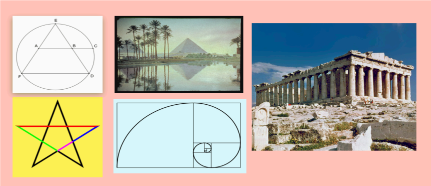
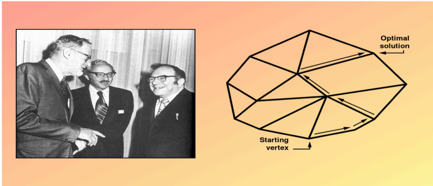
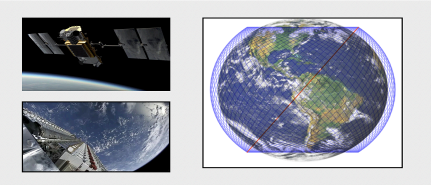
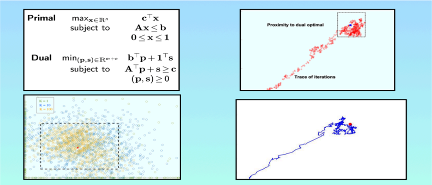
From 0.618 to Mathematical Optimization

Yinyu Ye, Stanford University

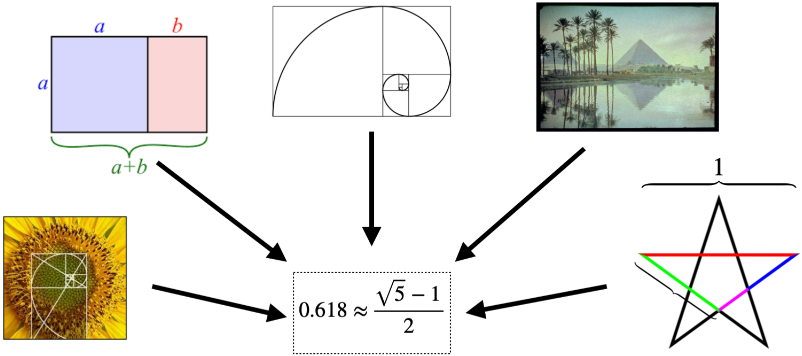








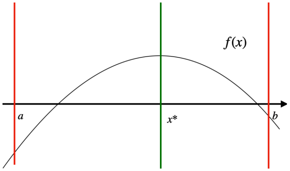
0.618 is Around Us: Art and Science

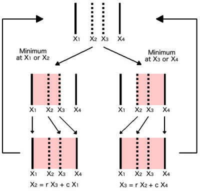


$$0.618 \approx \frac{\sqrt{5}-1}{2}$$

0.618 and Optimization

Finding the maximum/minimum of a unimodal function:
Optimal strategy: using golden ratio 0.618





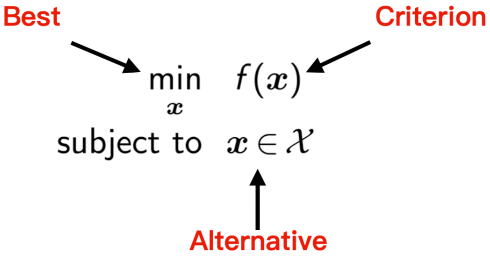
- Optimal strategy/decision is desired
- Optimization finds such decision
- Optimization is *natural* and *intuitive*

Table of Contents

What is Optimization?

Definition

Mathematical optimization is the selection of a best element, with regard to some criterion, from some set of available alternatives.



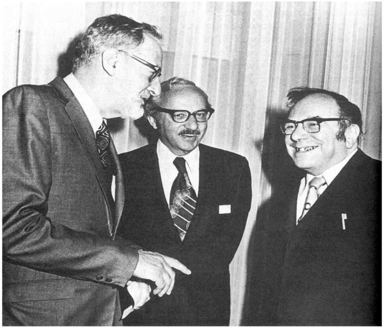
Mathematical Optimization deals with different f and $\mathcal{X}$

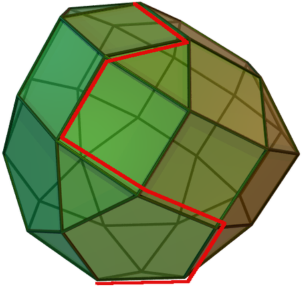
Advance of Optimization in the Last Century

Johan Ludwig William Valdemar Jensen	Harris Hancock	Leonid Kantorovich	John von Neumann	George Bernard Dantzig
				
Convex Functions	Theory of Maxima and Minima	LP–model and Algorithm for solving it	Duality for Linear Programming	Simplex Method Father of Linear Programming

Breakthrough in the 20th century: Linear Programming

George Dantzig and the Simplex Method



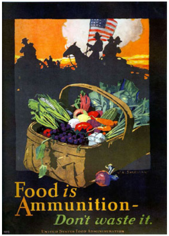


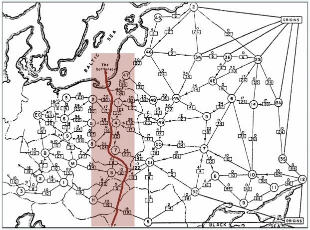
In the simplex method, we move among the vertices until reaching the optimal basic feasible solution

