

# An Introduction to ODH-CPLEX and Recent Computational Results

Alkis Vazacopoulos  
Robert Ashford  
Optimization Direct Inc.  
April 2017

# Summary

- Challenges of Large Scale Optimization
- Exploiting parallel hardware
- The ODHeuristics approach
  - ODHeuristics Engine
  - ODH-CPLEX Optimizer
- Scheduling, supply chain and telecomms examples

# The Problem: Large Scale Optimization

- Customers and prospective customers present increasing larger and more complex models
- Standard optimization technology fails
  - Cannot find a solution at all
    - Satellite scheduling, tournament planning
  - Solutions too poor to have any value
    - Telecomms
  - Usable solutions take too long to find
    - Supply chain

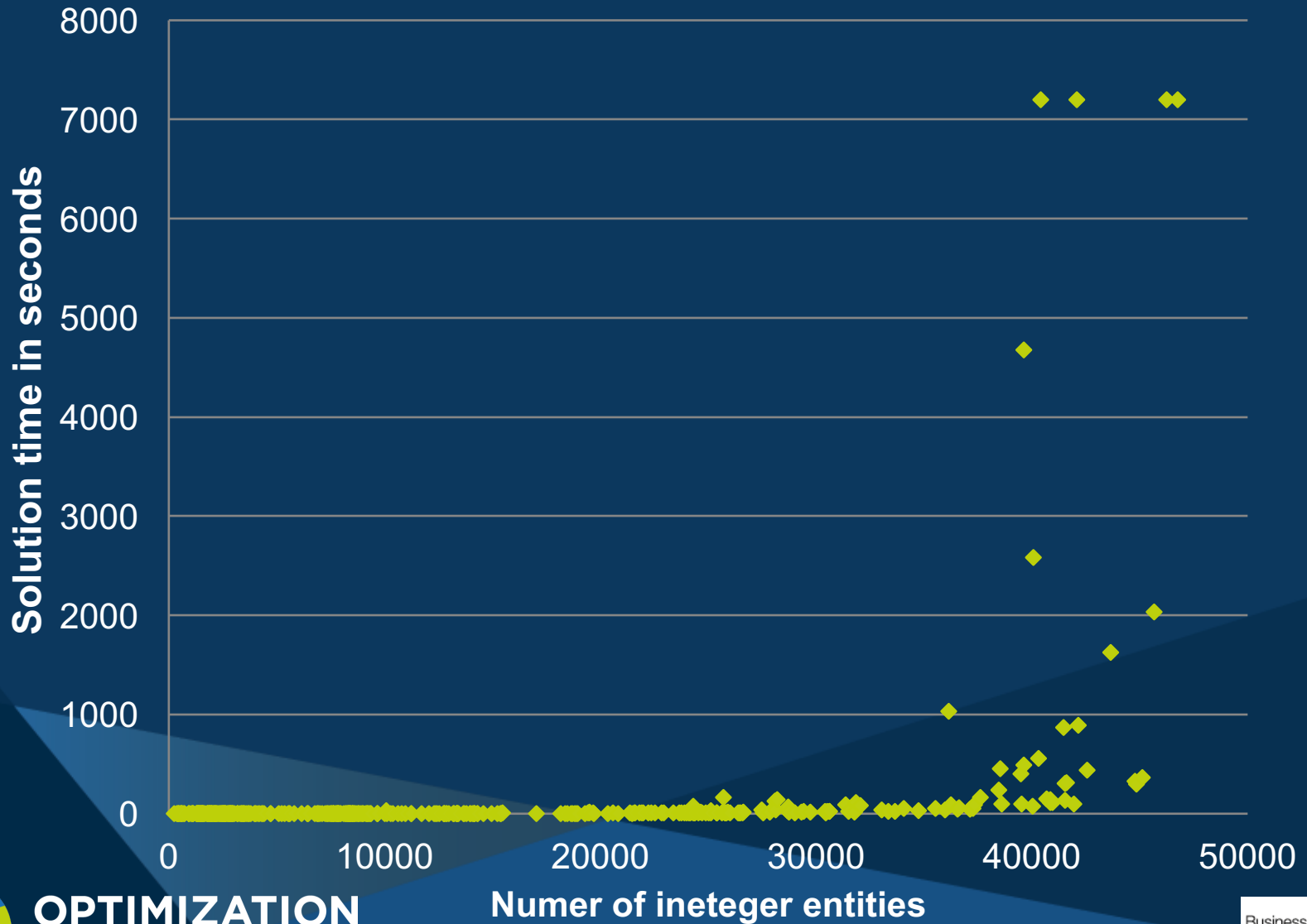
# Why Is There a Problem?

