

Solving combinatorial problems at scale

Srikrishna Sridhar

www.cs.wisc.edu/~srikris

Joint work with

Victor Bittorf, Ji Liu, Stephen J. Wright, Chris Ré

One Slide Summary

- Applications
- Optimization toolbox Example
 - Linear Programming
 - Support Vector Machines (SVM)
 - Linear Systems
- Asynchronous Algorithms
- Performance Evaluation
- Future work

One Slide Motivation

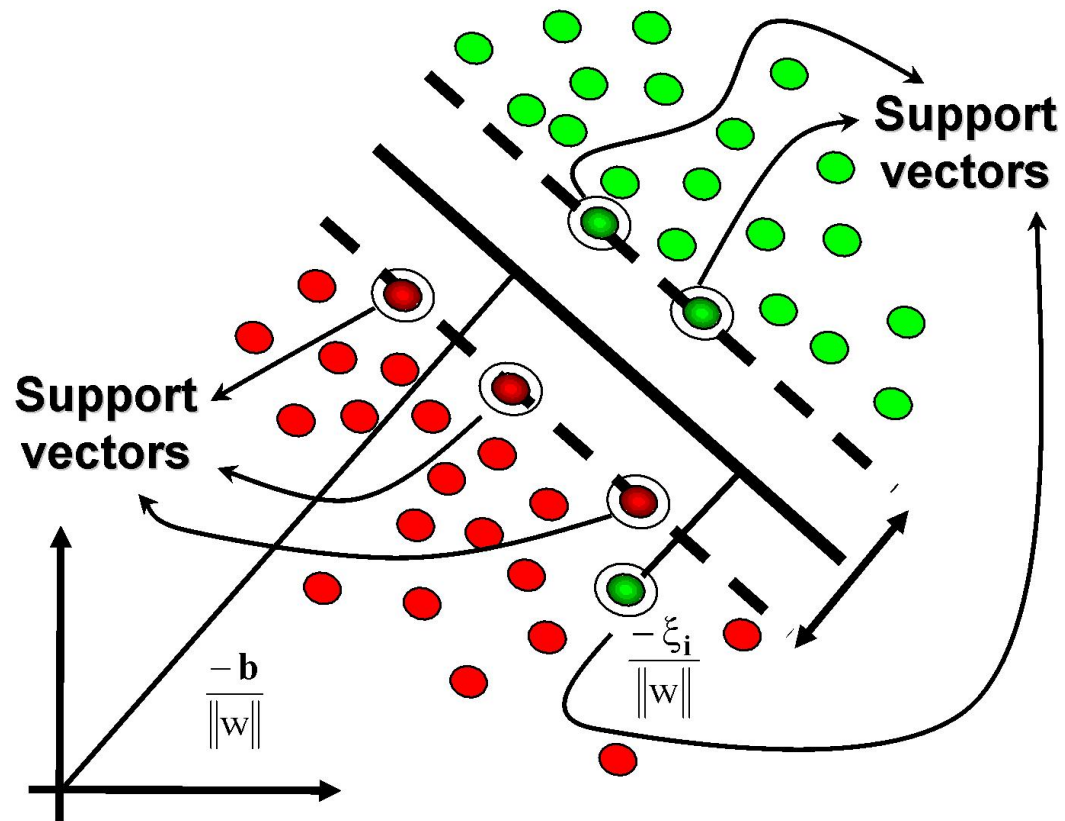
Optimization provides a powerful toolbox for data analysis and machine learning problems.

Systems interactions with optimization research
Multicore and clusters

Take away message

Asynchronous algorithms are the key to speedups in modern architectures.

Classification: SVM

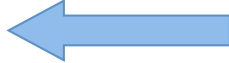


Matrix Completion

Movie



User

[illegible]

Matrix Completion

	4							1	
14									
	4								
52									
4									
					1				
1	2		4						
5									
								2	

=

Linear Systems

Seismic imaging process

1. Data collection from seismic survey
2. Data pre-processing
3. Choosing modeling methodology
4. Inverse model to fit data

