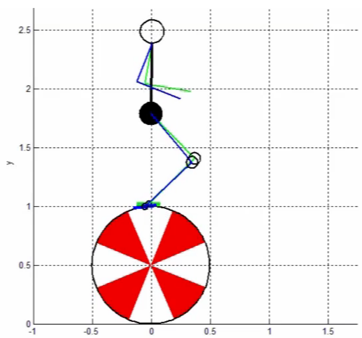
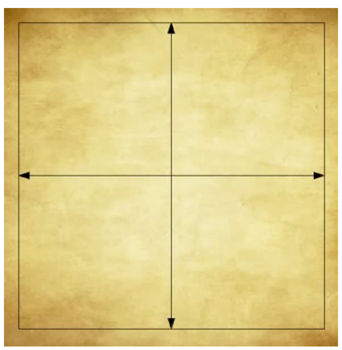
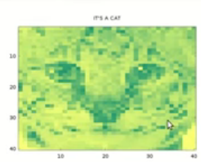
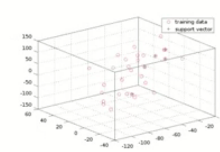
Optimize!

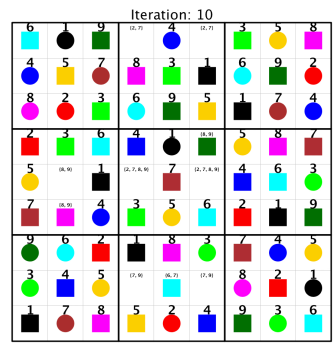
What do mobile robots, cat videos, Sudoku, and protein folding have in common?

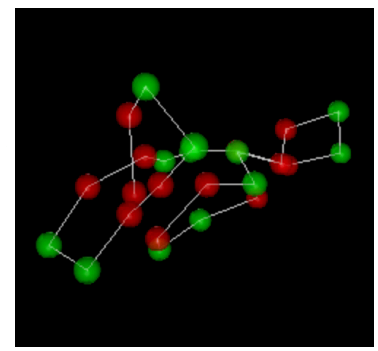












Research Takes a Village

Javier Alonso-Mora
TU Delft

Charles Mathy
Analog Devices

Jonathan Yedidia
Analog Devices

Veit Elser
Cornell

Ning Hao
Oracle

AmirReza Oghbaee
Northeastern

Mohammad Rostami
UPenn

José Bento
Boston College

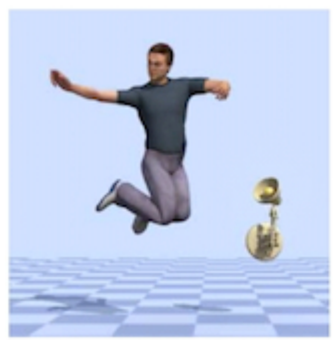
Display Swarm/Pixel Bots

One Dimension "Higher"

Many Problems…

Automated Manufacturing

Scene Reconstruction



Animation

Computer Vision

A Unifying Approach: Optimization

1. Model with an **objective function**
   - A mathematical representation of what you hope to achieve
   - Meaning, if the function value is minimized/maximized, you have found an *optimal* solution to your problem

2. Find the function inputs that lead to the minimum/maximum function value

To Be More Precise…

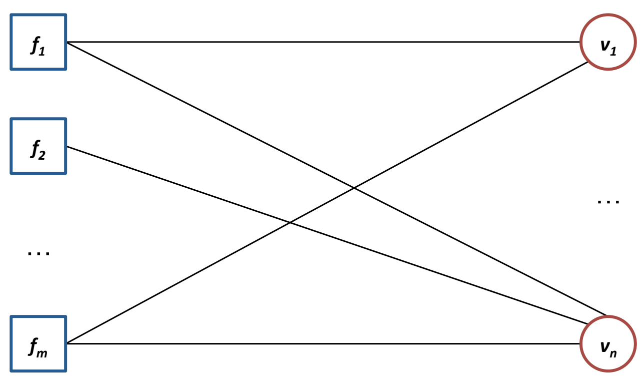
Soft

$$\underset{\boldsymbol{v}\in\mathbb{R}^n}{\text{minimize}} : f(\boldsymbol{v}) = f_1(v_1, v_2, \ldots) + f_2(\ldots) + \ldots$$

$$+ \sum c_k(v_1, v_2, \ldots) = \begin{cases} 0 & \text{constraint met} \\ \infty & \text{else} \end{cases}$$

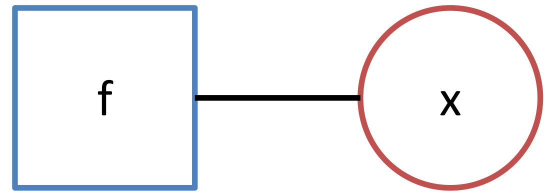
Hard

Objective Function

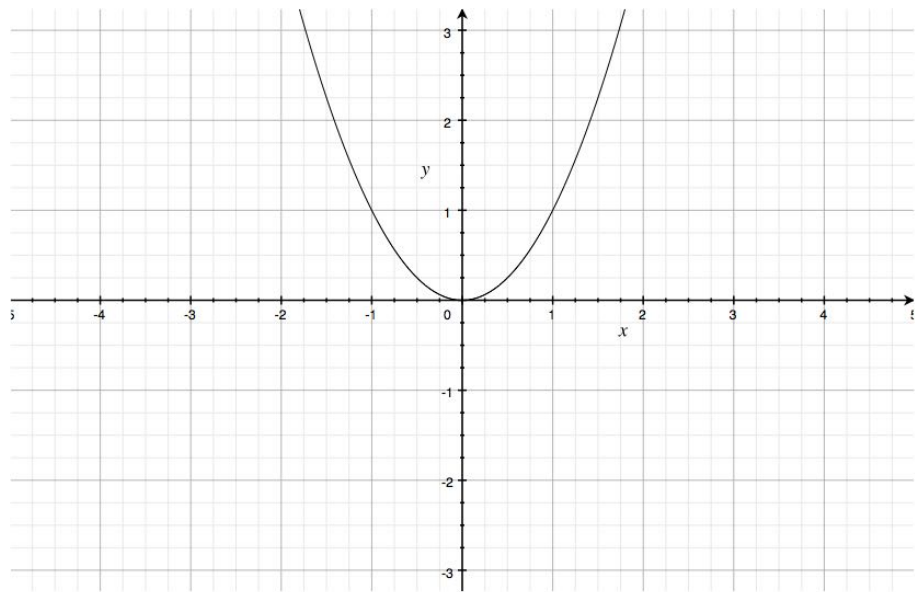


Factor Graph

An Example: $f(x) = x^2$



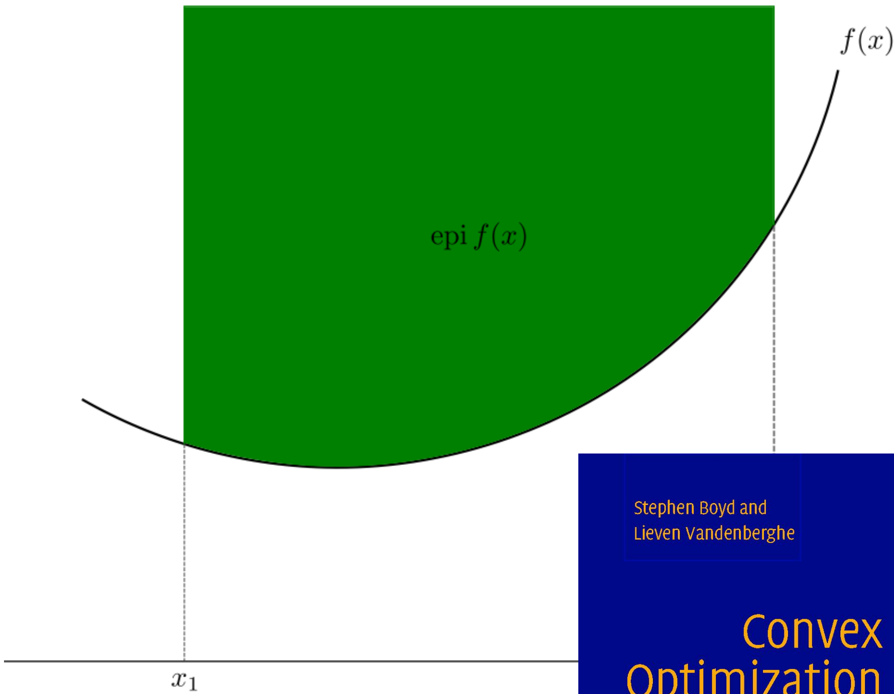
- What is the minimum value?
- What is the value of x that achieves that value?

