Optimal Control

Tommaso Caneva
PhD in Physics

Optimal Control

how to optimize tasks

Optimal Control

how to optimize tasks

Speed

Optimal Control

Precision

Eff ciency

How?

Training

How?

Training

How?

Too long or too complex?

Training

How?

Too long or too complex?

Optimal control

Teach machines!

Framework: repeatable task

System

Control

Target

Framework: repeatable task

System

Control

Target

First time...

Framework: repeatable task

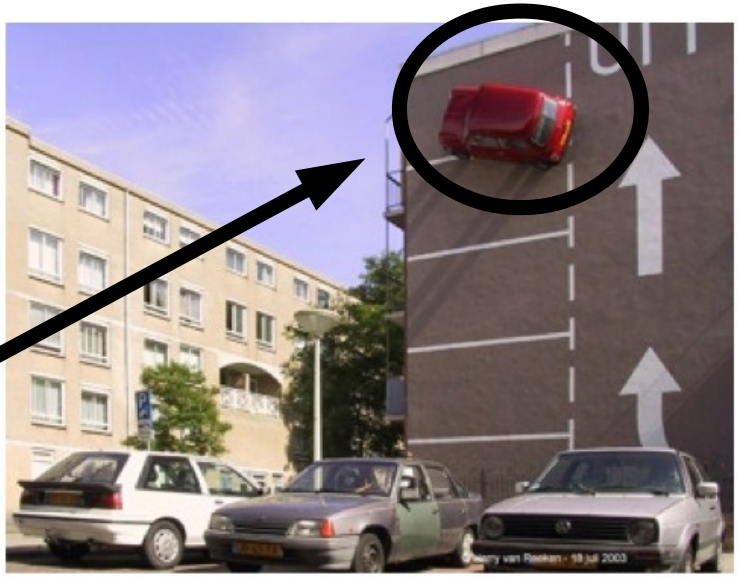
System

Control

Target

First time...

...but after years of driving experience...



Framework: repeatable task

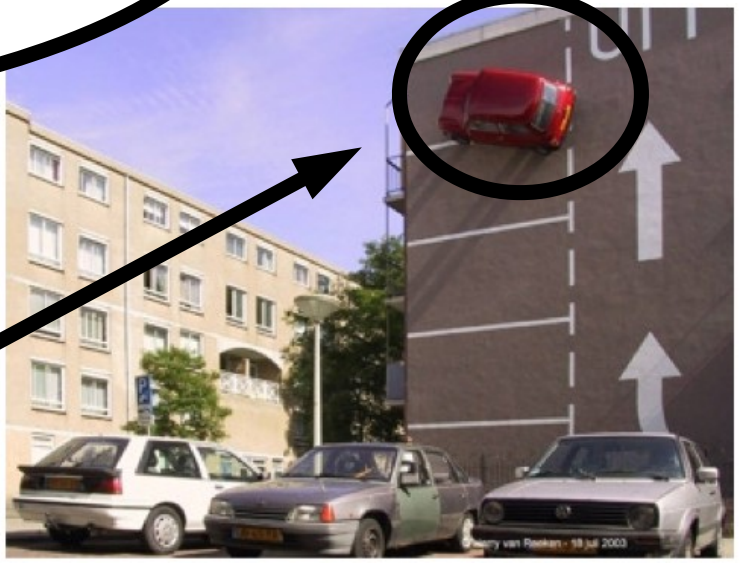
System

Control

Target

Repetita iuvant:
Learning from repetition of the task

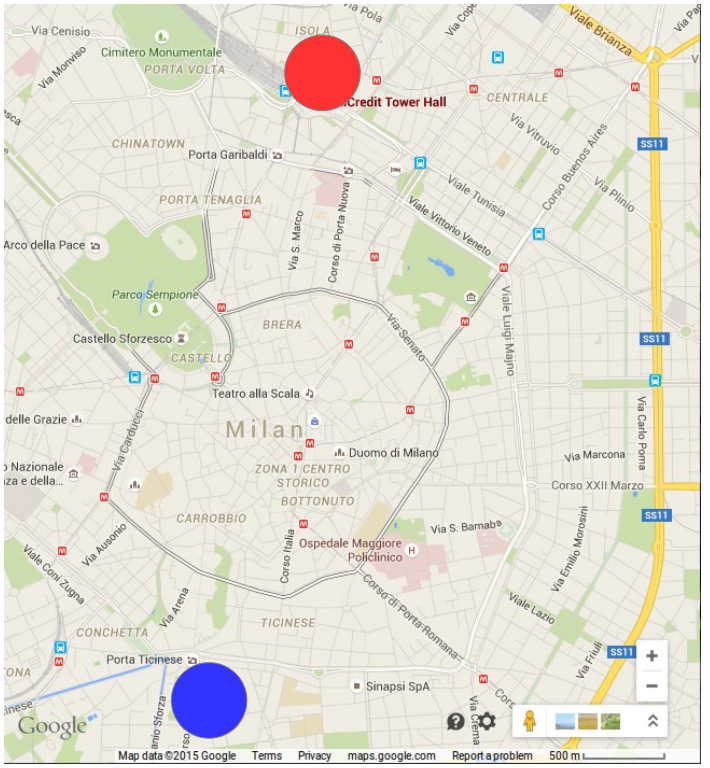
...but after years of driving experience...



