex twice. More precisely, at optimality of (13), the following conditions are always verified:

$$\sum_{j\in O_i} \varphi_{ij} \leq 1 - \delta_{ti}, \quad \sum_{j\in I_i} \varphi_{ji} \leq 1 - \delta_{si}, \quad \forall i \in V. \tag{20}$$

These conditions need not to hold for the convex relaxation of (13), whose optimal solution might involve a nonzero flow through the edges of a cycle and vertices with incoming flow greater than one; a simple example of this behavior is illustrated in Section IX-B. Enforcing (20) we explicitly rule out this undesired behavior, increasing the strength of our MICP. Note that it is not necessary to enforce both the conditions in (20): one is implied by the other and the conservation of flow (13d). We have then a total of $|V|$ additional linear inequality constraints for our problem.

Using Lemma 2, also the degree constraints (20) can be translated into spatial constraints. Applying (10), the first of the two inequalities in (20) yields the convex constraint

$$x_i - \delta_{ti} x_t - \sum_{j \in O_i} y_{ij} \in \left(1 - \delta_{ti} - \sum_{j \in O_i} \varphi_{ij}\right) X_i, \quad \forall i \in V.$$

Even if this constraint is not implied by the ones we already have, its addition to our problem formulation did not, empirically, yield a performance increase for the problem instances considered in the numerical experiments of this paper. For this reason, and in order to simplify the derivations below, we decide not to include it in our formulation.

VI. The Dual Program

In this section we derive the dual of the convex relaxation of the MICP (13), subject to the tightening constraints (19) and (20). This helps us gaining a deeper understanding of our problem formulation, and it is useful to prove bounds on its optimal value (see Section VII).

We start by defining the main tools employed in this section.

Definition 2. For a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, we call

$$f^*(x) = \sup_y (x^\top y - f(y))$$

the *conjugate function* of f.

Definition 3. For a set $S \subseteq \mathbb{R}^n$, we define the *indicator function* of S as

$$\iota_S(x) = \begin{cases} 0 & \text{if } x \in S \\ \infty & \text{if } x \notin S \end{cases}.$$

Definition 4. For a set $S \subseteq \mathbb{R}^n$, we define the *support function* of S as

$$\sigma_S(x) = \sup_{y \in S} x^\top y.$$

The conjugate function f^* is always convex. If the set S is convex, so is the indicator function ι_S. The support function is immediately recognized to be the conjugate of the indicator function ($\sigma_S(x) = \iota_S^*(x)$) and, as such, it is always convex.

Conjugate functions allow to easily derive the dual of a problem with convex objective and linear constraints only, such as

$$\min_{x,y} \quad f(x) \tag{21a}$$
$$\text{s.t.} \quad Ax + By = c, \tag{21b}$$
$$Fx + Gy \leq h. \tag{21c}$$

Retracing the steps from [37, Section 5.1.6], the dual of problem (21) is easily seen to be

$$\max_{\pi,\rho} \quad -f^*(-A^\top \pi - F^\top \rho) - c^\top \pi - h^\top \rho \tag{22a}$$
$$\text{s.t.} \quad B^\top \pi + G^\top \rho = 0, \tag{22b}$$
$$\rho \geq 0, \tag{22c}$$

where π is the Lagrange multiplier for the equality constraint (21b) and ρ for the inequality (21c).

Our objective function (13a) is convex, and among the constraints of (13), together with (19) and (20), the only nonlinear conditions are (13b) and (13c). We then move these two constraints to the objective function by adding their indicator functions as penalties:

$$\sum_{(i,j) \in E} (\tilde{\ell}_{ij}(\varphi_{ij}, y_{ij}, z_{ij}) + \iota_{\Lambda_i}(\varphi_{ij}, x_i, y_{ij}) + \iota_{\Lambda_j}(\varphi_{ij}, x_j, z_{ij})). \tag{23}$$

Problem (13) is unchanged, but it now consists in the minimization of a convex function subject to linear constraints only. The recipe (22) can be then applied directly.

There are a couple of additional observations that facilitate the derivation of our dual:

- Following (22a), we would need to derive the conjugate of the primal objective (23). This would not yield a neat expression since, in general, the conjugate of the sum of functions is not the sum of the conjugates. However, if we have a sum of independent functions $\sum_{i=1}^n f_i(x_i)$ it is trivially verified that its conjugate is $\sum_{i=1}^n f_i^*(x_i)$. We leverage this observation as follows. Every time we move one of the constraints (13b) or (13c) to the objective using its indicator function we: i) add a copy of the variables involved in this constraint; ii) add a constraint enforcing the equality between this copy and the original decision variables; and iii) let the copies, and not the original variables, be the arguments of the indicator function. In this way, the objective (23) becomes a sum of independent functions and its conjugate can be taken one addend at the time.
- The following result from [40, Proposition 2.3(iv)] is useful when deriving the conjugate of the addends in (23). For a closed proper convex function f, the conjugate of the perspective function is

$$(\tilde{f})^*(\lambda, x) = \iota_C(\lambda, x), \tag{24}$$

where $C := \{(\lambda, x) : \lambda + f^*(x) \leq 0\}$.

The dual of our program is reported in equation (25); its decision variables are the following Lagrange multipliers. For $i \in V$, we associate $\alpha_i \in \mathbb{R}$, $\beta_i \in \mathbb{R}^d$, $\gamma_i \in \mathbb{R}$ to the conservation of flow (13d), its spatial version (19), and the first degree constraint in (20), respectively. As just discussed, each one of the constraints in (13b) and (13c) requires the addition of an equality constraint, and consequently a multiplier, for each one of its arguments. For $(i,j) \in E$, we let $\mu_{ij}^\varphi \in \mathbb{R}$, $\mu_{ij}^x \in \mathbb{R}^d$, and $\mu_{ij}^y \in \mathbb{R}^d$ be the multipliers corresponding to each argument in (13b). Similarly, for constraint (13c) we use the multipliers $\nu_{ij}^\varphi \in \mathbb{R}$, $\nu_{ij}^x \in \mathbb{R}^d$, and $\nu_{ij}^z \in \mathbb{R}^d$.

A. Comments on the dual program

We start by analyzing the structure of problem (25). We compare it to the dual of the LP formulation (4) of the classical SPP:

$$\max \quad \alpha_t - \alpha_s \tag{26a}$$
$$\text{s.t.} \quad c_{ij} \geq \alpha_j - \alpha_i, \qquad \forall (i,j) \in E. \tag{26b}$$

$$\max \quad \alpha_t - \alpha_s - \sum_{i \in V - \{t\}} \gamma_i - \sum_{(i,j) \in E} (\sigma_{\Lambda_i}(\mu^\varphi_{ij}, \mu^x_{ij}, \mu^y_{ij}) + \sigma_{\Lambda_j}(-\nu^\varphi_{ij}, -\nu^x_{ij}, -\nu^z_{ij})) \tag{25a}$$

$$\text{s.t.} \quad -\ell^*_{ij}(-\mu^y_{ij} - \beta_i, \nu^z_{ij} + \beta_j) \geq \alpha_j - \alpha_i - \gamma_i + \nu^\varphi_{ij} - \mu^\varphi_{ij}, \qquad \forall (i,j) \in E, \tag{25b}$$

$$\sum_{j \in O_i} \mu^x_{ij} - \sum_{j \in I_i} \nu^x_{ji} = \delta_{si}\beta_s - \delta_{ti}\beta_t, \qquad \forall i \in V, \tag{25c}$$

$$\gamma_i \geq 0, \qquad \forall i \in V. \tag{25d}$$

Here, as in (25), α_i is the multiplier of the conservation of flow (4b) at vertex i. These multipliers are well-known to be interpretable as potentials: the objective asks to maximize the potential jump between the source s and the target t, the constraint ensures that the potential jump along each edge does not exceed the length of the edge itself. Despite the many additional terms, problem (25) has a similar interpretation.

The objective (25a) still maximizes the potential jump $\alpha_t - \alpha_s$, but it also includes the multipliers of the degree constraints (20) (which would be redundant in the classical SPP) and the support functions conjugate to the indicators of the primal constraints (13b) and (13c).

Constraint (25b) comes from conjugating the perspective of the edge length in (23): according to (24) this yields an indicator function in the dual objective, hence a dual constraint. This constraint clearly resembles (26b): the potential jump $\alpha_j - \alpha_i$, together with some extra terms, is upper bounded by a term associated to the edge length. The connection is even more evident noticing that, for $\ell_{ij}(x_i, x_j) := c_{ij}$, we have $\ell^*_{ij}(x_i, x_j) = -c_{ij}$ and the left-hand sides of (25b) and (26b) coincide.

Finally, we have constraint (25c) which yields a nice primal-dual symmetry: notice that this condition coincides with the spatial conservation of flow (19) if we identify μ^x_{ij} with y_{ij}, ν^x_{ji} with z_{ij}, β_s with x_s, and β_t with x_t.

Problem (25) is always feasible and its cost is nonnegative. This is seen by setting all the multipliers to zero: all the constraints are verified and the cost is zero. By weak duality, the optimal value of problem (25) bounds from below the cost of the convex relaxation of (13) subject to (19) and (20).