Optimization in the Early Days

Optimization was initially applied to

- Small or medium-sized problems
  hundreds of variables
- Simple scenarios
  dieting, transportation...
  mostly Linear Programs



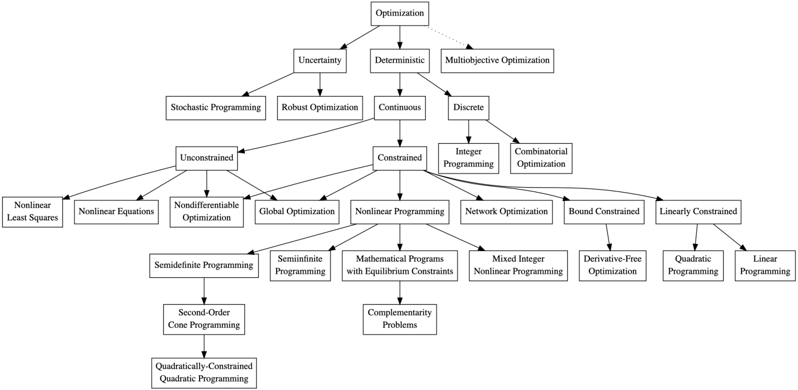


But now, things have gone far beyond simple LPs

Millions of Variables + Various Optimization Models

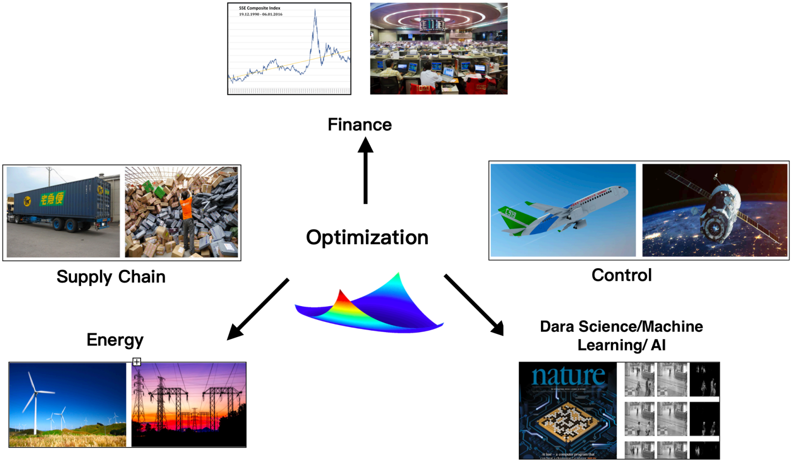
Types of Optimization Problems by Now

Beyond LP, there are now many new mathematical optimization models.



They solve models arising from different kinds of applications

Optimization Nowadays



Optimization is Everywhere

Table of Contents

Mixed Integer Programming and Combinatorial Optimization

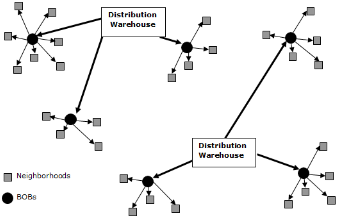
Facility Location and Vehicle Routing Problem

Facility Location Problem

Senario

- Potential sites for locating facilities
- Demands from consumers must be covered
- Total cost (transportation + opening) should be minimized

minimize	Total cost
subject to	Demand of customers
Decision	Whether to open
	$x \in \{\text{No}, \text{Yes}\} \Rightarrow \{0, 1\}$

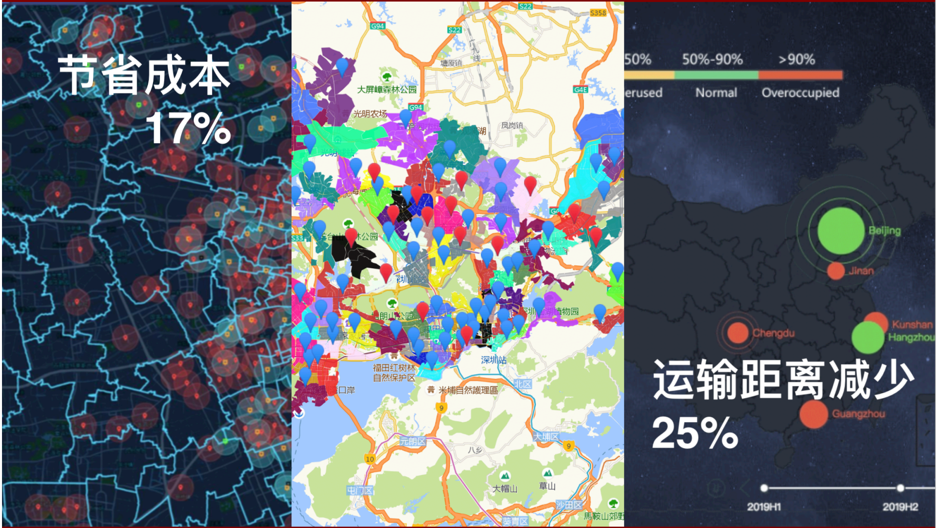


The problem

- is NP-Hard
- can be formulated as an Integer Program

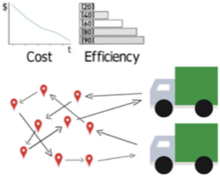
智能门店/网点选址优化系统

杉数科技 Cardinal Operations



Vehicle Routing Problem

There is growing need for efficient traveling.