General formulation of the problem

Ingredients

- Initial state:
- Task: go to in 20 minutes
- Tunable sytem: car + indications, G

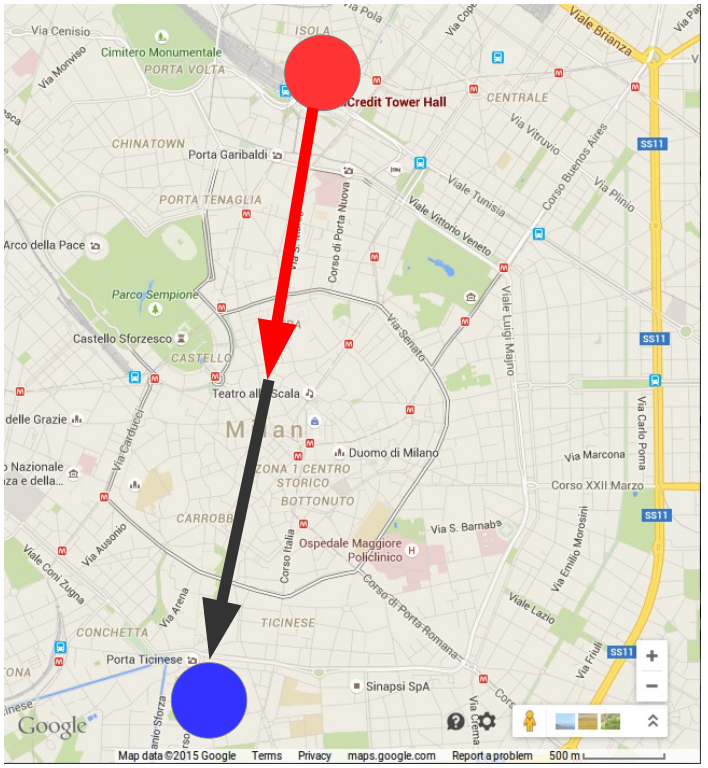


General formulation of the problem

Ingredients

- Initial state:
- Task: go to in 20 minutes
- Tunable sytem: car + indications, G

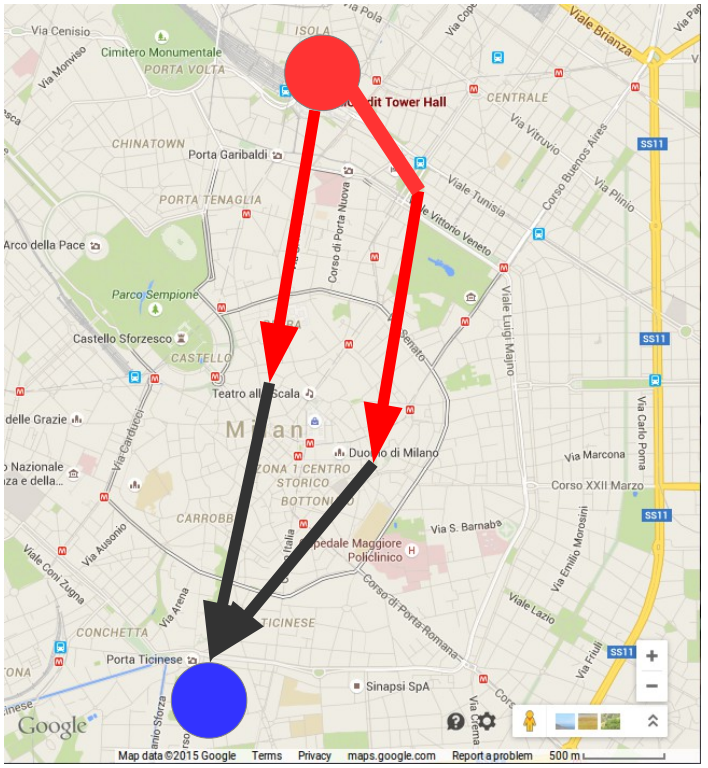
- Cost function: F(G)



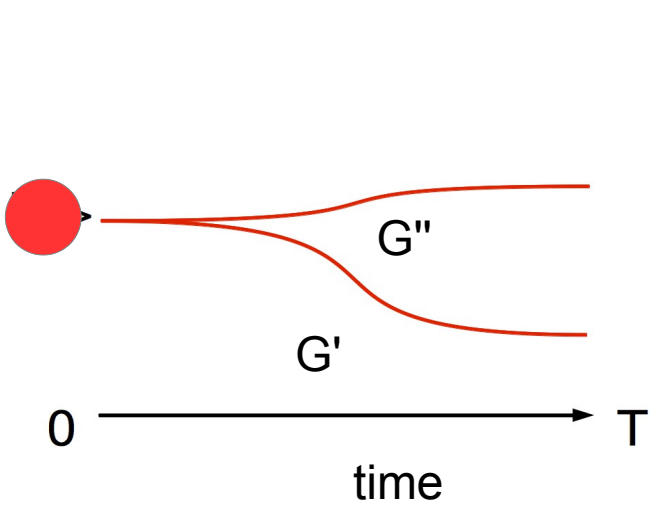
General formulation of the problem

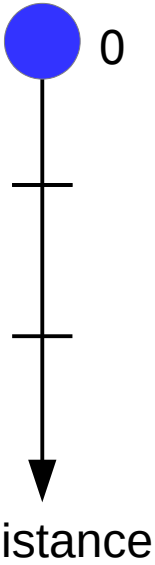
Ingredients

- Initial state:
- Task: go to in 20 minutes
- Tunable sytem: car + indications, G



