

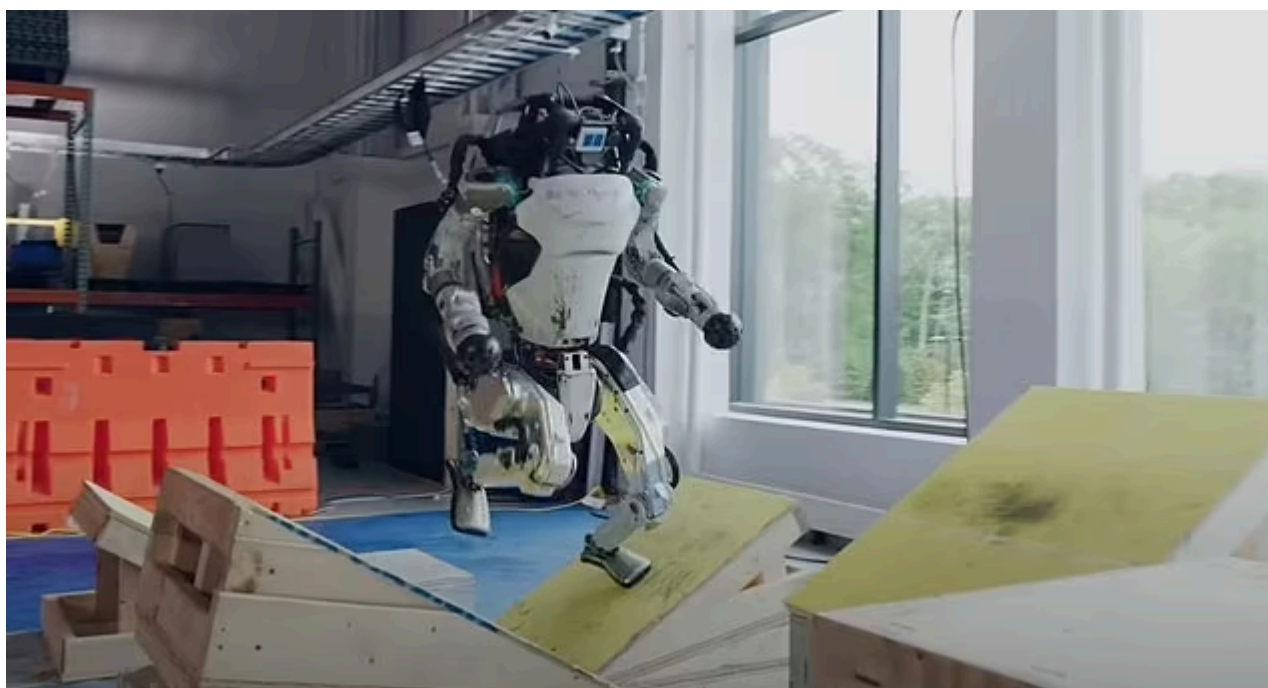
# GraphsOfConvexSets.jl



*C. Van Hees and B. Legat*

# Motivation

---

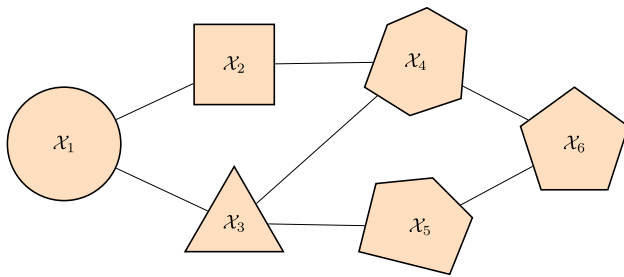


© Boston Dynamics



© Durbuy maze

# Graph of Convex Sets



On the **vertices**:

1. Convex set:

$$\mathbf{x}_i \in \mathcal{X}_i, \quad i = 1, \dots, 6$$

2. Convex objective

On the **edges**:

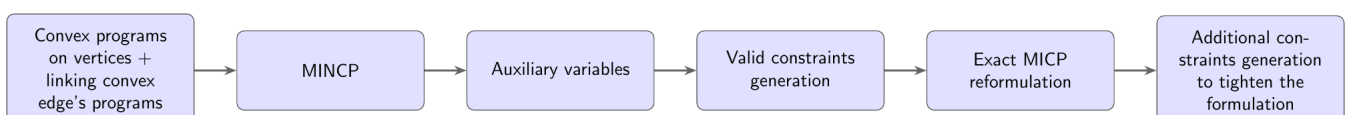
1. Convex constraints on the variables of its endpoints
2. Convex objective

## Combining graph problems and convex optimization

$$\begin{aligned} \min \quad & \sum_{v \in \mathcal{W}} f_v(\mathbf{x}_v) + \sum_{e=[v,w] \in \mathcal{F}} f_e(\mathbf{x}_v, \mathbf{x}_w) \\ \text{s.t.} \quad & H = (\mathcal{W}, \mathcal{F}) \in \mathcal{H}, \\ & \mathbf{x}_v \in \mathcal{X}_v, & v \in \mathcal{W}, \\ & (\mathbf{x}_v, \mathbf{x}_w) \in \mathcal{X}_e, & e = [v, w] \in \mathcal{F}. \end{aligned}$$