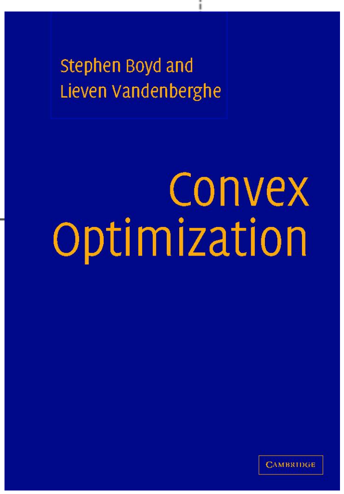


What Makes Optimization Hard

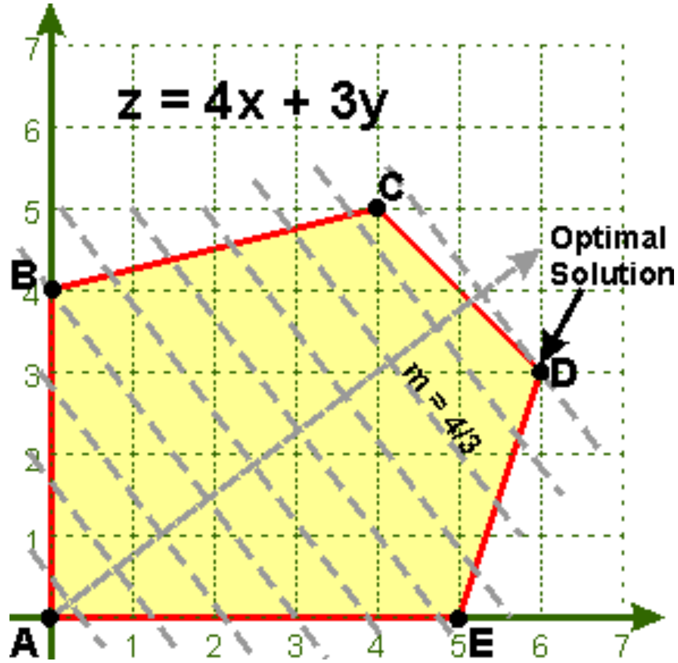
- Finding an appropriate objective function
- Simple: known methods to solve, but may not sufficiently capture your problem
  - Convex, Linear Programming (LP)
- Complex: may be difficult to find a good solution in a reasonable amount of time

"Easy" Formulations

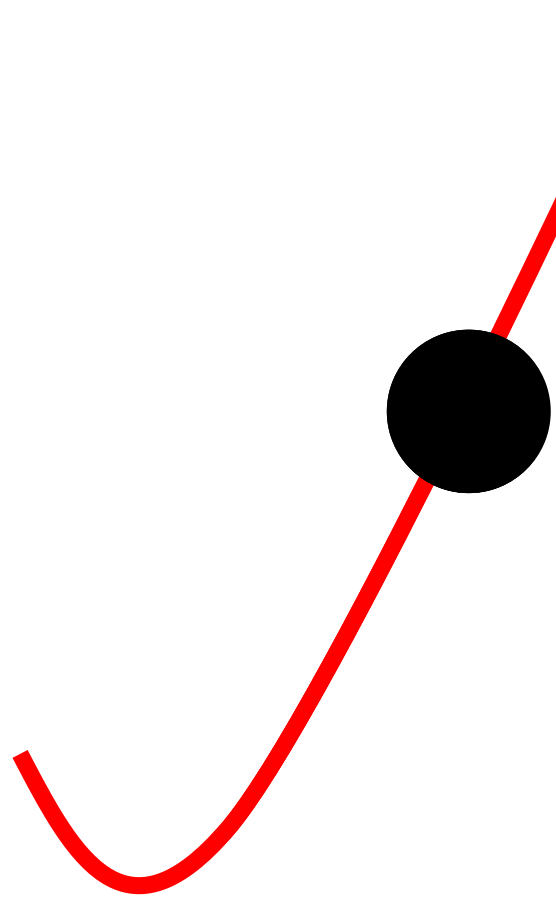




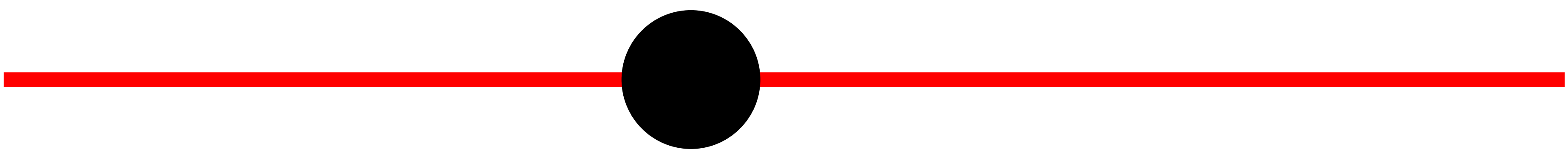
$$
\begin{array}{ll}
\text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{subject to} & A\mathbf{x} \leq \mathbf{b} \\
\text{and} & \mathbf{x} \geq \mathbf{0}
\end{array}
$$



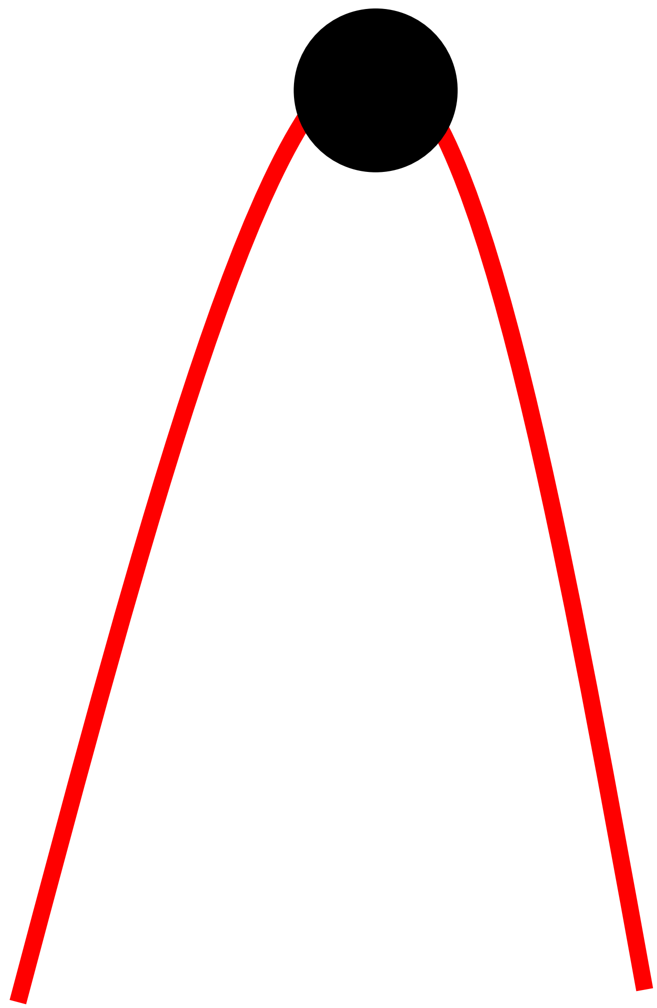
Which Way Do I Go, George?



