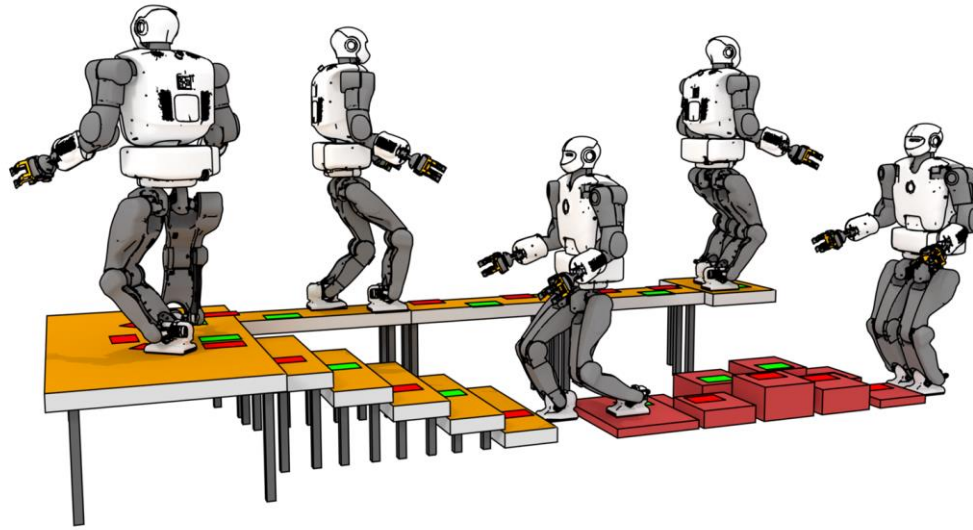


The "contact planning problem" for legged robots: A cardinality minimisation approach

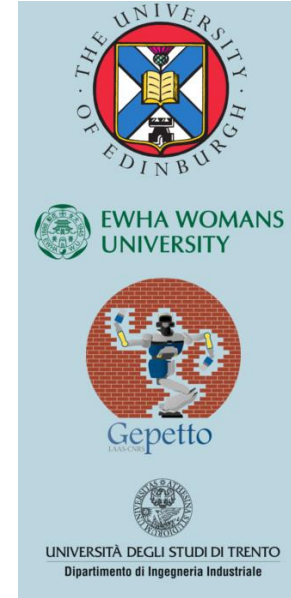


Collaborators for this work

Daeun Song and Young J. Kim
Ewha Womens University

Pierre Fernbach And Nicolas Mansard
LAAS-CNRS

Andrea Del Prete
Trento University

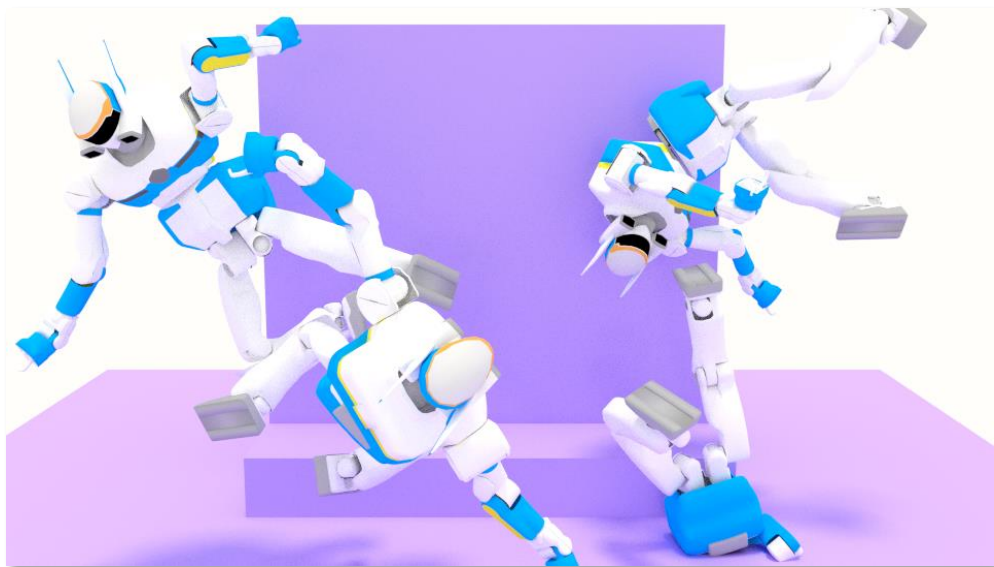


Research goal: autonomous legged locomotion

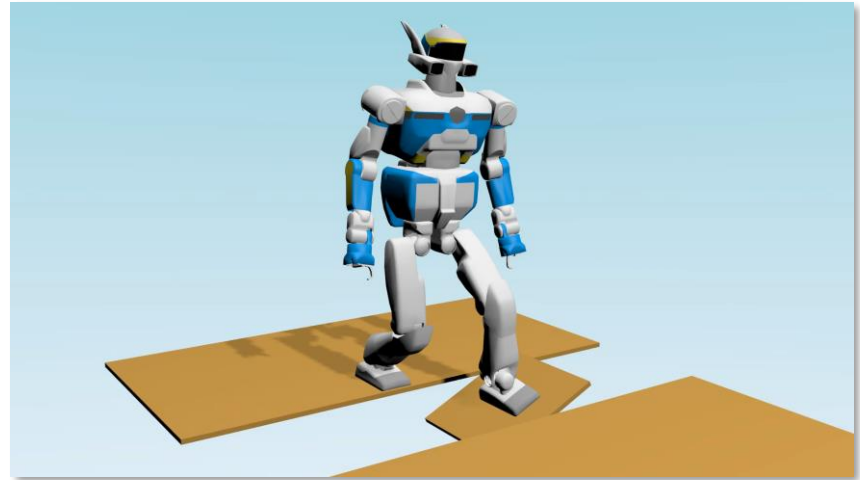
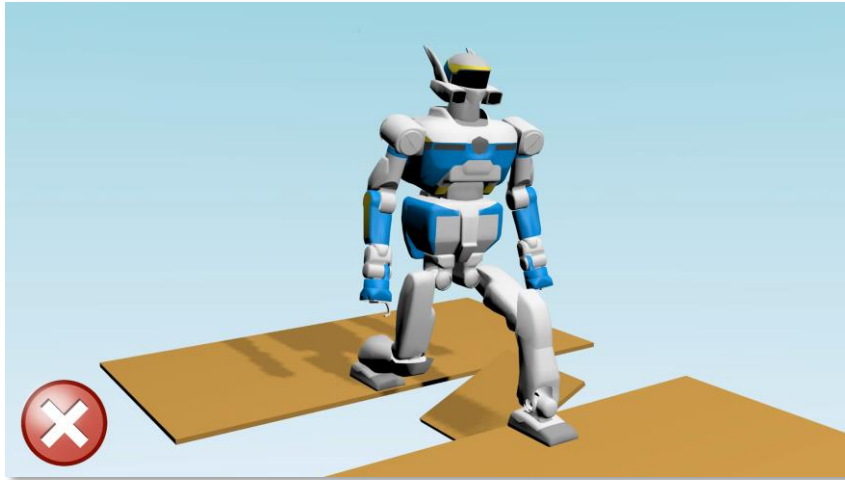


Carpentier, Tonneau, Naveau, Stasse et Mansard, ICRA 16

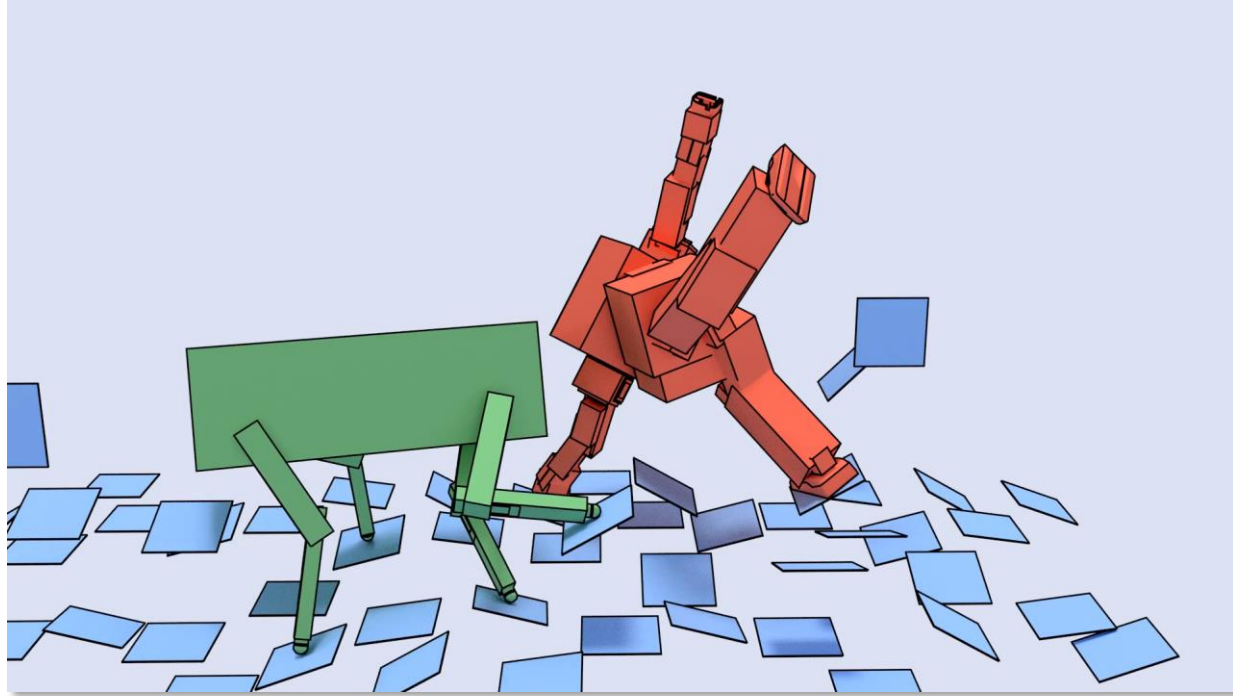
Contact postures in high-dimensional space ?



