OPTIMIZATION: PAST, PRESENT AND FUTURE

Robert Ashford
Optimization Direct Inc.

INFORMS International Meeting 2024

OPTIMIZATION DIRECT

Summary

What is Optimization, where is it used and what use is it?

The Mathematical Model (MILP) and how to solve it

Evolution of Optimization and the hardware on which it runs

Solver Performance

The Cutting Edge

- Multi-threading
- New Methods, e.g. ODHeuristics

Applications that push the envelope

Optimization: The Future

OPTIMIZATION DIRECT

What is Optimization?

Minimization (or maximization) of function of a set of decision variables

- Usually linear
- Could be quadratic
- Non-linear sometimes now practical

Subject to constraints on variables

- set of [in]equalites
- membership of (possibly discrete) sets

Mostly concerned with Mixed Integer Linear Programs - MILPs

OPTIMIZATION DIRECT

Where is Optimization Used?

Airlines	crew management, scheduling, yield management
Brewing	blending, production planning, distribution
Car industry	production planning/organization, model launch
Chemicals/Powders	distillation, production planning, distribution
Defense	scheduling logistics
Electric power	generation, transmission, storage, network design
Finance	capital mngt, trading rules, investment selection
Food indsutry	production scheduling
Forestry	what to plant, where, when to harvest
Gas distribution	network design and management, purchasing
Medical	resource scheduling
Mining	extraction planning
Oil	shipping, pipeline operation, refining, distribution
Retail	store grouping, purchasing
Sports scheduling	fixture management
Steel Manufacturing	production planning, furnace operation
Telecommunications	network design, frequency selection
Water	storage management, waste management

Common Optimization Tasks

Those were some of the areas I in which I worked

Almost all industries and many government agencies

Optimization tasks include:

- (intermediate) product/material processing
- yield management
- transport/distribution
- organization/design
- planning
- scheduling

Helpful to classify models according to their time horizons

OPTIMIZATION DIRECT

Model Time Horizons: Short Term

A week or day or even less

- scheduling/operation – do it now
- very accurate
- engineering activity
- easy to sell
- hard to do – competing technology, e.g.
    - constraint programming
    - heuristics

Tactical

OPTIMIZATION DIRECT

Model Time Horizons: Medium Term

Typically a month
- e.g. refinery/production planning
- engineering/management activity
- less accurate
- not-so-easy-to-sell
- easier to do
- sometimes embedded in culture e.g. refinery planning

Effective operational/management tool despite limitations of inaccuracy

OPTIMIZATION DIRECT

Model Time Horizons: Long Term

- Typically a year or more
  - Design, e.g.
    - telecomms /gas/electricity networks
    - distribution
  - Investment
  - Hard to sell
  - Easy(ier) to do
- Huge benefits
- *Strategic*

OPTIMIZATION DIRECT

What Use is Optimization?

Short term

- Tells you what to do

Medium term

- Gives you an idea

Long term

- Informs strategioes

Analyzes data and gives control

Only ‘inteligelligent’ (logic based) way of stress testing data

Simple Example: Wire Pulling

BICC factory in Liverpool, UK

Processes 8mm copper rod through series of dies and coats them with varnish

Produces drums of wire for use in electrical industry, typically motor manufacturers

Simple annual model looked at fulfillment of orders

Some cost several times more than others

“get rid of Black and Decker”

$8M annual loss became $5M profit

OPTIMIZATION DIRECT

The Mathematical Model (MILP)

Minimize (over x): $c^T x$

Subject to:

$Ax = b$

$l \leq x \leq u$

$x_j \in \mathcal{Z}$, some j

$x, c, l, u \in \mathcal{R}^n$; $b \in \mathcal{R}^m$; $A \in \mathcal{R}^{m \times n}$

Can have more exotic integrality requirements

OPTIMIZATION DIRECT

How To Solve It

Presolve

Solve LP relaxation

Add cuts

restart

Branch and Cut

Run heuristics at all stages

Stop:

- when incumbent solution sufficiently close to best possible one (“best bound”), say 0.01%
- maximum time

How To Solve It: Presolve

Successively tighten variable and row bounds

e.g. $x_1 + x_2 \leq 10;\ 0 \leq x_1 \leq 4;\ 0 \leq x_2 \leq 5$

$\Rightarrow x_1 + x_2 \leq 9$, remove the row

$x_1 + x_2 \leq 2;\ 1 \leq x_1 \leq 2;\ 1 \leq x_2 \leq 2$

$\Rightarrow x_1 \leq 1,\ x_2 \leq 1$ fix (remove) x_1 and x_2

Remove duplicate variables and rows

Aggregate: e.g. $- x_1 + x_2 + x_3 = 0;\ x_i \geq 0$

$\Rightarrow$ replace x_1 by $(x_2 + x_3)$ everywhere

Other reductions possible, for example:

use dual (cost) arguments to tighten variable bounds

infer and tighten dual bounds, remove rows

How To Solve It: Presolve

Integer tighten variable bounds and rows

e.g. $x \leq 2.4;\ x \in \mathcal{Z} \Rightarrow x \leq 2$

Tighten matrix coefficients

$x - 100\,\delta \leq 0;\ \delta \in \{0,1\};\ x \leq 50 \Rightarrow x - 50\,\delta \leq 0$

Other integer reductions/changes possible

Repeat

One (set of) reductions enables another

OPTIMIZATION DIRECT

How To Solve It: The LP relaxation

Relax the integrality conditions

Solve with primal or dual simplex or barrier (interior point) method and cross-over

Best method depends on:

- whether have some kind of starting solution
- hardware characteristics
- the LP itself

Do several methods simultaneously "concurrent solve"

OPTIMIZATION DIRECT

How To Solve It: Cutting

Make the LP feasible region closer to the convex hull if the MIP

This is the smallest convex region that contains all the integer feasible points

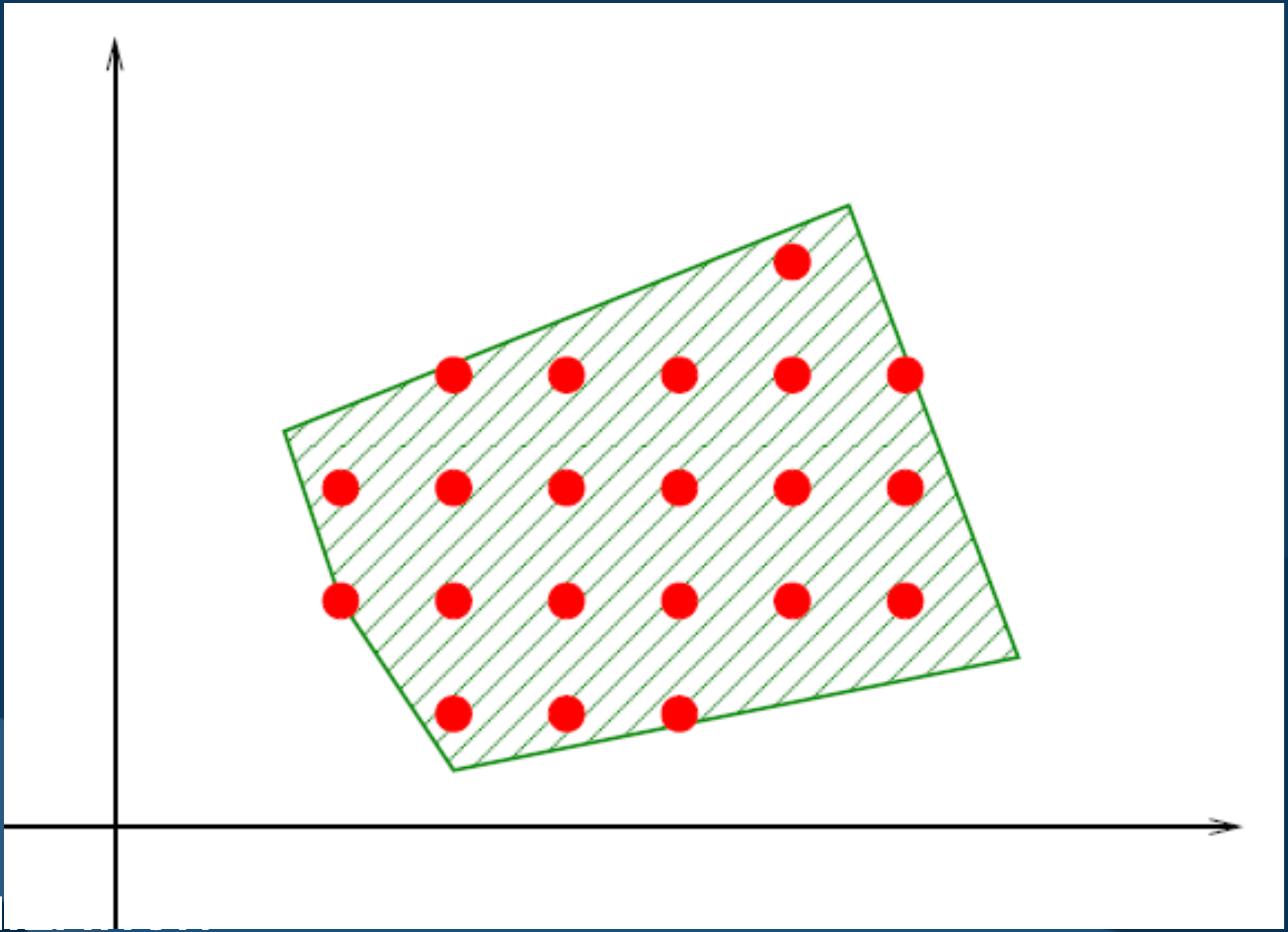
If could actually derive convex hull, would only need to solve the LP