$$Ax = b$$

Slide Source: Rashmi Raghu

Combinatorial Optimization

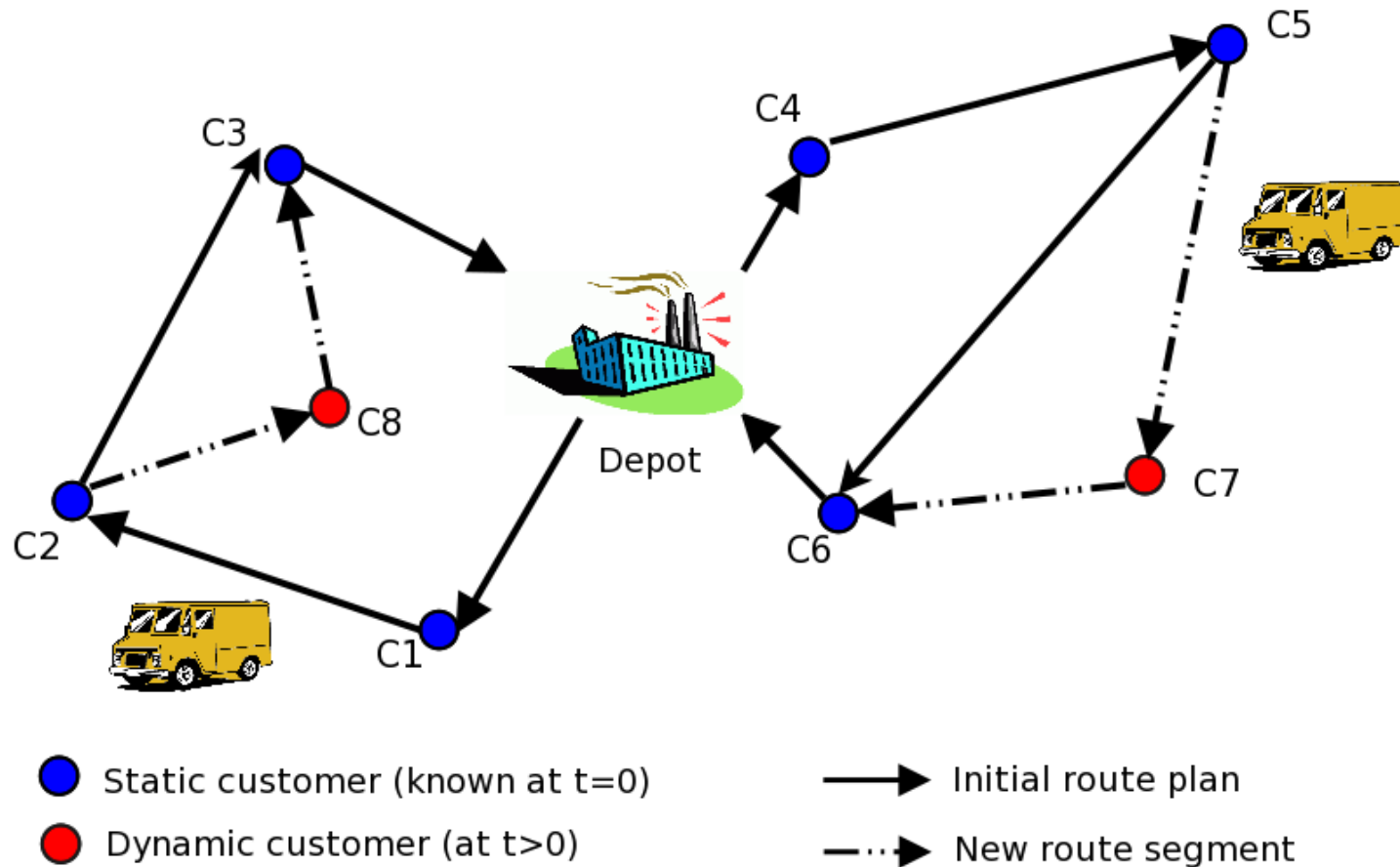
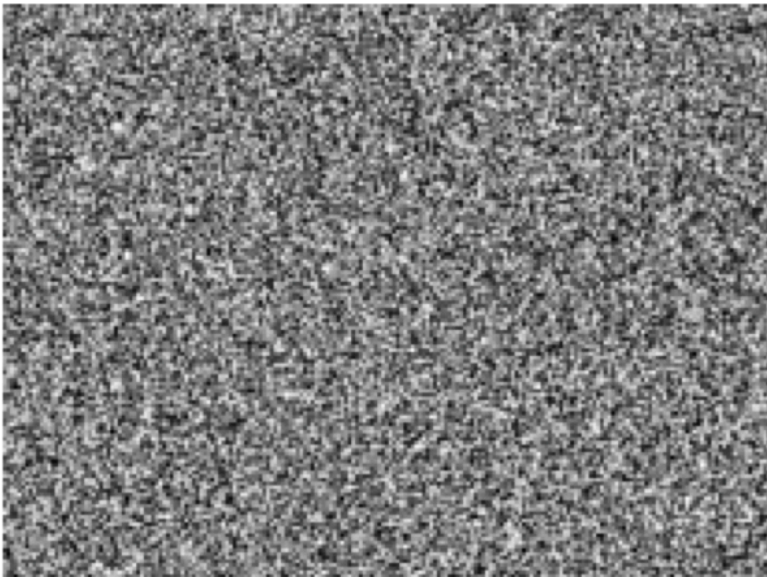


Image Processing

Pictures of natural objects are not random!
They usually have areas of non-constant intensity with sharp edges.



Optimization

Several problems in data analysis, machine learning and predictive analytics can be posed as an optimization problem

Optimization 101

Data: Observations (typically known) which we used to aid decision making.

Decision Variables: The set of decisions that we are seeking to optimize

Objective: A mathematical quantification of the quality of outcomes made by decisions.

Model: The relationship between the decisions we are trying to make, the outcome and the

Optimization in Analytics

- **BIG DATA**
- **Iterative:** Need to make several passes over the data.
- **Accuracy:** Sometimes, approximate is good enough.
- **Structure:** Seek solutions with a desired structure (like simplicity, robustness etc.)