Contact-dependent, discontinuous, non-linear dynamics / geometric constraints



Contact interactions without collisions ?



Legged locomotion is too hard

We need to cheat !

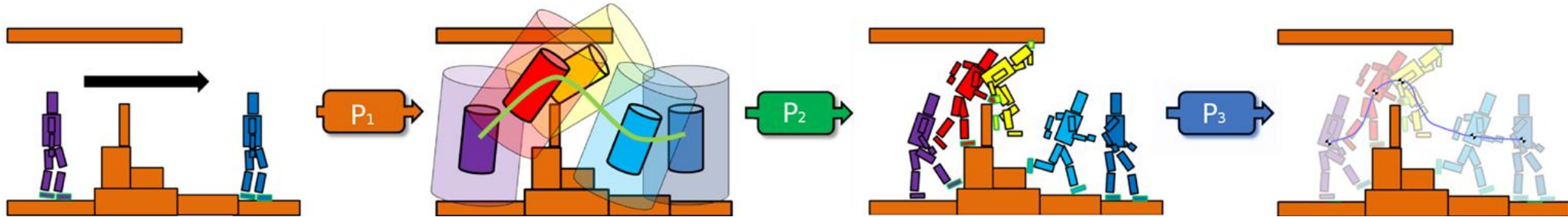
This has a cost ... not discussed today

A divide and conquer approach [Tonneau et al. 15]

Global path planner

Contact
Planner

Whole body motion
generator

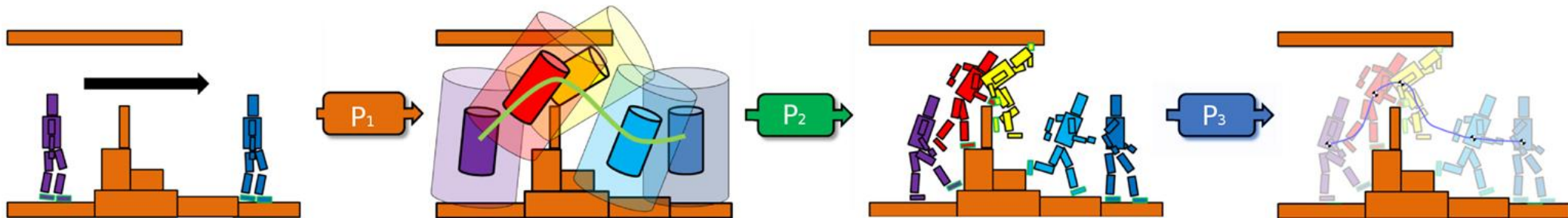


A divide and conquer approach [Tonneau et al. 15]

Global path planner

Contact
Planner

Whole body motion
generator



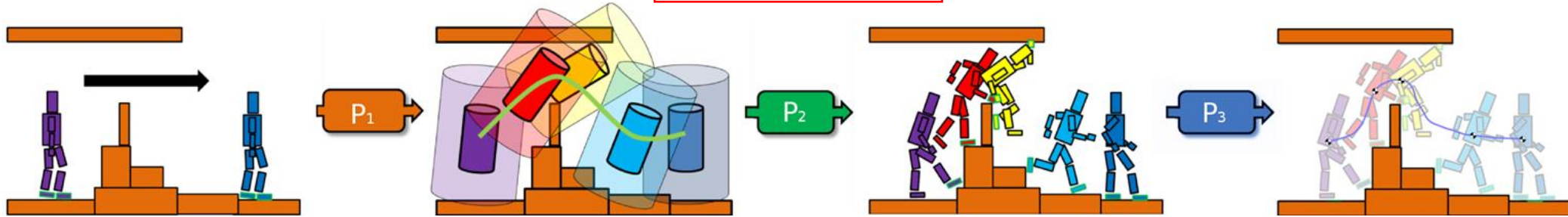
Solving $\mathcal{P}_i$ in the feasibility domain of $\mathcal{P}_j, j > i$?

A divide and conquer approach [Tonneau et al. 15]

Global path planner

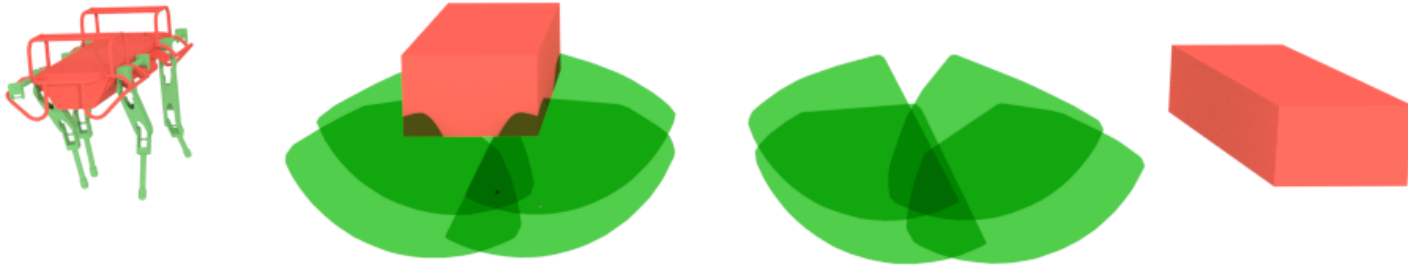
Contact
Planner

Whole body motion
generator



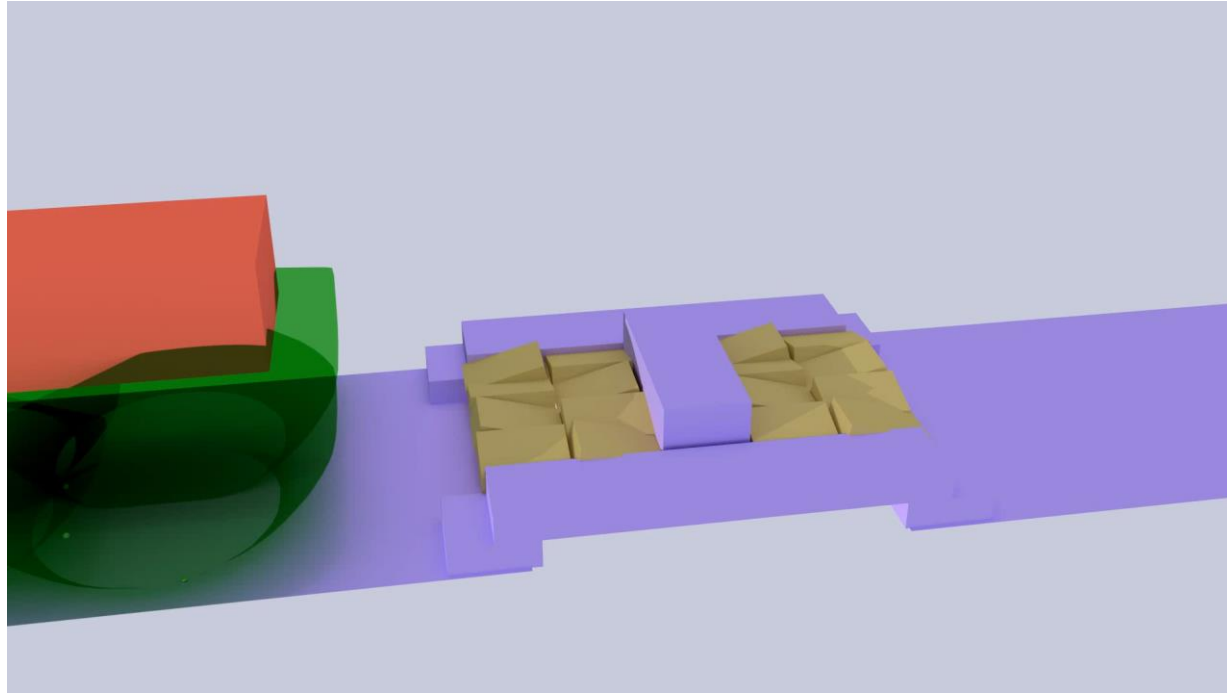
Solving $\mathcal{P}_i$ in the feasibility domain of $\mathcal{P}_j, j > i$?