Example: 2 integer variable model

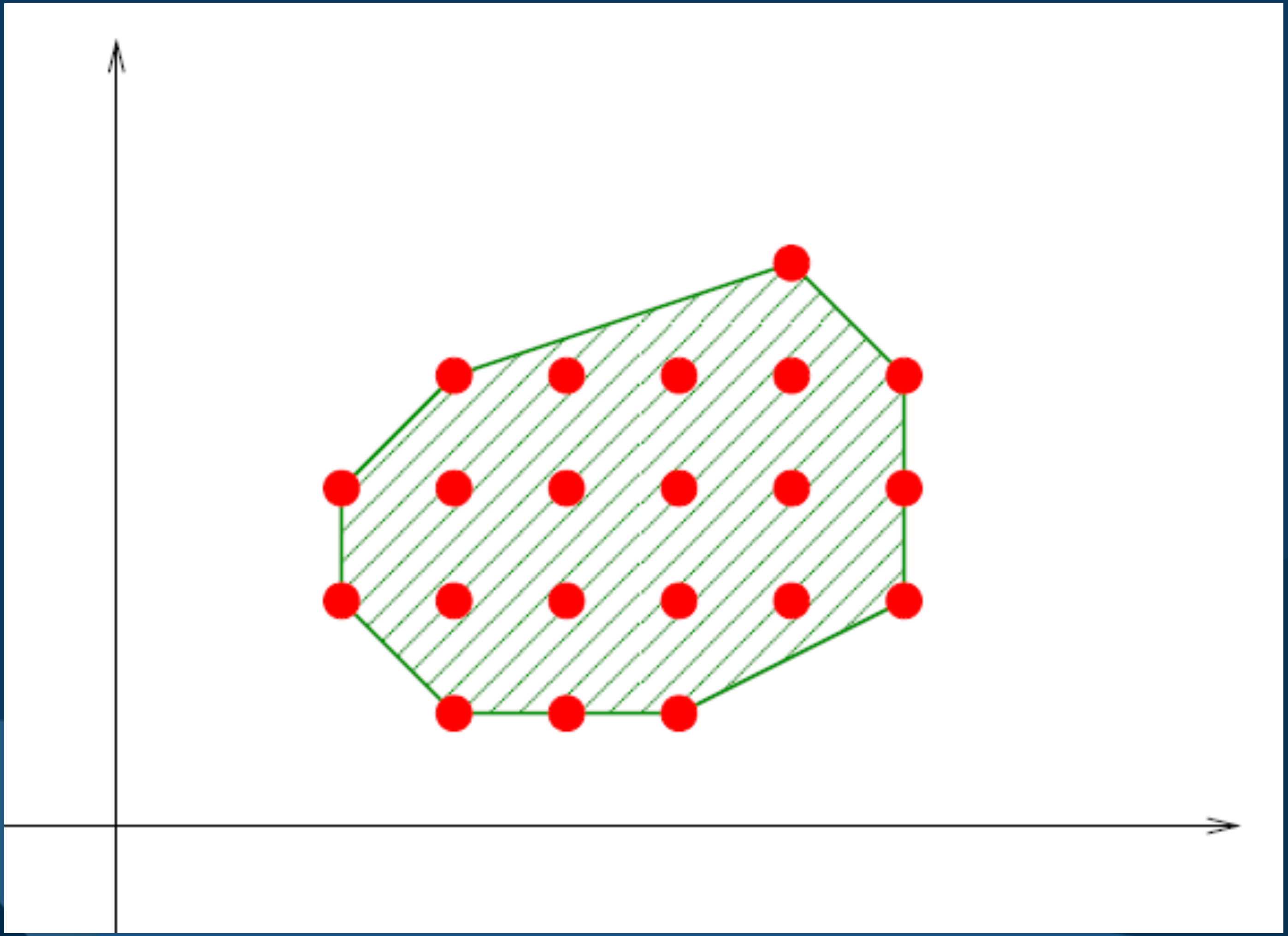
- LP feasible region (green lines)
- integer feasible region (red spots) like:

OPTIMIZATION DIRECT

MIP Feasible Regaion



Convex Hull



Cutting

"Snip" pieces away from feasible region

Cuts derived from the constraints.

