ODH – AN OPTIMIZER FOR HARD MIPS

Robert Ashford
Alkis Vazacopoulos
Optimization Direct Inc.

INFORMS Indianapolis 2022

OPTIMIZATION
DIRECT

Summary

- Challenges of Large Scale Optimization
- The ODHeuristics approach
  - ODHeuristics Engine
  - ODH Optimizer
- New features in release 7
- Scheduling, supply chain, and telecomms examples
- MIPLIB Open-v7 Models

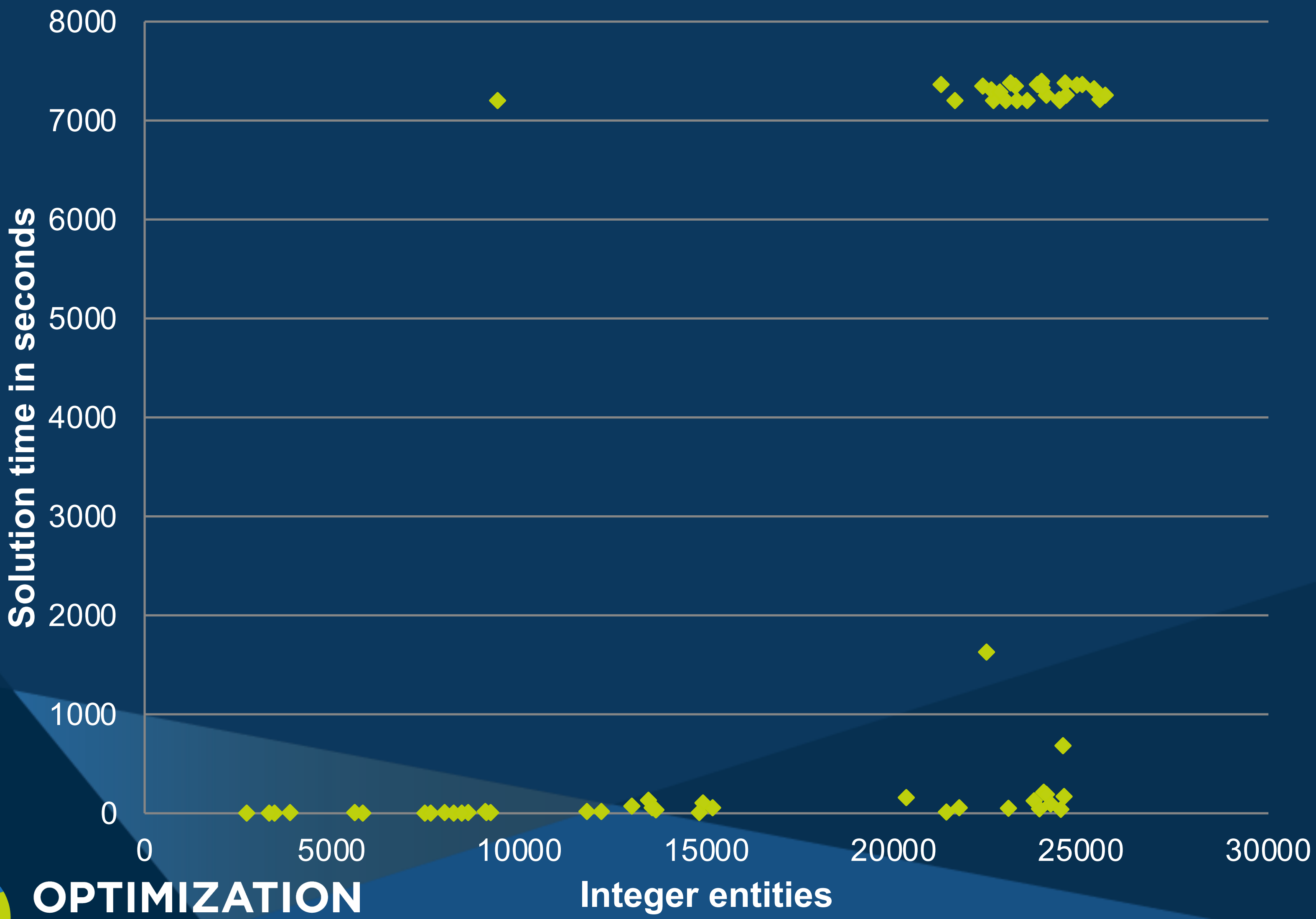
OPTIMIZATION DIRECT

The Problem: Large Scale Optimization

- Models becoming larger and more complex
- Standard optimization technology stretched/fails
- Super-linear solve time growth often supposed
- The reality is worse
  - See how solve time varies with integers after presolve (CPLEX)

OPTIMIZATION
DIRECT

Solution time vs Size



OPTIMIZATION DIRECT

ODHeuristics: What Is It?

- Tools for
  - handling large and/or difficult MIPs
  - exploits parallel hardware
    - typical server/workstation architecture
  - produces good solutions
    - uses CPLEX or Gurobi for solving sub-models
- ODHeuristics Engine
  - can be used on its own to find solutions
  - But doesn’t give an optimality guarantee (gap)