VII. A Lower Bound for the Convex Relaxation

We have discussed how the addition of the constraints (19) and (20) can increase the tightness of the convex relaxation of the MICP (13). The result presented in this section can be seen as a "sanity check" for the resulting formulation: the discussion from Section II-A suggests a simple lower bound on the optimal cost of problem (1), here we show that the constraints (19) and (20) are sufficient for the convex relaxation of our problem to always recover this bound.

Consider a problem setup in which all the edges $(i,j) \in E$ share the same length function

$$\ell_{ij}(x_i, x_j) := \ell(x_j - x_i), \tag{27}$$

which only depends on the difference $x_j - x_i$, and not on x_i and x_j independently. Assume also that $\ell(0) = 0$. Momentarily, we focus our attention on the case in which the source and target sets are single points θ_s and θ_t. With the goal of lower bounding the optimal cost of problem (1), we can drop the constraints $x_i \in X_i$ for all $i \in V - \{s,t\}$. Similarly to the proof of Lemma 1, an optimal solution for this relaxed problem is obtained by first detecting an s-t path $P = (i_k)_{k=0}^K$ with maximum number K of edges, and then by arranging the points x_{i_k} at equal distance along the segment Γ connecting θ_s and θ_t. The cost of this arrangement is

$$K\ell\left(\frac{\theta_t - \theta_s}{K}\right) = \tilde{\ell}(K, \theta_t - \theta_s).$$

On the other hand, any s-t path can contain at most $K = |V| - 1$ edges, therefore a simple lower bound on the optimal cost of (1) is $\tilde{\ell}(|V| - 1, \theta_t - \theta_s)$.

More generally, when the sets X_s and X_t are allowed to be full dimensional, we have the following result.

Proposition 5. *Assume the edges $(i,j) \in E$ to have a common length function* (27)*. The optimal cost of the convex relaxation of the MICP* (13)*, subject to the constraints* (19) *and* (20)*, is not smaller than*

$$\min_{x_s \in X_s, x_t \in X_t} \tilde{\ell}(|V| - 1, x_t - x_s). \tag{28}$$

Before proving this proposition, we specialize the dual program (25) to the edge length (27). In this case, the term $\ell^*_{ij}(-\mu^y_{ij} - \beta_i, \nu^z_{ij} + \beta_j)$ in the dual constraint (25b) becomes

$$\sup_{x_i, x_j} ((-\mu^y_{ij} - \beta_i)^\top x_i + (\nu^z_{ij} + \beta_j)^\top x_j - \ell(x_j - x_i)). \tag{29}$$

If $\mu^y_{ij} + \beta_i \neq \nu^z_{ij} + \beta_j$ this supremum would be infinite and the problem infeasible (this is seen by setting $x_i = x_j$ and recalling that $\ell(0) = 0$). Therefore, in case of the edge length (27), we have an additional dual constraint

$$\mu^y_{ij} + \beta_i = \nu^z_{ij} + \beta_j, \qquad \forall (i,j) \in E. \tag{30}$$

Using this constraint, the conjugate (29) becomes

$$\sup_{x_i, x_j} ((\nu^z_{ij} + \beta_j)^\top (x_j - x_i) - \ell(x_j - x_i)) = \ell^*(\nu^z_{ij} + \beta_j),$$

and constraint (25b) reads

$$-\ell^*(\nu^z_{ij} + \beta_j) \geq \alpha_j - \alpha_i - \gamma_i + \nu^\varphi_{ij} - \mu^\varphi_{ij}, \quad \forall (i,j) \in E. \tag{31}$$

Proof of Proposition 5. The plan is to synthesize a dual feasible solution whose cost coincides with (28). The thesis is then implied by weak duality.

Let $s' \in O_s$ be any vertex directly connected to the source s. Similarly, let $t' \in I_t$. We consider the following assignment for the Lagrange multipliers of (13d), (19), and (20):

$$\alpha_i := 0, \qquad \beta_i := \beta, \qquad \gamma_i := \ell^*(\beta), \qquad \forall i \in V,$$

where β is a decision variable and, noticing that $\ell(0) = 0$ implies $\ell^* \geq 0$, we see that γ_i is nonnegative as required by (25d). We set to zero the multipliers of (13b) and (13c) corresponding to the primal variables φ, y, and z:

$$\mu^{\varphi}_{ij} := \nu^{\varphi}_{ij} := 0, \qquad \mu^{y}_{ij} := \nu^{z}_{ij} := 0, \qquad \forall (i,j) \in E;$$

while for the ones associated to the variable x we let

$$\begin{aligned} &\mu^{x}_{s\hat{s}} := \nu^{x}_{\hat{t}t} := \beta, \\ &\mu^{x}_{ij} := \nu^{x}_{ij} := 0, \qquad \forall (i,j) \in E - \{(s,s'),(t',t)\}. \end{aligned}$$

We substitute these multipliers in the dual problem (25), with the additional constraint (30) and with (25b) replaced by (31). The dual problem is reduced to

$$\max_{\beta}(-(|V|-1)\ell^*(\beta) - \sigma_{\Lambda_s}(0,\beta,0) - \sigma_{\Lambda_t}(0,-\beta,0)).$$

We notice that

$$\sigma_{\Lambda_s}(0,\beta,0) = \sup_{(\varphi,x,y)\in\Lambda_s} \beta^\top x = \sup_{x \in X_s} \beta^\top x = \sigma_{X_s}(\beta),$$

where the second equality follows from Proposition 2. Analogously, $\sigma_{\Lambda_t}(0,-\beta,0) = \sigma_{X_t}(-\beta)$. Our dual program is then reduced to

$$\max_{\beta}(-(|V|-1)\ell^*(\beta) - \sigma_{X_s}(\beta) - \sigma_{X_t}(-\beta)).$$

This can be recognized to be the dual of the minimization in (28). Thus, by strong duality, these two problems have the same optimal value. □

Note that in the proof above both the multipliers β_i and γ_i of (19) and (20) are nonzero. Meaning that these two constraints are essential for our problem formulation to pass the sanity check from Proposition 5.

VIII. Application to Optimal Control of Dynamical Systems

We now show how multiple optimal control problems for discrete-time dynamical systems can be formulated as the SPP described in Section II.

As a first example, we consider the problem of driving a linear system across a disjunctive subset of state space. We analyze both the cases in which the time horizon of the control problem is to be optimized or it is fixed a priori. Problems of this form are pervasive in motion planning of robots [5], [41], [8] and autonomous vehicles [2], [4], [7]; see also the recent review [42].

In Section VIII-C, we consider the more general problem of controlling a PieceWise-Affine (PWA) system [43]. This broad class of hybrid systems has seen application in a multitude of areas: automotive [44], power electronics [45], robotics [46], [47], and many more [48]. Since the seminal work by Bemporad and Morari [9], mixed-integer control algorithms for hybrid systems have been widely developed. At present, computation times are the main limitation to a widespread application of these techniques, and their decrease has been a central theme recently [49], [50], [51], [52], [53]. The approach described below is fundamentally different from today's most efficient mixed-integer formulations of these problems [15], and it might enable the design of higher-performance control algorithms.

A. Constrained linear systems

Consider the constrained linear control system

$$\zeta(k+1) = A\zeta(k) + Bu(k),$$

where $k \in \mathbb{Z}_{\geq 0}$ is the discrete time, $\zeta \in Z \subset \mathbb{R}^{d_\zeta}$ is the system state, and $u \in U \subset \mathbb{R}^{d_u}$ is the control input. We assume the constraint set Z to be the union of convex compact sets $\{Z_i\}_{i\in V}$ for some finite index set V, which will represent the vertex set of our graph. For simplicity, here we assume the control set U to be convex and compact. Our goal is to find a feasible control sequence that transfers the system from the given initial state $Z_s := \{\zeta(0)\}$ to a target set Z_t, while minimizing some cost function. The time K at which the system reaches the target set is not specified a priori.

We might allow the system to transition from any set Z_i to any set Z_j, in which case we let $E := \{(i,j) \in V^2 : i \neq j\}$. Otherwise, we can prevent an undesired transition by not including the edge (i,j) in E. As in Remark 1, self-transitions can be allowed by introducing copies of the sets Z_i. We let $G := (V,E)$ be the graph underlying problem (1). To each vertex $i \in V$ we associate the convex compact set $X_i := Z_i \times U$, and each set X_i contains the point $x_i := (\zeta_i, u_i) \in \mathbb{R}^d$, where $d := d_\zeta + d_u$.

We define the edge-length function ℓ_{ij} so that state transitions that do not agree with the system dynamics incur infinite cost. This is done as in Section IV-G:

$$\ell_{ij}(x_i,x_j) := \begin{cases} \ell'_{ij}(x_i,x_j) & \text{if } \zeta_j = A\zeta_i + Bu_i \\ \infty & \text{otherwise} \end{cases}. \tag{32}$$

As an example, in case of a minimum time problem, where we want to reach the target Z_t as soon as possible, we can define $\ell'_{ij}(x_i,x_j) := 1$. Otherwise, another popular class of cost functions in control are positive semidefinite quadratic forms

$$\sum_{k=0}^{K-1} (\zeta^\top(k) Q \zeta(k) + u^\top(k) R u(k)).$$

For $Z_t = \{0\}$, this objective balances the magnitude of the control effort and the distance of the system from the target. In this case, we let $\ell'_{ij}(x_i,x_j) := \zeta_i^\top Q \zeta_i + u_i^\top R u_i$.