e.g. $4x_1 + 3x_2 \leq 5; x_i \geq 0$ and integer

$\Rightarrow x_1 + x_2 \leq 1$

Many different methods, some use multiple constraints

Look around current solution to "cut" LP to derive new cuts

More cuts make the LP harder to solve

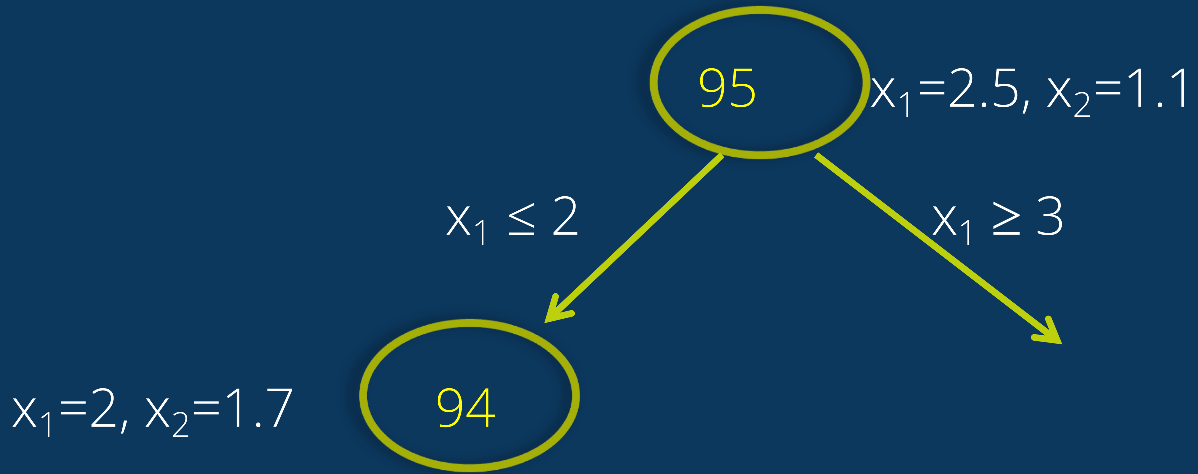
OPTIMIZATION DIRECT

Branch and Bound

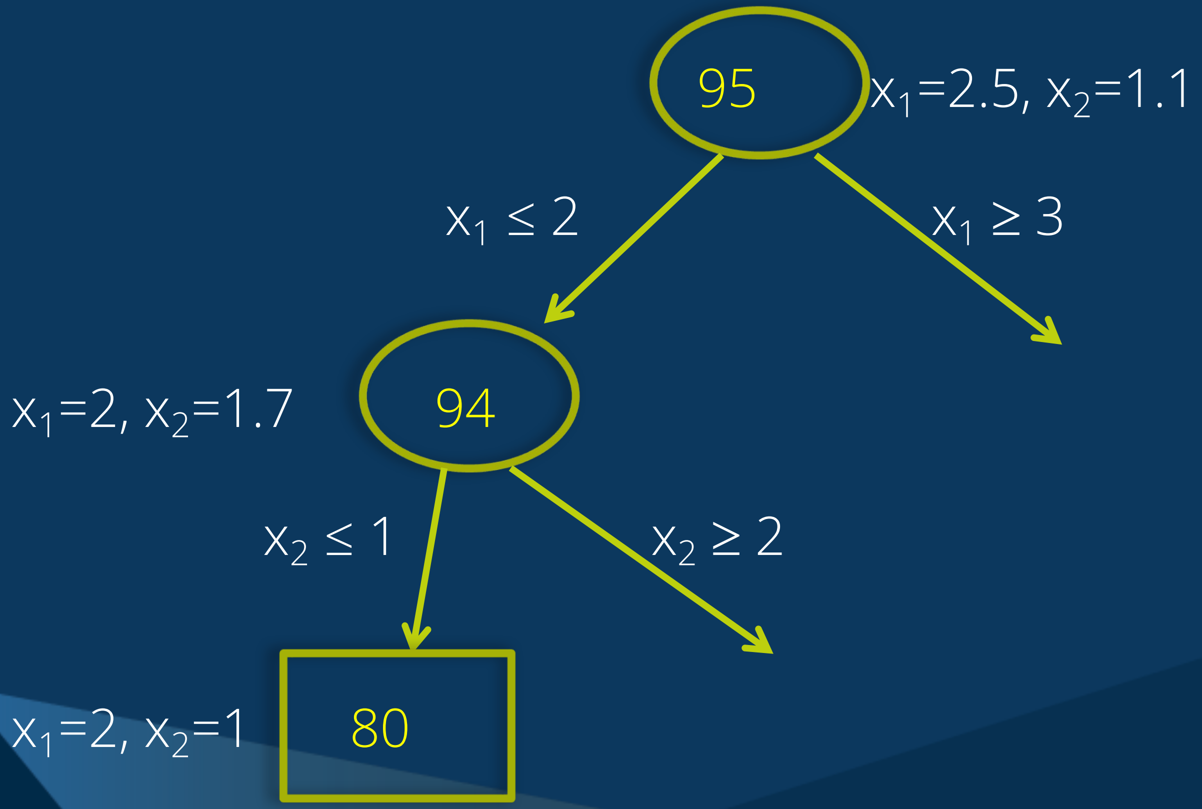
95 x_1=2.5, x_2=1.1

OPTIMIZATION DIRECT

Branch and Bound

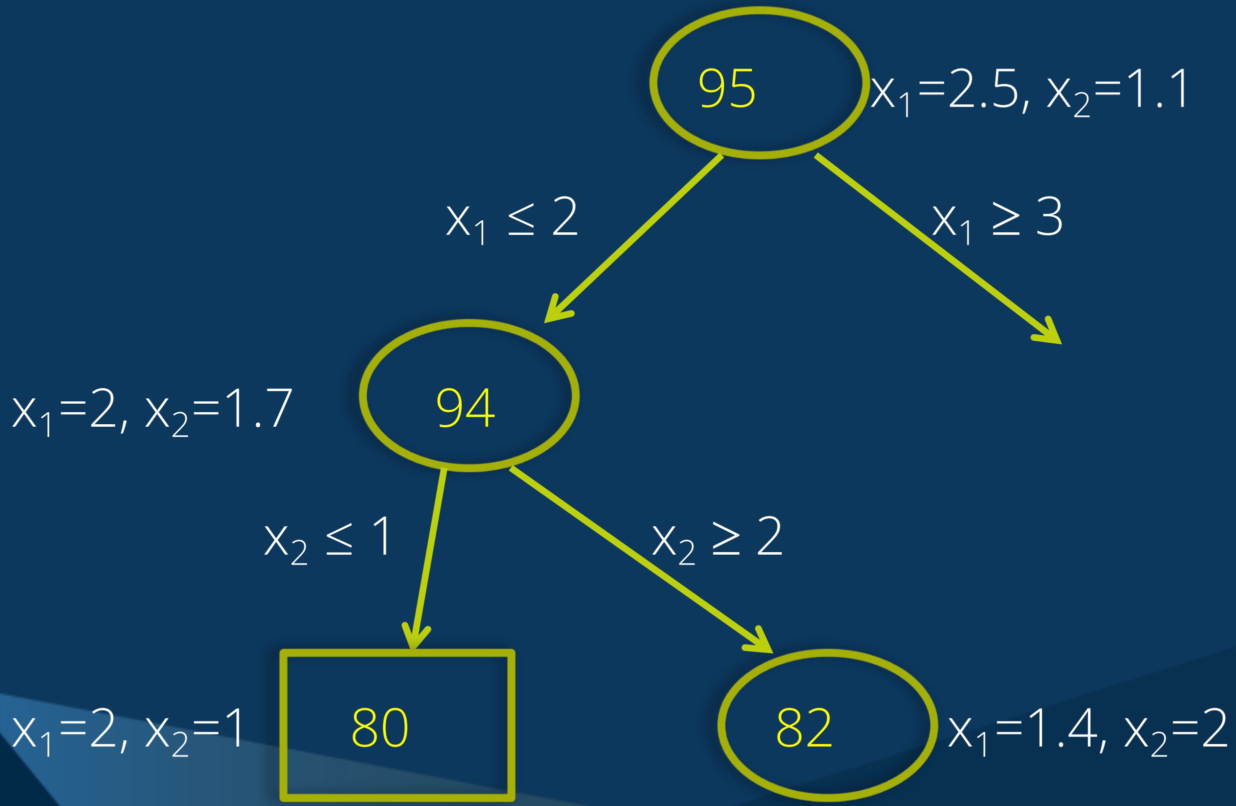


Branch and Bound

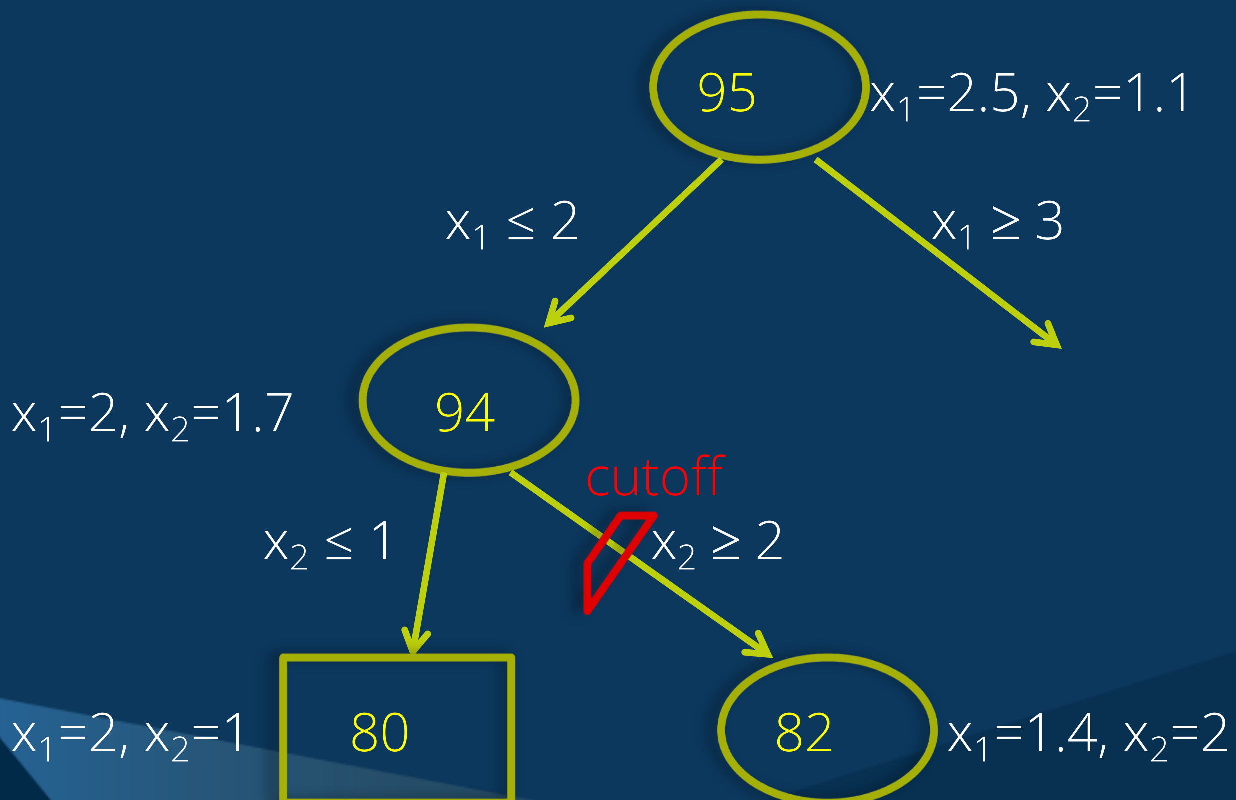


OPTIMIZATION DIRECT

Branch and Bound

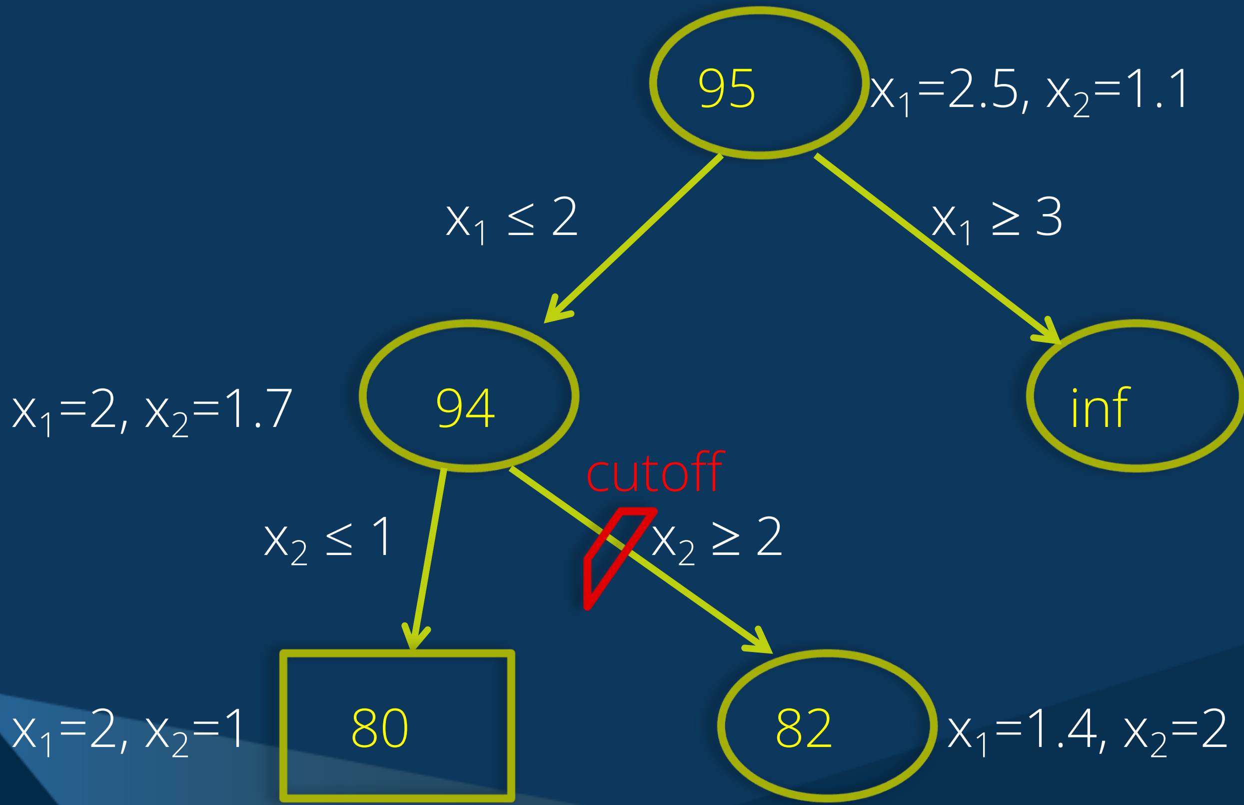


Branch and Bound



OPTIMIZATION DIRECT

Branch and Bound



OPTIMIZATION DIRECT

Branch and Cut

Nodes form an (upside down) tree

Maximum number of nodes is 2^n if we have n binary variables – can get large

Effective to generate cuts at nodes and 'lift' them so as to be cuts for the whole tree

Can parallelize the tree search

Effective to re-start when get new good incumbent for fresh presolve

Heuristics

- Methods for getting an integer feasible solution quickly
- Many techniques
  - diving
  - rounding
  - RINS, etc.
- Prunes the tree
- Good incumbent helps make better decisions
  - branching
  - start for next heuristic, etc.

Evolution of Optimization

1950-1970	LPs	mainframes	Primal Simplex	100-1000 row models
1970s	MIP begins	+ Mini-computers	Branch and bound	1000+ row models
1980s		+ workstations and PCs	Simple cuts	End of 'white coats'
1990s		Intel/AMD Servers Powerful PCs	Branch-and-cut Heuristics	Big Bang
2000s		Multi-processing		

OPTIMIZATION DIRECT

Some Key LP/MILP Methods

Began		Who	First Software
1947	Introduced LP (Primal simplex)	Dantzig	
1951	Computer impl. of simplex algorithm	National Bureau of Standards	
1954	Dual simplex	Lemke	
1957	Cutting plane algorithms	Gomory	
1960	Branch and Bound	Land & Doig	LP/90/94 (1965)
1972	Sparse updating	Forrest & Tomlin	UMPIRE
1973	Better simplex pivot choice	Harris	UMPIRE
1987	Effective Root Cuts	Wolsey, Chvátal,..	CPLEX, Xpress