How to get the most efficient routing strategy?

Vehicle Routine Problem

What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?

$$\begin{array}{rc} \text{minimize} & \text{Total Travelling Distance} \\ \text{subject to} & \text{Arrival at Destination} \\ \text{Decsion} & \text{Whether to go} \\ & \boldsymbol{x} \in \{\text{No}, \text{Yes}\} \Rightarrow \{0, 1\} \end{array}$$

The problem as well as its variants

- are NP-hard
- possibly solved efficiently using specially designed algorithms

Application: Truck Scheduling System

福佑卡车
FUYOUKACHE.COM

To Accelerate Freight Transportation in China

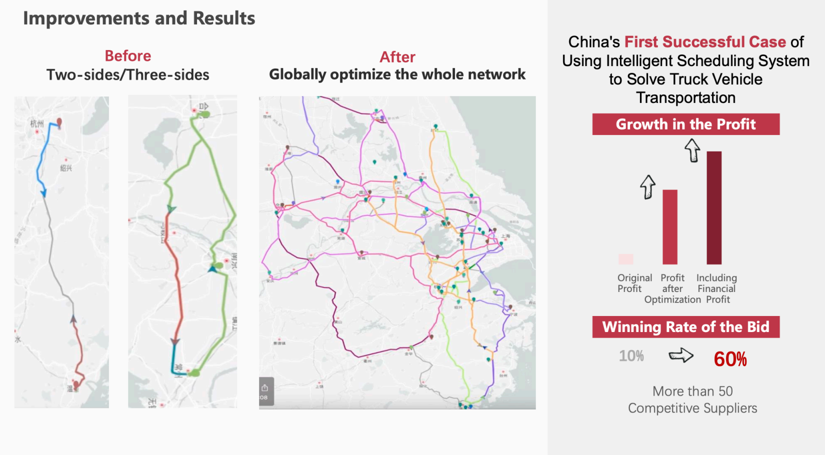
ForU chose to work with intermediaries in the transport sector so that drivers can get orders via agents and conventional intermediaries can get more orders.

ForU sent operation team to monitor each deal to avoid possible corruption problem between shipper representative and drivers, which bought true value for its clients and marked its core competence was the offline operation ability.

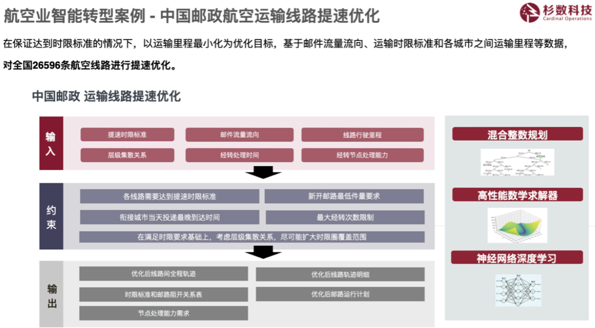
508,593	83,638	507,294	4.2 Billion
Truck Drivers	Clients	Orders	Revenue per Year

Strategy and Improvement

Real-time Scheduling + Multi-objective Optimization



Application: Chinese Postman Problem



中国邮政
CHINA POST

- A special case of vehicle routing
  sending mails to the destinations
- solved efficiently by state-of-the-art solvers
  COPT, Gurobi, CPLEX

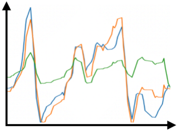
Robust Optimization

Robust Optimization and Uncertainty

Uncertainty and Robust Optimization

Many real-life applications are faced with uncertainty

- Production scheduling
  how much is needed?
- Autonomous driving
  will there be pedestrians?



Optimal decision for one scenario may be unacceptable for another

- High cost of a sub-optimal decision
  danger in driving, loss of sales...
- We need an acceptable solution in all the cases

Solution: We optimize against the controllably worst situation

Uncertainty and Robust Optimization

- A solution that works in the worst case works in all cases
- We apply Distributionally Robust Optimization

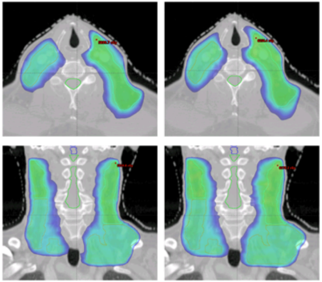
(Distributionally) Robust Optimization

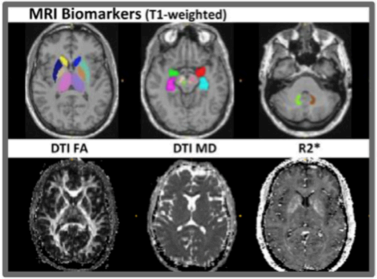
Given a decision variable, consider the expected value of the objective based on the worst-case possible distribution:

$$\min_{x \in \mathcal{X}} \left[\max_{f_\zeta \in D} \ \mathbb{E}_{f_\zeta}(h(x, \zeta)) \right]$$

The robust optimization model that exploit the historic data and

- remains computable
- has some interesting theoretical/statistical properties
- can be applied to many problems
  portfolio decision, image recognition, medical diagnosis, etc.