Cheating assumption: low-dimensional space



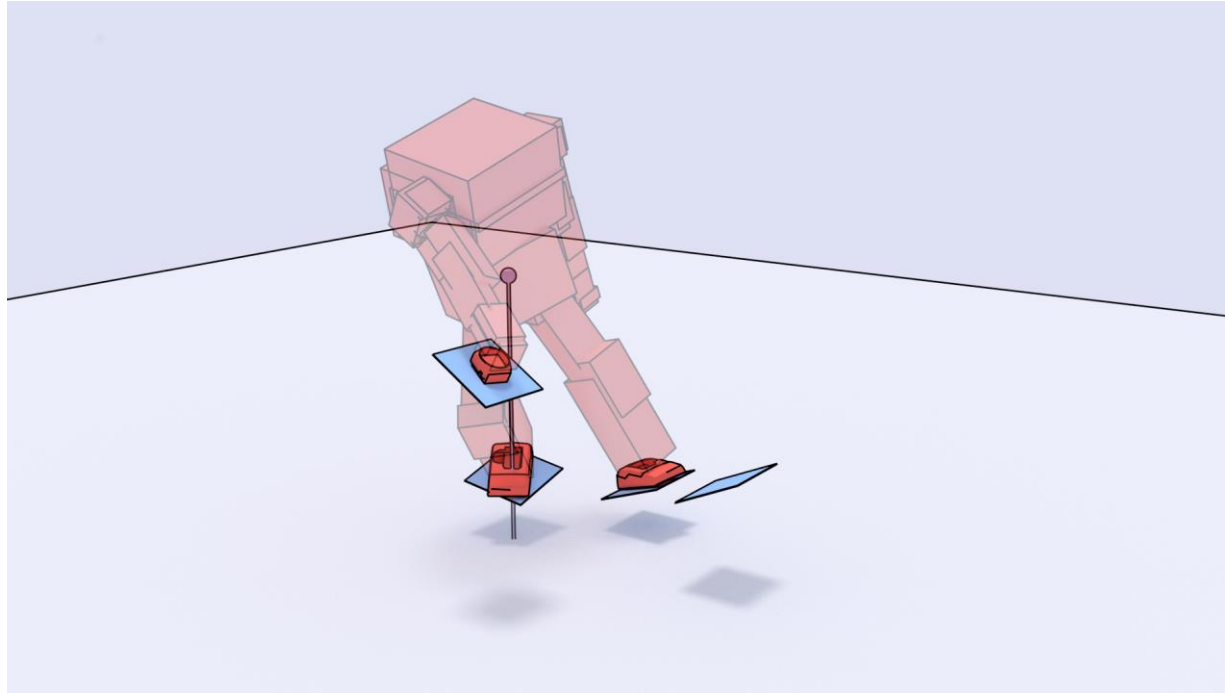
Dimensionality reduction for contacts / collision constraints [Tonneau et al. 2014]

Cheating assumption: low-dimensional space



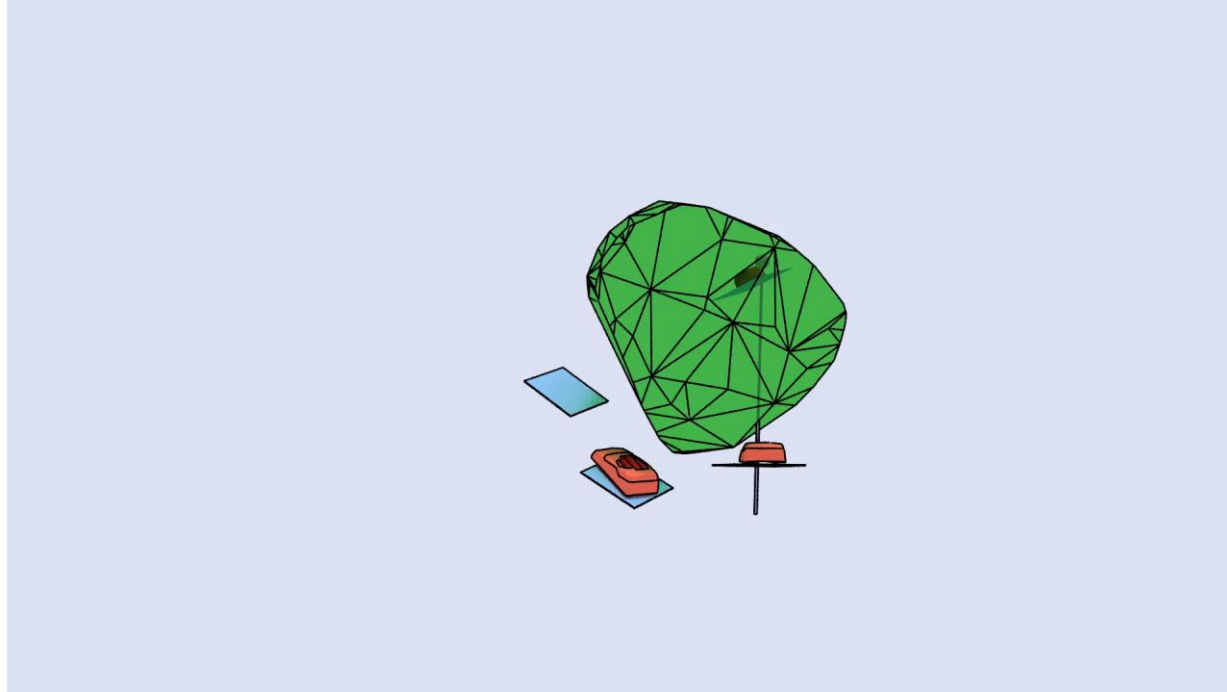
Low dimensional root path planner (RBPRM) [Tonneau et al. 15]

Cheating assumption: low-dimensional space



Dimensionality reduction (Centroidal model) [kajita 03, Orin 09]

Cheating assumption: linear kinematic constraints



Linear geometric constraints [Tonneau et al. 18]

~~Cheat..~~ conservative assumption: linear dynamics

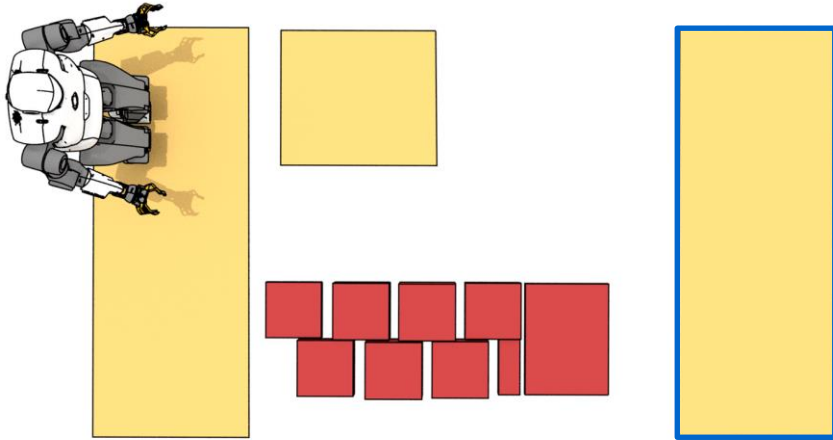


CROC [Fernbach, Tonneau et al. 2020]

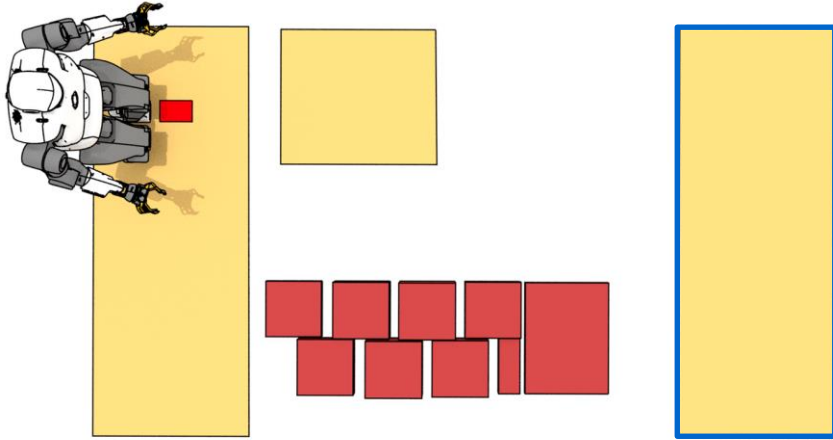
What's left ?

A combinatorial problem of large polynomial complexity...