- Cost function: F(G)

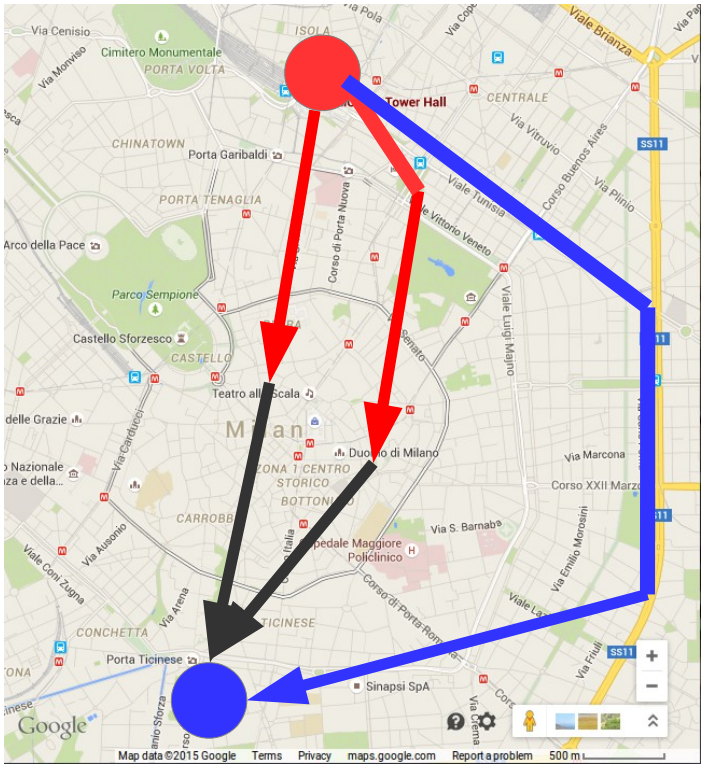




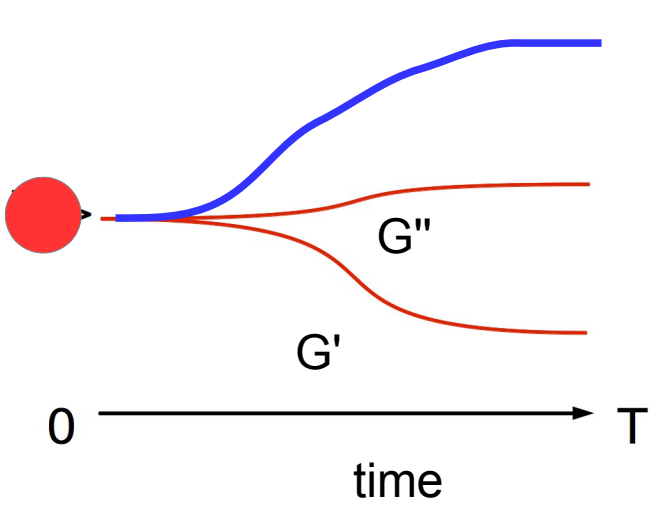
General formulation of the problem

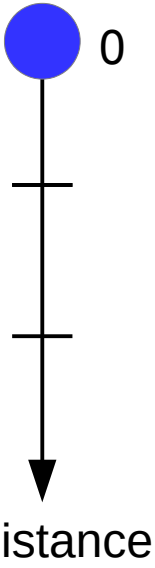
Ingredients

- Initial state:
- Task: go to in 20 minutes
- Tunable sytem: car + indications, G



- Cost function: F(G)