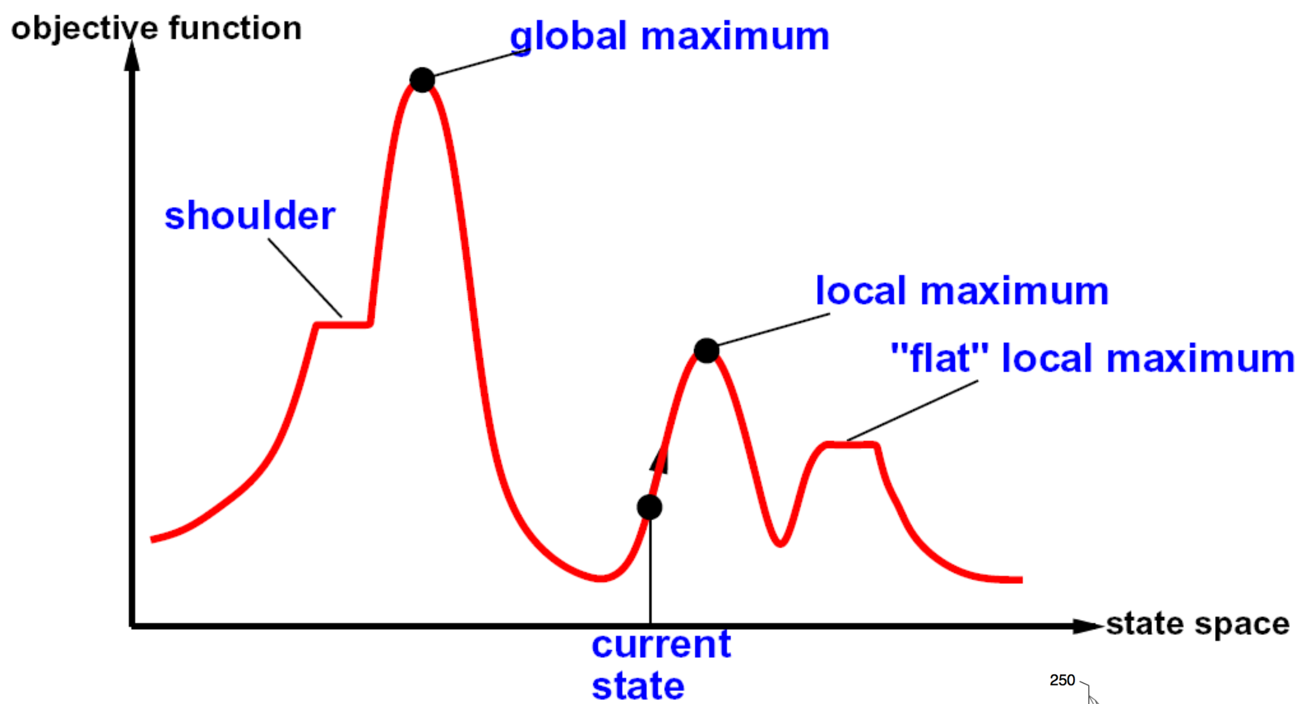
Which Way Do I Go?

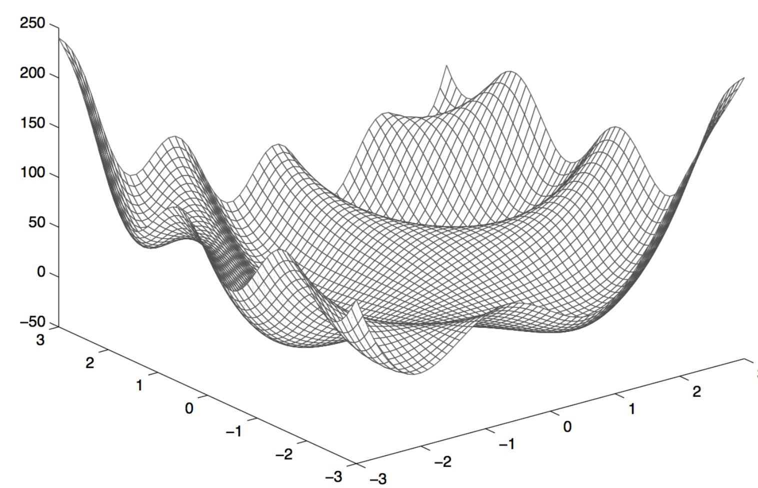


Am I Done Yet?



In General…


objective function
global maximum
shoulder
local maximum
"flat" local maximum
current
state
state space


250
200
150
100
50
0
−50
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

ADMM

Alternating Direction Method of Multipliers [Boyd et al. ‘11]

General

- <u>Arbitrary</u> objective functions, constraints, and variables
- Global minimum for *convex* problems
- If converges, yields a *feasible* solution (all hard constraints met)

Interruptible

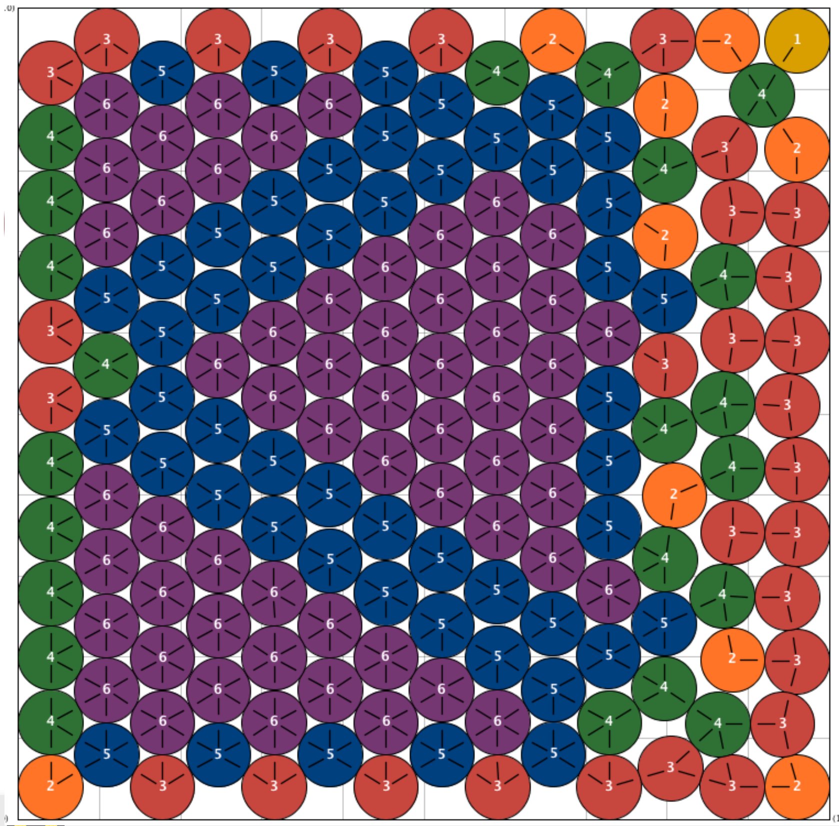
- Iterative algorithm; intermediate results can serve as heuristic start for complementary approaches