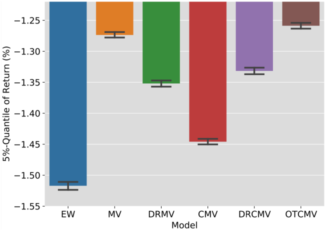
Application: Portfolio Decision in Stock Market

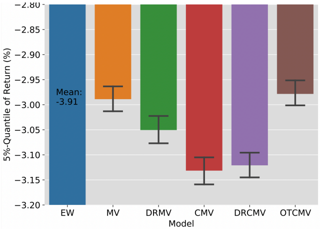
Scenario: decide a portfolio allocation maximizing the expected revenue.

- Decide facing high uncertainty
- Little perturbation causes huge loss

maximize	Return−Risk of Loss
subject to	Possible Portfolio Decisions
Decision	Portfolio Allocation

Solution: apply distributionally robust optimization via Optimal Transport (OTCMV) (Nguyen, Zhang, Blanchet, Delage & Ye, 2021)





Figures: simulated return for the US. and Chinese stock market

Application: Scheduling Facing Uncertainty

最佳实践：百威英博供应链E2E数智化升级案例

杉数科技 Cardinal Operations　COPT Cardinal Optimizer

项目背景 某ICT行业巨头多工厂协同生产排程场景，随业务发展，原来单工厂、逻辑复杂且不透明的计划排产系统已经不能满足客户业务的需求，需要大量的人工干预，严重影响生产效率，所以需要重新梳理业务需求，重构多工厂加工计划排产引擎。

客户挑战

定制化需求

- 原有单工厂排产系统已不能满足业务增长需求
- 大量人工干预严重影响排产和生产效率
- 亟需重构多工厂排产引擎实现协同生产

复杂业务场景

- 220个工厂
- 5万个+item
- “28天+10周”订单+预测需求
- 上亿种可能计划
- 千万级限制条件

杉数方案价值点

支持多工厂属性：产能约束、物料约束、加工流程等可按照各工厂设置

支持分工厂协议生产：对每一产品，不同工厂按协议比例进行排产

支持智能订单延迟分析：智能排查订单延迟原因，辅助业务快速定位问题

支持原料替代/试产：A材料紧缺用B替代，C为新上线零件部分试产

支持成品替代交付：相似产品需求不均匀、原料供应不均匀

支持多订单优先级：不同订单优先满足顺序可自定义

实施效果

20%	30%	70%	1.5倍
订单满足率提升	产能损失率降低	人工干预降低	仿真速度提升

- 所有工厂之间协作生产
- 工厂、仓库之间按最优路径转运
- 按物料清单给定的标准流程排产
- 统筹管理所有SKU

Semi-definite Programming

Satellite Localization and Starlink

Satellite and Starlink

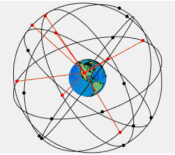
Wide application of satellite

- GPS, Vehicle Locationing
- Wireless Connection, Starlink
- Massive satellites (upto 10^4)

Technically, locating ground objects is

- **based** on location of satellite
- relatively easy

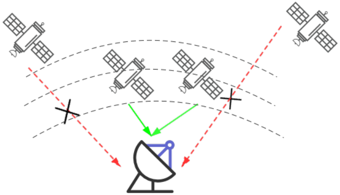
But how to locate the satellites?



Network Localization for Satellites

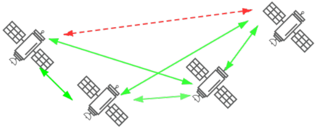
Generally, we have access to

- the location of a few satellites
  These satellites are called **anchors**
- insufficient to locate the rest of them
  The rest are called **sensors**



But fortunately,

- Satellites **know each other when close**
- They form a large network in space
- Sensor Network Localization Problem



Sensor Network Localization

Satellite Localization

Given coordinates of some of the satellites and certain distance between satellites, find the coordinates of satellites that realize the distance

Sensor Network Localization

Given coordinates of the anchors and certain distance between anchors and sensors, find the coordinates of sensors that realize the distance

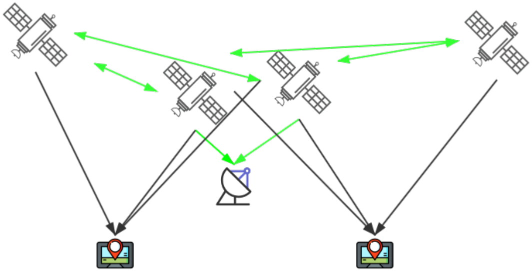
Given m anchor points $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{R}^d$ whose locations are known and n sensor points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ whose locations we wish to determine. Further more, we are given the Euclidean distance $\bar{d}_{ij}$ between $\boldsymbol{a}_k$ and $\boldsymbol{x}_j$ for some k, j and d_{ij} between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ for some i, j. The Problem is to find a realization of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ such that

$$\|\boldsymbol{a}_k - \boldsymbol{x}_j\|^2 = \bar{d}_{kj}^{\,2}, \qquad \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = d_{ij}^2, \forall i, j, k$$