- ODH Optimizer
  - Commercial MIP optimizer (CPLEX or Gurobi) with the ODHeuristics engine inside
  - Good at getting solutions
  - Gives an optimality guarantee

ODHeuristics Engine

- Presented as a software library
  - For embedding into customer applications
  - Call-backs and controls
  - In C, C++/Concert, Java and Python (CPLEX)
  - Universal API (Gurobi)
  - Supports Windows and Linux
- Driver programs are supplied
  - For command line use
  - As examples of calling the library
- Short User Guide (PDF)
- Skeleton scripts for compiling callers and linking

ODHeuristics Engine: How Does it Work?

- Finds an initial solution
  - local search; and/or
  - ‘bigM’ and ‘phase1’ methods; or
  - using commercial MIP solver

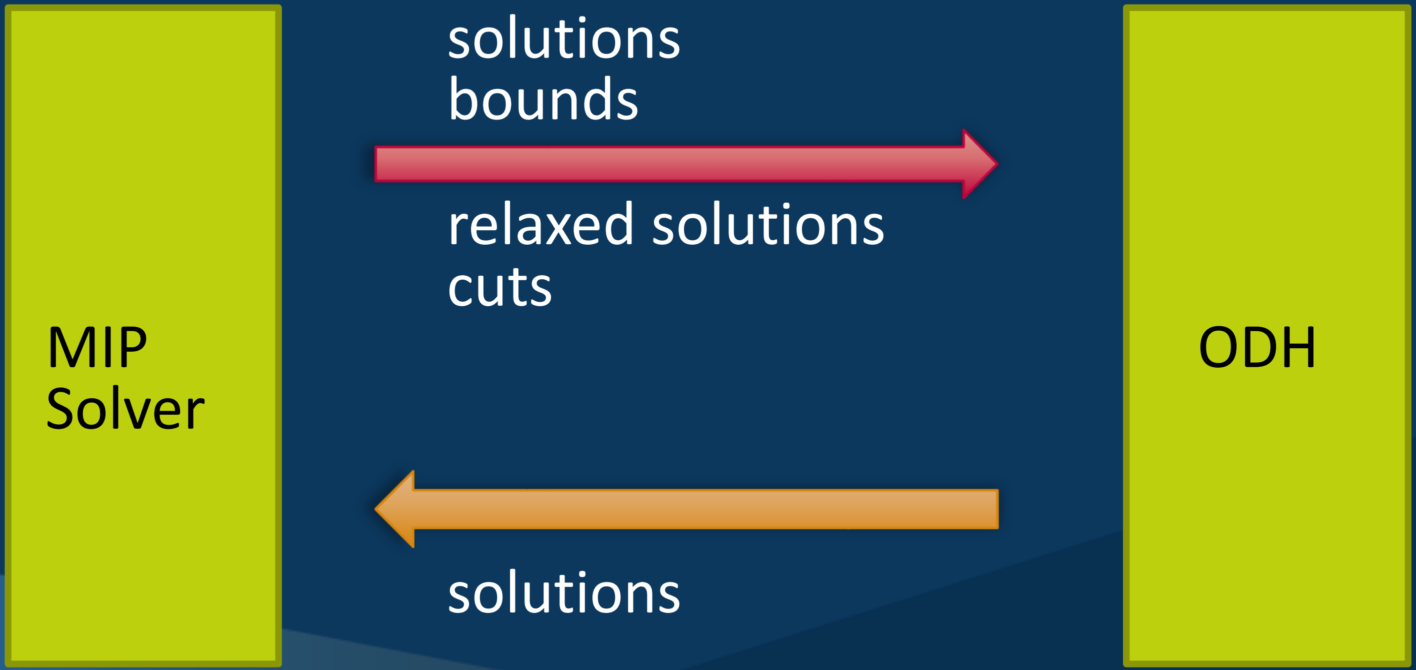
- Improves its current solution
  - Decomposes original model into sub-models
  - Finds better solution to sub-models (not necessarily optimal)
  - Each ODH thread solves its own set of sub-models
  - Combines the solutions across threads
  - Repeats with fresh decomposition
  - Progressively increases 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