- Many models now solved routinely which would have been impossible (“unsolvable”) a few years ago
- **BUT:** have super-linear growth of solving effort as model size/complexity increases
- **AND:** customer models keep getting larger
  - More and better data are available (“Big Data”)
  - Globalized business has larger and more complex supply chain
  - Optimization expanding into new areas, especially scheduling
  - Detailed models easier to sell to management and end-users

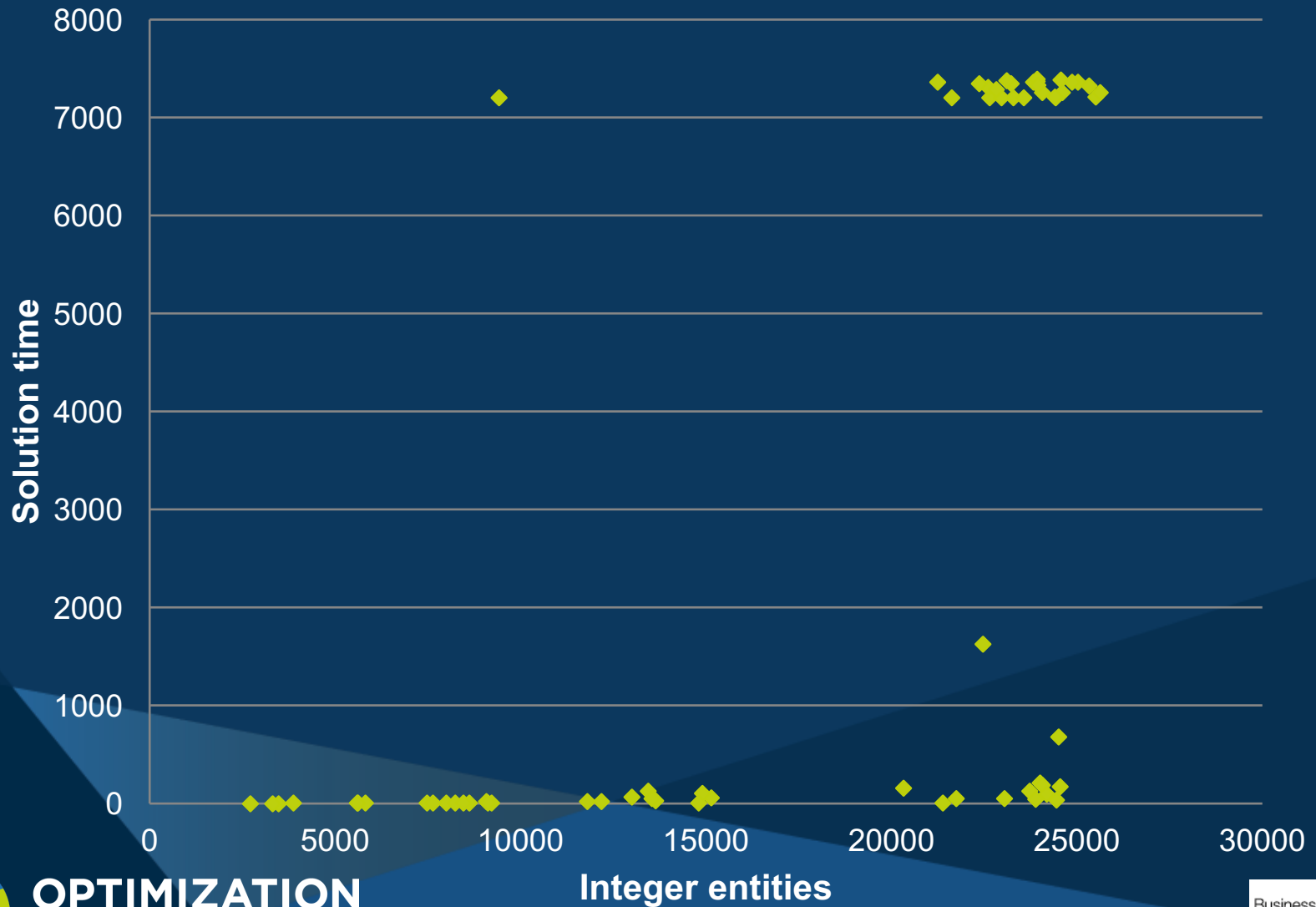
# The Curse of Dimensionality: Size Matters