SNL and Semi-definite Programming

The SNL problem

- is intractable even for $d = 1$
- can be relaxed as an SDP feasibility problem (So, M. C., and Y.Ye, 2007)
- the relaxed problem can be solved efficiently (Biswas and Y 2004) (Wang and Ding, 2008)



Online and Dynamic Programming

Dynamic Pricing Mechanism Design and Unmanned Warehouse

Online Combinatorial Auction

Senario:

- Fixed number of buyers
- Fixed inventory of goods
- Buyers require goods and bid

Decision:

- To sell or not

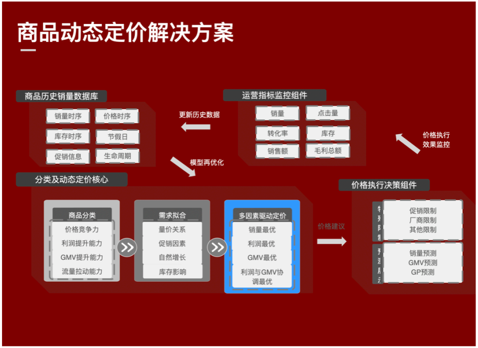
Objective:

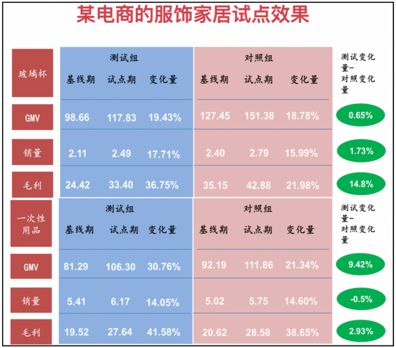
- Maximize the revenue

Application of the Online Algorithm

We can use the Online Algorithm to

- instruct the pricing strategy of sellers
- get better performance when combined with AI





某电商的服饰家居试点效果

玻璃杯	测试组 基线期	测试组 试点期	测试组 变化量	对照组 基线期	对照组 试点期	对照组 变化量	测试变化量-对照变化量
GMV	98.66	117.83	19.43%	127.45	151.38	18.78%	0.65%
销量	2.11	2.49	17.71%	2.40	2.79	15.99%	1.73%
毛利	24.42	33.40	36.75%	35.15	42.88	21.98%	14.8%
一次性用品	测试组 基线期	测试组 试点期	测试组 变化量	对照组 基线期	对照组 试点期	对照组 变化量	测试变化量-对照变化量
GMV	81.29	106.30	30.76%	92.19	111.86	21.34%	9.42%
销量	5.41	6.17	14.05%	5.02	5.75	14.60%	-0.5%
毛利	19.52	27.64	41.58%	20.62	28.58	38.65%	2.93%

Unmanned Warehouse

Senario

- Massive real-time decisions
  Hundreds of robots
- Complex constraint and objective
  Short path; no collision
- A series of large-scale problems

Highly efficient algorithms is needed

Unmanned Warehouse and Dynamic Programming

Solution

- Decompose large problems
  By region and path
- Dynamic programming approach

Efficiency: 3-4x faster than human

Algorithm: Core of Optimization

With so many applications of optimization, it is

- modeling that tells the problem
- algorithm that tells the **answer**

Large-scale Applications Demand Faster Algorithms

Table of Contents

Algorithms and Computation Complexity

We use

Zeroth order $f(x)$ First order $\nabla f(x)$ Second order $\nabla^2 f(x)$

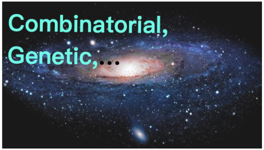
information to design numerical algorithms.

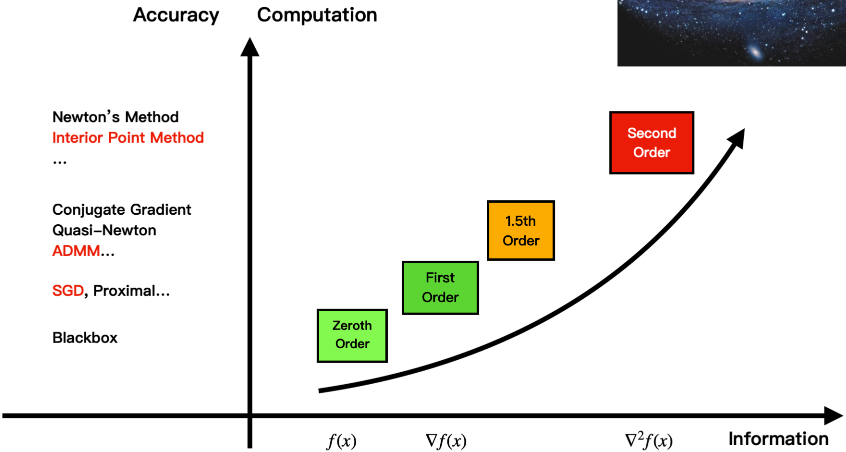
The more information we use,

- the more accurate solution is
- the more computation is needed

We choose algorithm by need

Numerical Algorithms





We apply different algorithms on different problems

First Order: Simple and Fast Online Algorithm

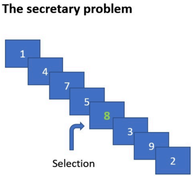
Offline Binary LP and Column Generation

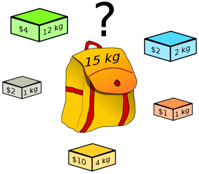
Binary Integer Linear Program

Consider the offline LP relaxation of a Binary Integer Linear Program

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \boldsymbol{r}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{0} \le \boldsymbol{x} \le \boldsymbol{1} \end{aligned} \tag{BILP}$$