1992	Effective dual simplex	Bixby	CPLEX
1992	Barrier (interior point) methed	Marsden, Lustig	OB1, CPLEX
1993	Presolve		CPLEX, Xpress
1995	Super-sparsity	Laundy	Xpress
1996	Parallel branch and bound	Laundy	Xpress
2000	Useful Heuristics		CPLEX
2000	Probing		CPLEX
2005	Branch and Cut		CPLEX
2007	MIP restarts		CPLEX
2014	LP folding	Grohe et al.	CPLEX, Xprs,Gurobi

Observations About LP/MIP Development

First LP was solved by pencil-and-paper

- 7 const, 77 vars and took 120 days (Laderman, 1947)

Theory often appeared before effective implementation

- until 1992
- cuts work in literature years before implemented commercially

Reluctance to publish post 1992

Large differences made by incremental developments

Many people contributed, not just ones mentioned before, e.g.

- Karmarkar did first efficient interior point method in 1984, Terlaky subsequently made major contibutions
- Many people worked on cutting planes: Van Roy, Balas, ..

Usability largely depends on modeling software

- Xpress LP-Model (Ashford) was the first commercially available in 1983
- Followed by GAMS, then AMPL, OPL, MPL, etc.

Some Commercial LP/MIP Software

Date	Software	Vendor	
1963	LP/90/94	CEIR	LP
1965	MPS/360	IBM	LP
1972	MPSX/370	IBM	LP, MIP from 1974
1974	UMPIRE	CEIR, Scicon	MIP
1976	Sciconic	Scicon	MIP
1984	Xpress	Dash Assoc, then FICO	LP, MIP from 1989
1991	CPLEX	CPLEX, then IBM	MIP
2009	Gurobi	Gurobi	MIP
2015	ODH	Optimization Direct	MIP
2021	COPT	Cardinal Software	MIP

OPTIMIZATION DIRECT

Some Typical Hardware

Date	Computer	Type	Bits (Addr)	Max Memory	Cores	Single Core Perf
1964	IBM/360	Mainframe	32 (24)	16MB	1	0.01165
1970	IBM/370	Mainframe	32 (24)	64MB	1	0.4458
1974	Intel 8080	PC	8(16)	64KB	1	0.02 *
1979	DEC VAX 11/780	Mini	32(32)	3MB	1	1
1983	Intel 8086/8087	PC	16(20)	2MB	1	0.25
1987	IBM PS/2 80	PC	32(32)	4MB	1	2.15
1998	Intel Xeon	Server	64(64)	4GB	1	623
2001	Intel Pentium 4	PC	32(32)	2GB	1	2495
2008	Intel i7-4790K	PC	64(64)	32GB	4	7549
2015	Intel Xeon E5	Server	64(64)	2TB	24	6113