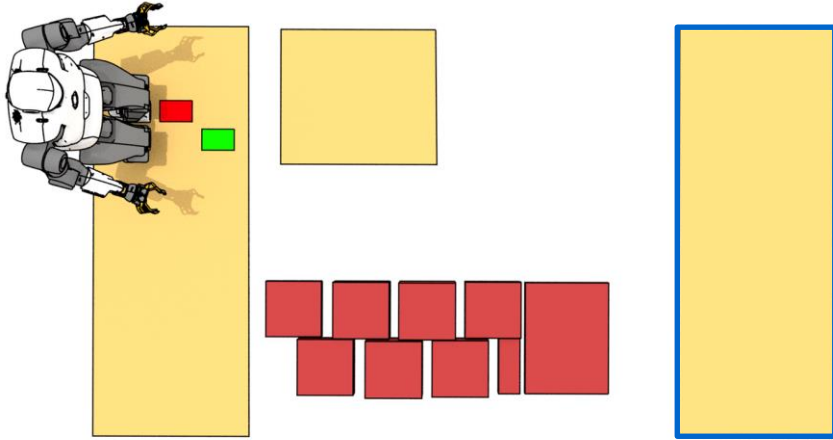
Example: how to reach the platform ?



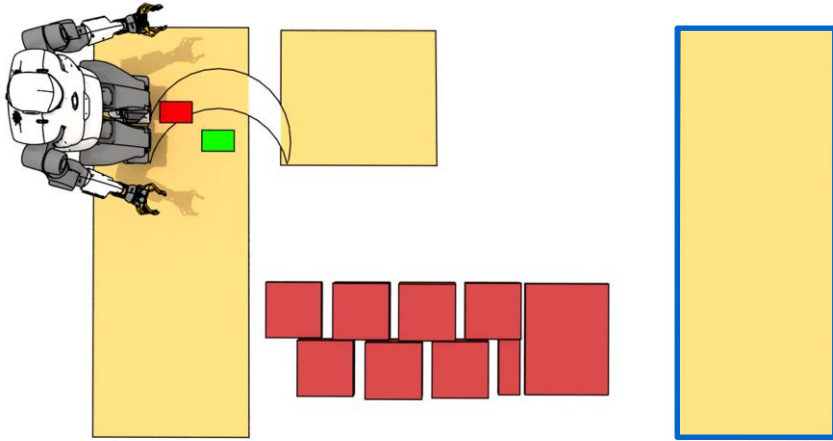
Combinatorial step planning



Combinatorial step planning



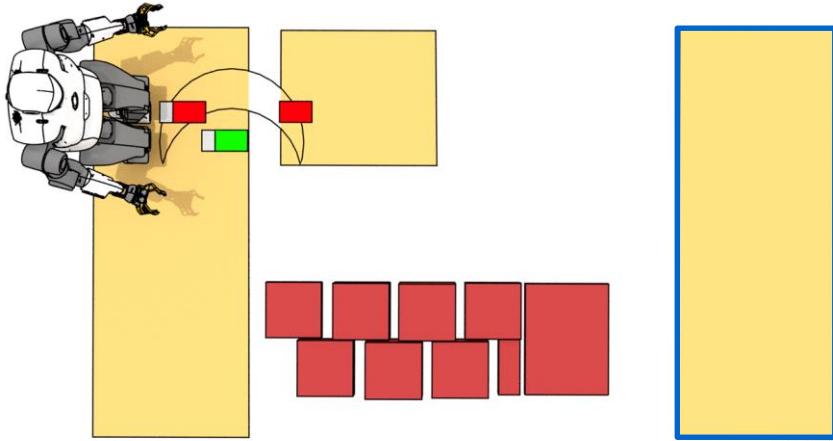
Combinatorial step planning



Feasibility $\mathcal{F}$:

Geometric constraints

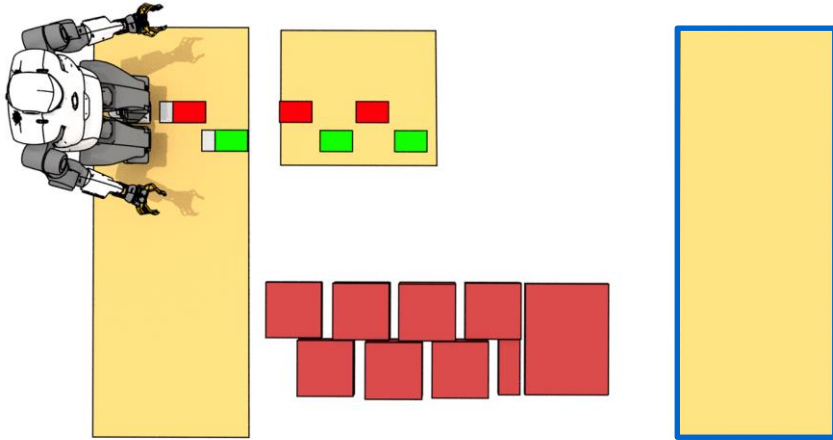
Combinatorial step planning



Feasibility $\mathcal{F}$:

Geometric constraints

Combinatorial step planning

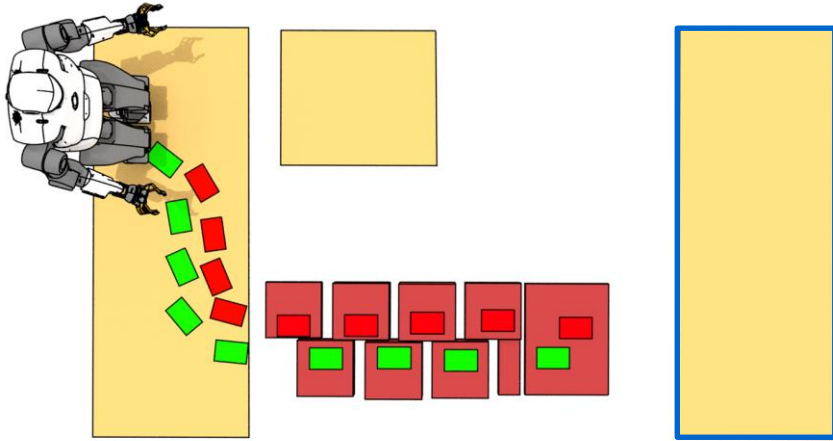


Global path search

Feasibility $\mathcal{F}$:

Geometric constraints

Combinatorial step planning

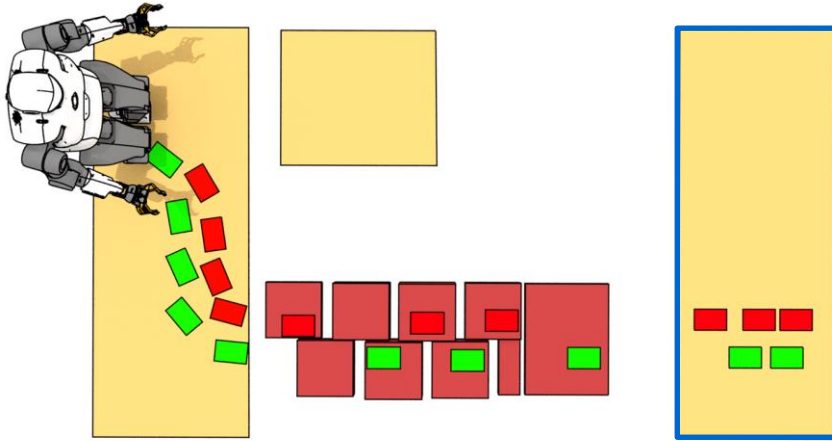


Global path search

Feasibility $\mathcal{F}$:

Geometric constraints

Combinatorial step planning



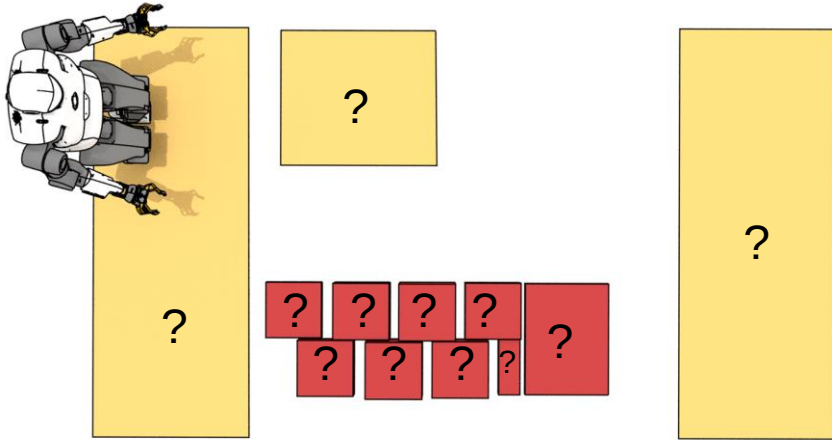
Global path search

Feasibility $\mathcal{F}$:

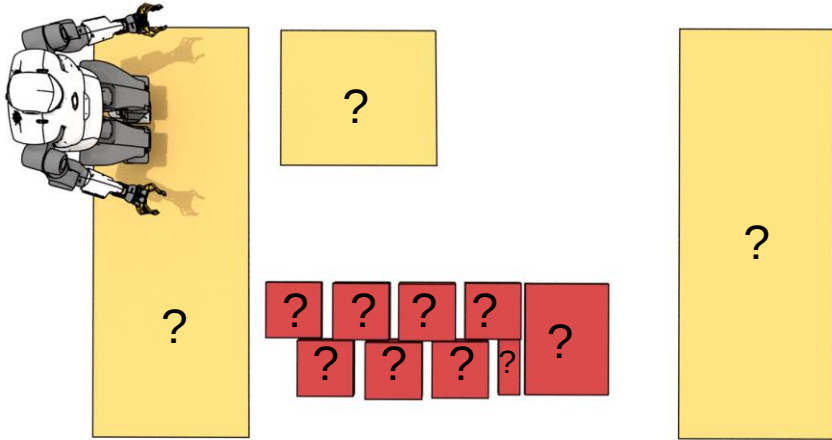
Geometric constraints

Dynamic constraints

Which contact surface ?