How big is BIG DATA?

Sl.	Variables	Size	Solve Time (s)			
			Cplex(B)	Cplex(S)	Gurobi(B)	Gurobi(S)
1	123	0.2 MB	0.031	0.065	0.081	0.03
2	12596	1.2 GB	5882.5	1690.8	3001.1	2707.8
3	129136	125 GB	--	--	--	--

S: Simplex

B: Barrier (Interior point method)

Note: -- indicates timed out at 2 hours

How big is BIG DATA?

Sl.	Variables	Size	Solve Time (s)			
			Cplex(B)	Cplex(S)	Gurobi(B)	Gurobi(S)
1	123	0.2 MB	0.031	0.065	0.081	0.03
2	12596	1.2 GB	5882.5	1690.8	3001.1	2707.8
3	129136	125 GB	--	--	--	--

S: Simplex

B: Barrier (Interior point method)

Note: -- indicates timed out at 2 hours

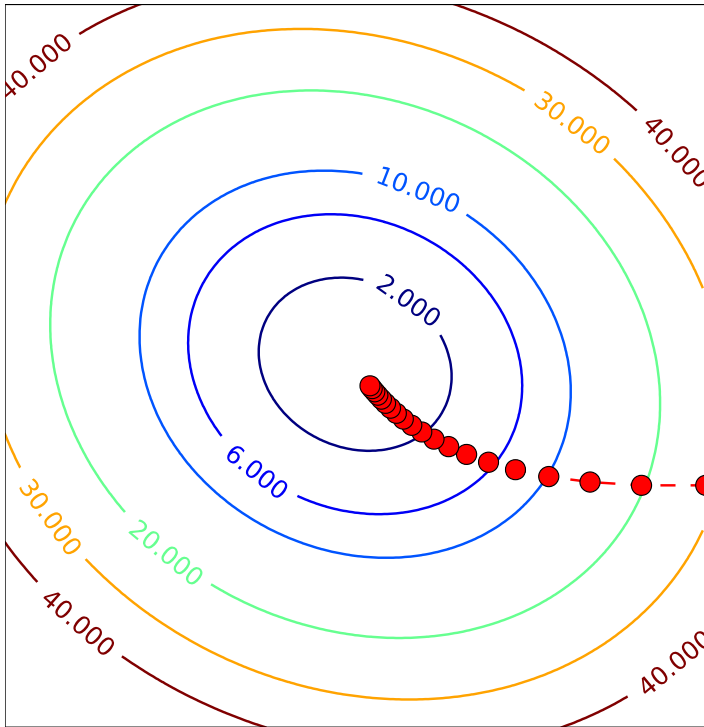


How big is BIG DATA?

Sl.	Variables	Size	Solve Time (s)			
			Cplex(B)	Cplex(S)	Gurobi(B)	Gurobi(S)
1	123	0.2 MB	0.031	0.065	0.081	0.03
2	12596	1.2 GB	5882.5	1690.8	3001.1	2707.8
3	129136	125 GB	--	--	--	--



Gradient Descent Methods



$$\min f(x)$$

Gradient

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Step Size

Gradient Methods

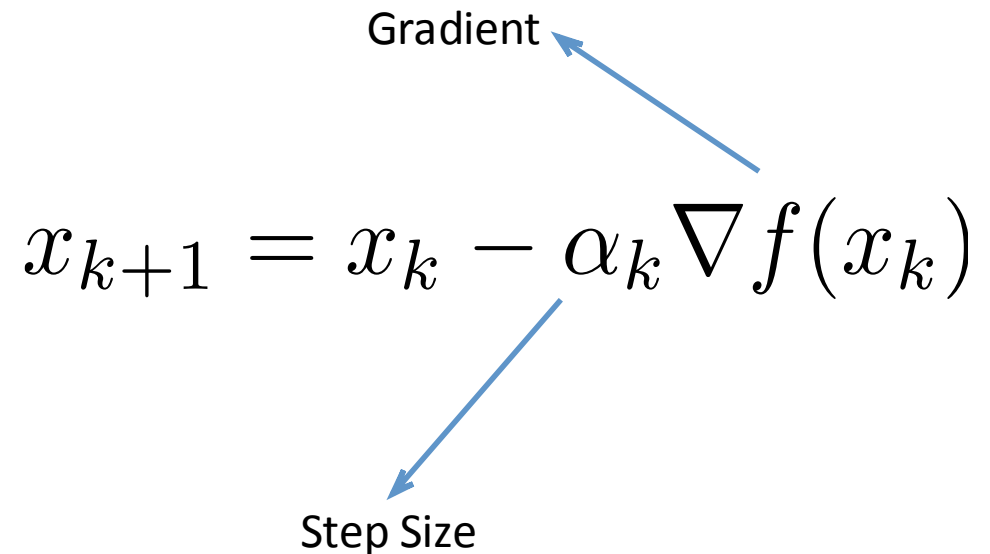
- Computing the gradient requires an **entire pass** on the data!
- We need to make several passes over the data.
- Hard to parallelize