After having solved the MICP (13), and recovered the shortest path $P = (i_k)_{k=0}^K$, the optimal control sequence is $u(k) = u_{i_k}$ for $k = 0,\dots,K-1$, while the optimal state trajectory is $\zeta(k) = \zeta_{i_k}$ for $k = 0,\dots,K$. (Note that the control input u_t associated with the target set X_t is a redundant variable in these optimizations.)

B. Problems with fixed time horizon

Choosing a fixed time horizon K for a control problem is a tricky compromise between performance and computational efficiency. Furthermore, letting K be a decision variable has also several technical advantages [54], [55], [56]. Nevertheless, in some cases, we might need the value of K to be fixed. With some extra effort, also fixed-horizon control problems can be formulated as the SPP (1).

Here we assume to have a finite collection of convex compact sets $Z_l \subset \mathbb{R}^{d_\zeta}$ indexed by $l \in L$. The initial state $\zeta(0)$ is given. At each time step $k = 1, \dots, K-1$ the state $\zeta(k)$ is allowed to lie in any of the regions Z_l. The final state $\zeta(K)$ must belong to the set $\bar{Z} \subset \mathbb{R}^{d_\zeta}$. As in the previous case, we require the controls $u(k)$ to lie in the set U for all k and we minimize some convex function of states and controls.

We construct the graph with $2+(K-1)|L|$ vertices shown in Figure 2. The source vertex s is associated with the region $X_s := \{\zeta(0)\} \times U$, the target t with $X_t := \bar{Z} \times U$. The remaining $(K-1)|L|$ vertices have the form $i = (k, l)$ for $k = 1, \dots, K-1$ and $l \in L$, and they encode the region Z_l in which the system might be at time k. For these vertices, the convex regions X_i are independent of time k and equal to $Z_l \times U$. The source s is connected by an edge to each vertex $(1, l)$ for $l \in L$ in the first layer. Similarly, the $|L|$ vertices $(K-1, l)$ are connected to the target t. For $k = 1, \dots, K-2$, we have an edge from vertex $i = (k, l)$ to vertex $j = (k+1, l')$ for all $(l, l') \in L^2$. This ensures that every state transition increases the time count by exactly one unit.

Since any s-t path in the graph we just constructed has exactly K edges, the time available to reach the target is fixed. For the remaining components of problem (1), the discussion from the previous subsection carries over without any modification.

When working with fixed-horizon problems, it is frequently useful to enforce a penalty on the magnitude of the terminal state $\zeta(K)$, e.g. $\zeta^\top(K) P \zeta(K)$ with P positive semidefinite. This is a fundamental tool to ensure closed-loop stability of the control system [57]. In our construction, this can be easily done via a proper definition of ℓ'_{ij} in (32) for $j = t$.

Remark 3. Overall, the size of the mixed integer program we construct scales linearly with the time horizon K and quadratically with the number $|L|$ of sets. Conversely, classical formulations for these problems [15] have size linear in both K and $|L|$. However, as we will see in Section IX-E, the higher strength of our approach is generally worth this price.

C. Piecewise-affine and hybrid systems

The analysis above extends immediately to PWA systems. This is a very broad class of hybrid dynamical systems: loosely speaking, almost any system whose nonlinearity is exclusively due to discrete logics can be written in PWA form [58]. A PWA system has the structure

$$\zeta(k+1) = A_l \zeta(k) + B_l u(k) + c_l, \quad \text{if } (\zeta(k), u(k)) \in D_l, \tag{33}$$

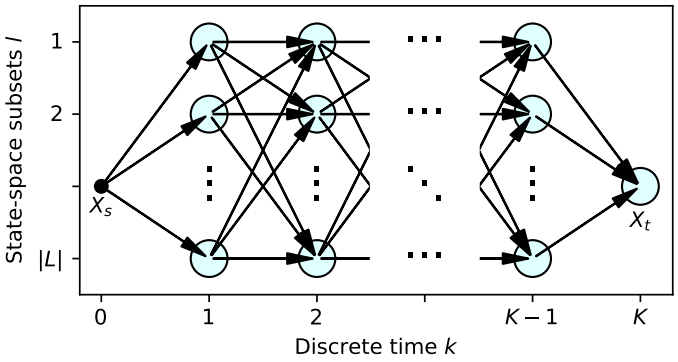


Fig. 2: Graph G for an optimal control problem with fixed time horizon K.

with $l \in L$. In words, we have a collection of $|L|$ affine dynamics, each of which applies in a different (convex compact) portion D_l of the state and control space.

The optimal control of PWA systems can be easily cast in shortest-path form either with free time horizon or, constructing a graph as in Figure 2, with fixed time horizon. The only difference lies in the definition of the edge length (32). Assume for simplicity that the initial mode l_s of the system is uniquely determined by $\zeta(0)$, i.e. l_s is the unique $l \in L$ such that $(\zeta(0), u) \in D_l$ for some u. To enforce the correct dynamics at time $k = 0$, we modify (32) requiring ℓ_{sj} to have finite value on the set $\zeta_j = A_{l_s}\zeta_s + B_{l_s}u_s + c_{l_s}$. Similarly, for $k = 1, \dots, T-1$, we let ℓ_{ij} have finite value if $\zeta_j = A_l \zeta_i + B_l u_i + c_l$, where $i = (k, l)$ and $j = (k+1, l')$.

IX. Examples

We collect in this section multiple numerical examples. The first two (Sections IX-A and IX-B) are toy examples meant to show the effects of the tightening constraints from Section V. Then we move to a more challenging but still relatively small example in Section IX-C. Section IX-D contains a statistical analysis of the performance of our formulation in case of large-scale random instances of the SPP. Finally, in Section IX-E, we demonstrate the applicability of our problem formulation in the context of optimal control of PWA systems.

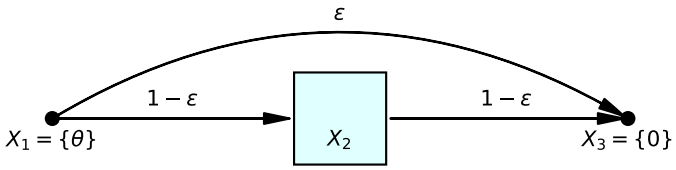
A. Effects of the spatial conservation of flow

We provide a simple example to demonstrate how the addition of (19) to the constraints of the MICP (13) can have a dramatic effect on the tightness of the convex relaxation.

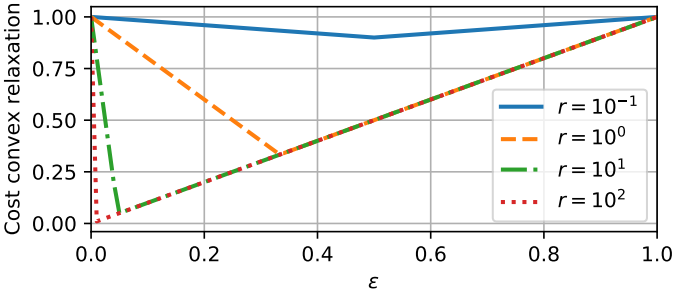
As shown in Figure 3a, we let $X_1 := \{\theta \in \mathbb{R}^d\}$ and $X_3 := \{0 \in \mathbb{R}^d\}$, while X_2 is a full-dimensional set, e.g. a cube, centered at $\theta/2$. We set $s := 1$ and $t := 3$. The edge set is $E := \{(1,2), (2,3), (1,3)\}$, and all the edges have 2-norm length function (2). We analyze the tightness of the convex relaxation of problem (13) as the cube X_2 grows in volume.

Independently of the volume of X_2, the shortest path has length $\|\theta\|_2$. This is achieved either by letting $\varphi_{13} = 1$ or, since X_2 intersects the line between X_1 and X_3, by letting $\varphi_{12} = \varphi_{23} = 1$ and, e.g., $x_2 = \theta/2$.

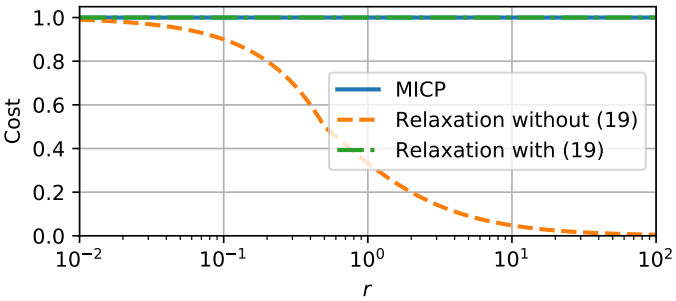
We start by showing that, in case constraint (19) is not enforced, the cost of the convex relaxation can be made



(a) Graph G and sets X_i. Feasible flows parameterized by the continuous variable $\varepsilon \in [0, 1]$.



(b) Cost of the convex relaxation of problem (13) as a function of the flow parameter ε for different sizes r of the set X_2. Constraint (19) is not enforced.



(c) Optimal cost of the MICP (13) and of its convex relaxation, without and with constraint (19), as functions of the size r of the set X_2.

Fig. 3: Numerical results from Section IX-A.

arbitrarily small by increasing the size of X_2. Note that, for the problem in Figure 3a, any flow that verifies the conservation law (13d) also verifies the degree constraint (20). Thus, constraint (20) is irrelevant in this analysis.