Assumptions:

1. Bounded sets
2. Linear objectives



# Homogenization

$$\min \sum_{v \in \mathcal{V}} y_v f_v(x_v) + \sum_{e=[v,w] \in \mathcal{E}} y_e f_e(x_v, x_w)$$

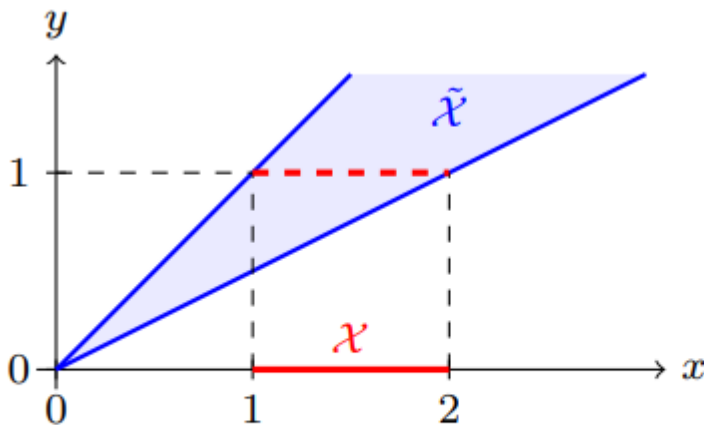
$$\text{s.t. } y \in \mathcal{Y}_{bin},$$

$$y_v(x_v, 1) \in \tilde{\mathcal{X}}_v,$$

$$y_e(x_v, x_w, 1) \in \tilde{\mathcal{X}}_e,$$

$$v \in \mathcal{V},$$

$$e = [v, w] \in \mathcal{E}.$$



Homogenization  $\tilde{\mathcal{X}}$  of the closed interval  $\mathcal{X} = [1, 2]$ .

© T. Marcucci

## Valid constraints generation

If the following linear inequality is valid for  $\mathcal{Y}_{bin}$  at some vertex  $v \in \mathcal{V}$ :

$$ay_v + \sum_{e \in \delta_v} b_e y_e \geq 0;$$

then the following convex constraint is valid for the MINCP:

$$a(z_v, y_v) + \sum_{e \in \delta_v} b_e(z_v^e, y_e) \in \tilde{\mathcal{X}}_v.$$

Valid constraints  
on the ILP



Valid constraints  
on the MINCP

# MICP reformulation

---

$$\min \sum_{v \in \mathcal{V}} c_v^T z_v + \sum_{e=[v,w] \in \mathcal{E}} c_e^T (z_v^e, z_w^e)$$

$$\text{s.t. } y \in \mathcal{Y}_{bin},$$

$$(z_v^e, y_e) \in \tilde{\mathcal{X}}_v,$$

$$v \in \mathcal{V}, e \in \delta_v,$$

$$(z_v - z_v^e, y_v - y_e) \in \tilde{\mathcal{X}}_v,$$

$$v \in \mathcal{V}, e \in \delta_v,$$

$$(z_v^e, z_w^e, y_e) \in \tilde{\mathcal{X}}_e,$$

$$e = [v, w] \in \mathcal{E}.$$

# GraphsOfConvexSets.jl

---

```
struct GraphModel <: Graphs.AbstractGraph
    graph <: Graphs.AbstractGraph
    model <: JuMP.AbstractModel
end
```

```
g = GraphModel({0, 0} directed simple Int64 graph, A JuMP Model, solver: GCS with Paiarito)
```

## Vertices' convex sets

---

```
1 # Define the vertices
2 Graphs.add_vertices!(g, 5);
```

```
1 # Define variables on the vertices
2 begin
3     x = [@variable(GCS.Vertex(g,v), [1:2]) for v in Graphs.vertices(g)]
4
5     ## OR
6
7     @variable(GCS.Vertices(g, 1:5), y[1:2])
8     y = collect.(eachrow(stack(y)))
9 end;
```

```
Vector{JuMP.VariableRef}[[-[1], -[2]], [-[3], -[4]], [-[5], -[6]], [-[7], -[8]], [-[9], -[10]]]
```

```
1 # Add constraints on the vertices
2 begin
3     @constraint(GCS.Vertex(g, 5), [1; x[5][:] - c[5, :]] in SecondOrderCone());
4
5     ## OR
6
7     c(x) = JuMP.build_constraint(error, [1; x], SecondOrderCone())
8     JuMP.add_constraint.(GCS.Vertices(g, 2:4), c.(y[2:4]))
9 end;
```