- Super-linear solve time growth often supposed
- The reality is worse
- Few data sets available to support this
- Look at randomly selected sub-models of two scheduling models
  - Simple basic model
  - More complex model with additional entity types
  - Two hour time limit on each solve
  - 8 threads on 4 core hyperthreaded Intel i7-4790K
- See how solve time varies with integers after presolve

# Simple model



# Complex model



# Parallelization and Hardware

- CPU clock speed is not improving
  - Power consumed (and heat generated)  $\sim$  speed<sup>2</sup>
- Memory speed is not improving (much)
- Can get wider registers (vectorization)
  - of limited use in sparse optimization
- Multi-core (processor) machines
  - Can fit more processors onto a single chip
  - 24 cores now on inexpensive servers
  - Exploited by ODHeuristics and CPLEX multi-threading
  - Cannot always use full processor capability if using many cores
  - Performance limited by bus speed ( $\sim$ 20MB/sec)



# ODHeuristics: What Is It?

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

# ODHeuristics Engine

- Presented as a software library
  - For embedding into customer applications
  - Call-backs and controls
  - In C, C++/Concert and Java
  - 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 CPLEX
- 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
  - Automatically inferred from matrix structure
- Deterministic or Opportunistic

# Examples: Large Scale Scheduling, Supply Chain and Telecomms Models

<b>Model</b>	<b>entities</b>	<b>rows</b>	<b>cols</b>	<b>integers</b>
Easy	314	299288	57804	57804
Medium	314	389560	94200	94200
Difficult	406	371964	149132	149132
Large	302965	2836736	4892396	1827140
Huge	27000	2577916	12944400	12944400