We parameterize the feasible flows in the graph with $\varphi_{13} = \varepsilon \in [0, 1]$ and, by the conservation law, we have $\varphi_{12} = \varphi_{23} = 1-\varepsilon$ (see Figure 3a). Constraints (13b) and (13c) allow to solve for a subset of the continuous variables: $y_{12} = (1-\varepsilon)\theta$, $y_{13} = \varepsilon\theta$, and $z_{13} = z_{23} = 0$. Using (14), the objective function of the convex relaxation can be written as

$$\sum_{(i,j)\in E} \|z_{ij} - y_{ij}\|_2 = \|z_{12} - (1-\varepsilon)\theta\|_2 + \|y_{23}\|_2 + \|\varepsilon\theta\|_2. \tag{34}$$

The remaining conditions from (13b) and (13c) are

$$z_{12} \in (1-\varepsilon)X_2, \qquad x_2 - z_{12} \in \varepsilon X_2, \tag{35a}$$
$$y_{23} \in (1-\varepsilon)X_2, \qquad x_2 - y_{23} \in \varepsilon X_2. \tag{35b}$$

The optimal cost of the convex relaxation is the minimum of (34) with respect to ε, x_2, z_{12}, y_{23} and subject to (35). This constrained minimization is difficult to solve by hand; nevertheless, for any fixed $\varepsilon \in (0, 1)$, we can assume the set X_2 to be large enough so that all the constraints in (35) are redundant. In this case, the unconstrained minimum of (34) with respect to the spatial variables is $\varepsilon\|\theta\|_2$ and is achieved for $z_{12} = (1-\varepsilon)\theta$ and $y_{23} = 0$. We conclude that, as the volume of the set X_2 goes to infinity, the optimal choice for the flow ε tends to zero and the cost of the convex relaxation approaches zero arbitrarily close.

In Figure 3 we report a two-dimensional numerical example where $\theta := (-1, 0)$, and the size of X_2 is parameterized via the scalar $r > 0$ as $X_2 := \{\theta/2 + x : \|x\|_\infty \leq r\}$. Figure 3b shows the optimal cost of the convex relaxation as a function of ε, for different sizes r of X_2. As predicted, for large sets X_2, the minimum cost is achieved for ε close to zero, and is sensibly smaller than the optimal cost $\|\theta\|_2 = 1$ of the MICP which is obtained for $\varepsilon \in \{0, 1\}$. The dependence of the cost of the convex relaxation on the size r of the set X_2 is depicted more neatly in Figure 3c.

We now add constraint (19) which, for vertex $i = 2$, reads $y_{23} = z_{12}$. Using the triangle inequality

$$\|z_{12} - (1-\varepsilon)\theta\|_2 + \|z_{12}\|_2 \geq \|(1-\varepsilon)\theta\|_2,$$

the cost in (34) is lower bounded by $\|\theta\|_2$. Therefore, with the addition of (19), the problem formulation has zero relaxation gap, independently of the volume of X_2. This is confirmed numerically in Figure 3c, where the curves of the MICP and the convex relaxation with (19) overlap for all values of r.

B. *Effects of the degree constraints*

We now describe the effects of the degree constraint (20) on the strength of the MICP (13). We assume that constraint (19) is already included in the problem formulation.

Unfortunately, the usefulness of (20) is only observed in graphs more involved than the one from Section IX-A. One example is depicted in Figure 4a. The sets $X_1 = X_s := \{\theta\}$ and $X_4 = X_t := \{0\}$ are singletons, while X_2 and X_3 are cubes centered at $2\theta/3$ and $\theta/3$, respectively. The edge set is

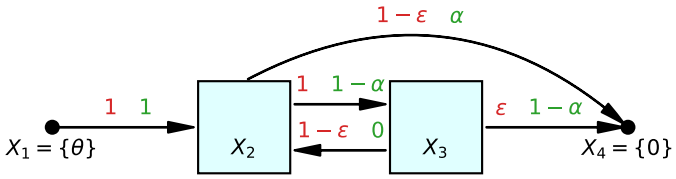
$$E := \{(1, 2), (2, 3), (2, 4), (3, 2), (3, 4)\}.$$

The edge-length function is the squared 2-norm (3). Similarly to the previous example, we analyze the cost of the convex relaxation as the sets X_2 and X_3 grow in volume.

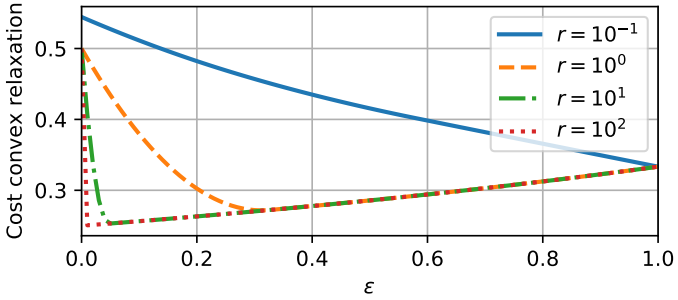
Independently of the size of the sets, the optimal flow is easily seen to be $\varphi_{12} = \varphi_{23} = \varphi_{34} = 1$ and $\varphi_{24} = \varphi_{32} = 0$, while the optimal continuous variables are $x_2 = 2\theta/3$ and $x_3 = \theta/3$. This choice corresponds to three steps of equal length and has cost $3\|\theta/3\|_2^2 = \|\theta\|_2^2/3$.

We first analyze the problem without constraint (20) showing that the resulting convex relaxation is very loose. To this end, we consider a specific flow (red values in Figure 4a): $\varphi_{12} = \varphi_{23} = 1$, $\varphi_{24} = \varphi_{32} = 1-\varepsilon$, $\varphi_{34} = \varepsilon$, with $\varepsilon \in (0, 1)$. From the constraints (13b) and (13c) we have

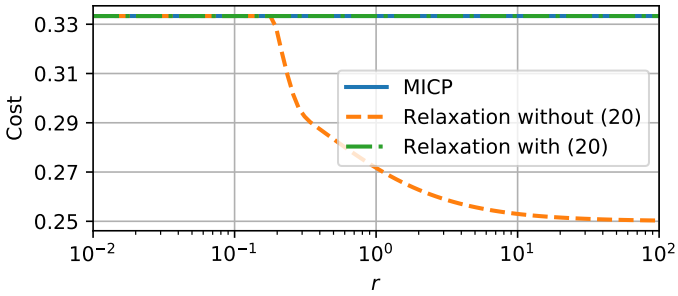
$$y_{12} = \theta, \quad y_{23} = z_{12} = x_2, \quad z_{23} = x_3, \quad z_{24} = z_{34} = 0.$$



(a) Graph G and sets X_i. The red flow, parameterized by ε, is used to show the looseness of the convex relaxation when constraint (20) is not enforced. The feasible flows after the addition of (20) are parameterized by α and written in green.



(b) Cost of the convex relaxation of problem (13) as a function of the flow parameter ε for different sizes r of the sets X_2 and X_3. Constraint (19) is enforced, constraint (20) is not.



(c) Optimal cost of the MICP (13) and of its convex relaxation, without and with constraint (20), as functions of the size r of the sets X_2 and X_3. Constraint (19) is enforced.

Fig. 4: Numerical results from Section IX-B.

Whereas, constraint (19) for $i = 2, 3$ yields $y_{24} = z_{32}$ and $y_{32} + y_{34} = x_3$. Using property (15), we can express the cost of this flow assignment as

$$\sum_{(i,j)\in E} \frac{\|z_{ij} - y_{ij}\|_2^2}{\varphi_{ij}} = \|x_2 - \theta\|_2^2 + \|x_3 - x_2\|_2^2 + \frac{\|y_{24}\|_2^2 + \|y_{24} - y_{32}\|_2^2}{1-\varepsilon} + \frac{\|x_3 - y_{32}\|_2^2}{\varepsilon}. \quad (36)$$

As in the previous example, for a fixed $\varepsilon \in (0, 1)$, assuming X_2 and X_3 to be sufficiently large, the value of the variables x_2, x_3, y_{24}, y_{32} can be picked arbitrarily. A straightforward, but a little tedious, unconstrained minimization of (36) with respect to these variables gives the optimal cost $\|\theta\|_2^2/(4-\varepsilon)$. Therefore, as X_2 and X_3 grow in volume, the optimal choice for ε converges to zero and the cost of this convex program decreases to $\|\theta\|_2^2/4$: a value significantly smaller than the optimal value $\|\theta\|_2^2/3$ of the MICP.

In Figure 4 we show the numerical results in case $\theta := (-1, 0)$, $X_2 := \{2\theta/3 + x : \|x\|_\infty \le r\}$, and $X_3 := \{\theta/3 + x : \|x\|_\infty \le r\}$. Figure 4b shows the cost of the convex relaxation as a function of ε for different sizes r of X_2 and X_3. Figure 4c reports the cost of the convex relaxation as a function of the size r of X_2 and X_3. As predicted, the minimum cost is achieved for ε close to zero and approaches $\|\theta\|_2^2/4 = 1/4$ asymptotically.