Which contact surface ?



Convex contact surfaces:

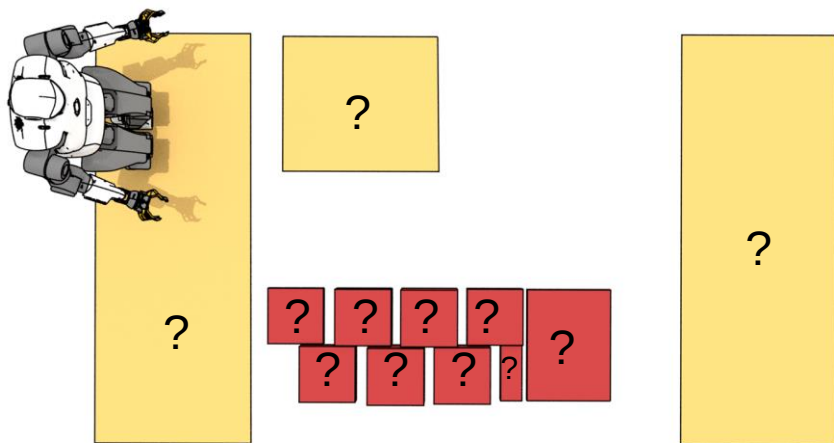
$$\mathcal{S}_j : \{\mathbf{p}, \mathbf{S}_j \mathbf{p} \leq \mathbf{s}_j\}$$

$$1 \leq j \leq n$$

Footsteps positions:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Which contact surface ?



Convex contact surfaces:

$$\mathcal{S}_j : \{\mathbf{p}, \mathbf{S}_j \mathbf{p} \leq \mathbf{s}_j\}$$

$$1 \leq j \leq n$$

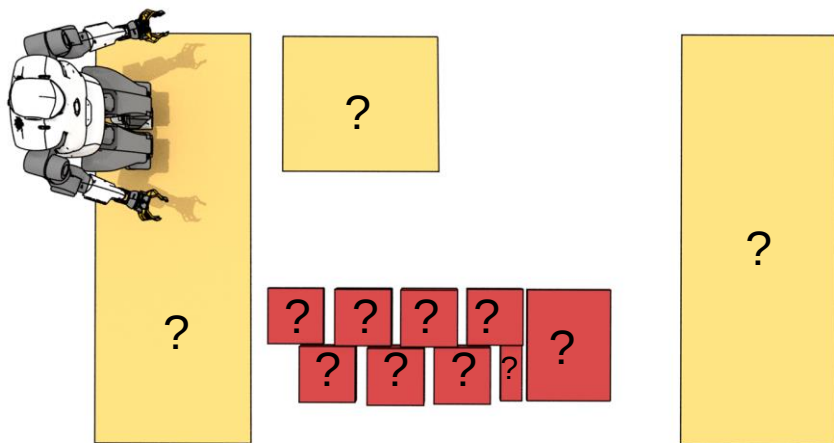
Footsteps positions:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Combinatorial:

$$n^m$$

Which contact surface ?



Convex contact surfaces:

$$\mathcal{S}_j : \{\mathbf{p}, \mathbf{S}_j \mathbf{p} \leq \mathbf{s}_j\}$$
$$1 \leq j \leq n$$

Footsteps positions:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Combinatorial:

$$n^m$$

Contact planning as a feasibility problem

find $\mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m]$

s.t. $\mathbf{X} \in \mathcal{F}$ // Feasibility

$\mathbf{X} \in \mathcal{I} \cap \mathcal{G}$ // Initial and goal conditions

$\forall i :$

$\mathbf{p}_i \in \mathcal{S}_1 \vee \dots \vee \mathbf{p}_i \in \mathcal{S}_n$

Contact planning as a feasibility problem

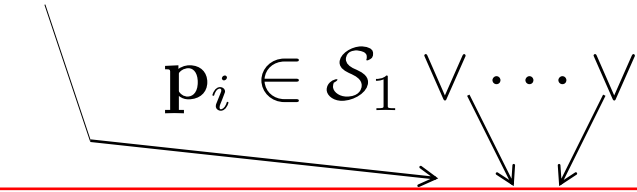
find $\mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m]$

s.t. $\mathbf{X} \in \mathcal{F}$ // Feasibility

$\mathbf{X} \in \mathcal{I} \cap \mathcal{G}$ // Initial and goal conditions

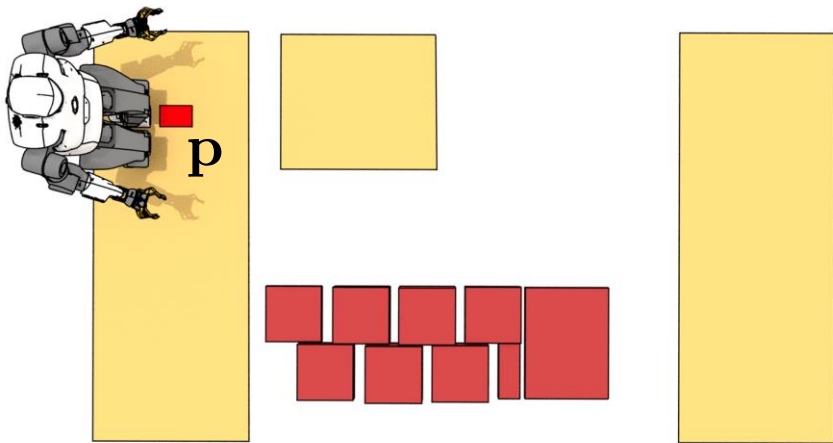
$\forall i :$

$\mathbf{p}_i \in \mathcal{S}_1 \vee \dots \vee \mathbf{p}_i \in \mathcal{S}_n$



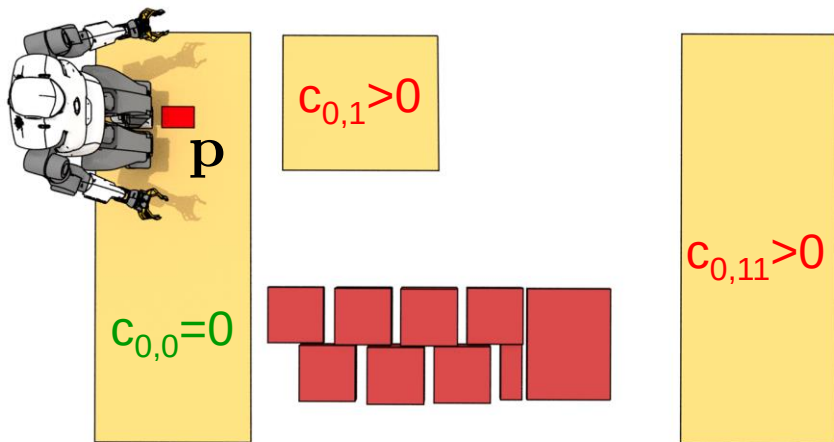
How to tackle the combinatorics ?