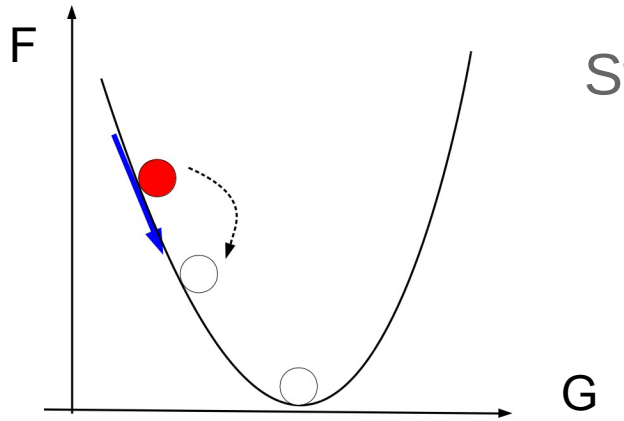
Optimal Control

$F(G) = 0$

minimization cost function

Recipe 1: gradient method

Single variable function minimization

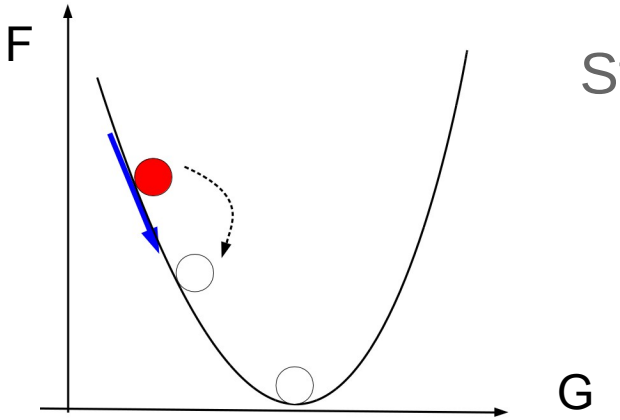


Steepest descent

gradient

Recipe 1: gradient method

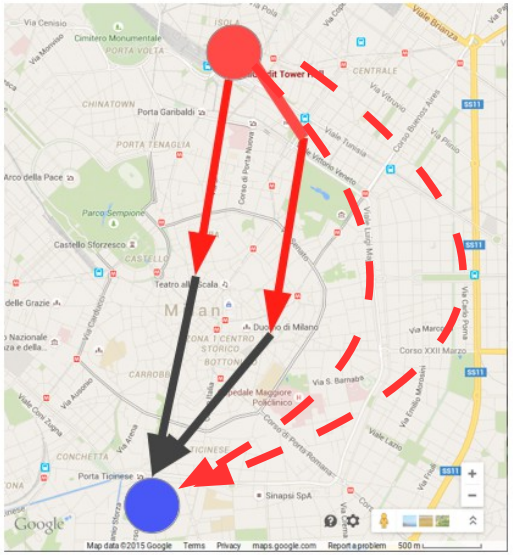
Single variable function minimization



Steepest descent

gradient

Many variable function minimization



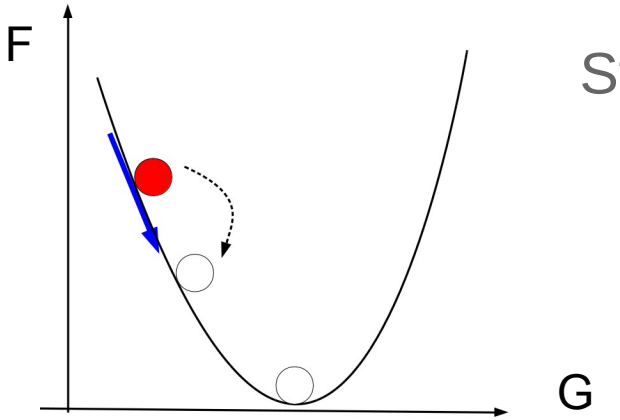
- In principle inf nite paths

- Computationally slow
(~ Neural Network training)

- Maybe not necessary: close
to mimum is enough

Recipe 1: gradient method

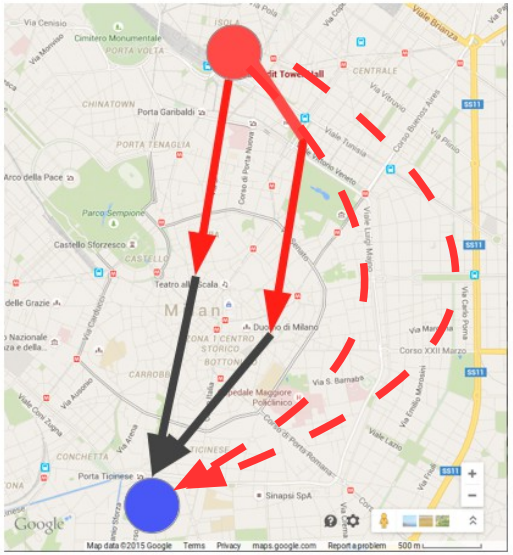
Single variable function minimization



Steepest descent

gradient

Many variable function minimization



- In principle inf nite paths

- Computationally slow
(~ Neural Network training)

- Maybe not necessary: close
to mimum is enough

Alternatives?

Looking for alternatives: an important dinner

You have an important guest;

you decide to prepare an impressive dinner...

Looking for alternatives: an important dinner

You have an important guest;

you decide to prepare an impressive dinner...

That is what you f ȵd in your kitchen:

- a tomato

Looking for alternatives: an important dinner

You have an important guest;

you decide to prepare an impressive dinner...

That is what you f ȵd in your kitchen:

- a tomato
- a bottle of water

Looking for alternatives: an important dinner

You have an important guest;

you decide to prepare an impressive dinner...

That is what you f ȵd in your kitchen:

- a tomato
- a bottle of water
- ...sorry, nothing else...

Looking for alternatives: an important dinner

You have an important guest;

you decide to prepare an impressive dinner...