We add the degree constraints (20) to our problem formulation, and we show that this is sufficient to make the convex relaxation of the MICP tight. By the conservation of flow, the edge $(1, 2)$ must always be traversed by a unit flow, and (20) implies $\varphi_{32} = 0$. We parameterize all the remaining feasible flows with $\varphi_{24} = \alpha \in [0, 1]$ and, consequently, $\varphi_{23} = \varphi_{34} = 1 - \alpha$ (green values in Figure 4a). For $\alpha = 0$ we get the minimum of the MICP, while we know that the choice $\alpha = 1$ is suboptimal. We can then restrict our attention to the open interval $\alpha \in (0, 1)$. A subset of the constraints (13b) and (13c) yields

$$y_{12} = \theta, \quad z_{12} = x_2, \quad z_{24} = z_{34} = 0. \quad (37)$$

While, for $i = 2, 3$, condition (19) reads $y_{24} + y_{23} = x_2$ and $y_{34} = z_{23}$. The cost of this flow assignment is

$$\|x_2 - \theta\|_2^2 + \frac{\|y_{34} - y_{23}\|_2^2 + \|y_{34}\|_2^2}{1-\alpha} + \frac{\|y_{23} - x_2\|_2^2}{\alpha}. \quad (38)$$

This should be minimized subject to the constraints in (13b) and (13c) that have not already been considered in (37). On the other hand, the unconstrained minimum of (38) with respect to the continuous variables x_2, y_{23}, y_{34} is easily found to be $\|\theta\|_2^2/(3-\alpha)$. Since for $\alpha \in (0, 1)$ this is strictly greater than the optimal cost $\|\theta\|_2^2/3$ of the MICP, we conclude that $\alpha = 0$ is also the optimal solution of the convex relaxation. Thus, constraint (20) leads to an MICP with zero relaxation gap for the problem at hand, independently of the size of X_2 and X_3. This is confirmed in Figure 4c.

Remark 4. This example shows that, even when condition (19) is enforced, the degree constraint (20) can still improve the strength of our problem formulation. The example from Section IX-A shows that the inverse is also true: for that problem constraint (20) is always redundant, whereas (19) makes an arbitrarily loose formulation into a perfectly tight one.

C. A two-dimensional example

We now move to a more involved two-dimensional example: this is described in Figure 5 and has been carefully constructed to show that the convex relaxation of our formulation needs not to be tight in the regime where the sets X_i are large. We have a graph with $|V| = 9$ vertices and $|E| = 19$ edges. As before, the source $X_s = \{\theta_s\}$ and target $X_t = \{\theta_t\}$ sets are single points, while the remaining regions are full dimensional. The geometry of the sets X_i and the edge set E can be deduced from Figure 5.

We consider two edge-length functions: the 2-norm (2) and the squared 2-norm (3). The corresponding shortest paths are shown in Figure 5 as a blue dashed line and a red dash-dotted line, respectively. As expected, the first path is almost

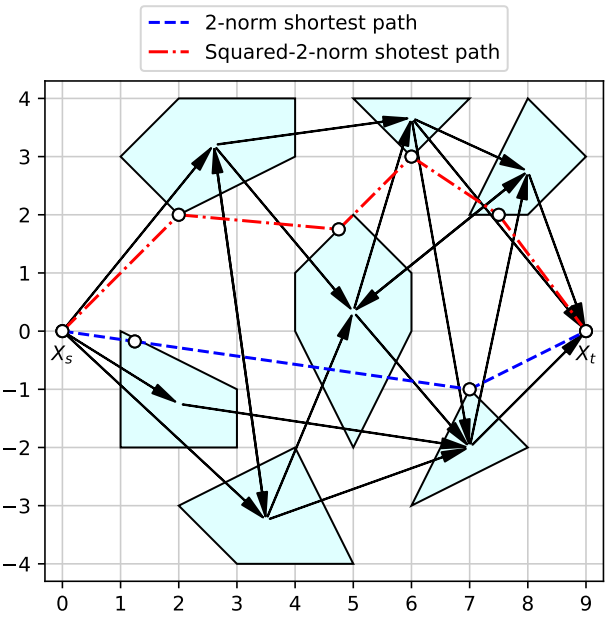
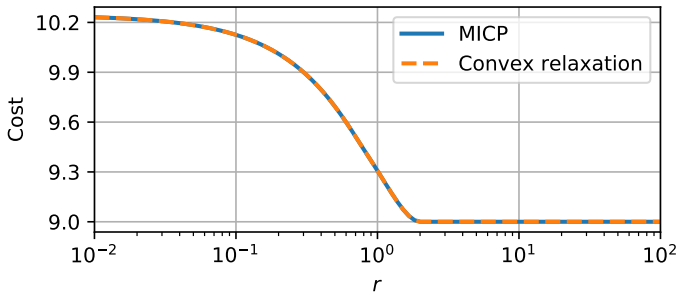
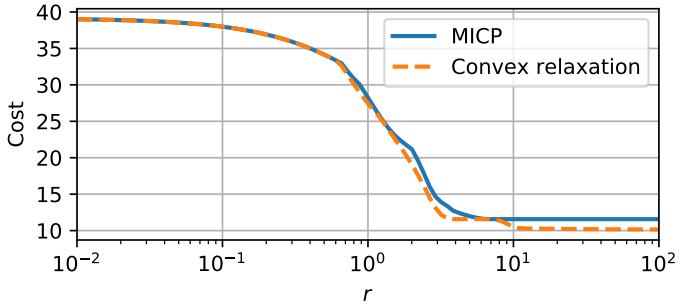


Fig. 5: Graph G and sets X_i for the example in Section IX-C. The blue dashed (red dash-dotted) line is the shortest path in case of 2-norm (squared-2-norm) edge-length function.



(a) 2-norm edge length (2).



(b) Squared-2-norm edge length (3). (This example is carefully built to make the convex relaxation loose in the regime of large r.)

Fig. 6: Numerical results from Section IX-C. Optimal cost of the MICP (13) and of its convex relaxation as functions of the size r of the sets X_i.

straight while the length of the segments in the second is better balanced.

In Figure 6 we compare the cost of the MICP (13) and of its convex relaxation, equipped with constraints (19) and (20). As in the previous examples, we analyze these values as functions of the size of the regions X_i. We parameterize the scale of these sets using the scalar r: we fix the position of the center (centroid of the vertices) of each polytope X_i, and we scale by r the position of each vertex relative to the center. The case depicted in Figure 5 is obtained for $r = 1$.

Figure 6a reports the results for the 2-norm edge length: despite the many regions and the presence of cycles, the convex relaxation is tight for all values of r.

For the edge length (3), Figure 6b shows that the convex relaxation is not always tight, even though the relaxation gap is small even in the worst case. In agreement with the results from Section IV-A, when the size of the sets converges to zero the convex relaxation of problem (13) becomes tight. A very small relaxation gap can be noticed for r approximately equal to one (as in Figure 5): this shows that our formulation is not necessarily tight when the sets X_i do not overlap. In the regime of large r, approximately $r \geq 10$, the convex relaxation becomes looser and, in agreement with the proof of Lemma 1, the asymptotic cost of the MICP equals $\|\theta_t - \theta_s\|_2^2/K = 11.6$, where $K = 7$ is the maximum number of edges traversed by an s-t path in Figure 5. Furthermore, a closer inspection of Figure 6b reveals that the curve of the convex relaxation converges to $\|\theta_t - \theta_s\|_2^2/(|V| - 1) = 10.1$, which corresponds to the lower bound derived in Proposition 5.

Recalling the discussion from Section II-A and noticing that the graph in Figure 5 is not Hamiltonian, it is to be expected that the relaxation of our MICP might be loose in the regime of large sets X_i. If our MICP was always tight in these scenarios, we would have a polynomial-time algorithm for the HPP, which is NP-complete. On the other hand, it is not even true that for these problem instances our convex relaxation always yields the trivial lower bound from Proposition 5. As an example, the removal of the edge connecting the top-left set to the bottom set in Figure 5 does not make the graph Hamiltonian, but it is sufficient to close the asymptotic relaxation gap in Figure 6b.

D. Large-scale random instances

So far we have only discussed simple examples with a handful of vertices and edges. Moreover, we have mostly focused on analyzing the strength of our formulation as a function of the size of the sets X_i. We now move to problems of larger scale, and we analyze the impact of multiple parameters on the efficiency of our formulation. We formulate a large number of random problem instances and we analyze the resulting solution statistics.

Generating random graphs representative of the "typical" SPP on convex sets we might encounter in practice is a difficult operation. Restrictions such as requiring the source s to be connected to all the vertices in the graph introduce strong biases in the topology of the graph. Inevitably, the instances we describe below are not completely representative and our algorithm might perform worse or better on different classes of random graphs. With the results reported in this section we do not want to make any claim regarding, e.g., the average strength of our formulation. Our purpose is instead to show that the applicability of our formulation is not limited to small-scale problems.

	Relaxation gap (%)				Convex-relaxation solve time (s)				MICP solve time (s)			
	2-norm		Squared 2-norm		2-norm		Squared 2-norm		2-norm		Squared 2-norm	
Increased parameters ($\times 5$)	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max
None (nominal)	0.0	0.6	0.3	11.6	0.04	0.08	0.04	0.07	0.2	0.5	0.2	0.8
Dimensions d	0.0	0.4	9.6	35.8	0.24	0.70	0.20	0.37	1.5	11.0	20.5	209.8
Edges $\lvert E\rvert$	0.0	0.5	15.3	31.2	0.53	1.02	0.47	0.63	4.5	19.5	53.5	184.2
Vertices $\lvert V\rvert$ and edges $\lvert E\rvert$	0.0	0.0	0.2	9.0	0.21	0.60	0.18	0.41	1.7	3.3	1.6	6.6
Volume Δ	0.0	0.1	1.1	10.1	0.05	0.15	0.04	0.08	0.2	1.0	0.4	1.1

TABLE I: Relaxation gap and computation times, in the average and worst case, for the random problem instances described in Section IX-D. In each row we increase part of the problem parameters by a factor of 5, we generate 100 random problem instances, and we report the statistics of their solution. The nominal value of the parameters is $d = 4$, $|E| = 100$, $|V| = 50$, and $\Delta = 0.01$. Two edge-length functions are considered: the 2-norm (2) and the squared 2-norm (3). These statistics show that our formulation can be used to tackle problems of significant size. However, given the random nature of these instances, these values might be unrepresentative of the average performance of our formulation.