CPU Price Performance 1944-2003 - John McCallum

cpu.userbenchmark.com

** Estimated*

OPTIMIZATION DIRECT

Observations About Hardware

'Big Bang' occurred when clock-time-to-solve on very cheap hardware matched typical mainframe:1987 with IBM PS/2-80 (Intel 386/387)

The hardware drives the maths

Computers don't speed up uniformly – some operations speed up more than others

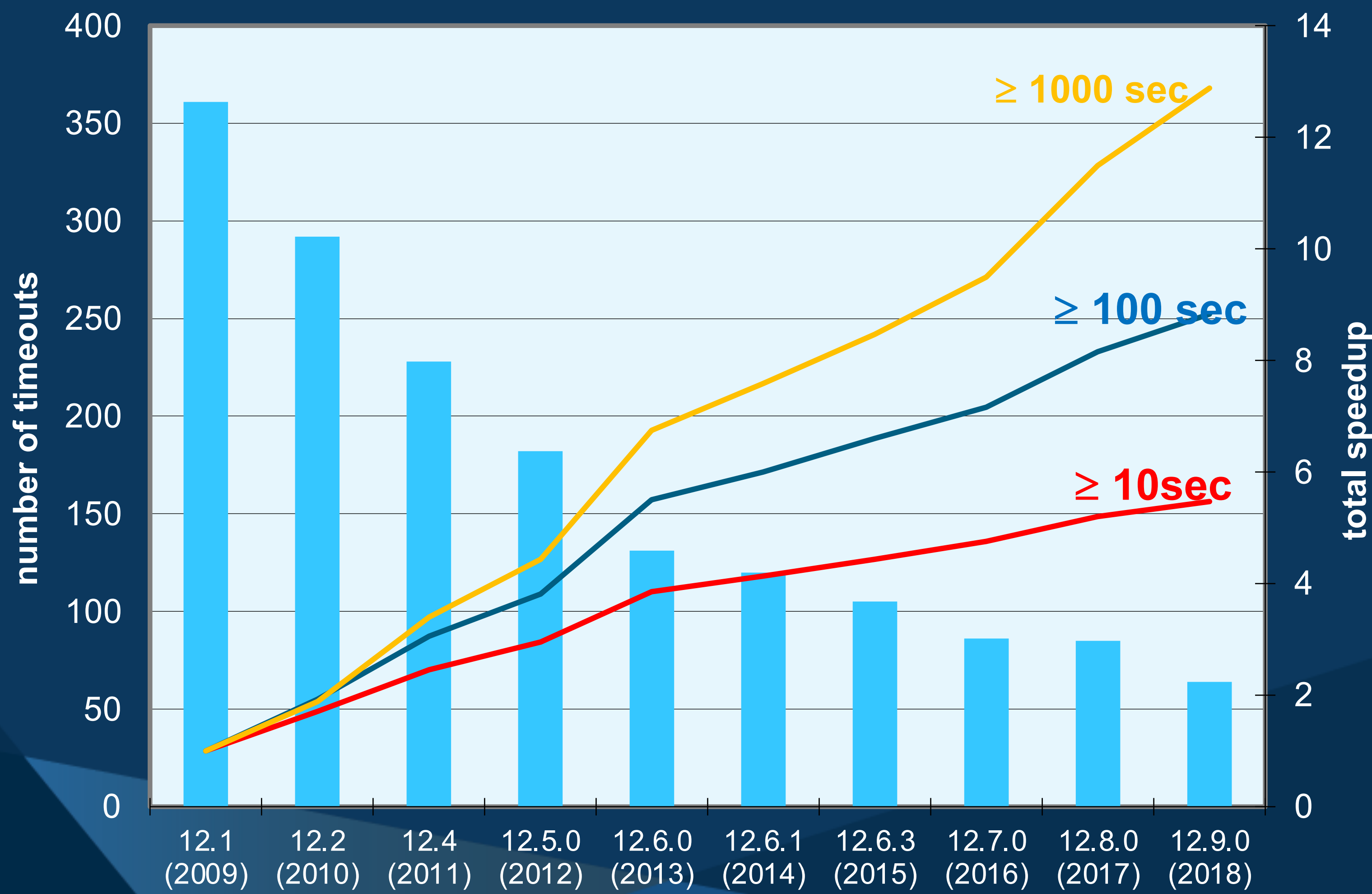
- FP multiply was 4300 X faster † on Intel Pentium 4 than IBM PS/2, but memory access only 2 X faster‡
- Now constrained by bus speeds – a real bottleneck for parallel processing

Rate of improvement now slow

- Effort has gone in to bit-coin mining and AI

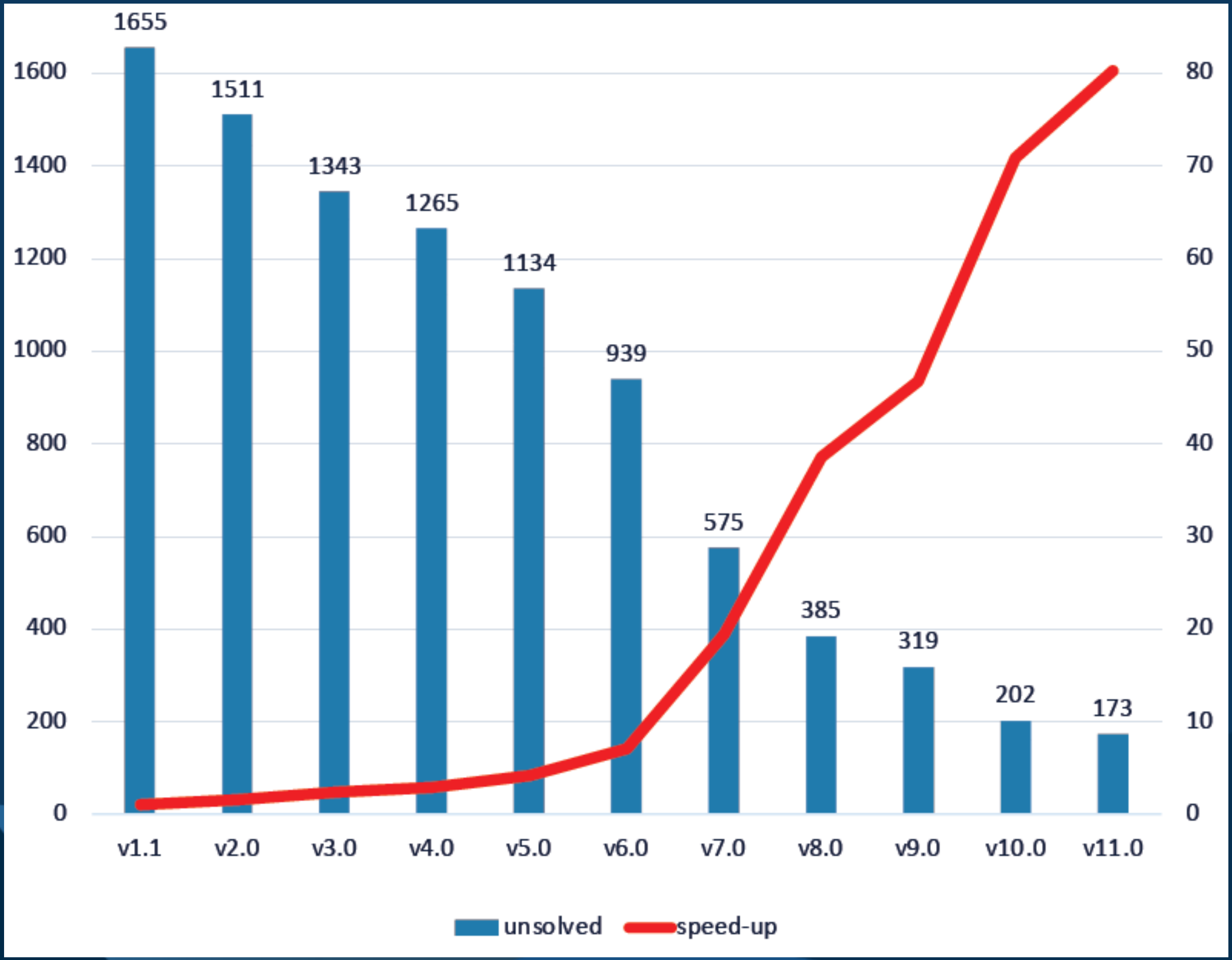
† 35,000 X faster with vector facility
‡ if the L2 cache is missed