# Heuristic Results

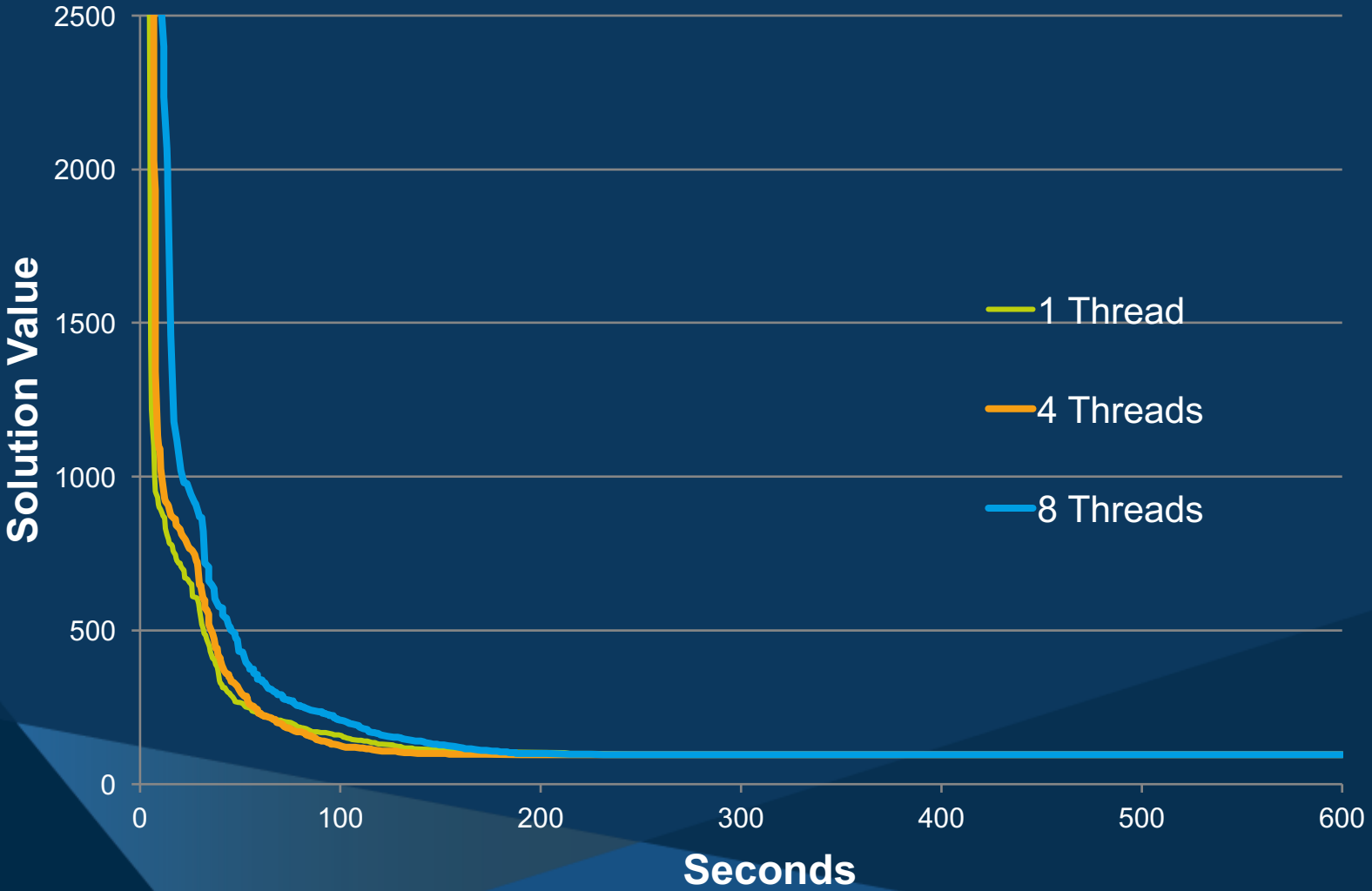
8 Threads on Intel 24 core Xeon E5-2690v3 3GHz