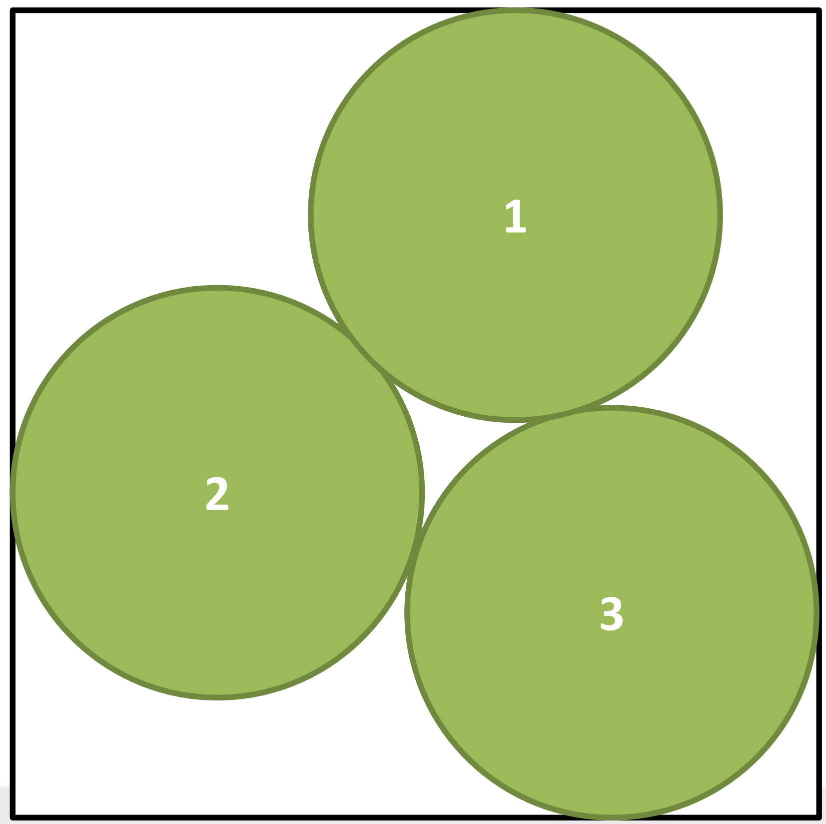
Scalable and Parallelizable

- Naturally concurrent (e.g. multi-core, GPU, MapReduce)

Example Problem: Circle Packing

Fit *n* circles of radius *r* in a square of side-length *s* without overlap (non-convex, NP-hard, ∞ solutions)

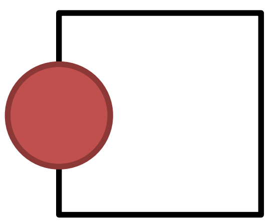




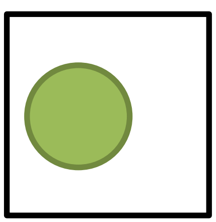
Objective Function

$$\underset{x_1, x_2, x_3}{\text{minimize}} : \text{Box}(x_1) + \text{Box}(x_2) + \text{Box}(x_3) + \text{Collision}(x_1, x_2) + \text{Collision}(x_2, x_3) + \text{Collision}(x_1, x_3)$$

$$\text{Box}(x)$$

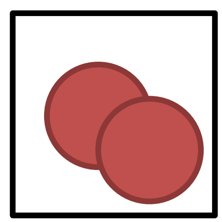


$$= \infty$$

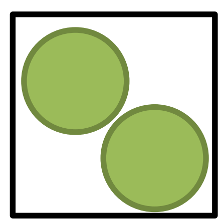


$$= 0$$

$$\text{Collision}(x, x')$$



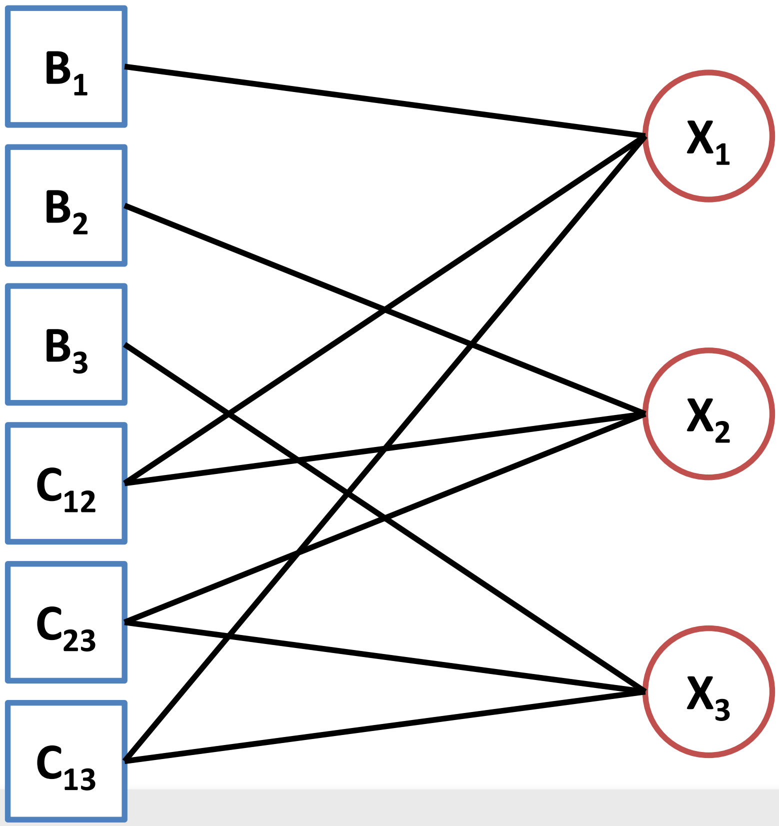
$$= \infty$$



$$= 0$$

Factor Graph

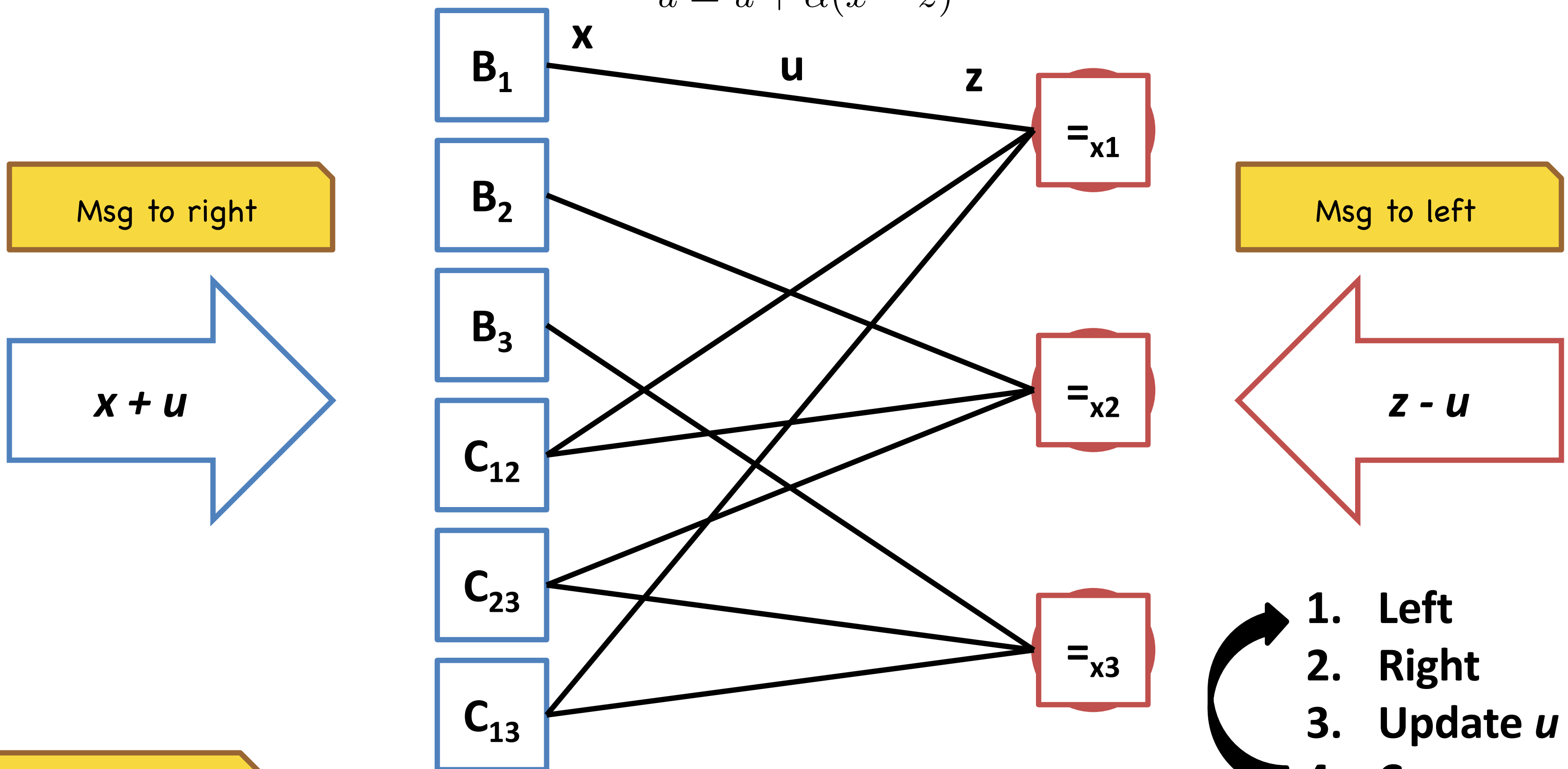
$$\underset{x_1,x_2,x_3}{\text{minimize}} : \text{Box}(x_1) + \text{Box}(x_2) + \text{Box}(x_3) +$$
$$\text{Collision}(x_1, x_2) + \text{Collision}(x_2, x_3) + \text{Collision}(x_1, x_3)$$