$$\min f(x)$$

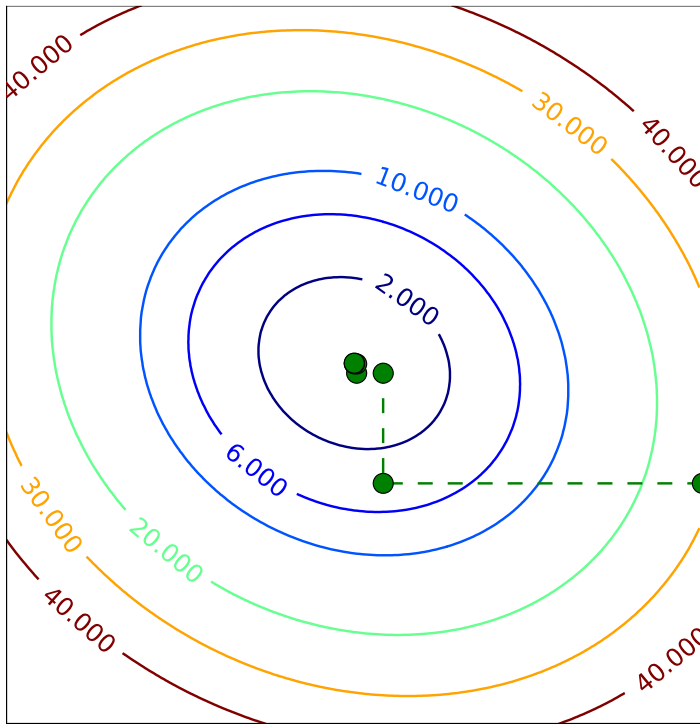
$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

Gradient

Step Size

A diagram illustrating the gradient method update equation. The equation is $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$. A blue arrow points from the word "Gradient" to the term $\nabla f(x_k)$. Another blue arrow points from the text "Step Size" to the term α_k .

Co-ordinate Descent Methods (Nestrov 2010)



$$\min f(x)$$

Gradient along a co-ordinate

$$x_{k+1}^i = x_k^i - \alpha_k [\nabla f(x_k)]_i$$

Update only a single co-ordinate

Co-ordinate Descent Methods

- Computing the ‘partial’ gradient requires only a **single row (or column)** of the data!
- Converges slowly. We need several passes over the data.
- Possible to parallelize

$$\min f(x)$$

Gradient along a co-ordinate

$$x_{k+1}^i = x_k^i - \alpha_k [\nabla f(x_k)]_i$$

Update only a single co-ordinate

Co-ordinate Descent Methods

- Computing the ‘partial’ gradient requires only a **single row (or column)** of the data!
- Converges slowly. We need several passes over the data.
- Possible to parallelize

$$\min f(x)$$

Gradient along a co-ordinate

$$x_{k+1}^i = x_k^i - \alpha_k [\nabla f(x_k)]_i$$

Update only a single co-ordinate

Stochastic Gradient Methods

- Computing the partial gradients are cheap!
- Converges very slowly. We need several passes over the data.
- Possible to parallelize