An equivalent cardinality minimisation problem



$$\mathbf{S}_j \mathbf{p}_i \leq \mathbf{s}_j \Leftrightarrow \mathbf{p}_i \in \mathcal{S}_j$$

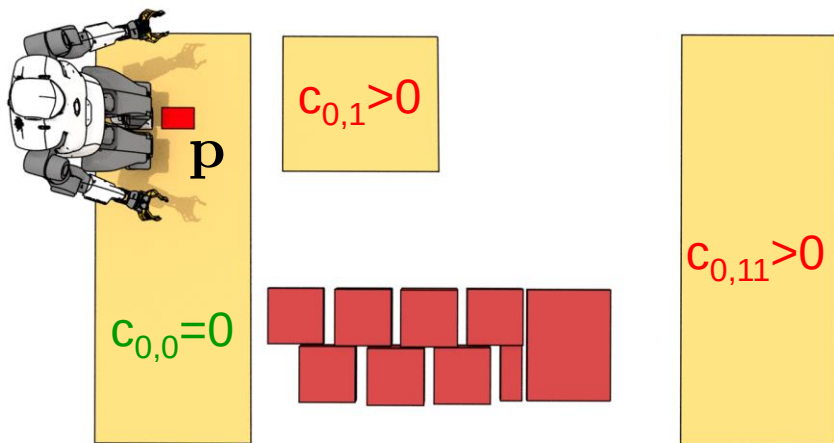
An equivalent cardinality minimisation problem



Slack variables $c_{i,j} \in \mathbb{R}^+$

$$\left. \begin{array}{l} \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \\ c_{i,j} = 0 \end{array} \right\} \Rightarrow \mathbf{p}_i \in \mathcal{S}_j$$
$$\mathbf{c}_i = [c_{i,1}, \dots, c_{i,n}]$$

An equivalent cardinality minimisation problem



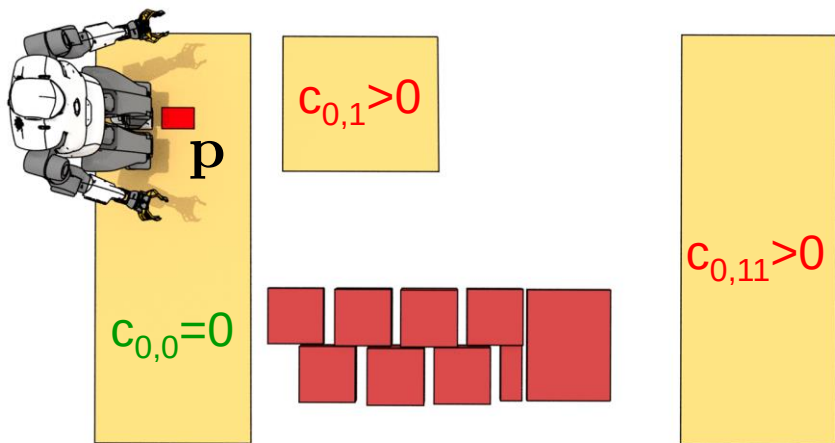
Slack variables $c_{i,j} \in \mathbb{R}^+$

$$\left. \begin{array}{l} \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \\ c_{i,j} = 0 \end{array} \right\} \Rightarrow \mathbf{p}_i \in \mathcal{S}_j$$

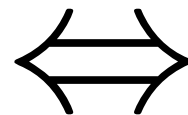
$$\mathbf{c}_i = [c_{i,1}, \dots, c_{i,n}]$$

$$\#NonZeros(\mathbf{c}_i) = n - 1 \Rightarrow \mathbf{p}_i \text{ on a surface}$$

An equivalent cardinality minimisation problem



find $\mathbf{p}_i$
s.t. $\mathbf{p}_i \in \mathcal{S}_1 \vee \dots \vee \mathbf{p}_i \in \mathcal{S}_n$



find $\mathbf{p}_i, \mathbf{c}_i$
min $card(\mathbf{c}_i)$
s.t. $\mathbf{S}_j \mathbf{p}_i - \mathbf{1}_{c_{i,j}} \leq \mathbf{s}_j \quad \forall j$

An equivalent cardinality minimisation problem

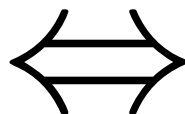
$$\text{find } \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m]$$

$$\text{s.t. } \mathbf{X} \in \mathcal{F}$$

$$\mathbf{X} \in \mathcal{I} \cap \mathcal{G}$$

$\forall i :$

$$\mathbf{p}_i \in \mathcal{S}_1 \vee \dots \vee \mathbf{p}_i \in \mathcal{S}_n$$



$$\text{find } \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m]$$

$$\mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n]$$

$$\text{min } \sum_{i=1}^m \text{card}(\mathbf{c}_i)$$

$$\text{s.t. } \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j$$

$$\mathbf{X} \in \mathcal{F}$$

$$\mathbf{X} \in \mathcal{I} \cap \mathcal{G}$$

Convex relaxation with sparsity inducing norm

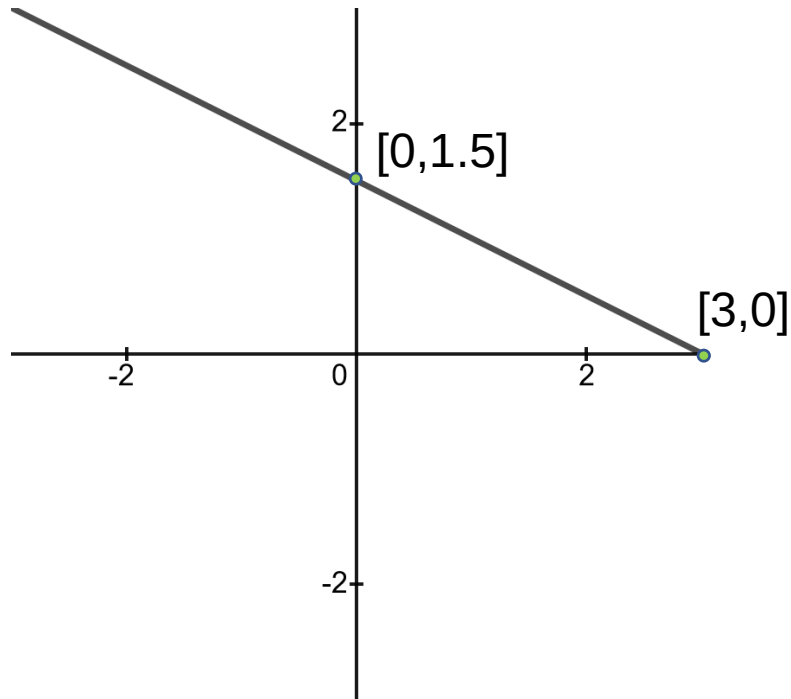
$$\begin{aligned} \text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\ & \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\ \min \quad & \sum_{i=1}^m \text{card}(\mathbf{c}_i) \\ \text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\ & \mathbf{X} \in \mathcal{F} \\ & \mathbf{X} \in \mathcal{I} \cap \mathcal{G} \end{aligned}$$

$\approx$

$$\begin{aligned} \text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\ & \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\ \min \quad & \sum_{i=1}^m \|\mathbf{c}_i\|_1 \\ \text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\ & \mathbf{X} \in \mathcal{F} \\ & \mathbf{X} \in \mathcal{I} \cap \mathcal{G} \end{aligned}$$

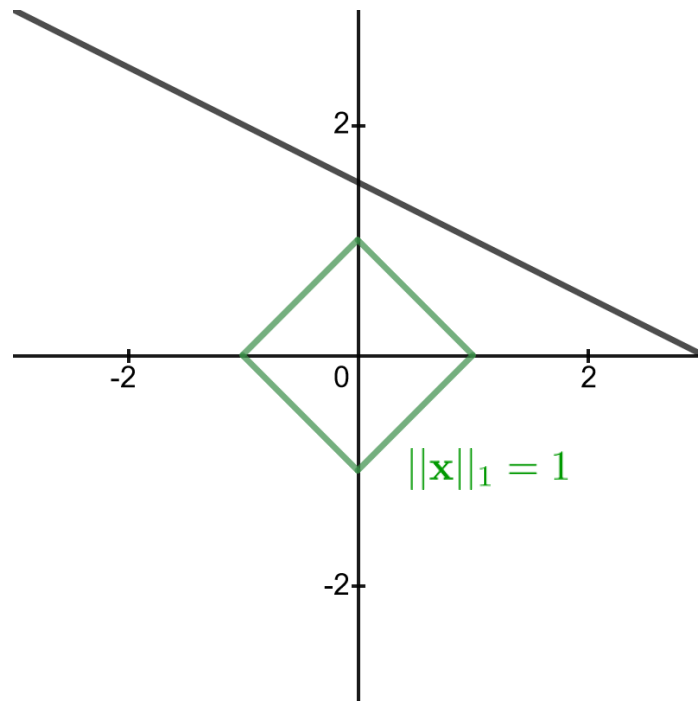
L1 Norm induces sparsity

$$\begin{array}{ll}\text{find} & \mathbf{x} = [x_1, x_2] \\ \text{min} & \text{card}(\mathbf{x}) \\ \text{s.t.} & x_2 = 1.5 - 0.5 x_1\end{array}$$



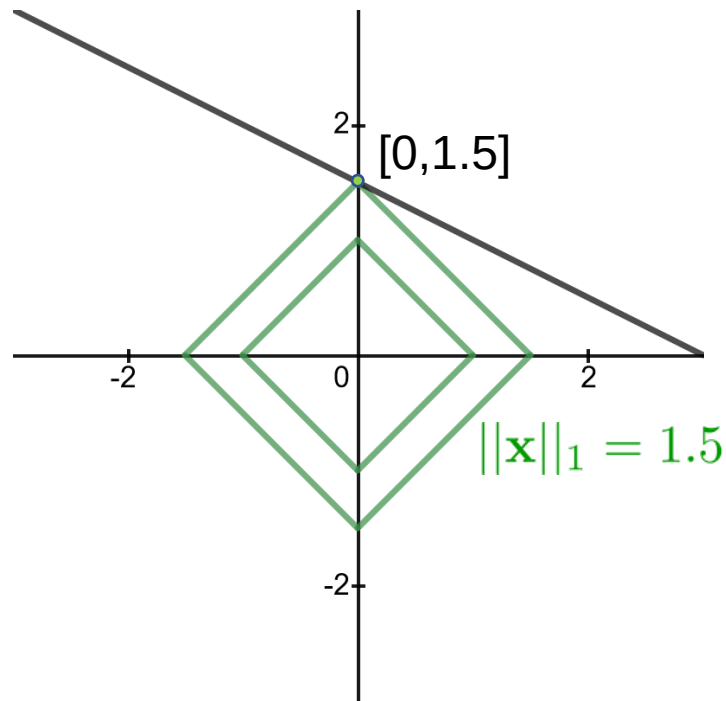
L1 Norm induces sparsity

$$\begin{array}{ll}\text{find} & \mathbf{x} = [x_1, x_2] \\ \text{min} & \cancel{\text{card}(\mathbf{x})} \|\mathbf{x}\|_1 \\ \text{s.t.} & x_2 = 1.5 - 0.5 x_1\end{array}$$