Message-Passing ADMM

Cumulative Disagreement

$$u = u + \alpha(x - z)$$



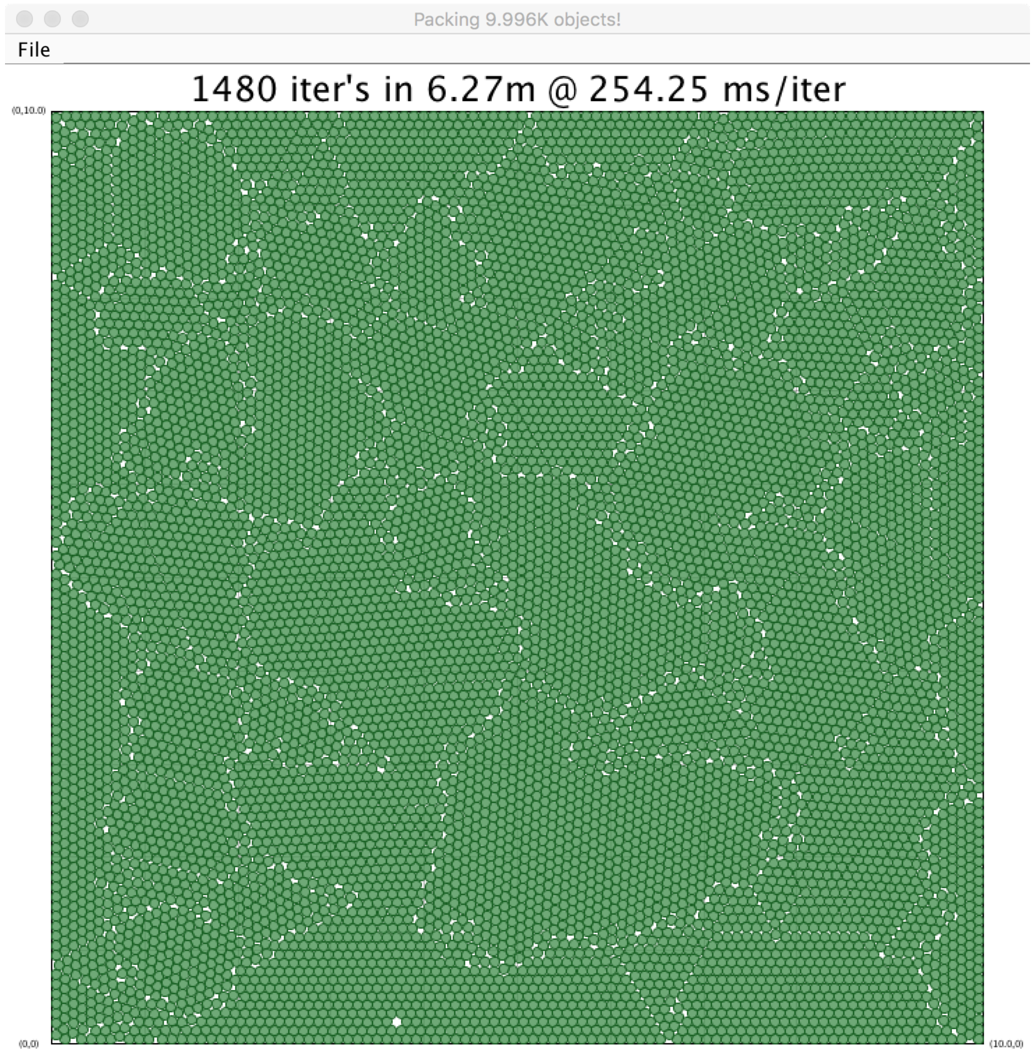
1. Left
2. Right
3. Update ***u***
4. Convergence check

Balance **function cost** and **projection cost** w.r.t. weight$_{in}$

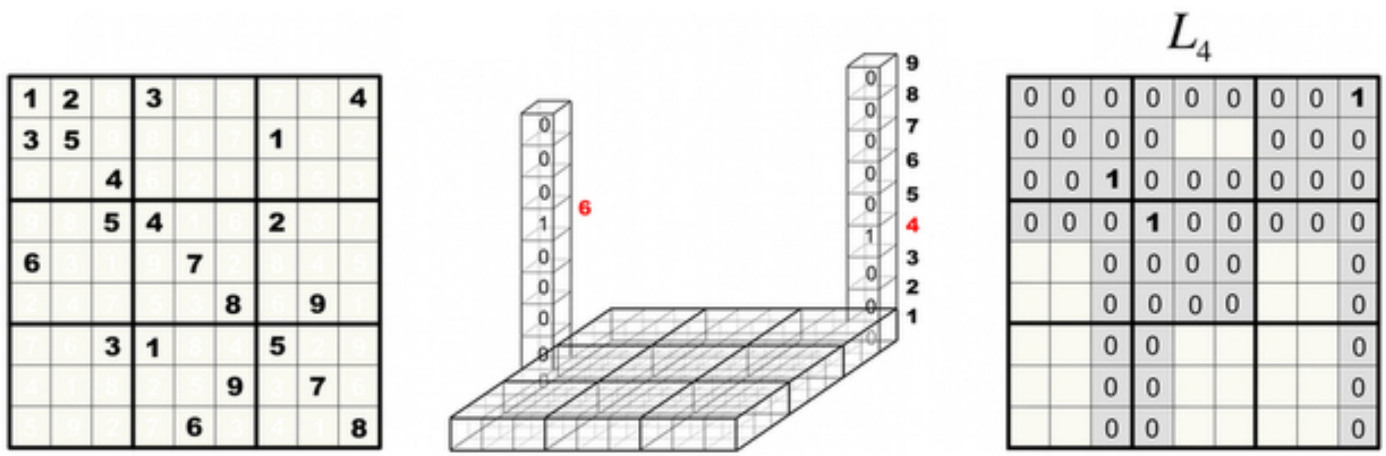
$$\underset{v}{\operatorname{argmin}} \left[f(v) + \frac{weight_{in}}{2}(v - msg_{in})^2 \right]$$