That is what you f ɲd in your kitchen:

- a tomato
- a bottle of water
- ...sorry, nothing else...

What are you going to do?

Looking for alternatives: an important dinner

You have an important guest;
you decide to prepare an impressive dinner...

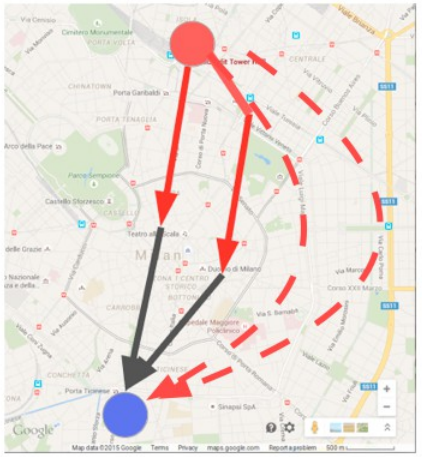
That is what you find in your kitchen:
- a tomato
- a bottle of water
- ...sorry, nothing else...

What are you going to do?

Try to get the best
of what you have at disposal!

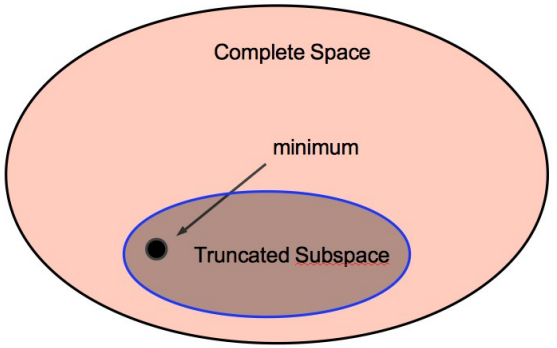
Recipe 2: approximate method

Initial state  → Dynamics G(t) → target state  , cost F



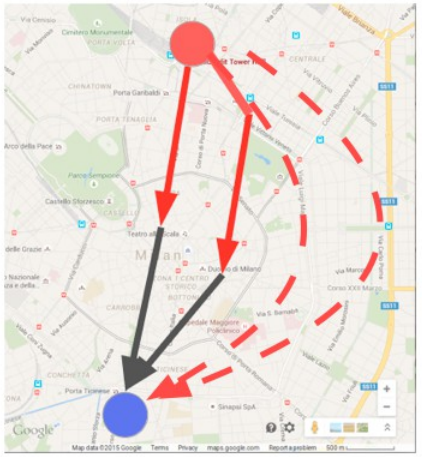
Problem: space too big

Solution: truncated space



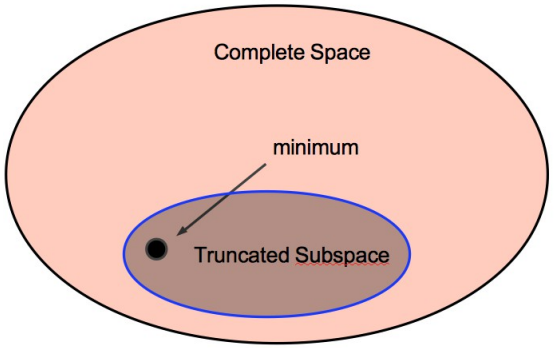
Recipe 2: approximate method

Initial state  → Dynamics G(t) → target state  , cost F



Problem: space too big

Solution: truncated space

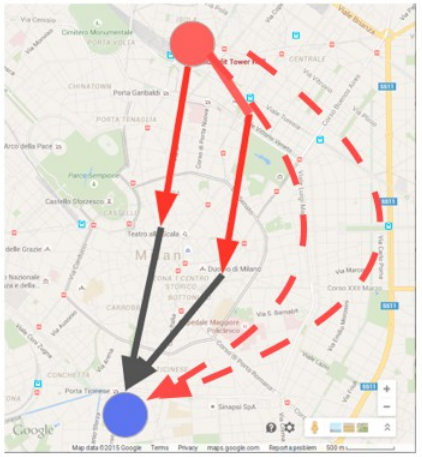


Example:

Draw a man: 

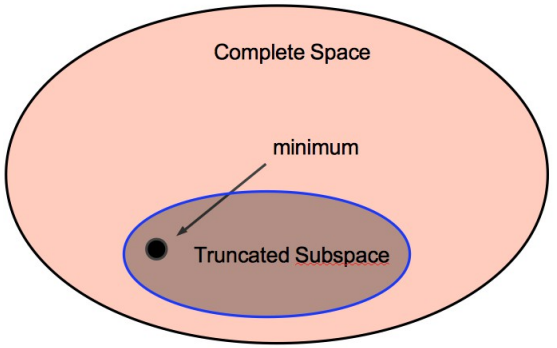
Recipe 2: approximate method

Initial state  → Dynamics G(t) → target state  , cost F

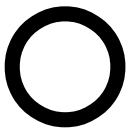
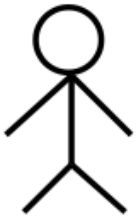