- Wide applications in reality
- Approximate solution is acceptable
- Sometimes too large for IPM



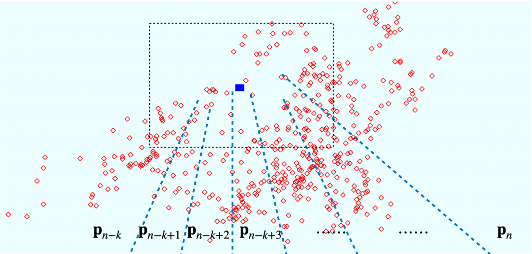


We use a simple and fast algorithm to solve the offline LP.

Simple and Fast Online Algorithm

Simple and Fast Online Algorithm (Li, Sun, & Ye (2020)) **simultaneously**

- Learn the dual price of LP using first order oracle for one pass
  An implementation of $O(\text{nnz}(\boldsymbol{A}))$ complexity
- Output the primal estimates using duality theory
  A good approximate solution $\hat{\boldsymbol{x}}$

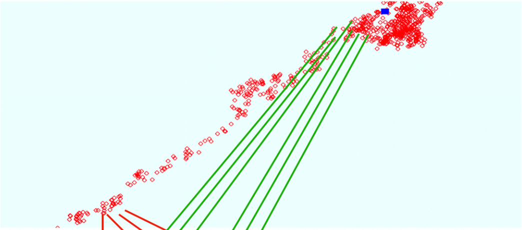


Variable j	...	$n-k$	$n-k+1$	$n-k+2$	$n-k+3$	...	$n-4$	$n-3$	$n-2$	$n-1$	n
$\mathbb{I}\{r_j > \mathbf{a}_j^\top \hat{\boldsymbol{p}}\}$	...	1	1	0	1	...	1	1	0	1	0
$\hat{x}_j$	...	1	1	0	1	...	1	1	0	1	0
x_j^*	...	1	1	0	1	...	1	1	0	1	0

Boosted Online Algorthm

A boosted version of online algorithm (BOA) (Gao, Sun, Ye, &Ye(2021))

- Learn the dual price for multiple passes with random permutation
- Output the primal estimates by taking average



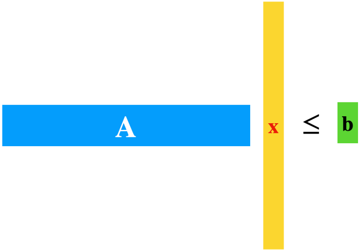
Variable j	1	2	3	4	…	$n-4$	$n-3$	$n-2$	$n-1$	n
$\hat{x}_{j1}$	1	1	0	1	…	1	1	0	1	0
$\hat{x}_{j2}$	0	0	0	0	…	1	1	0	1	0
$\vdots$	…	…	…	…	…	…	…	…	…	…
$\hat{x}_{jk}$	0	1	0	1	…	1	1	0	1	0
$\hat{x}_j$	0.02	0.88	0.01	0.99		0.92	0.98	0.03	0.67	0.21
$[\hat{x}_j]$	0	1	0	1		1	1	0	1	0
x_j^*	0	1	0	1	…	1	1	0	1	0

Applications

(Boosted) Online Algorithm can be applied to

- Approximately solve binary/integer linear programs
- Identify basic (non-zero) variables for "thin" LPs

Figure: A thin LP



Numerical Results for Binary LP

We generate data of Multi-knapsack problem as in Chu & Beasley but with $\boldsymbol{b} = O(\sqrt{n})$.

		V.R. Alg.	Gurobi
$m = 32, n = 4000, k = 50$	Time	0.048	> 60
	Cmpt. Ratio	89.4%	95.4%
$m = 32, n = 4000, k = 1000$	Time	0.720	> 60
	Cmpt. Ratio	90.6%	95.4%
$m = 64, n = 10^4, k = 50$	Time	0.392	> 60
	Cmpt. Ratio	89.9%	98.5%
$m = 64, n = 10^4, k = 1000$	Time	3.400	> 60
	Cmpt. Ratio	90.9%	98.5%
$m = 128, n = 10^5, k = 50$	Time	2.100	> 60
	Cmpt. Ratio	91.3%	98.6%
$m = 128, n = 10^5, k = 1000$	Time	45.000	> 60
	Cmpt. Ratio	94.9%	98.6%