$$\min \sum_{i=1}^n f_i(x)$$

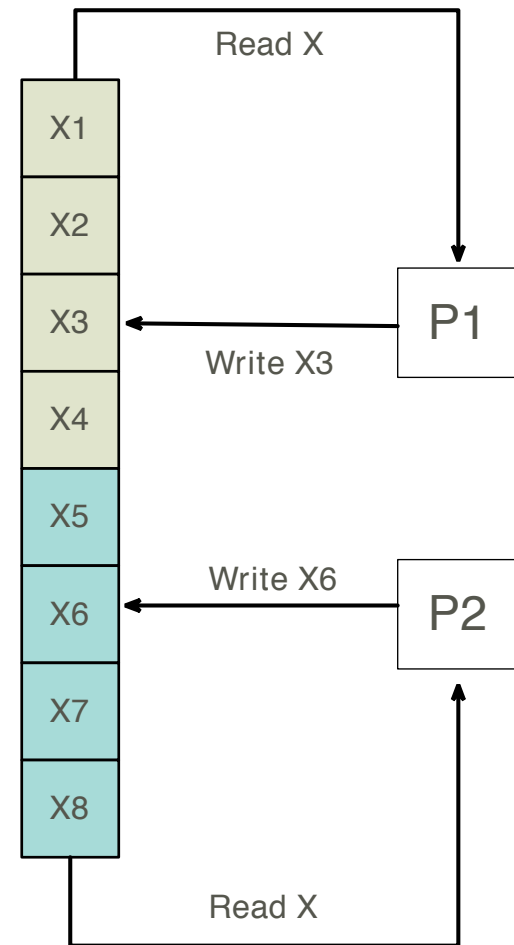
Gradient of a single function

$$x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$$

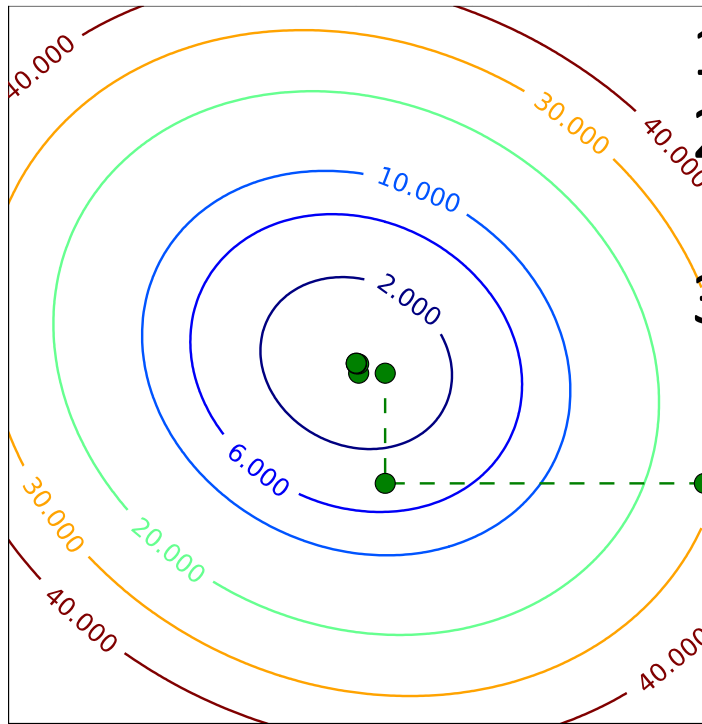
Update only a single co-ordinate

Parallel Co-ordinate Descent Methods

- Each core grabs a centrally located x , evaluates the gradient and then writes the update back to x .
- Updates may be old by the time they are applied.
- Processors don't overwrite each other's work.