ODHeuristics 2.14

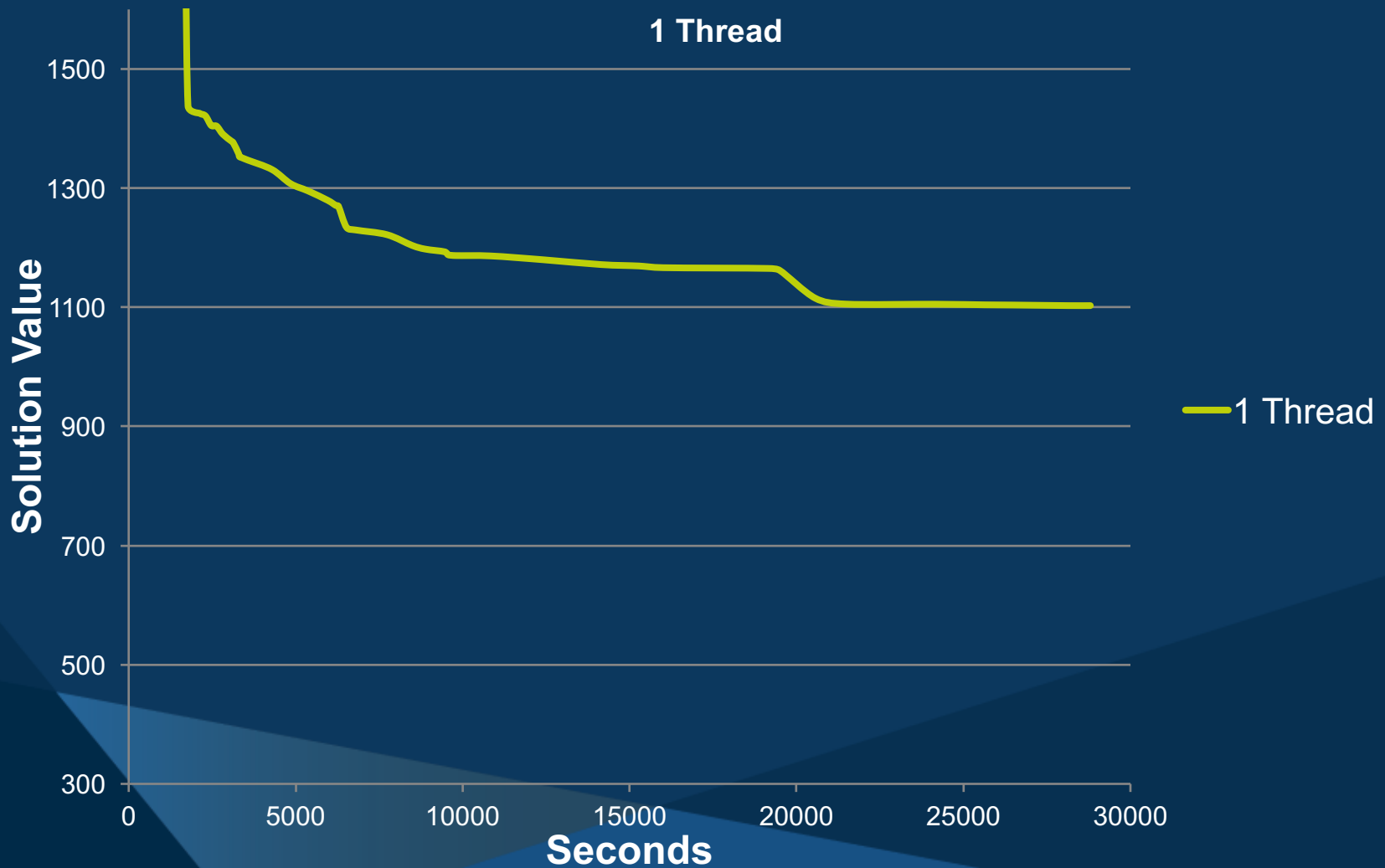
CPLEX 12.6.3

	Solution	Time	Gap	Solution	Time	Gap
Easy	96	4 mins	0%	96	4 ½ hrs	0%
Medium	113	8 hrs	≤ 13%	1161	3 ½ days	93%
Difficult	773.8	8 hrs	≤ 56%	106682	3 ½ days	99.7%
Large	1.149E+7	8 hrs	≤ 1%	1.420E+7	3 days	20%
Huge	370	8 hrs	≤ 61%	412	3 days	66%