CPLEX Performance (2009-2018)



Date:	26 October 2018
Testset:	MILP: 4061 models
Machine:	Intel X5650 @ 2.67GHz, 24 GB RAM, 12 threads, deterministic
Timelimit:	10,000 sec

Gurobi Performance (2009-2024)



Time limit: 10,000 sec.
Intel Xeon CPU E3-1240 v5 @ 3.50GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 7766 models:
- 714 discarded due to inconsistent answers
- 2124 discarded that none of the versions can solve
- speed-up measured on >100s bracket: 2892 models

The Cutting Edge

Non-Linear

- Quadratic objectives and constraint handling now mature (MIQCQP)
- Functions of a single argument, f(x) where x ∊ $\mathcal{R}$ have been approximated for decades and now some can be handled internally by solver
- Can even get **globally optimal** solutions to non-convex models with commercial software

Parallel processing

- Multi-machine solving possible though not popular, but
- Vector processing in barrier now transparent and ubiquitous, as is
- Multi-threading during most of the solve, esp. branch and cut

Novel methods

The Cutting Edge: Multi-threading

Would like to go n X faster with n cores, but reality is harder

Useful work division limited by inherently sequential nature of optimization methods

presolve $\rightarrow$ *root solve* $\rightarrow$ *cutting* $\rightarrow$ *search*

although parallelization possible within methods

Tasks need to mutually communicate

Tasks compete for resources

Cores, memory bus capacity

OPTIMIZATION DIRECT

Determinism

Threads must be synchronized to get deterministic behavior

Synchronization costs time at

- Sync points; or
- Accessing information pool

Depends on your model, hardware, program quality and number of threads

OPTIMIZATION DIRECT

Determinism: Costs

Typically using 8 threads to solve a MIP on a 4 core SMT (hyper-threaded) workstation costs ~ 20%

Cost rises with number of threads

25 user models, 2hr time limit, ODH|CPLEX

Threads	Computer	Cores	Synchronization time		
			Average	Spread	Max
8	i7-4790K	4	19%	11%	50%
12	E5-2690 v3	24	23%	12%	59%
24	E5-2690 v3	24	30%	15%	67%

OPTIMIZATION DIRECT

Synchronization

Can have specific synchronization points, but better to synchronize the passing of information between the threads

Need deterministic measure of work (“time”)

- “CPU time” not deterministic
- Use retired instructions or some counter

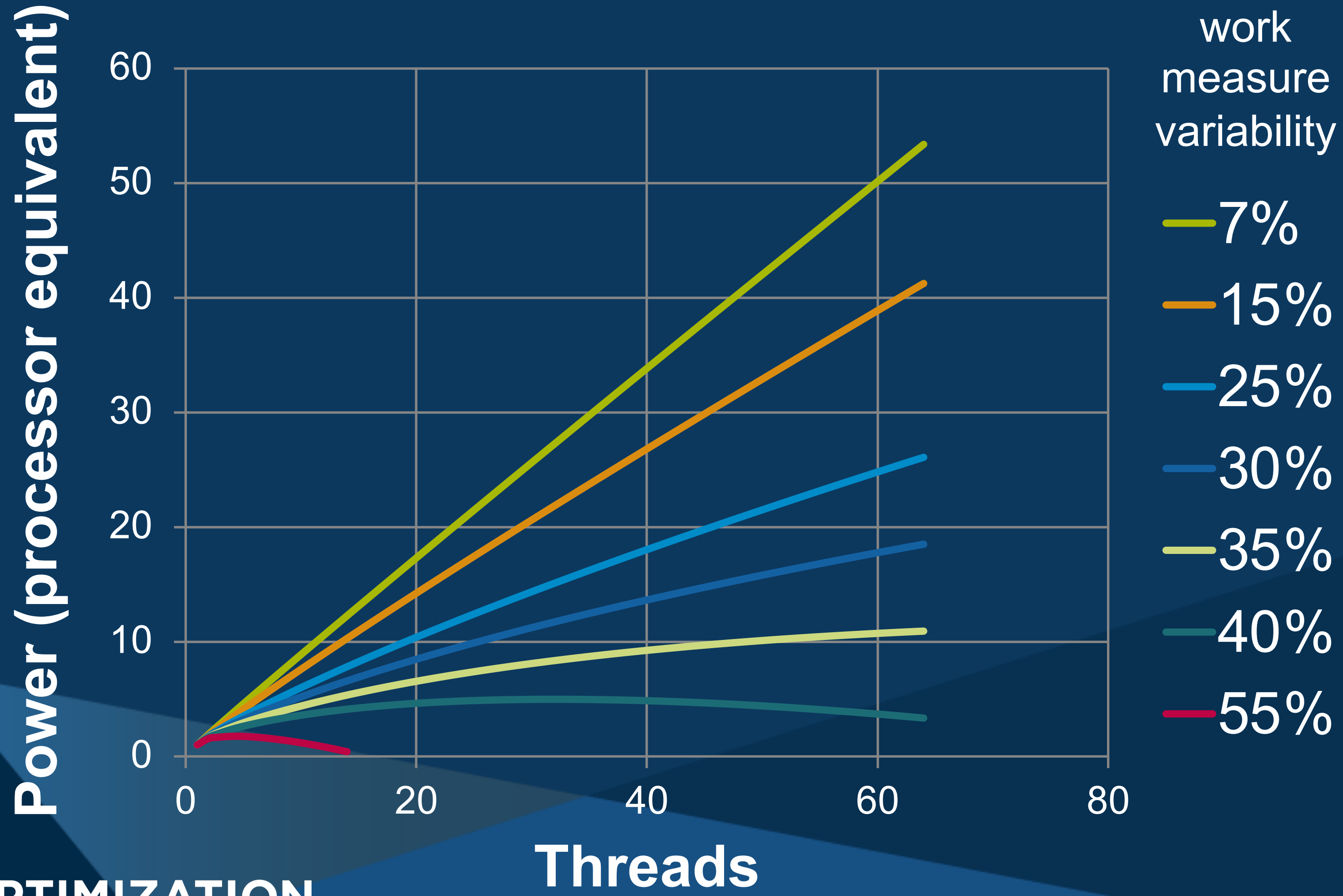
Variability in

- Actual work done for a given count
  - varies according to model size and algorithmic activity
- Resource allocation to threads

Work measure varies 7% - 60%

Synchronization

Theoretically, *ignoring bus contention*



OPTIMIZATION DIRECT

Determinism: Pros and Cons

Pros: repeatability

- Emotionally good to get same answer from repeated runs
- Easier to analyze and QA models
- Easier to tune solver parameters

Cons: slower

- Waste computer resources
- Wait longer for e.g. solution quality to be hit

OPTIMIZATION DIRECT

Determinism: Users

Most OR optimizer users prefer determinism

‘Performance’ users prefer non-determinism

- Users of very large and/or difficult models
- Meteorological modelers
  - solve Navier-Stokes equations fast
  - ‘determinism is for wimps’

Future is non-determinism

- No way out of sync overhead
- Number of cores is increasing, speed is not
- Greater issue as bus (memory speeds) improve