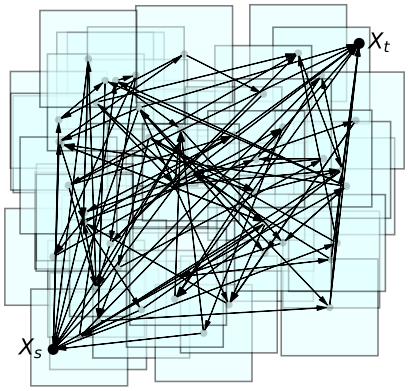


Fig. 7: Projection onto two dimensions of a random instance of the SPP from Section IX-D for a nominal value of the problem parameters.

We construct a random instance of problem (1) as follows. We set $X_s := \{0 \in \mathbb{R}^d\}$ and $X_t := \{1 \in \mathbb{R}^d\}$. Each of the remaining $|V| - 2$ sets X_i is a cube of volume Δ with center drawn from the uniform distribution over $[0, 1]^d$. For a given number of edges $|E|$, we construct the edge set E in two steps. First we generate multiple s-t paths such that each vertex in $V - \{s, t\}$ is traversed exactly by one path. The number of these paths is drawn uniformly from the interval $[1, |V| - 2]$ and the vertices they traverse are determined by generating a random partition of the set $V - \{s, t\}$: both the number of sets in the partition and their cardinality are uniform random variables. Finally, we expand the edge set E by drawing edges uniformly at random from the set $\{(i, j) \in V^2 : i \neq j\}$ until the desired cardinality $|E|$ is achieved. We use the following nominal parameters: $d = 4$ dimensions, $|E| = 100$ edges, $|V| = 50$ regions, and a volume $\Delta = 0.01$ for the regions X_i. To give an idea of what these problems look like, the projection onto two dimensions of a random instance generated with these parameters is shown in Figure 7.

We consider both the 2-norm (2) and the squared-2-norm (3) edge lengths. For each edge length, first we solve 100 random problem instances with nominal parameters. Then we consider four subgroups of the parameters: for each subgroup, we multiply the value of the parameters in it by 5, and we solve another 100 random instances. Table I shows the statistics of these trials. The first group of columns records the average and maximum relaxation gap (percentage difference between the cost of the MICP and of its convex relaxation). The second and third groups of columns record the average and worst-case solution times for the convex relaxation and the MICP.[3] In support of the analysis below, we recall that, including (19) and (20), these MICPs have $|E|$ binaries, $O(d(|V| + |E|))$ continuous variables, and $O(d|V| + |E|)$ constraints.

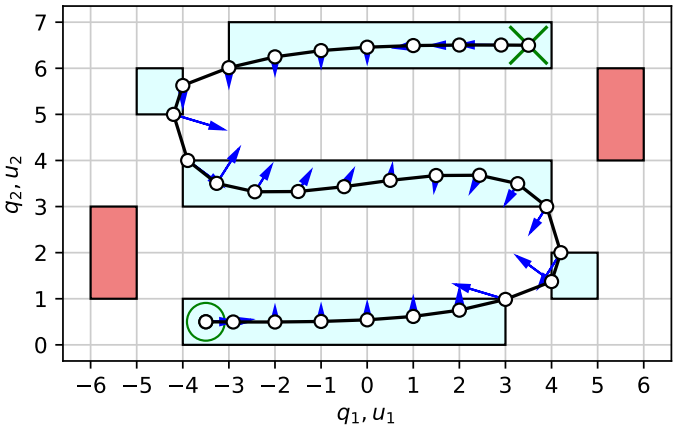
Overall, the 2-norm edge length (2) results in much simpler optimizations: the relaxation gaps never exceed 0.6% and computation times are relatively low.

The squared-2-norm length (3) leads to more challenging problems even though, in the nominal case, the average relaxation gap is still very low and the MICP computation times are always within 0.8 s. The growth of the space dimension to $d = 20$ increases the size of our programs, and also deteriorates the tightness of the convex relaxation. In the worst case, we have a relaxation gap of 35.8% and an MICP solution time greater than 3 min. A similar analysis applies when the number $|E|$ of edges is increased from 100 to 500: in general, we have found our MICP formulation to struggle with graphs of high density $|E|/|V|$ of edges. (Recall that $|E|$ equals the number of binaries in our MICP, while the bigger $|V|$ the more constrained the binaries are.) To show this, in the fourth row we keep 500 edges but we increase the vertices to $|V| = 250$. This has the effect of reducing the edge density and, even if the resulting MICPs are bigger than the ones from the previous case, the relaxation gap and the computation times are reduced. Finally, we increase the size of the cubes X_i from $\Delta = 0.01$ to $\Delta = 0.05$: these sets have now a total volume of $|V|\Delta = 2.5$, which is significantly larger than the unit cube containing them. Despite this, the performance of our formulation does not differ significantly from the nominal case. Notice that this is not in contradiction with the previous examples, where we were analyzing the regime of extremely large sets X_i. In addition, recall that the volume of the sets does not affect the size of these programs.

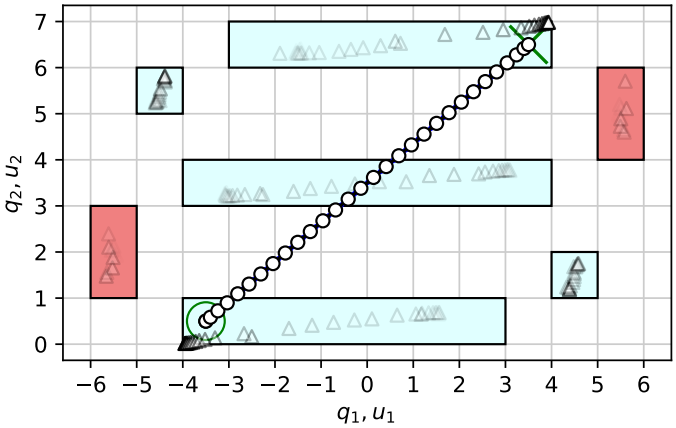
E. Optimal control of a piecewise-affine system

We conclude illustrating the discussion from Section VIII via the optimal-control problem shown in Figure 8. We consider a mechanical system with position $q \in \mathbb{R}^2$ and velocity

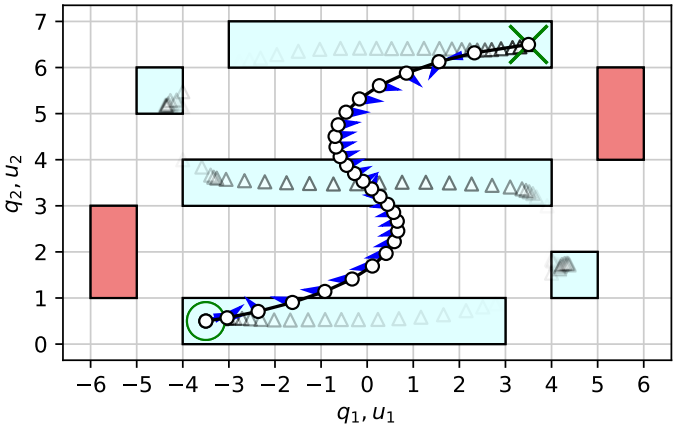
[3]Problems are solved with the commercial solver Mosek 9.0.96 with default options on a 2.4 GHz 8-Core Intel Core i9.



(a) Optimal solution of the MICP.



(b) Solution of the convex relaxation of the formulation [14], [15]. Relaxation gap is 93%: the corresponding MICP is solved in 175 s.



(c) Solution of the convex relaxation of the proposed formulation. Relaxation gap is 20%: the corresponding MICP is solved in 2.5 s.

Fig. 8: Control problem of driving a second-order system from start (green circle) to goal (green cross). In the light-blue regions the system is highly controllable ($\eta = 1$); in the red regions controllability is low ($\eta = 0.1$). Optimal positions $(q(k))_{k=0}^{K}$ are white circles; optimal controls $(u(k))_{k=0}^{K-1}$ are blue arrows. The triangles represent the auxiliary variables $q_l(k)$ whose convex combination yields $q(k)$. The opacity of the triangles equals the optimal value of the indicator variables $b_l(k)$ which serve as weights in this convex combination.