- Deterministic or Opportunistic

ODH : How Does it Work?

- MIP solver and ODH run concurrently
- Information is exchanged



MIP Solver

solutions
bounds

relaxed solutions
cuts

ODH

solutions

Some Challenges

- ODH works best with ‘exact’ solutions
  - Most solvers use an integer feasibility tolerance of 10^{-5} but a feasibility tolerance of 10^{-6}
  - Consider e.g. $x \le M\,\delta$ where $0 \le x \le M$ and $\delta \in \{0,1\}$, could have

    $M = 1000$, $\delta = 10^{-5}$ (treated as 0) but $x = 0.1$
    when want $\delta = 0 \;\Rightarrow\; x = 0$

    ODH tries cleaning solutions from solver
- ODH works in initial presolved space and solver’s presolved space may change

  ⇒solver solutions may not be feasible to ODH and vice-versa
- Thread synchronization

OPTIMIZATION
DIRECT

What's New in Release 7

- Performance enhancements
  - More aggressive decomposition
  - Improved RINS heuristic
  - Improved handling of large variable bounds (>1e+9)
  - Improved cut management
- Support for Gurobi 9.5
- Support for Xpress-MP coming soon
- Can use to run CPLEX in MATLAB

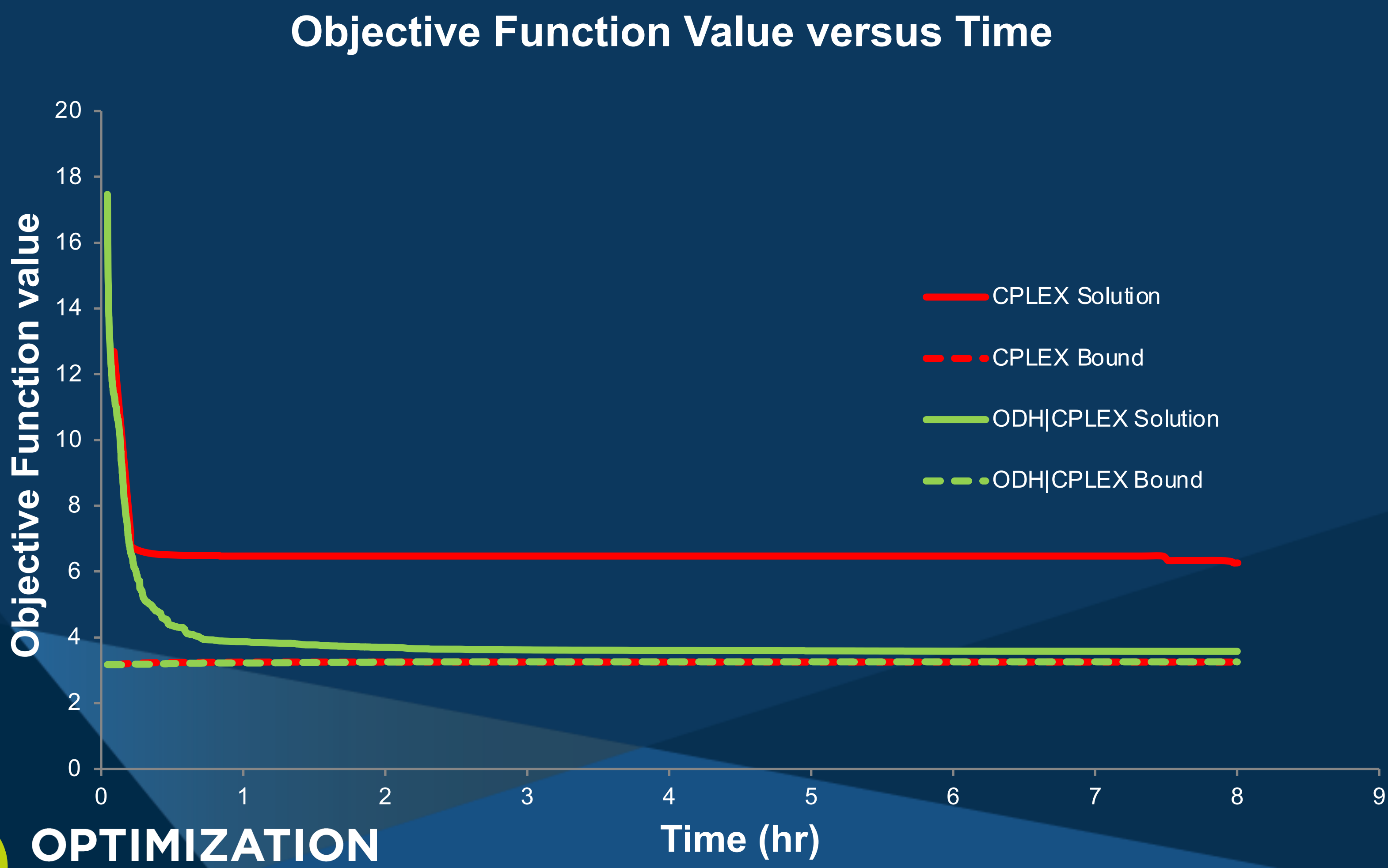
Coming Soon: Release 7

- Integration with other solvers