The Cutting Edge: New Methods: ODH

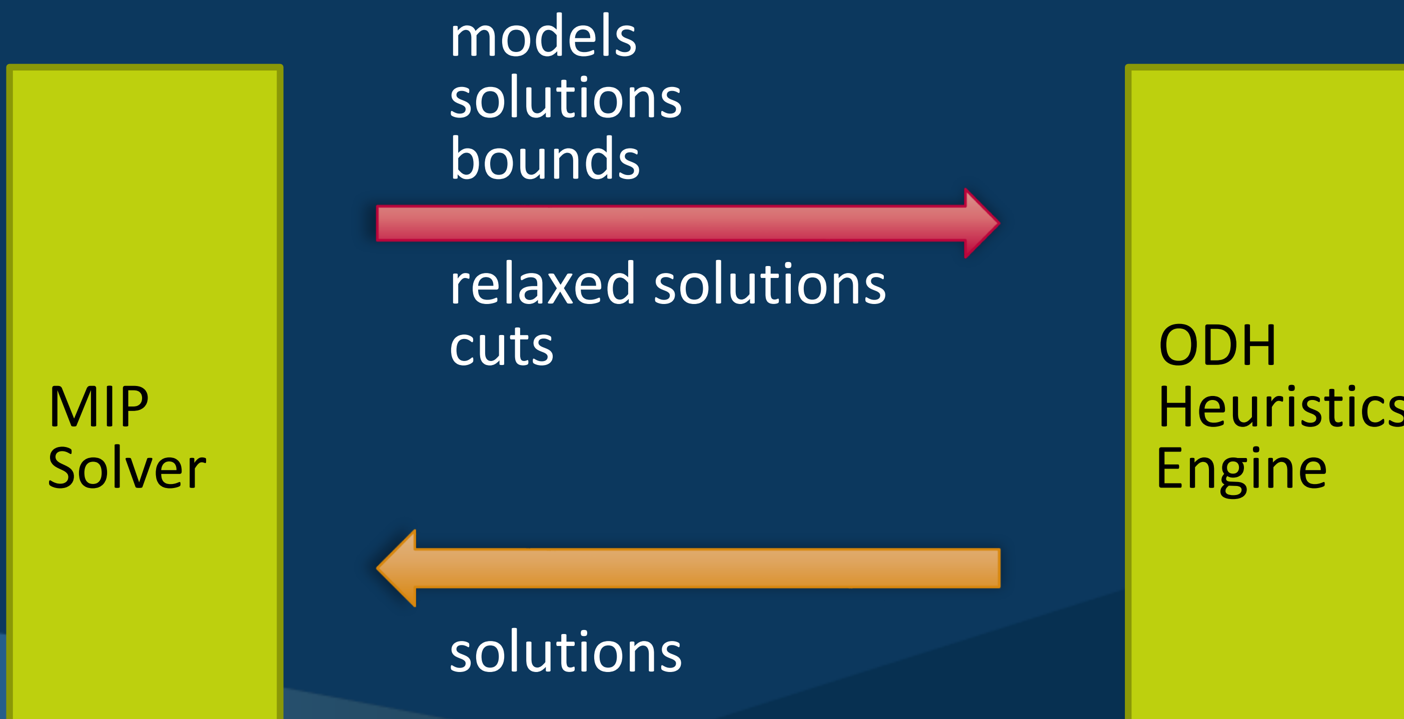
- Push the envelope of what can be usefully optimized
  - Try other methods concurrently with traditional solver
    - Use available threads (cores) more effectively
  - Get useful information by solving smaller models
    - Avoid the 'curse of dimensionality'
  - Example is Optimization Direct Heuristics (ODH)
- Accept that with most users' data aiming for 0.01% accuracy (gap) is pointless

OPTIMIZATION DIRECT

ODH : How Does it Work?

MIP solver (CPLEX, Xpress, Gurobi) and ODH run concurrently

Information is exchanged:



OPTIMIZATION DIRECT

ODHeuristics Engine

Finds a (possibly infeasible) **initial solution** with local search

Improves its **current solution**

- Decomposes original model into sub-models
- Finds better solution to sub-models (not necessarily optimal)
- phaseI or bigM if infeasible
- Each ODH thread solves its own set of sub-models
- Combines the solutions across threads
- Repeats with fresh decomposition
- Dynamically adjusts sub-model size

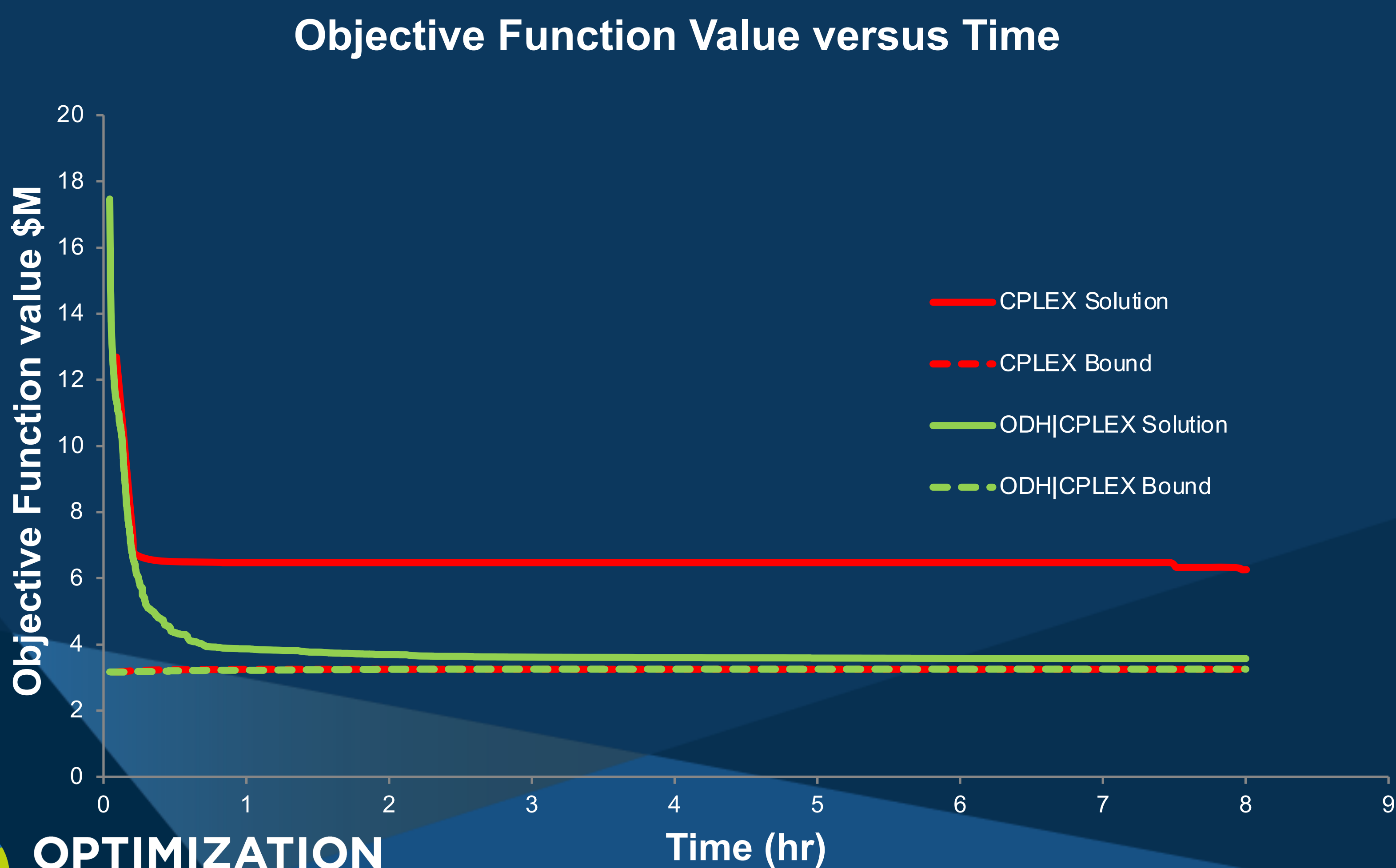
Decomposition

- Uses structure inferred from variable names and user-supplied pattern or matrix partition information; or
- Using user call-back; or
- *Automatically inferred from matrix structure*

OPTIMIZATION DIRECT

Recent Customer Model (ODH|CPLEX)

740K binaries and 12M non-zeros



OPTIMIZATION DIRECT

ODH Effectiveness

Randomly selected 100 model sub-set of 850 customer models, *Intel i4790K, 8 threads, 2 hour limit*

	ODH|CPLEX	CPLEX
Solved	23	20
Feasible	88	84
Average gap	19%	27%

i.e. 30% average reduction in gap

MIPLIB Open-v7 Models: public collection of 286 models to which an optimal solution has not been proven, feasible solution found to 257 models, none to 29

Proves optimality on 16 models

Finds better solutions than the 'best known' to 116 (45%)