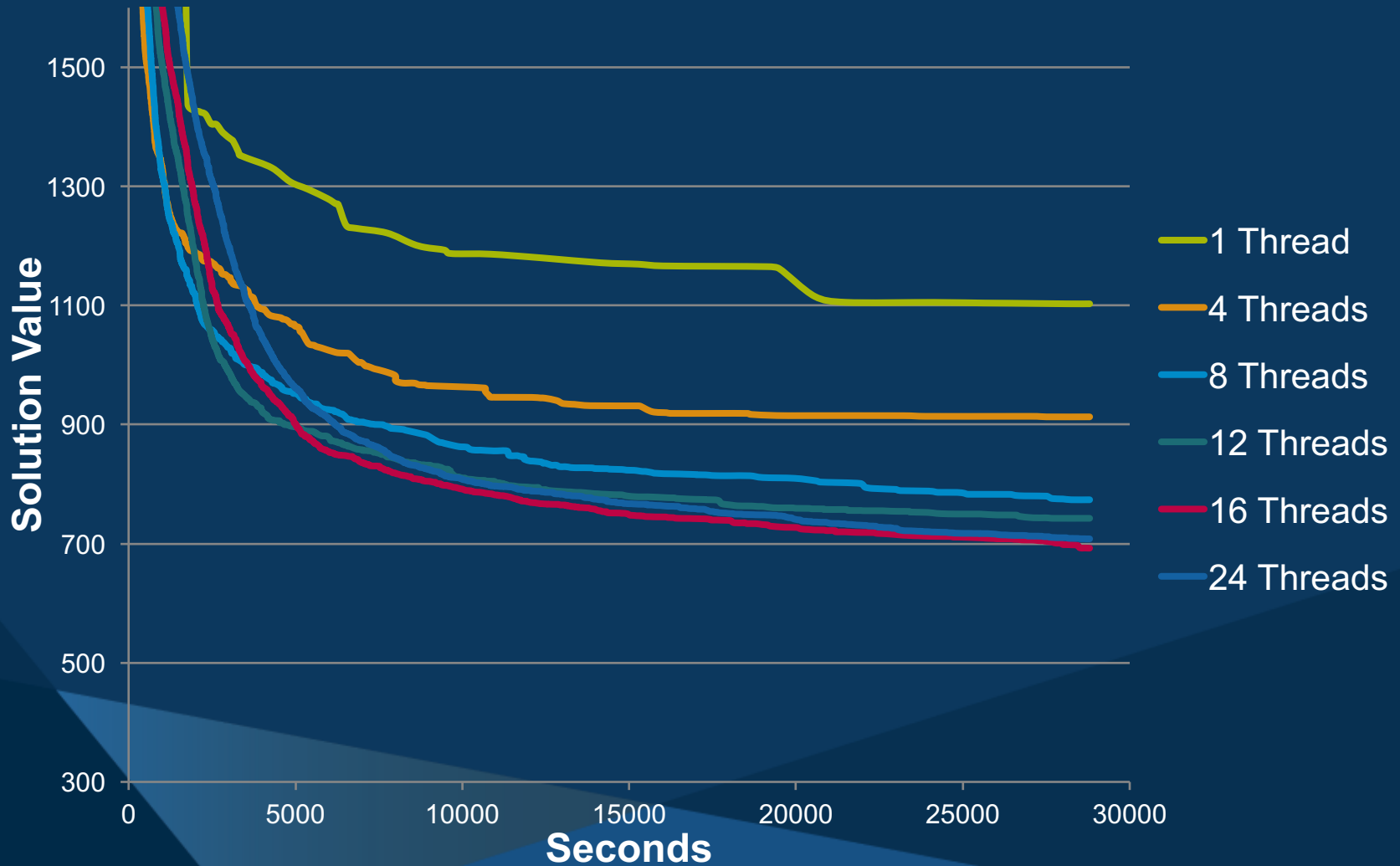
# Easy Model Heuristic Behavior



# Difficult Model