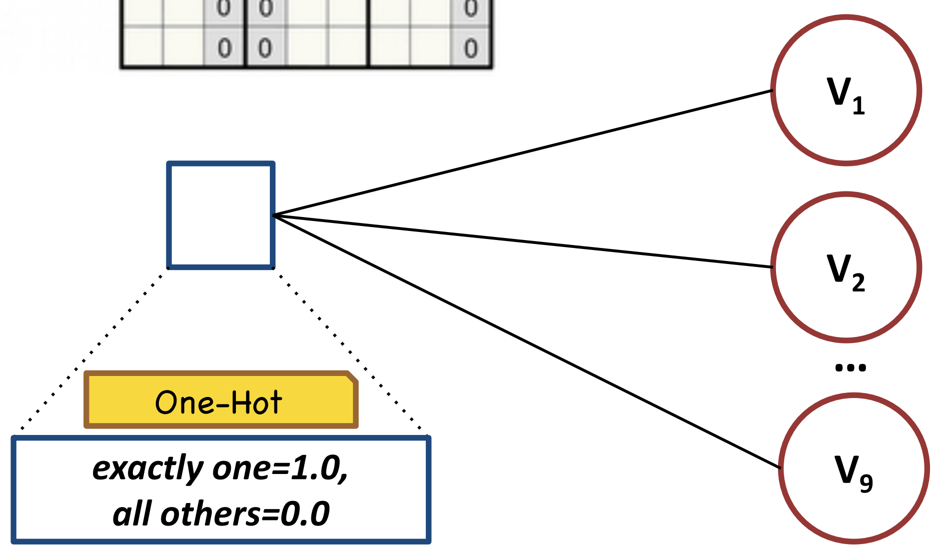
Example Run

Let's Break a World Record Live :)

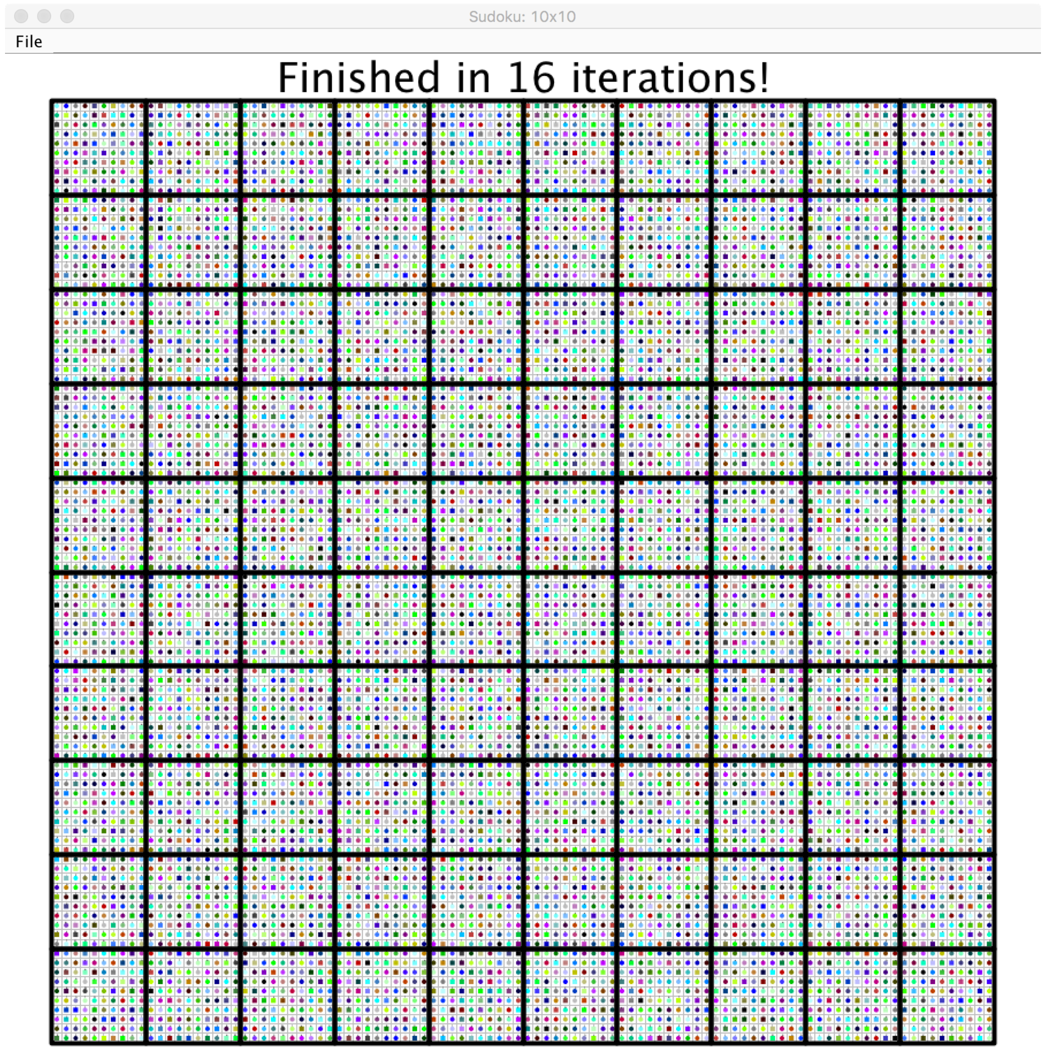


Sudoku is Optimizing Too!



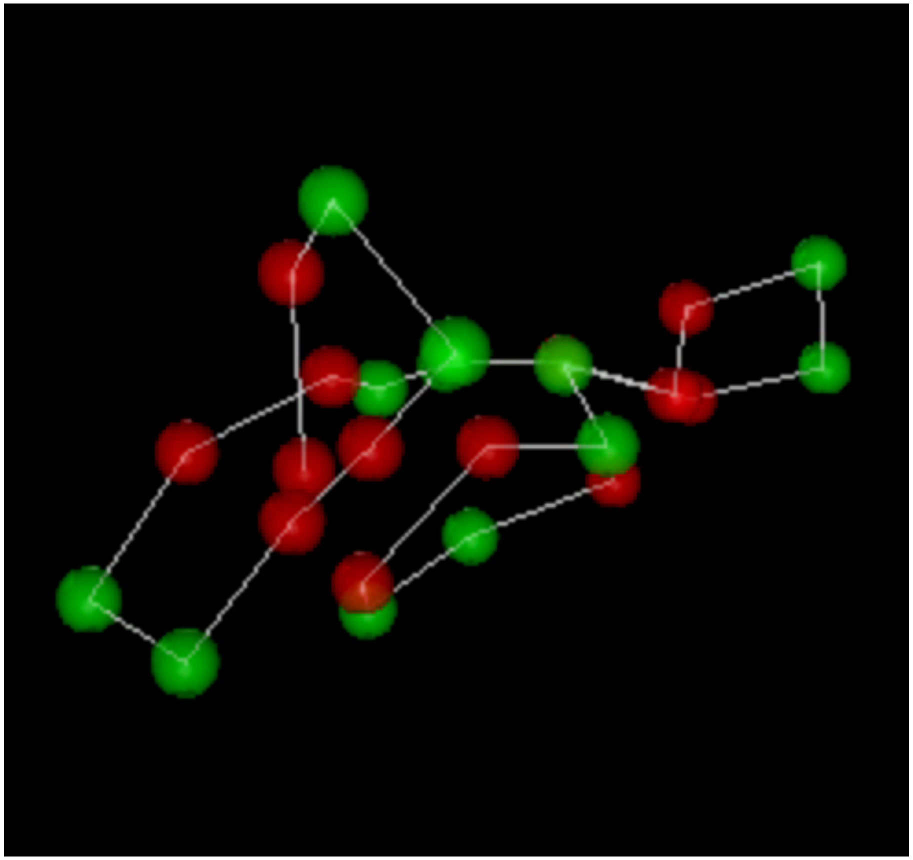


Let's Solve BIG Puzzles :)