Finds solutions to 5 models where no solution found before

Intel Xeon E5-2690v3, 16 threads, 2 hour limit

Applications that push the envelope

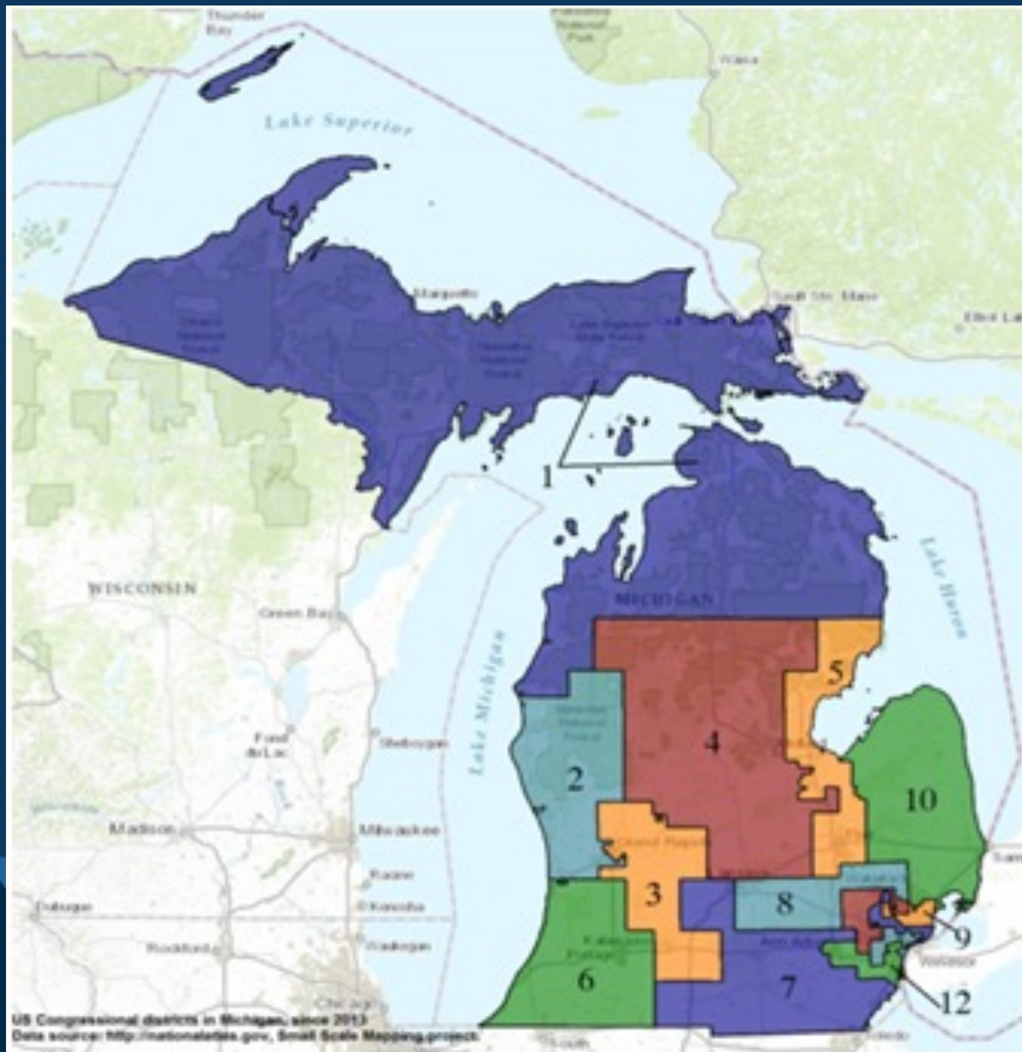
ODH is necessary for applications in areas as diverse as satellite management, forestry, retail and fiber optic network design.

Recently (2022) used for redistricting:

- Models exceptionally large:
  - 20M cons, 35M (5M binary) vars and 130M elts is midsized
  - Have used on models 5X larger.
- Usually have a (possibly poor) starting solution
- Aim for 5% gap
- Run times up to 8 hours on 24 core Xeon E5-2690v3

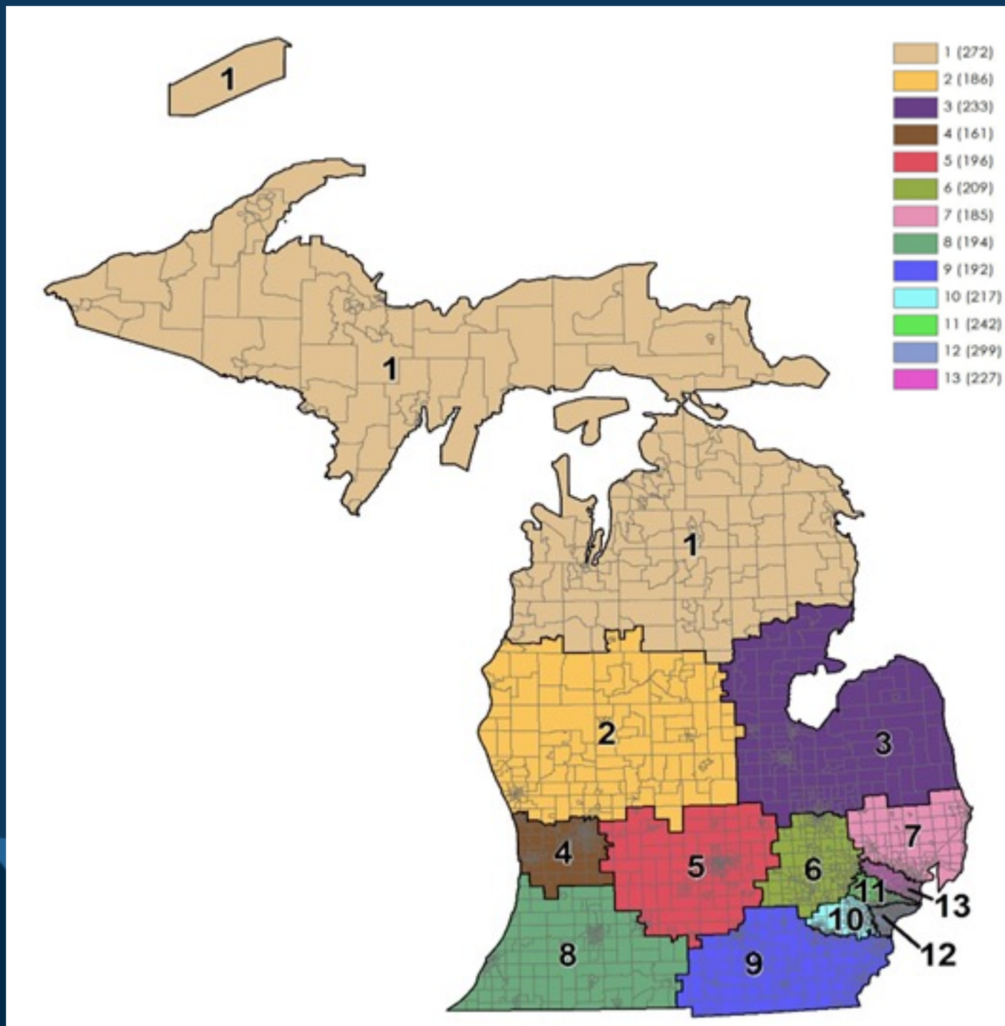
OPTIMIZATION DIRECT

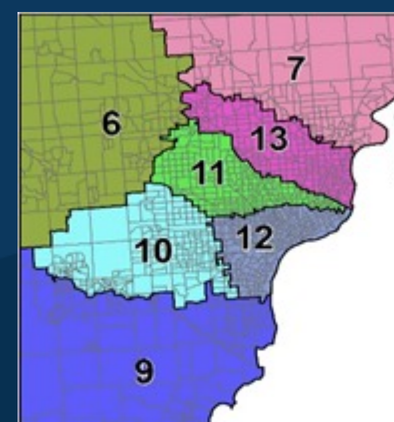
Michigan Congressional Districts 2010



OPTIMIZATION DIRECT

Optimized Michigan Congressional Districts





OPTIMIZATION DIRECT

Optimization: The Future

Non-linear and global optimization will mature

Concurrent co-working with alternative technologies

- Heavy primal heuristics, e.g. ODH
- Constraint programming, etc.
- Abandon determinism
- *Especially if bus speeds improve*

More automation in model building with AI

OPTIMIZATION DIRECT

Conclusions

Looked at what optimization is

How models are solved and how methods and hardware have evolved over last 77 years

Given an idea of methods which are pushing the envelope of its use

Looked at what the future might hold

OPTIMIZATION DIRECT

Thanks for listening

Robert Ashford

rwa@optimizationdirect.com

www.optimizationdirect.com

OPTIMIZATION DIRECT