  - Xpress-MP
- Significant software engineering exercise
  - ~1600 calls to solver library
  - ~ 230 unique calls
  - Solvers have different data paradigms

OPTIMIZATION
DIRECT

Recent Customer Model (ODH|CPLEX)

- 740K binaries and 12M non-zeros



OPTIMIZATION DIRECT

Examples: Large Scale Scheduling, Supply Chain and Telecomms Models

Model	Application	entities	rows	cols	integers
Easy	Scheduling	314	299288	57804	57804
Mixed	Supply Chain	89177	553715	496455	153183
Medium	Scheduling	314	389560	94200	94200
Difficult	Scheduling	406	371964	149132	149132
Large	Supply Chain	302965	2836736	4892396	1827140
Phase1	Scheduling	1275	421650	155336	154828
Huge	Telecomms	27000	2577916	12944400	12944400

ODH|Gurobi Results

8 Threads on Intel 4 core i7-4790K 4GHz

	ODH|Gurobi Optimizer 7.00			Gurobi 9.52		
	Solution	**Time**	**Gap**	**Solution**	**Time**	**Gap**
Easy	96	6’	17%	96	45’	17%
	96	55’	0%	96	2hr 10’	0%
Mixed	3.1634e+06	5’	1%	3.1578e+06	5’	1%
	3.1530e+06	8hrs	0.02%	3.1529e+06	8hrs	0.01%
Medium	123	8hrs	37%	none	8hrs	inf
Difficult	852.595	8hrs	63%	none	8hrs	inf
Large	1.1444e+07	8hrs	0.8%	3.0642e+09	8hrs	99.6%
Phase1	30854.549	8hrs	1.4%	none	8hrs	inf
Huge	419	8hrs	66%	458	8hrs	69%