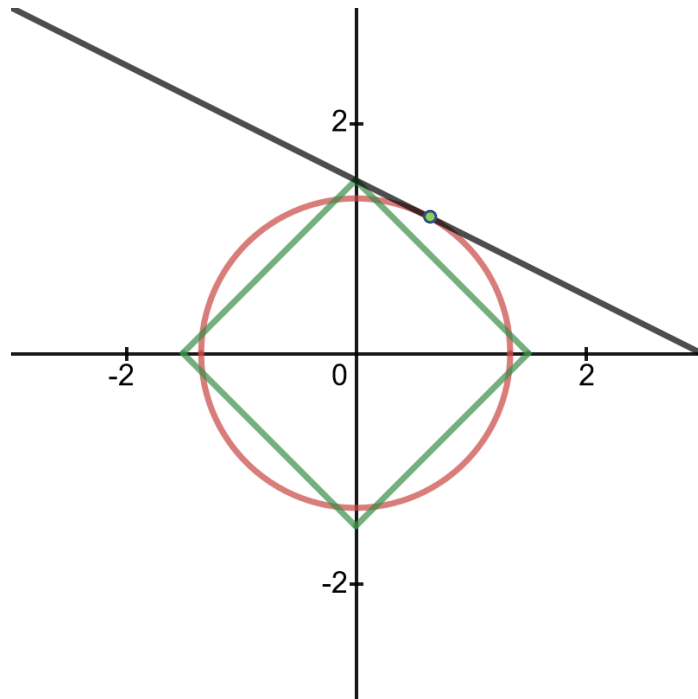
L1 Norm induces sparsity

$$\begin{array}{ll}\text{find} & \mathbf{x} = [x_1, x_2] \\ \text{min} & \cancel{\text{card}(\mathbf{x})} \|\mathbf{x}\|_1 \\ \text{s.t.} & x_2 = 1.5 - 0.5 x_1\end{array}$$



Euclidian norm does not lead to sparsity

$$\begin{array}{ll}\text{find} & \mathbf{x} = [x_1, x_2] \\ \text{min} & \cancel{\text{card}(\mathbf{x})} \quad ||\mathbf{x}||_2 \\ \text{s.t.} & x_2 = 1.5 - 0.5 x_1\end{array}$$



Convex relaxation with sparsity inducing norm

$$\begin{aligned} \text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\ & \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\ \min \quad & \sum_{i=1}^m \text{card}(\mathbf{c}_i) \\ \text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\ & \mathbf{X} \in \mathcal{F} \\ & \mathbf{X} \in \mathcal{I} \cap \mathcal{G} \end{aligned}$$

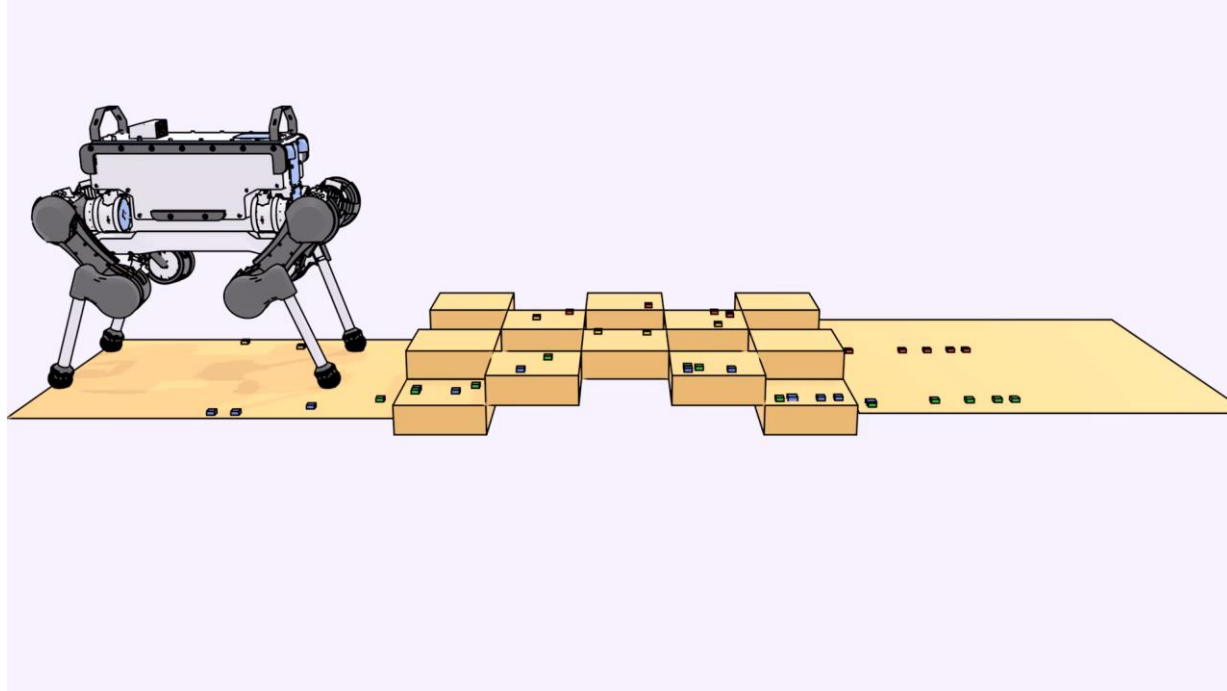
$\approx$

$$\begin{aligned} \text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\ & \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\ \min \quad & \sum_{i=1}^m \|\mathbf{c}_i\|_1 \\ \text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\ & \mathbf{X} \in \mathcal{F} \\ & \mathbf{X} \in \mathcal{I} \cap \mathcal{G} \end{aligned}$$

Some results



Ongoing work: extension to quadrupeds



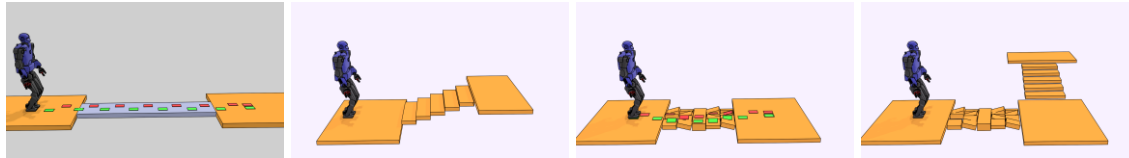
L1 relaxation is faster, but less reliable

L1 up to 10x faster than Mixed-Integer (MI) solvers...

What if relaxed problem does not converge to sparse solution ?

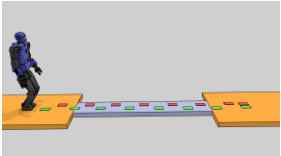
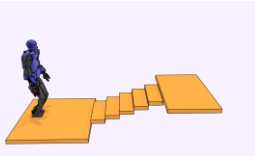
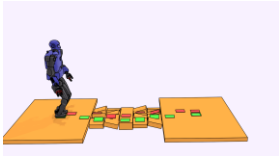
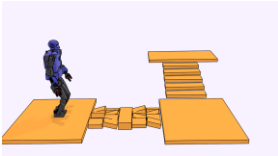
- Can we guarantee convergence ?
- A more general question: Is this really a combinatorial problem?