Numerical Result: Basic Variable Identification

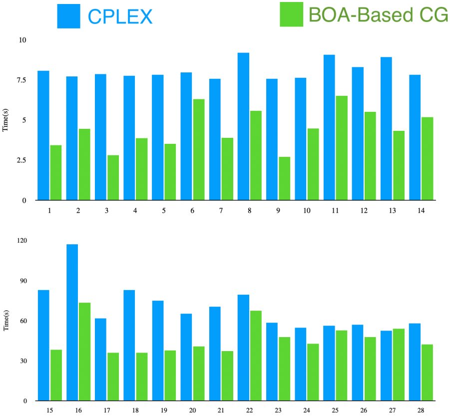
Datasets from MIPLIB 2017

ID	Dataset	Accuracy	Size Reduction
1	rail507	271/301	11862/62171
2	rail516	121/138	8572/46978
3	rail582	325/347	12465/54315
4	rail2586	1536/1672	145373/909940
5	rail4284	1951/2042	348135/1090526
6	scpm1	2754/2754	10352/500000
7	scpn2	3411/3411	20860/1000000
8	scpl4	1149/1149	5718/200000
9	scpj4scip	552/552	3635/99947
10	scpk4	930/930	4077/100000
11	s82	1992/3020	52383/1687859
12	s100	150/487	835/364203
13	s250r100	415/747	3080/270323

Table: Basic variable identification

Numerical Result: Basic Variable Identification

We generate sparse Multi-knapsack instances following Chu & Beasley and initialize Column Generation by boosted online algorithm. Result is compared against CPLEX.



3/2 th Order: ADMM and Parallel Optimization

Security Constrained Unit Commitment Problem

ADMM is an algorithm that

- tackles hard problems using the block structure
- is friendly to variable splitting and parallelism
- can be applied to Mixed Integer Linear Programs
- is an 1.5th order algorithm (access 2nd order information once)

ADMM has been applied vastly in recent years

- Statistical and Machine learning
  Lasso, robust PCA, tensor completion
- Optimal control
  System estimation, beamforming
- Security constrained Unit commitment
- ...

Application: SCUC

Security Constrained Unit Commitment Problem

- determines the startup/shutdown of a series of generators
- is constrained by the load demand and security

minimize	Cost of generating Power
subject to	Thermal Power+ Hydro Power = Load
	Security Constraints
Decision	Whether to startup/shutdown
	$x \in \{\text{No}, \text{Yes}\} \Rightarrow \{0, 1\}$

A natural block structure between thermal and hydro power

案例：国家电网机组组合优化

杉数科技 Cardinal Operations　COPT Cardinal Optimizer

国家电网

水火电联合机组组合优化

与国家电网电力科学研究院某分部合作，开发针对**四川水火电联合模型**的机组组合优化，调度优化等核心功能的专用求解器，做到高效、稳健、自主，**全面替换**目前的解决方案（目前底层均使用美国求解器）

项目目标

提高资产利用效率，有效降低用户用电成本，降低电价，加快电力生产企业有序竞争。

以国家电网公司为代表的大型电力企业，承担着保障安全、经济、清洁、可持续供电的基本使命，关系国民经济命脉和国家能源安全。

核心目标是交付一套适合安全约束机组组合的高效求解器，并基于求解器对模型进行针对性算法加速，结合图并行计算技术整体提升电力系统优化的求解效率和稳定性，进而达到高效、稳健、自主、可控的电力系统优化求解的目标。

- Dominating advantage over conventional method
- Upto 10x faster than original model and solver
- Full support over parallelism

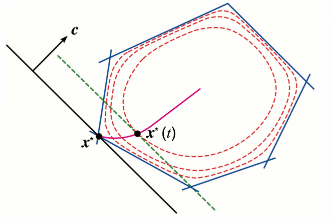
Second Order: Interior Point Method and Smart Crossover

Solve a Puzzle 100 Times Faster

Interior Point Method and Linear Programming

For Linear Programming, the Interior Point Method (IPM)

- generates iterates in the interior of primal-dual space
- is a second order algorithm
- converges towards basic optimal solution

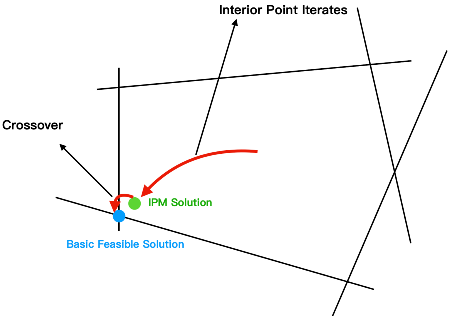


IPM is

- one of the most efficient algorithms for LP
- one of the most beautiful and elegant numerical algorithms

Crossover: Last Step for the IPM

- IPM gives a solution in the interior of primal-dual space
  A strictly feasible solution close to some basic optimal solution
- We are often more interested in basic solutions
  Network problems, decision variables, MIP root problem...



Crossover does this last job

Smart Crossover (Ge, Wang, Xiong, &Ye)

For problems with certain network structure, one can

1. utilize the structure to detect the basic variables accurately
2. initiate re-optimization using the simplex method

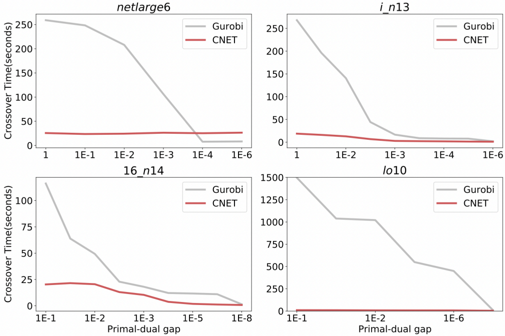


Figure: Results on Benchmark Datasets

Perturbation Crossover Method and Puzzle Problem

One “horrible” problem for crossover exists in the benchmark

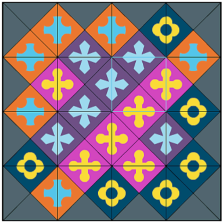
```
==========================================================================
problem      MOSEK  MATLAB  Gurobi     CLP   TULIP     COPT   MDOPT  KNITRO
==========================================================================
......
**************************************************************************
datt256        255      61     514     238      24        7      26      18
**************************************************************************
......
```