Parallel Co-ordinate Descent Methods



Each processor

1. Pick a co-ordinate **i**
2. Read the current state of **x_k**
3. Compute the gradient along the

$$[\nabla f(x_k)]_i$$

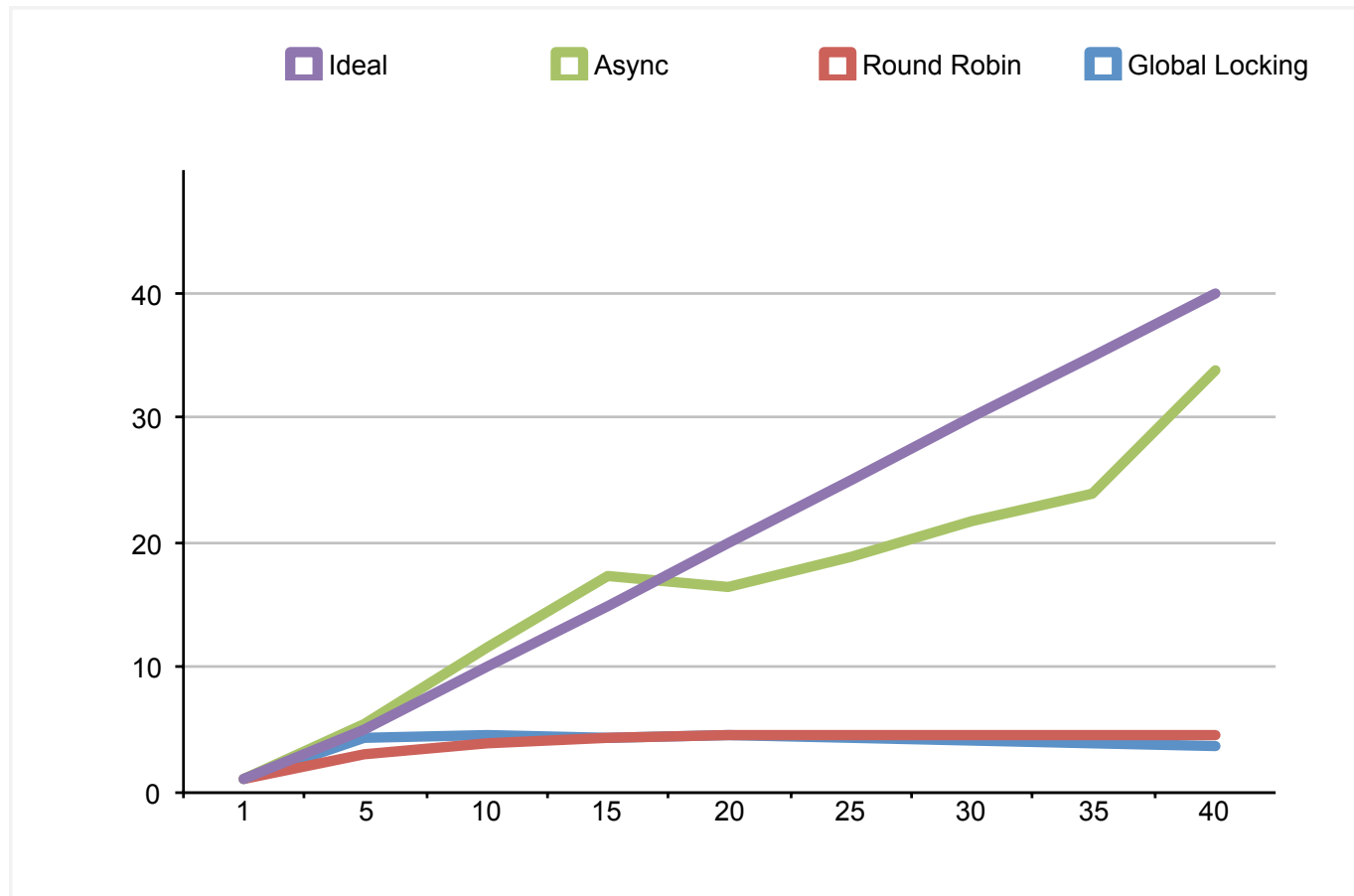
Parallel Co-ordinate Descent Methods

Global locking: Lock the shared memory x for reading and writing operations. Cores acquire the lock in a round robin fashion. (Langford 2009)

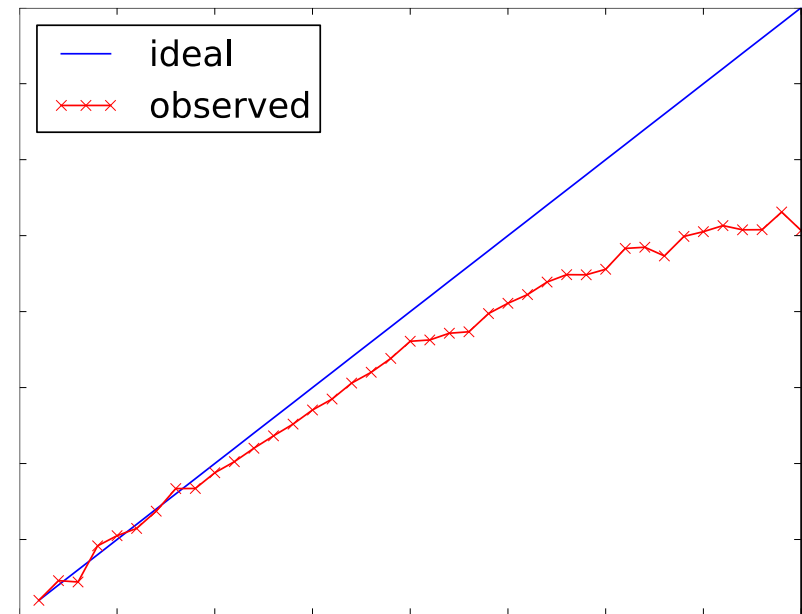
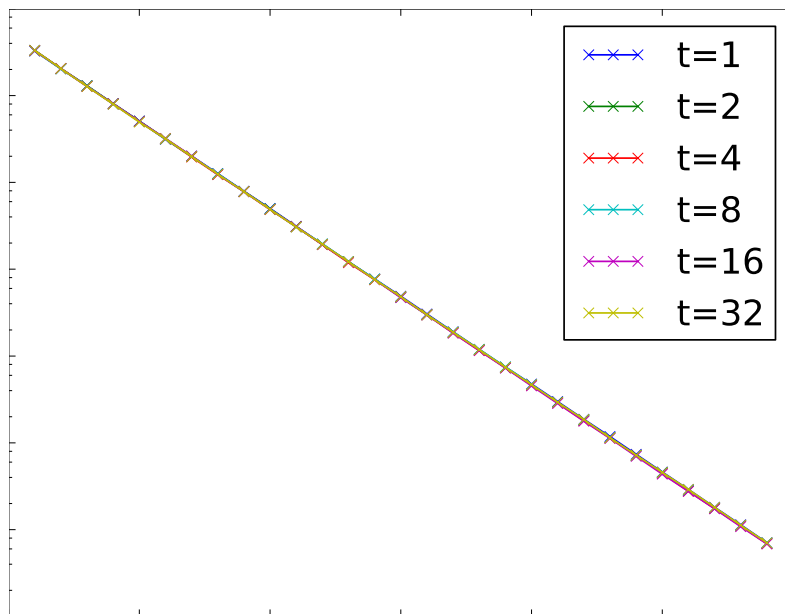
Global locking: Lock the shared memory x for reading and writing operations. Cores acquire a lock in any order.

Asynchronous: No locking! Cores may overwrite

Comparison of Parallel SCD schemes



Algorithmic & Implementation Speedups



What does Async buy us?