$v \in \mathbb{R}^2$. The force $u \in \mathbb{R}^2$ serves as control input. The system has the dynamics of a double integrator

$$q(k+1) = q(k) + v(k), \quad v(k+1) = v(k) + \eta u(k),$$

where η is a scalar parameter that regulates the controllability of the system. Representing the system state via $\zeta := (q, v)$, the transition matrices are

$$A := \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \eta & 0 \\ 0 & \eta \end{bmatrix}.$$

At time $k = 0$, the system is in position $q(0) := (-3.5, 0.5)$ (bottom-left green circle in Figure 8) with zero velocity $v(0)$. At each time step $k = 1, \dots, K-1$, the position vector $q(k)$ is allowed to be in one of the seven regions depicted in Figure 8, while the velocity and the controls are limited by the constraints $\|v(k)\|_\infty \leq 1$ and $\|u(k)\|_\infty \leq 1$. The goal is to reach the configuration $q(K) := (3.5, 6.5)$ (top-right green cross in Figure 8) with zero velocity $v(K)$ in $K := 30$ time steps. In doing this, we want minimize the quadratic form

$$\sum_{k=0}^{K-1} (\|v(k)\|_2^2/5 + \|u(k)\|_2^2).$$

We let the controllability parameter η vary between the regions. For the regions included in the range $-5 \leq q_1 \leq 5$ (light blue in Figure 8) we set $\eta = 1$, and the system is highly controllable. In the remaining two regions (red in Figure 8) we let $\eta = 0.1$, making it very expensive to apply any significant force. Since the matrix B varies with the state, the resulting system is PWA and the control problem falls into the class considered in Section VIII-C. The graph G beneath this problem (see Figure 2) has $|V| = 205$ vertices and $|E| = 1386$ edges, the sets X_i live in a space of $d = 6$ dimensions.

Figure 8a shows the optimal trajectory $(q(k))_{k=0}^{K}$ (white circles) with the optimal controls $(u(k))_{k=0}^{K-1}$ (blue arrows). Geometrically, the red regions would represent a shortcut towards the goal, but the reduced controllability of these areas would make it very expensive not to fall out of the feasible set. The optimal strategy is then to follow the S-shaped path and incur a cost of 9.37.

As a baseline for our method, we first solve this problem using the strongest MICP formulation available in the control literature: this also employs perspective functions and has been presented in [14] and further developed in [15, Section 5.2.2]. For each time step k, it uses $|L|$ indicator variables $b_l(k) \in \{0, 1\}$ to select which one of the affine dynamics in (33) should be applied. This is done by decomposing the state and the controls at time k into the convex combination of $|L|$ auxiliary variables $(\zeta_l(k), u_l(k)) \in D_l$:

$$(\zeta(k), u(k)) = \sum_{l \in L} b_l(k)(\zeta_l(k), u_l(k)), \tag{39}$$

where $\sum_{l \in L} b_l(k) = 1$.[4] Each copy $(\zeta_l(k), u_l(k))$ is used to predict the next state according to the lth dynamics, and the

[4]Equations (39) and (40) are convexified using the method from [59], [10].

actual state of the system at time $k+1$ is recovered as the convex combination of these predictions:

$$\zeta(k+1) = \sum_{l \in L} b_l(k)(A_l \zeta_l(k) + B_l u_l(k) + c_l). \tag{40}$$

When the binaries are relaxed, $b_l(k) \in [0,1]$, the system evolves according to a convex combination of the various affine dynamics.

Figure 8b shows the solution of the convex relaxation of the formulation [14], [15]. It reports the position $q(k)$, the (barely visible) controls $u(k)$, and the auxiliary copies $q_l(k)$ of the position vector whose convex combination yields $q(k)$. The latter have triangle markers with opacity equal to the value of the indicator $b_l(k)$. At each time step k, the solver is allowed to select the best convex combination of the $|L|$ affine dynamics: it decides to reach the goal with a perfectly-straight trajectory and incur a cost of 0.67, which is only 7% of the MICP cost (93% relaxation gap). Note that this behavior is completely insensitive to the geometry of the problem. The auxiliary variables are also uninformative: given the wide variety of convex combinations of the affine dynamics that yield a straight trajectory, the values of $q_l(k)$ and $b_l(k)$ make it impossible to guess in which region the system should be at a given time. The MICP resulting from this problem formulation required 175 s to be solved to global optimality (with the machine and solver reported in Section IX-D).

The convex relaxation of our formulation is much tighter: its optimal value is 7.46, which is 80% of the MICP cost (20% relaxation gap). This has a dramatic effect on computation times which are now reduced to 2.5 s. To generate a plot comparable with 8b we leverage the structure of the graph G in Figure 2. For each time step $k = 1, \dots, K-1$, this graph has vertices $i = (k, l)$ and the total flow

$$b_l(k) := \sum_{j \in O_i} \varphi_{ij}$$

traversing vertex i takes the role of the binary indicator from the formulation [14], [15].[5] In fact, at optimality of the MICP (13) we have $(\zeta(k), u(k)) \in D_l$ if and only if $b_l(k) = 1$, while the conservation of flow (13d) is easily seen to imply $\sum_{l \in L} b_l(k) = 1$. Recalling that the spatial variables x_i are obtained by stacking the state and the controls, and provided that $b_l(k) > 0$, the role of the auxiliary continuous variables from [14], [15] can be taken by

$$(\zeta_l(k), u_l(k)) := \sum_{j \in O_i} y_{ij}/b_l(k) \in D_l,$$

where the inclusion on the right holds even for nonbinary flows and follows directly from (13b). Reconstructing the system state and controls as in (39), we obtain

$$(\zeta(k), u(k)) = \sum_{i \in \{k\} \times L} \sum_{j \in O_i} y_{ij}.$$

[5]By the conservation of flow (13d) and its spatial version (19), we could equivalently have defined $b_l(k) := \sum_{j \in I_i} \varphi_{ji}$ and $(\zeta_l(k), u_l(k)) := \sum_{j \in I_i} z_{ji}/b_l(k)$.

The values just described are depicted in Figure 8c. The system trajectory $(q(k))_{k=0}^{K}$ and the controls $(u(k))_{k=0}^{K-1}$ obtained from the proposed convex program resemble the S-shaped optimal solution in Figure 8a much more closely than the formulation [14], [15]. For the auxiliary variables, we note that all the markers in the regions with low controllability are invisible, meaning that our convex relaxation identifies these as regions of high cost, and sets to zero the corresponding indicators $b_l(k)$. All the visible points $q_l(k)$ are clustered along the optimal trajectory of the MICP, suggesting that our convex relaxation contains detailed information on the optimal path to reach the goal.

X. Conclusions and Future Works

We have presented a generalization of the SPP in which the position of each vertex in the graph is a continuous decision variable lying in a convex set, and the length of an edge is a convex function of the position of the vertices it connects. Our main contribution is a strong mixed-integer convex formulation for the solution of this NP-hard problem. A wide variety of numerical tests show that the convex relaxation of this MICP is often very tight. We have focused part of our attention on control systems: many mixed-integer control problems turn out to be interpretable as generalized SPPs and, in our tests, the proposed MICP outperforms state-of-the-art techniques for their solution.

One of our future goals is the development of approximation algorithms for this generalized SPP: the similarities between our MICP and the LP formulation of the classical SPP might allow us to leverage a massive literature to progress in this direction. In terms of applications, we plan to conduct a thorough test of the performance of the proposed method in the context of robot motion planning. Finally, we underline that the core techniques from this paper are not bound to the SPP and can be extended to many other classical problems in combinatorics; the bipartite-matching and the traveling-salesman problems being two examples.

Appendix A
Perspective of Commonly-Used Functions

In this appendix we derive the perspective functions employed in this paper. Below we assume $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}$.

A. Constants

Consider the constant function $f(x) := b$. Using Definition 1, for $\lambda \geq 0$, the perspective of f is given by

$$\tilde{f}(\lambda, x) = \begin{cases} \lambda b & \text{if } \lambda > 0 \\ \lim_{\tau \downarrow 0} \tau b & \text{if } \lambda = 0 \end{cases} = b\lambda.$$

B. Positively homogeneous functions

Let $f(x)$ be positively homogeneous, i.e., $f(ax) = af(x)$ for all x and $a \geq 0$. For example, linear functions $f(x) := Ax$ and norms $f(x) := \|x\|$ enjoy this property. Any positively homogeneous function must verify $f(0) < \infty$, in fact, if not, we would have the contradiction $\infty = f(0) = 0f(0) = 0 \cdot \infty$. Letting $\bar{x} = 0$ in Definition 1, it is immediately verified that $\tilde{f}(\lambda, x) = f(x)$ for all $\lambda \geq 0$.

C. Affine functions

Assume $f(x) := Ax + b$. We notice that the perspective of the sum of two real-valued functions is the sum of their perspectives. For $\lambda \geq 0$, this implies $\tilde{f}(\lambda, x) = Ax + b\lambda$.

D. Norms

Consider a function of the form $f(x) := \|Ax + b\| + c$. Exploiting the positive homogeneity of the norm, the perspective of $\|Ax + b\|$ is immediately verified to be $\|Ax + b\lambda\|$. Therefore, we have $\tilde{f}(\lambda, x) = \|Ax + b\lambda\| + c\lambda$.