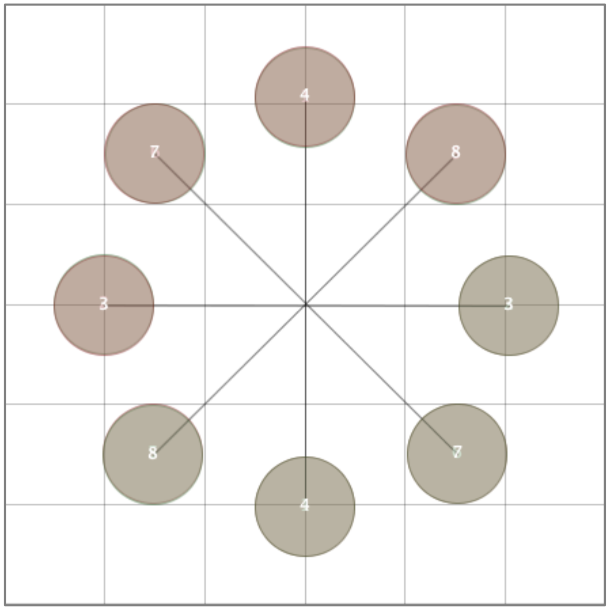


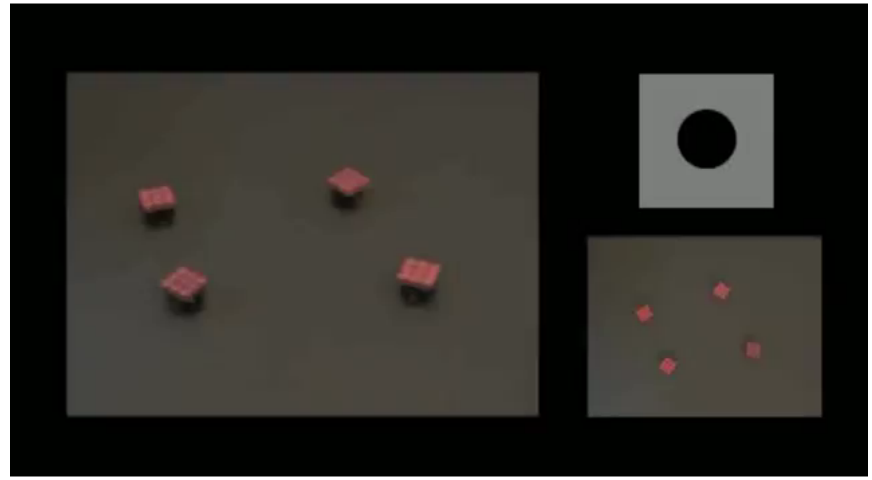
Simple Protein Folding

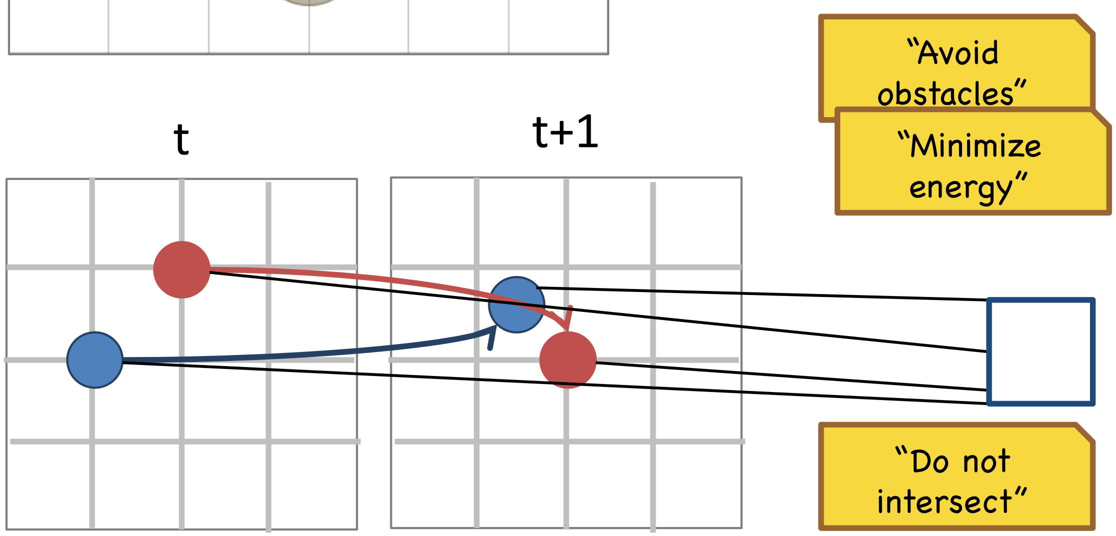
- Variables
  - Particle positions
- Constraints
  - Neighboring distance
  - Energy via distance
  - Total energy < k

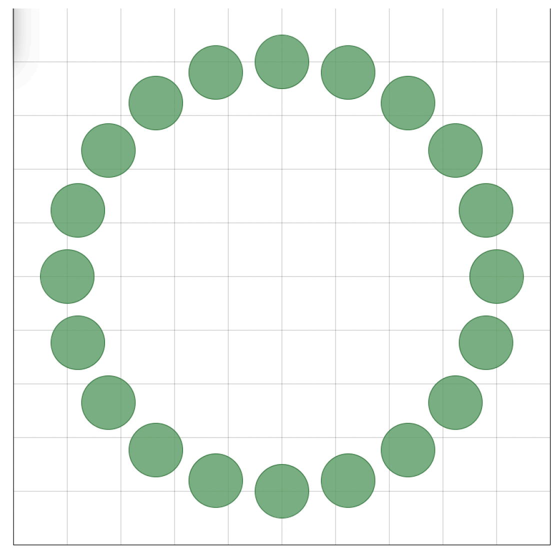


Planning Robot Trajectories





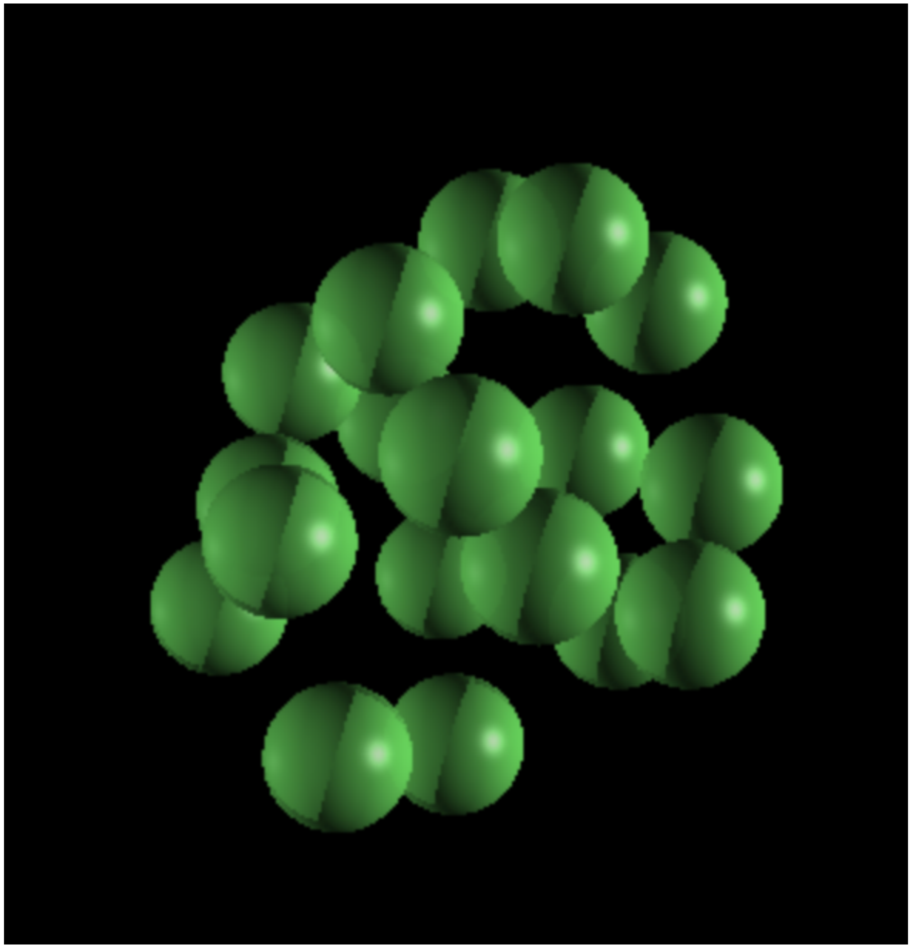




Piecewise Linear: t=1…T

Optimize!