# Edges' convex sets

---

```
1 # Define the edges
2 begin
3     edges = [(1, 3), (1, 4), (3, 4), (3, 5), (4, 5), (4, 2), (5, 2)]
4     for (u,v) in edges
5         Graphs.add_edge!(g, u, v)
6     end
7 end
```

```
1 # Define variables on the edges
2 edge_cost_1_3 = @variable(GCS.Edge(g, 1, 3)); # precise src -> dst
```

```
1 # Define constraints on the edges
2 @constraint(GCS.Edge(g, 1, 3), [edge_cost_1_3; x[:,3] - x[:,1]] in
SecondOrderCone());
3 # This corresponds to cost >= ||x_dest - x_src||
```

```
1 # Set objective on the edges
2 JuMP.set_objective_function(GCS.Edge(g, 1, 3), edge_cost_1_3);
```

# Extract the solution

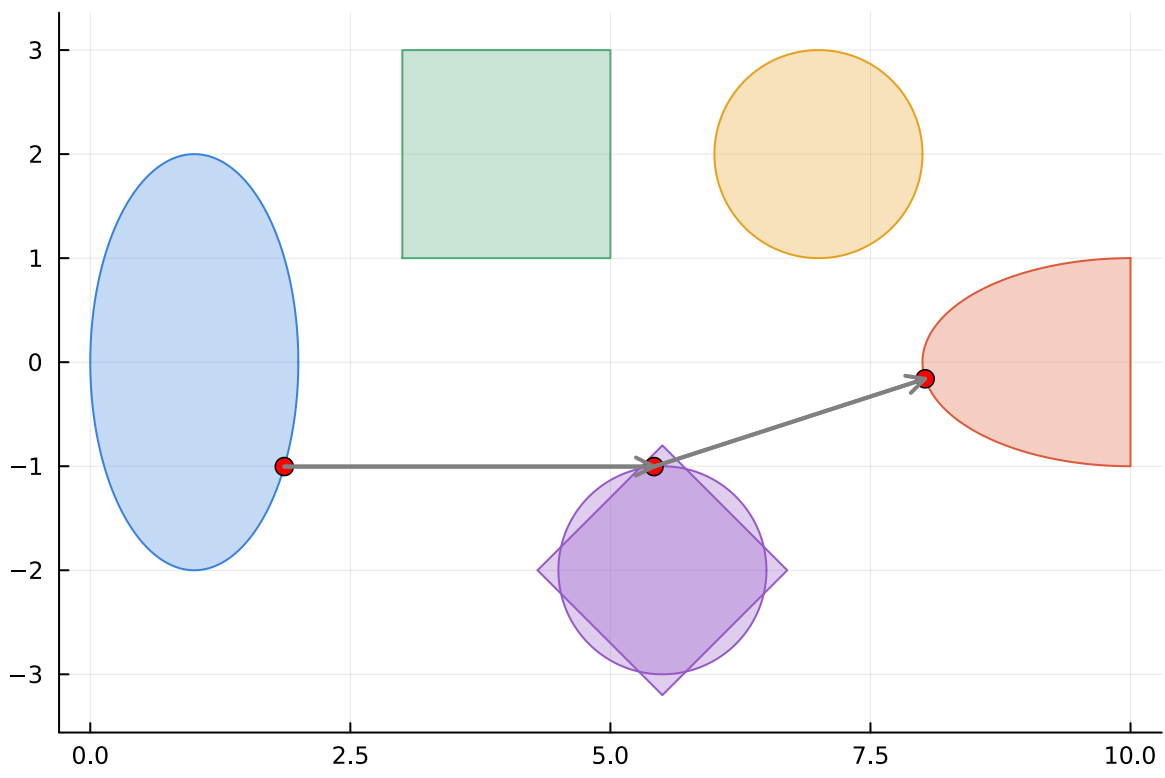
---

```
1 # Compute the shortest path between vertices 1 and 2
2 GCS.shortest_path(g, 1, 2);
```

`sol = {5, 2}` directed simple Int64 graph

```
1 # Get the optimal subgraph
2 sol = MOI.get(g.model, GCS.SubGraph());
```

```
1 # Get the optimal values of the continuous variables
2 xv = value.(x);
```



# More information

---

- [Mar25] T. Marcucci. A Unified and Scalable Method for Optimization over Graphs of Convex Sets. October 2025.
- <https://github.com/CharlesVanHees/GraphsOfConvexSets.jl>

## Bibliography

- [LDD23] M. Lubin, O. Downson, J. Dias Garcia, J. Huchette, B. Legat and J.P. Vielma. {JuMP} 1.0: Recent improvements to a modeling language for mathematical optimization. In *Mathematical Programming Computation*, 2023.
- [FBS21] J. Fairbanks, M. Besançon, S. Simon, J. Hoffiman, N. Eubank and S. Karpinski. JuliaGraphs/Graphs.jl: an optimized graphs package for the Julia programming language. 2021.