E. Positive semidefinite quadratic forms

Assume $f(x) := \|Ax\|_2^2$. For $\lambda > 0$, we have $\tilde{f}(\lambda, x) = \|Ax\|_2^2/\lambda$. For $\lambda = 0$ and $\bar{x} = 0$, we obtain

$$\tilde{f}(0, x) = \lim_{\tau \downarrow 0} \tau \|Ax/\tau\|_2^2 = \begin{cases} 0 & \text{if } Ax = 0 \\ \infty & \text{otherwise} \end{cases}.$$

F. Functions with extended-real values

Consider

$$f(x) := \begin{cases} f'(x) & \text{if } x \in D \\ \infty & \text{otherwise} \end{cases},$$

where f' and D are closed and convex. Assume $\bar{x}$ to be such that $f(\bar{x}) < \infty$. For $\lambda > 0$, we have

$$\begin{aligned} \tilde{f}(\lambda, x) &= \lambda \begin{cases} f'(x/\lambda) & \text{if } x/\lambda \in D \\ \infty & \text{otherwise} \end{cases} \\ &= \begin{cases} \tilde{f}'(\lambda, x) & \text{if } x \in \lambda D \\ \infty & \text{otherwise} \end{cases}. \end{aligned}$$

For $\lambda = 0$, we have

$$\begin{aligned} \tilde{f}(0, x) &= \lim_{\tau \downarrow 0} \tau \begin{cases} f'(\bar{x} + x/\tau) & \text{if } \bar{x} + x/\tau \in D \\ \infty & \text{otherwise} \end{cases} \\ &= \begin{cases} \tilde{f}'(0, x) & \text{if } x \in D^\infty \\ \infty & \text{otherwise} \end{cases}, \end{aligned}$$

where D^∞ denotes the recession cone of D. Denoting with the symbol $+$ the Minkowski sum between two sets, and noticing that $\lambda D + D^\infty = \lambda D$ for $\lambda > 0$, we conclude that

$$\tilde{f}(\lambda, x) = \begin{cases} \tilde{f}'(\lambda, x) & \text{if } x \in \lambda D + D^\infty \\ \infty & \text{otherwise} \end{cases}.$$

ACKNOWLEDGMENT

The authors would like to thank Hongkai Dai for all the time spent improving the optimization-solver interface employed in the numerical examples presented in this paper.

This research was supported by the National Science Foundation, Award No. EFMA-1830901, and by the Department of the Navy, Office of Naval Research, Award No. N00014-18-1-2210. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

REFERENCES

[1] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*. Springer Science & Business Media, 2003, vol. 24.

[2] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 3. IEEE, 2002, pp. 1936–1941.

[3] A. Richards and J. How, "Mixed-integer programming for control," in *Proceedings of the 2005, American Control Conference, 2005.* IEEE, 2005, pp. 2676–2683.

[4] M. Vitus, V. Pradeep, G. Hoffmann, S. Waslander, and C. Tomlin, "Tunnel-MILP: Path planning with sequential convex polytopes," in *AIAA guidance, navigation and control conference and exhibit*, 2008, p. 7132.

[5] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS international conference on humanoid robots*. IEEE, 2014, pp. 279–286.

[6] ——, "Efficient mixed-integer planning for UAVs in cluttered environments," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 42–49.

[7] B. Landry, R. Deits, P. R. Florence, and R. Tedrake, "Aggressive quadrotor flight through cluttered environments using mixed integer programming," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1469–1475.

[8] M. R. Maximo and R. J. Afonso, "Mixed-integer quadratic programming for automatic walking footstep placement, duration, and rotation," *Optimal Control Applications and Methods*, 2020.

[9] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[10] S. Ceria and J. Soares, "Convex programming for disjunctive convex optimization," *Mathematical Programming*, vol. 86, no. 3, pp. 595–614, 1999.

[11] A. Frangioni and C. Gentile, "Perspective cuts for a class of convex 0–1 mixed integer programs," *Mathematical Programming*, vol. 106, no. 2, pp. 225–236, 2006.

[12] O. Günlük and J. Linderoth, "Perspective reformulations of mixed integer nonlinear programs with indicator variables," *Mathematical programming*, vol. 124, no. 1-2, pp. 183–205, 2010.

[13] ——, "Perspective reformulation and applications," in *Mixed Integer Nonlinear Programming*. Springer, 2012, pp. 61–89.

[14] N. Moehle and S. Boyd, "A perspective-based convex relaxation for switched-affine optimal control," *Systems & Control Letters*, vol. 86, pp. 34–40, 2015.

[15] T. Marcucci and R. Tedrake, "Mixed-integer formulations for optimal control of piecewise-affine systems," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 230–239.

[16] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge University Press, 2011.

[17] N. Deo and C.-Y. Pang, "Shortest-path algorithms: Taxonomy and annotation," *Networks*, vol. 14, no. 2, pp. 275–323, 1984.

[18] A. Frieze, "Minimum paths in directed graphs," *Journal of the Operational Research Society*, vol. 28, no. 2, pp. 339–346, 1977.

[19] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[20] D. E. Knuth, "A generalization of Dijkstra's algorithm," *Information Processing Letters*, vol. 6, no. 1, pp. 1–5, 1977.

[21] G. Ramalingam and T. Reps, "An incremental algorithm for a generalization of the shortest-path problem," *Journal of Algorithms*, vol. 21, no. 2, pp. 267–305, 1996.

[22] L. Boguchwal, "Shortest path algorithms for functional environments," *Discrete Optimization*, vol. 18, pp. 217–251, 2015.

[23] F. Li and R. Klette, *Euclidean shortest paths*. Springer, 2011.

[24] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[25] D.-T. Lee and F. P. Preparata, "Euclidean shortest paths in the presence of rectilinear barriers," *Networks*, vol. 14, no. 3, pp. 393–410, 1984.

[26] J. Canny and J. Reif, "New lower bound techniques for robot motion planning problems," in *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*. IEEE, 1987, pp. 49–60.

[27] C. H. Papadimitriou, "An algorithm for shortest-path motion in three dimensions," *Information processing letters*, vol. 20, no. 5, pp. 259–263, 1985.

[28] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1528–1538, 1995.

[29] J. Kim and J. P. Hespanha, "Discrete approximations to continuous shortest-path: Application to minimum-risk path planning for groups of UAVs," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 2. IEEE, 2003, pp. 1734–1740.

[30] A. Deshpande, "Exact geometry algorithms for robotic motion planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2019.

[31] M. Dror, A. Efrat, A. Lubiw, and J. S. Mitchell, "Touring a sequence of polygons," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003, pp. 473–482.

[32] S. Ntafos, "Watchman routes under limited visibility," *Computational Geometry*, vol. 1, no. 3, pp. 149–170, 1992.

[33] C. Wei-Pang and S. Ntafos, "The zookeeper route problem," *Information Sciences*, vol. 63, no. 3, pp. 245–259, 1992.

[34] W.-p. Chin and S. Ntafos, "Optimum watchman routes," in *Proceedings of the second annual symposium on Computational geometry*, 1986, pp. 24–33.

[35] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM (JACM)*, vol. 34, no. 3, pp. 596–615, 1987.

[36] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[37] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[38] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms I: Fundamentals*. Springer science & business media, 2013, vol. 305.

[39] L. Taccari, "Integer programming formulations for the elementary shortest path problem," *European Journal of Operational Research*, vol. 252, no. 1, pp. 122–130, 2016.

[40] P. L. Combettes, "Perspective functions: Properties, constructions, and examples," *Set-Valued and Variational Analysis*, vol. 26, no. 2, pp. 247–264, 2018.

[41] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic foundations of robotics XI*. Springer, 2015, pp. 109–124.

[42] D. Ioan, I. Prodan, S. Olaru, F. Stoican, and S.-I. Niculescu, "Mixed-integer programming in motion planning," *Annual Reviews in Control*, 2020.

[43] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on automatic control*, vol. 26, no. 2, pp. 346–358, 1981.

[44] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, "An MPC/hybrid system approach to traction control," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 541–552, 2006.

[45] T. Geyer, G. Papafotiou, and M. Morari, "Hybrid model predictive control of the step-down DC–DC converter," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1112–1124, 2008.

[46] T. Marcucci, R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake, "Approximate hybrid model predictive control for multi-contact push recovery in complex environments," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 31–38.

[47] W. Han and R. Tedrake, "Feedback design for multi-contact push recovery via LMI approximation of the piecewise-affine quadratic regulator," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 842–849.

[48] E. F. Camacho, D. R. Ramírez, D. Limón, D. M. De La Peña, and T. Alamo, "Model predictive control techniques for hybrid systems," *Annual reviews in control*, vol. 34, no. 1, pp. 21–31, 2010.

[49] V. V. Naik and A. Bemporad, "Embedded mixed-integer quadratic optimization using accelerated dual gradient projection," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 723–10 728, 2017.

[50] B. Stellato, V. V. Naik, A. Bemporad, P. Goulart, and S. Boyd, "Embedded mixed-integer quadratic optimization using the OSQP solver," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 1536–1541.

[51] P. Hespanhol, R. Quirynen, and S. Di Cairano, "A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2763–2768.

[52] T. Marcucci and R. Tedrake, "Warm start of mixed-integer programs for model predictive control of hybrid systems," *IEEE Transactions on Automatic Control*, 2020.

[53] J. Liang, S. Di Cairano, and R. Quirynen, "Early termination of convex QP solvers in mixed-integer programming for real-time decision making," *IEEE Control Systems Letters*, 2020.

[54] H. Michalska and D. Q. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on Automatic control*, vol. 38, no. 11, pp. 1623–1633, 1993.

[55] P. O. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions on Automatic control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[56] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 7, pp. 333–351, 2006.

[57] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[58] W. P. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.

[59] E. Balas, "Disjunctive programming: Properties of the convex hull of feasible points," *Discrete Applied Mathematics*, vol. 89, no. 1-3, pp. 3–44, 1998.