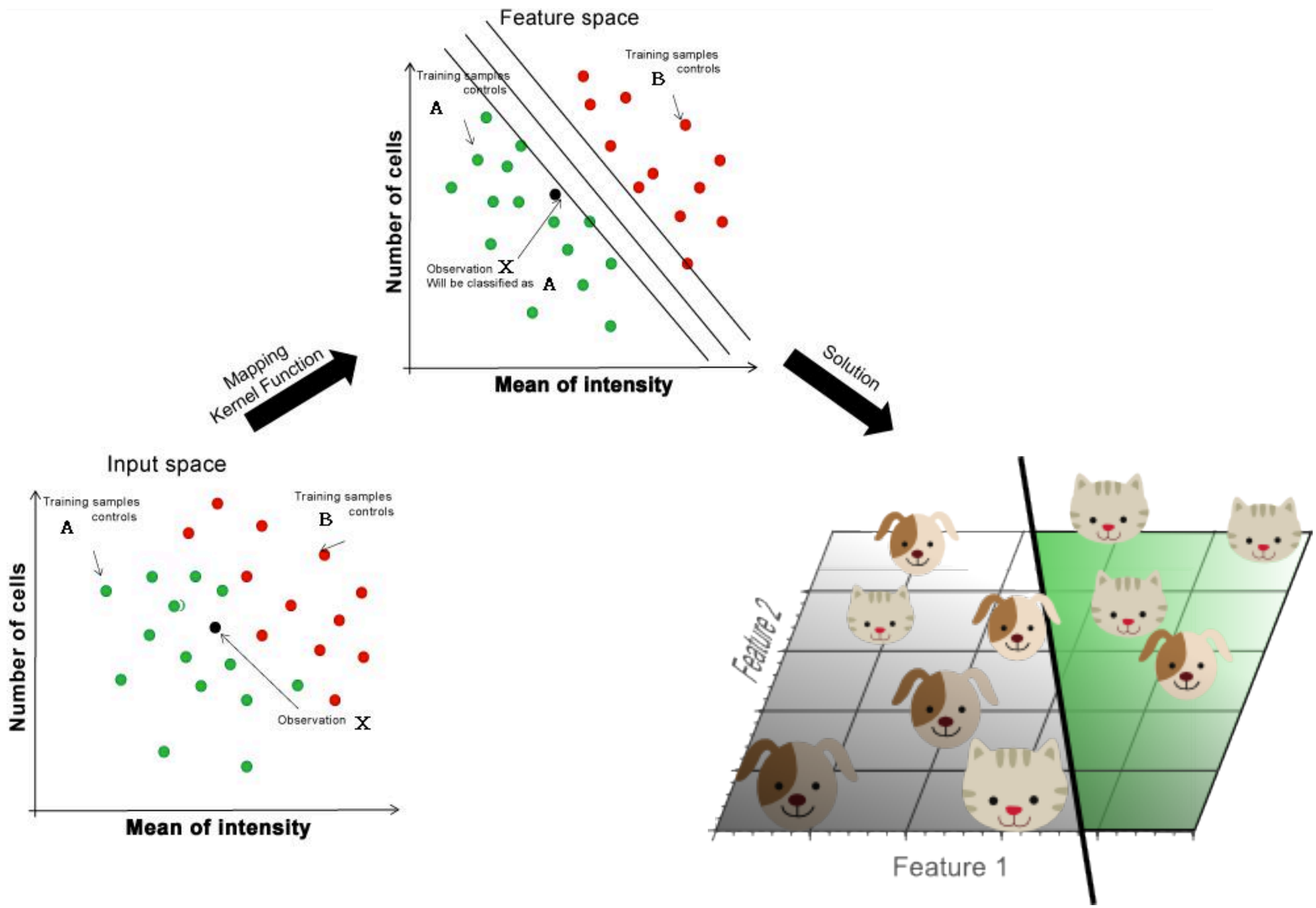
Let's Make Things Interesting :)



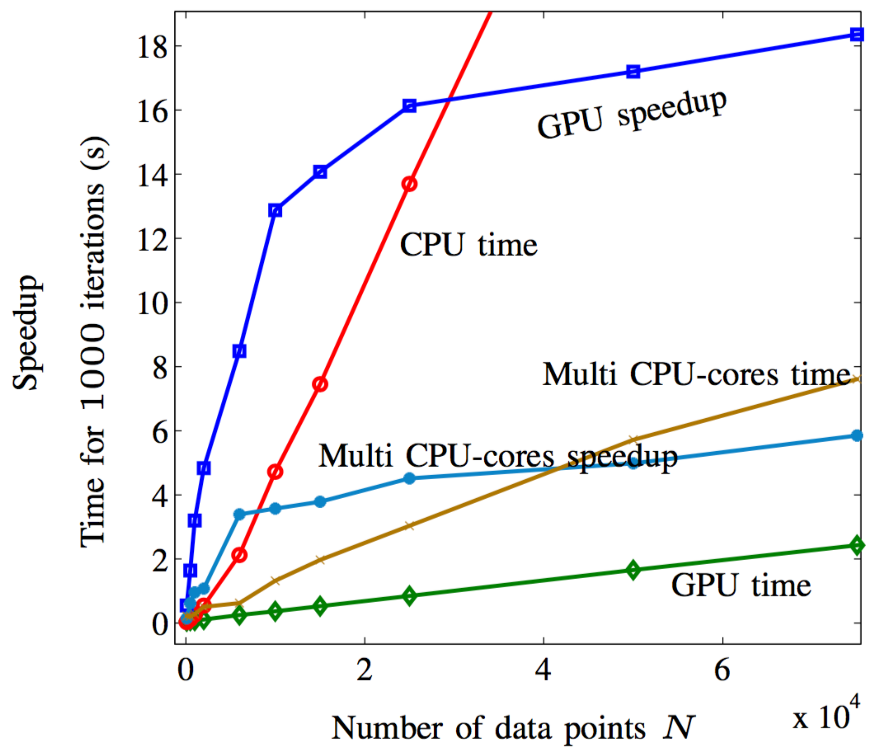
Scaling to Big Problems

- Today we have lots of computing power
  - FPGA, multi-core, GPU, multi-node
- ADMM naturally parallelizes many-variable and many-variable problems
  - Do the left all in parallel, then right
- Tools exist to make it easy to write parallelized versions with serial code

Support Vector Machines (SVM)



SVM Speedup



GPU speedup
CPU time
Multi CPU-cores time
Multi CPU-cores speedup
GPU time
Speedup
Time for 1000 iterations (s)
Number of data points N
x 10^4
0
2
4
6
8
10
12
14
16
18

Takeaways

- Optimization is a useful framework for solving a variety of problems
  - Formulate an objective function
  - Many methods exist for particular types of variables/constraints
- ADMM is an algorithm that has useful properties and easily parallelizes to solve large problems

Thanks :)

Questions?

Javier Alonso-Mora
TU Delft

Charles Mathy
Analog Devices

Jonathan Yedidia
Analog Devices

Veit Elser
Cornell

Ning Hao
Oracle

AmirReza Oghbaee
Northeastern

Mohammad Rostami
UPenn

José Bento
Boston College