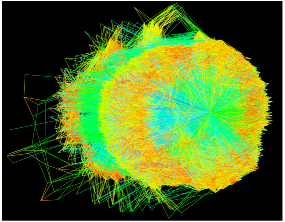
Gurobi spends 1.5s finding the interior optimal but 500s on crossover

```
                    Objective                Residual
Iter       Primal          Dual         Primal    Dual     Compl     Time
   0   4.47194709e+04  2.56000000e+02  2.31e+02 9.96e-14  5.70e-03     1s
   1   1.40975104e+03  2.44173005e+02  6.00e+00 5.55e-16  1.76e-04     1s
   2   3.74750808e+02  2.48268648e+02  6.18e-01 4.44e-16  3.19e-05     1s
   3   2.62724890e+02  2.54388185e+02  3.50e-02 4.44e-16  4.43e-06     1s
   4   2.56032589e+02  2.55997788e+02  1.69e-04 6.66e-16  6.68e-09     1s
   5   2.56000008e+02  2.55999998e+02  4.07e-08 7.77e-16  5.89e-12     1s

Barrier solved model in 5 iterations and 1.48 seconds
Optimal objective 2.56000008e+02

Crossover log...

    6288 DPushes remaining with DInf 0.0000000e+00                 2s
       0 DPushes remaining with DInf 0.0000000e+00                 4s

  143784 PPushes remaining with PInf 9.1721636e+00                 4s

A long time...

     892 PPushes remaining with PInf 0.0000000e+00               510s
       0 PPushes remaining with PInf 0.0000000e+00               514s
```

The problem comes from relaxation of the well-known puzzle

`Eternity II` (meaning "take eternity to solve")





- The problem has complicated network structure
- The original MIP is still unsolved...

COPT: State-of-the-art Optimization Solver

The First Commercial Solver for Mathematical Programming in China

SIMPLEX	CPLEX	XPRESS	GUROBI	CLP	SAS	COPT	MDOPT
2015-07	1.94	1.12	1.18	1			
2016-11	1.88	1.01	1	1.03			
2017-10	1.91	1.22	1	1.69			
2018-07	1.88	1.07	1	1.87	5.61		
2018-11	2.16	1.15	1	2.19	6.43		
2019-05				1.32	4.26	1	
2020-06				2.03	4.52	1	
2020-08				3.01	6.12	1.41	1
2020-09			1.28	2.83	6.10	1	1.19
2020-10			1.45	3.37	7.25	1	1.41
2020-11			1	4.01	8.63	1.19	1.68
2020-12a			1.25	5.01		1.20	1
2020-12b			1.60	6.43		1	1.28
2021-0608			1.60	6.43		1	1.28

Figure: Performance of COPT on Mittelman Hans' Simplex LP Benchmark

Current Best Barrier Solver on Benchmark

BARRIER	CPLEX	XPRESS	GUROBI	MOSEK	SAS	COPT	MDOPT
2011-11	1	1.52	1.15	1.47			
2012-10	1.08	1.35	1	1.64			
2013-07	1.15	1.01	1	1.54			
2015-07	1.64	1	1.07	1.55			
2016-11	1.56	1	1.01	1.95			
2017-10	1.84	1.03	1	2.29			
2018-07	1.78	1	1.10	2.69	10.70		
2018-11	1.92	1.08	1	3.31	11.60		
2019-10				1	1.22		
2020-11			1	2.36	5.26	2.39	
2020-12			1	2.34		1.58	2.00
2021-03			1	2.34		1.09	2.00
2021-06			1.22	2.85		1	2.45

Figure: Performance of COPT on Mittelman Hans' Barrier(IPM) LP Benchmark

The First Commercial MILP Solver in China

++

Unscaled and scaled shifted geometric means of run times

All non-successes are counted as max-time.
The third line lists the number of problems (240 total) solved.

1 thr	CBC	GLPK	LP_SOL	MATLAB	SCIP	Gurobi	COPT
unscal	2107	5044	5335	3301	1100	245	1029
scaled	8.59	20.5	21.7	13.5	4.48	1	4.19
solved	89	23	20	63	125	201	132

8 thr	CBC	FSCIP	Gurobi	COPT
unscal	1723	1065	146	754
scaled	11.8	7.30	1	5.17
solved	98	138	220	156

Figure: Performance of COPT on Mittelman Hans' MILP Benchmark

Summary

1. Many real-world problems, especially real-economy applications, demand mathematical optimization tools/solvers
2. The innovation of efficient optimization methods/algorithms should be driven by scientific/theoretical research, besides software engineering and coding
3. The development of mathematical programming solvers is best done by a small dedicated team whose members have passion and love in optimization

Stanford University

Thanks for Listening!