Sl.	Variables	Size	Solve Time (s)				
			Cplex(B)	Cplex(S)	Gurobi(B)	Gurobi(S)	Us
1	123	0.2 MB	0.031	0.065	0.081	0.03	0.05
2	12596	1.2 GB	5882.5	1690.8	3001.1	2707.8	6.9
3	129136	125 GB	--	--	--	--	686.9

S: Simplex

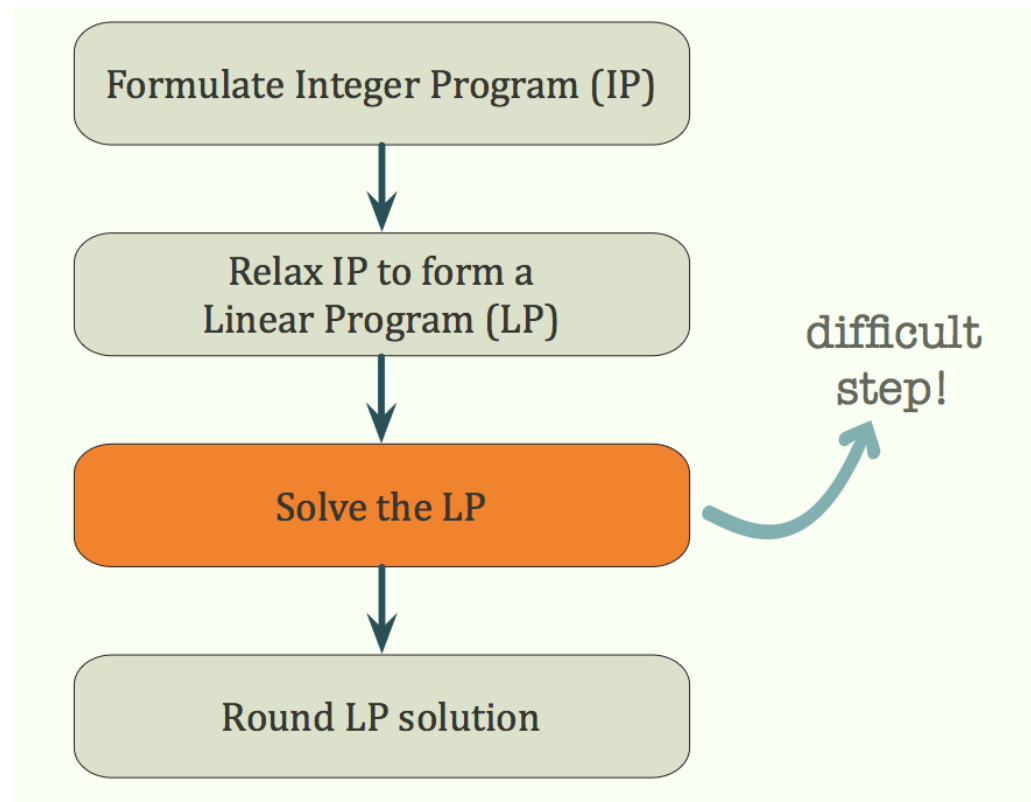
B: Barrier (Interior point method)

Note: -- indicates timed out at 2 hours

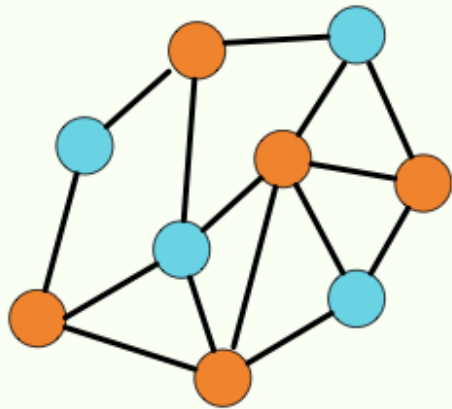
Extreme Linear Programming

Can we leverage async algorithms for large scale combinatorial problems?