OPTIMIZATION
DIRECT

ODH|Gurobi Results

12 Threads on Intel 24 core Xeon E5-2690v3 3GHz

	ODH|Gurobi Optimizer 7.00			Gurobi 9.52		
	Solution	**Time**	**Gap**	**Solution**	**Time**	**Gap**
Easy	96	5’	17%	96	2hr 0’	17%
	96	33’	0%	96	2hr 8’	0%
Mixed	3.1703e+06	2’	1%	3.1583e+06	6’	1%
	3.1530e+06	8hrs	0.01%	3.1528e+06	3hr 3’	0.01%
Medium	116	8hrs	34%	none	8hrs	inf
Difficult	776.769	8hrs	59%	none	8hrs	inf
Large	1.1464e+07	8hrs	0.9%	2.0345e+07	8hrs	44%
Phase1	30778.195	8hrs	1.1%	none	8hrs	inf
Huge	424	8hrs	67%	498	8hrs	72%

OPTIMIZATION
DIRECT

ODH|Gurobi Results

16 Threads on Intel 24 core Xeon E5-2690v3 3GHz

	ODH|Gurobi Optimizer 7.00			Gurobi 9.52		
	Solution	Time	Gap	Solution	Time	Gap
Easy	96	5’	17%	96	30’	17%
	96	33’	0%	96	2hr 0’	0%
Mixed	3.1659e+06	1’	1%	3.1818e+06	6’	1%
	3.1529e+06	6hrs 40’	0.01%	3.1529e+06	3hr 46’	0.01%
Medium	118	8hrs	35%	none	8hrs	inf
Difficult	755.1883	8hrs	58%	none	8hrs	inf
Large	1.1456e+07	8hrs	0.8%	3.0408e+09	8hrs	100%
Phase1	30779.279	8hrs	1.1%	none	8hrs	inf
Huge	407	8hrs	66%	493	8hrs	72%

OPTIMIZATION
DIRECT

ODH Customer Models

- Have collected 850 customer models
- Regularly test ODH on a randomly selected 100 model sub-set
- 8 threads on 4 core Intel i4790K, 2 hour time limit
- ODH useful tool on typical models:

	ODH|Gurobi	Gurobi
Solved	29	31
Feasible	90	83
Average gap	15%	23%

Miplib Open-v7 Models

- Public collection of 286 models to which an optimal solution has not been proven
- 257 models are known to have a feasible solution
- No solution found to 29 models
- Tried ODH|CPLEX on this set
- Proves optimality on 13 models [16 with 16 threads]
- Finds better solutions than the ‘best known’ in 2 hours to 101 (39%) of them with 8 threads [116 (45%) with 16 threads]
- Finds solutions to 5 models where no solution found before

Applictions

- ODH is necessary for applications in areas as diverse as satellite management, forestry, retail and fiber optic network design.
- Recently used for redistricting:
  - Models exceptionally large:
    20M rows, 35M (5M binary) cols 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
  - MB71 track on Monday, October 17, 11:00 am - 12:15 pm

Conclusions

- Customers now want to solve larger and larger models
- Hard size barriers to solve (to optimality) or even to getting a solution at all
- ODHeuristics can find good solutions
  - Useful on small(er) models too
- ODH can provide solutions of proven optimality quality
- Parallel solution methods best way of exploiting modern hardware (although limited by memory bus speeds)

Benchmarking and Evaluation

- If you think that ODHeuristics and/or ODH might work for you:
- send us your difficult matrices and we will send you the results
- request an evaluation copy

Thanks for listening

Robert Ashford
rwa@optimizationdirect.com
www.optimizationdirect.com

OPTIMIZATION
DIRECT