Problem: space too big

Solution: truncated space

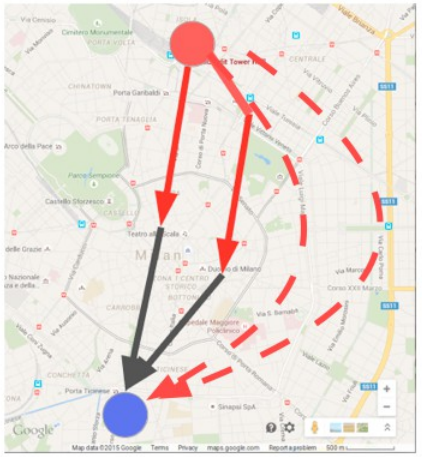


Example:

Draw a man: 1x  + 5x  = 

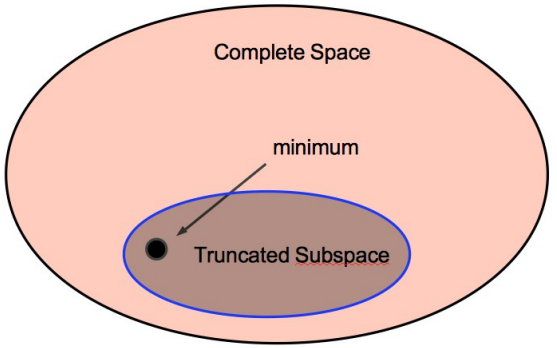
Recipe 2: approximate method

Initial state  → Dynamics G(t) → target state  , cost F

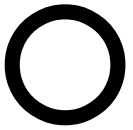
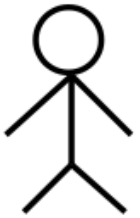


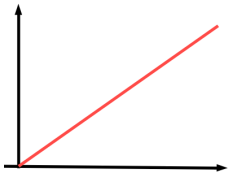
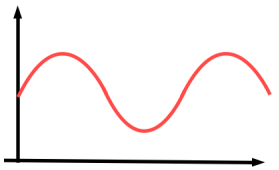
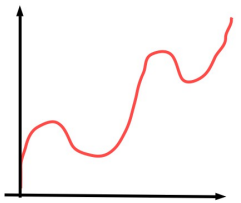
Problem: space too big

Solution: truncated space



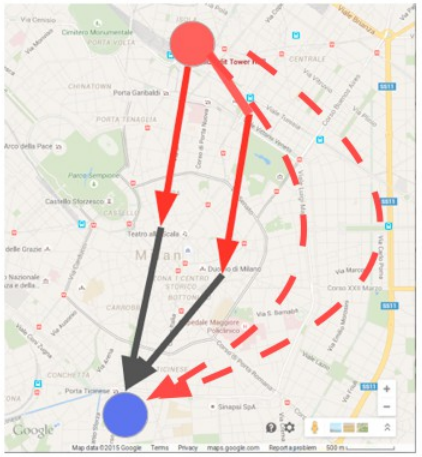
Example:

Draw a man: 1x  + 5x  = 

Draw Opt. path: 1x  + 5x  = 

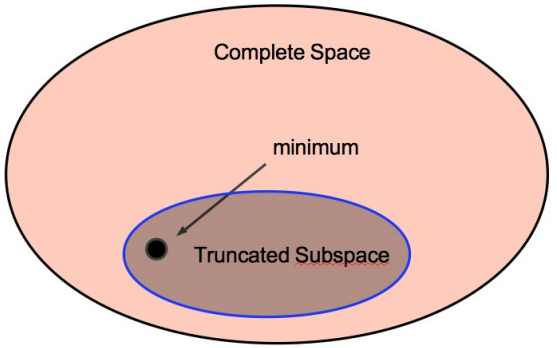
Recipe 2: approximate method

Initial state  → Dynamics G(t) → target state  , cost F



Problem: space too big

Solution: truncated space



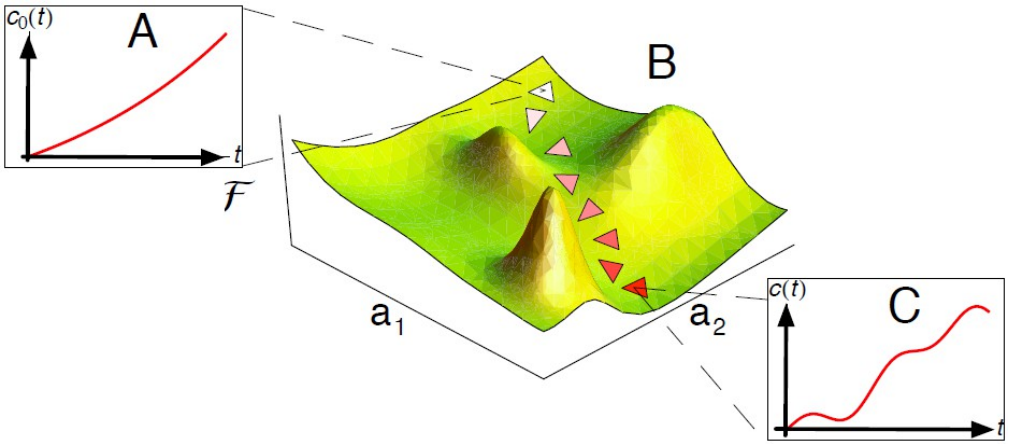
Algorithm CRAB:

- G(t) in truncated basis of functions:

$$G(t) = \sum_i w_i b_i(t)$$

- Cost: few variable function of weigths W

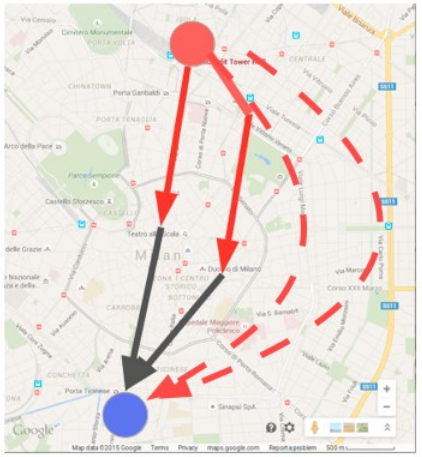
$$F[G(t)] = F(\vec{W})$$



P. Doria, T. Calarco and S. Montangero, PRL (2011); T. Caneva, T. Calarco and S. Montangero, PRA (2011)

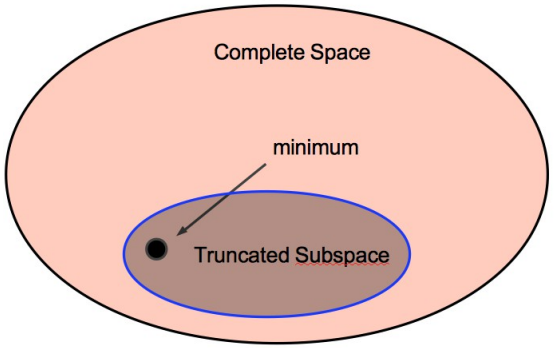
Recipe 2: approximate method

Initial state  → Dynamics G(t) → target state  , cost F



Problem: space too big

Solution: truncated space



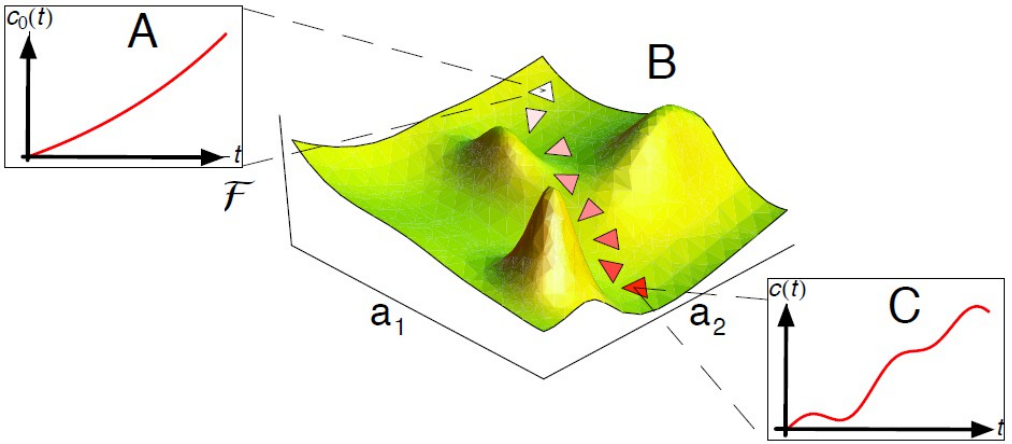
Algorithm CRAB:

- G(t) in truncated basis of functions:

$$G(t) = \sum_i w_i b_i(t)$$

- Cost: few variable function of weigths W

$$F[G(t)] = F(\vec{W})$$



P. Doria, T. Calarco and S. Montangero, PRL (2011); T. Caneva, T. Calarco and S. Montangero, PRA (2011)

Application: reversibility problem

?

- Is it possible to revert dynamics?
- Is the protocol robust?

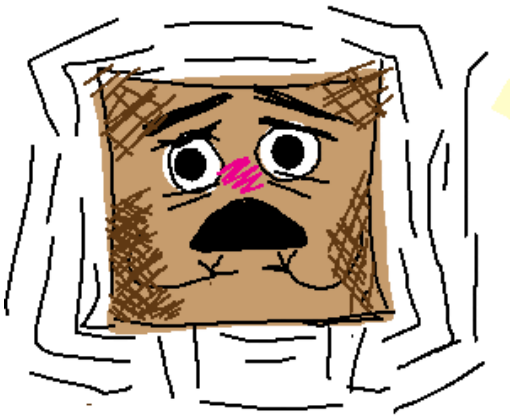
Application: reversibility problem

?

- Is it possible to revert dynamics?
- Is the protocol robust?