Let's look at the number of nodes explored by MI:



MIP (+cost)	0 (1)	0 (1)	383 (2742)	2724 (20452)
L1 success	V	V	X	X

Let's look at the number of nodes explored by MI:

				
MIP (+cost)	0 (1)	0 (1)	383 (2742)	2724 (20452)
L1 success	V	V	X	X

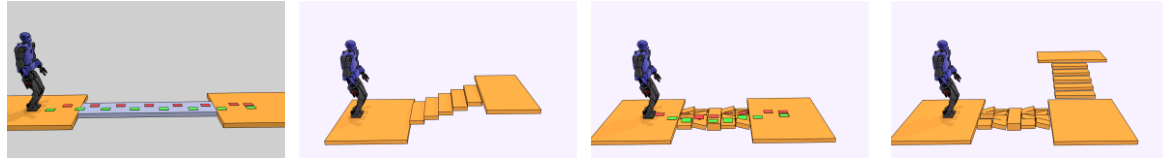
« Presolve » option:

does not change optimum

discard unfeasible solutions

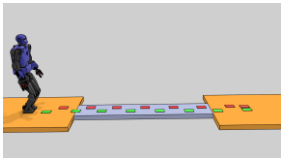
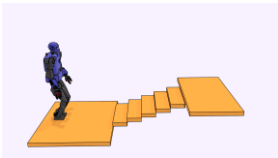
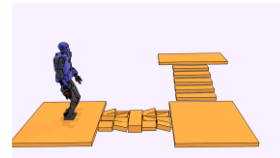
additional constraints => slower resolution time

With presolve, at most 1 node exploration for feasibility problem



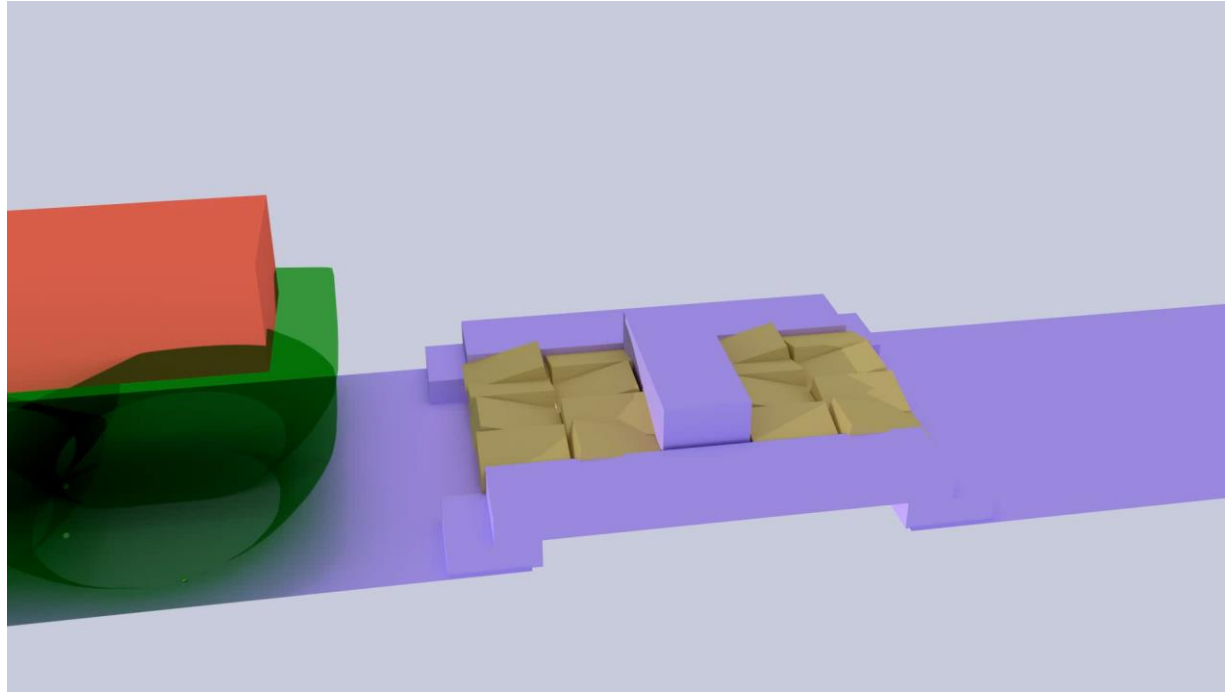
MIP - No presolve	0 (1)	0 (1)	383 (2742)	2724 (20452)
MIP - Presolve	0 (1)	0 (1)	1 (1)	1 (2960)

With presolve, at most 1 node exploration for feasibility problem

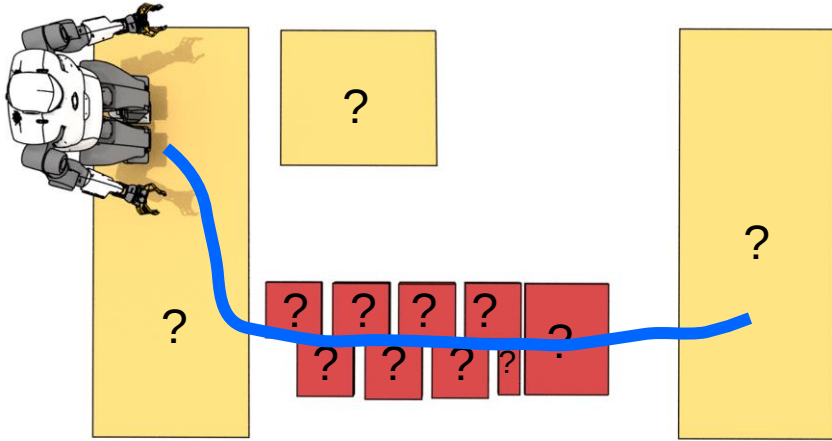
				
MIP - No presolve	0 (1)	0 (1)	383 (2742)	2724 (20452)
MIP - Presolve	0 (1)	0 (1)	1 (1)	1 (2960)

Can we use a « custom » presolve for L1 norm minimisation ?

COMPLEXITY CAN BE REDUCED WITH A GUIDE



COMPLEXITY CAN BE REDUCED WITH A GUIDE



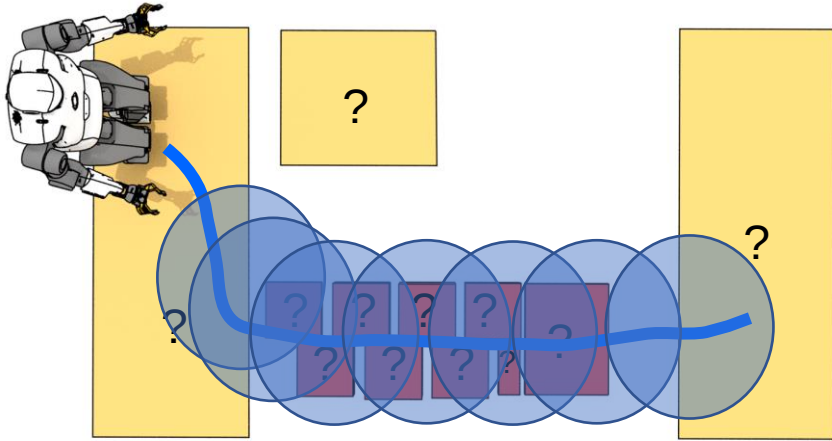
Contact surfaces

$$\mathcal{S}_j, 1 \leq j \leq n$$

Steps:

$$\mathbf{p}_i, 1 \leq i \leq m$$

COMPLEXITY CAN BE REDUCED WITH A GUIDE



Contact surfaces

$$\mathcal{S}_j, 1 \leq j \leq n$$

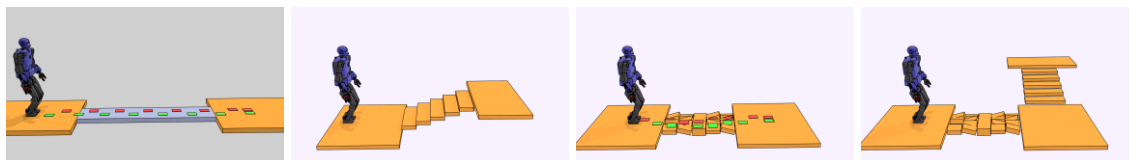
Steps:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Combinatorial:

$$n^h, h < m$$

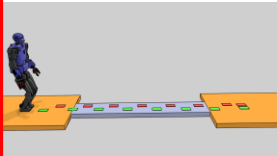
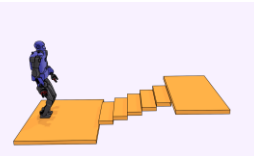
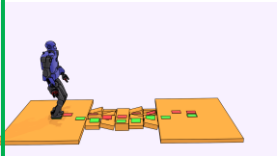
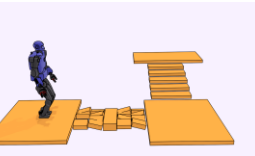
Guide-path improves slim convergence, but also MIP performances



L1 success	V	V	X	X
Guide + L1	V	V	V	V

Ok but why bother ? Computational time (ms)

Tradeoff guide cost (about 100 ms) / Solver time

				
Guide + L1	35	20	73	219
Guide + MIP	77	34	230	508
MIP	166	59	389	74710

Early conclusions for footstep planning

- Refining problem definition leads to vanishing combinatorics

Using domain-specific knowledge seems promising

Other issues to consider (what is the right path discretisation ?)

- L1-norm relaxation converges to feasible solution faster
- What about more challenging problems (gait selection) ?
- Optimality ? I think we do not care!

Overall conclusions for Contact planning

Should we embrace combinatorics or try to ignore it ?

What can we expect for more challenging tasks ?

ie. Can we « presolve » harder problems ?

Non-linear problems, unknown gait patterns

I believe those questions are challenging, important and fun

That's all for now

<https://stevetonneau.fr> for papers, videos, source code