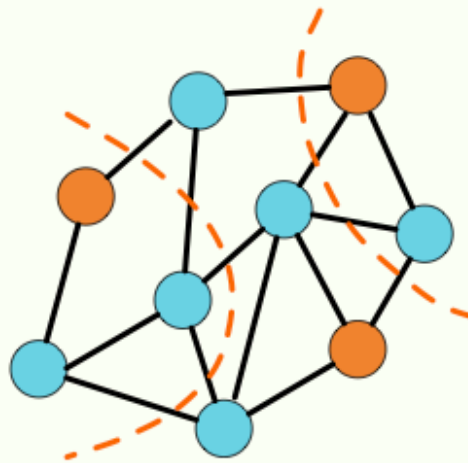
LP Rounding



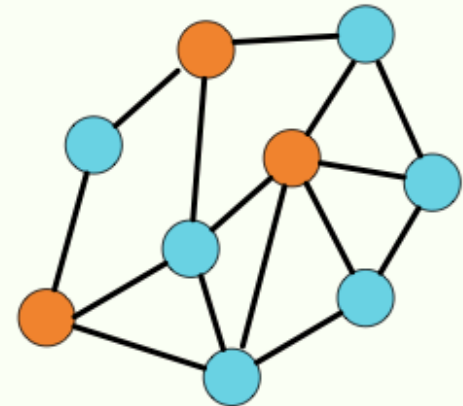
LP Rounding: Examples



vertex cover

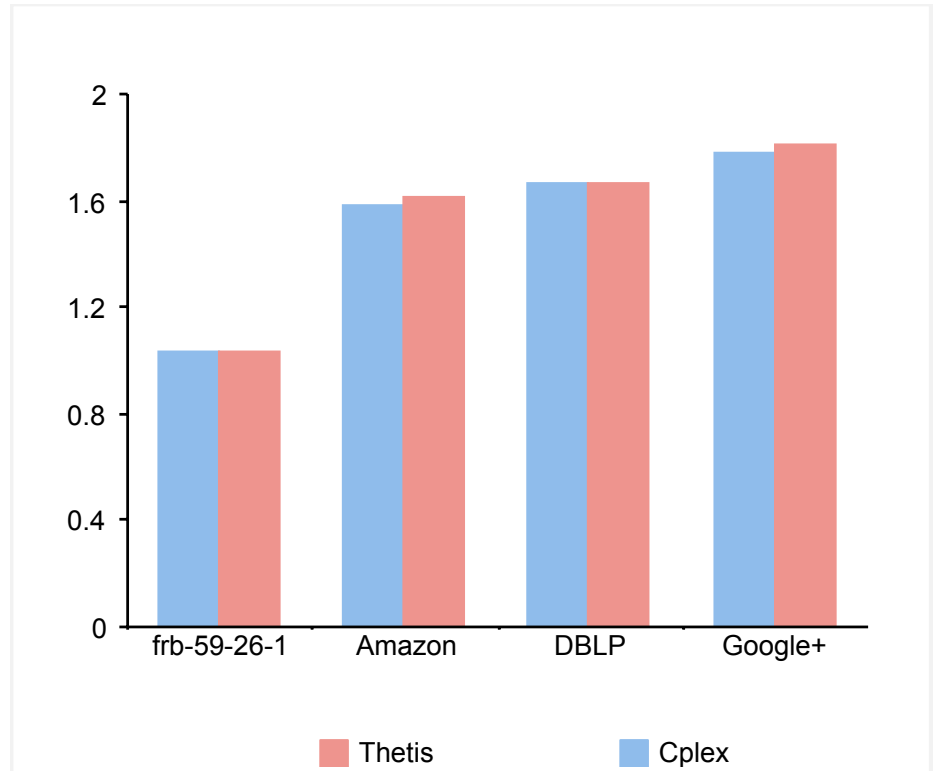
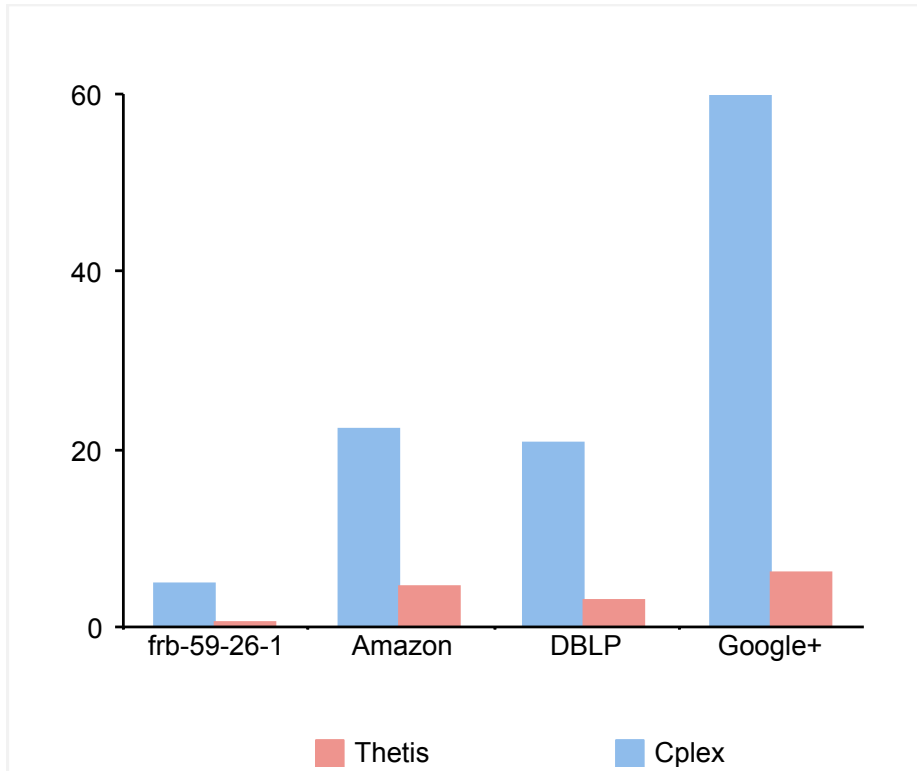


multiway cut



independent set

Results: Quality & Speed



Solution quality = Rounded solution / Optimal solution
Results reported on Vertex cover

Results: Quality & Speed

instance	type	Cplex IP	Cplex LP	Us
frb59-26-1	VC	-	5.1	0.65
Amazon	VC	44	22	4.7
DBLP	VC	23	21	3.2
Google+	VC	-	62	6.2
LiveJournal	VC	-	-	934
frb59-26-1	MC	54	360	29
Amazon	MC	-	-	131
DBLP	MC	-	-	158
Google+	MC	-	-	570

Note: - indicates timed out at 1 hours

Conclusion

Optimization provides a powerful toolbox for data analysis and machine learning problems.

Take away message

Asynchronous algorithms may be the key to speedups in modern architectures.