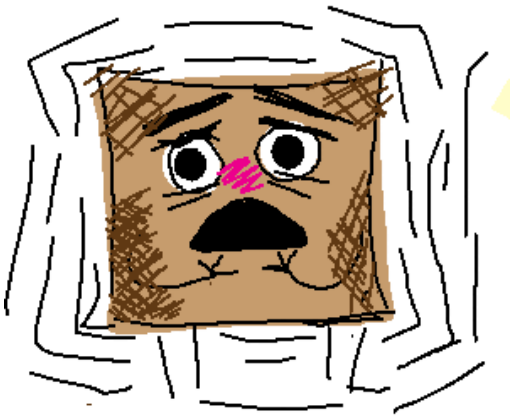
✔ Of course, use OC!
✔ Extremely robust

Application: reversibility problem



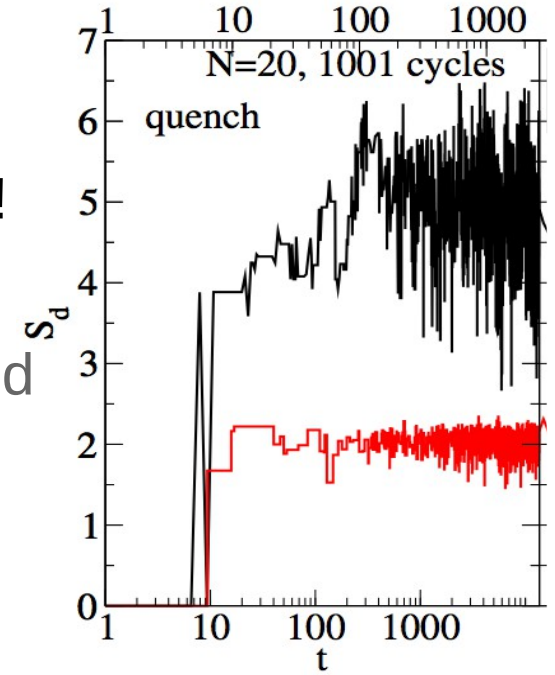
Shake it randomly!

Application: reversibility problem



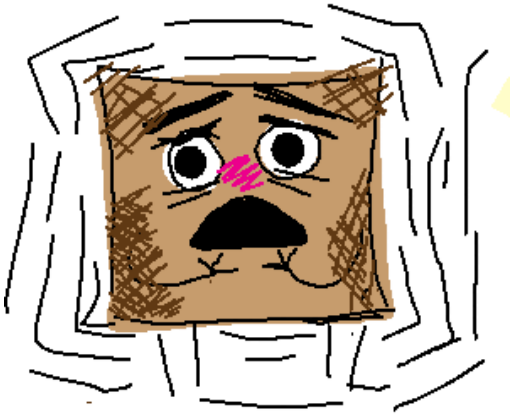
Shake it randomly!

Entropy grows and saturates



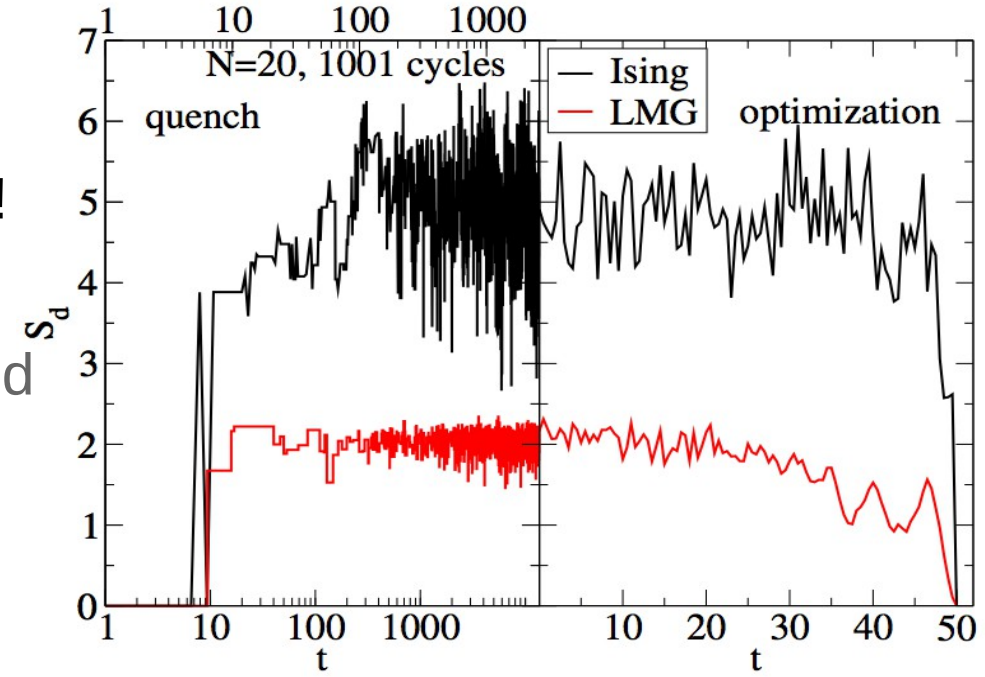
T. Caneva et al, PRA (2014)

Application: reversibility problem



Shake it randomly!

Entropy grows and saturates



Shake it with OC!

Entropy decreases, system goes back!

T. Caneva et al, PRA (2014)

Conclusions...

how to optimize tasks

Speed

Optimal Control

Precision

Eff ciency