Heuristic Behavior

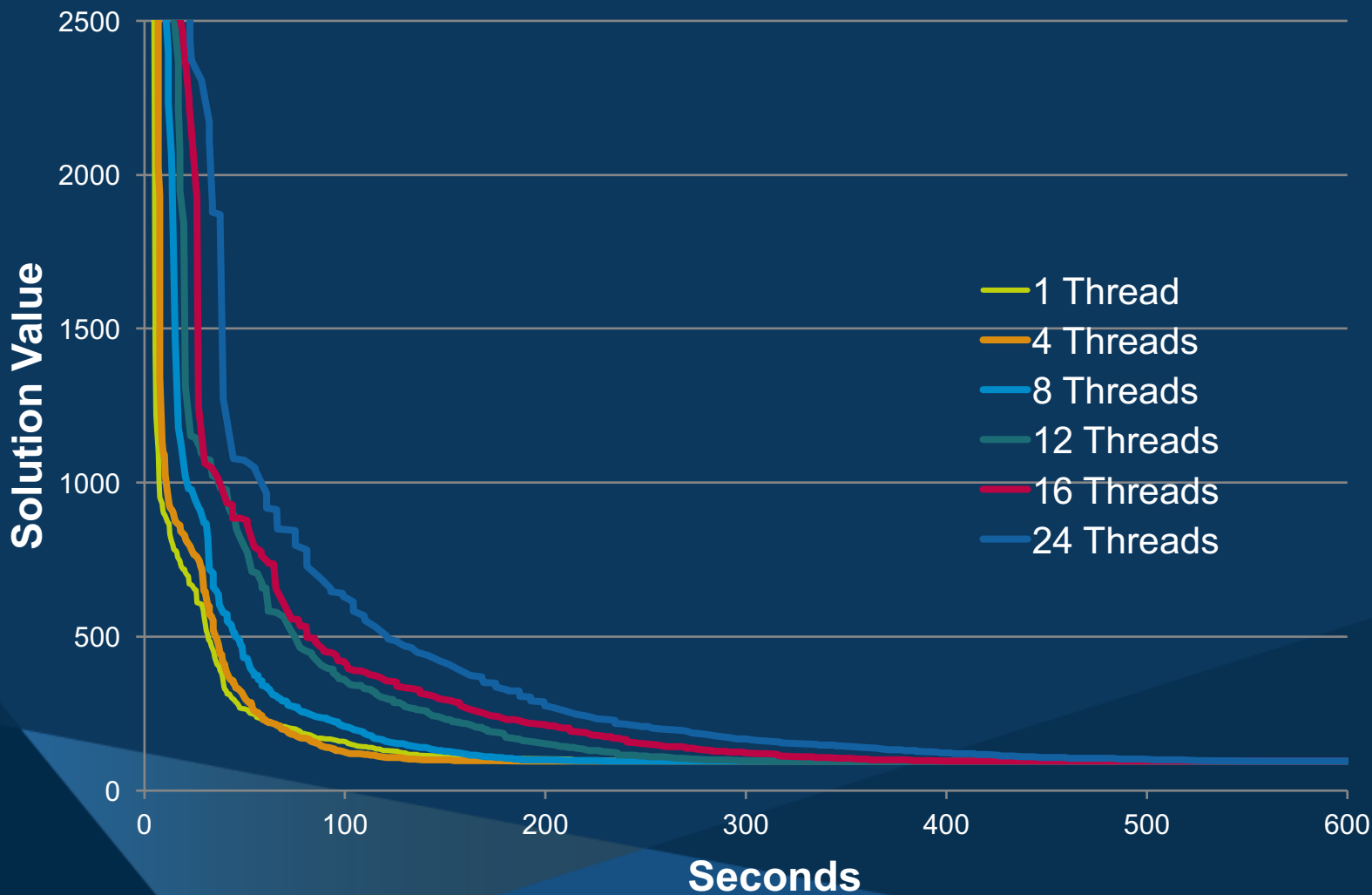


# Difficult Model Heuristic Behavior

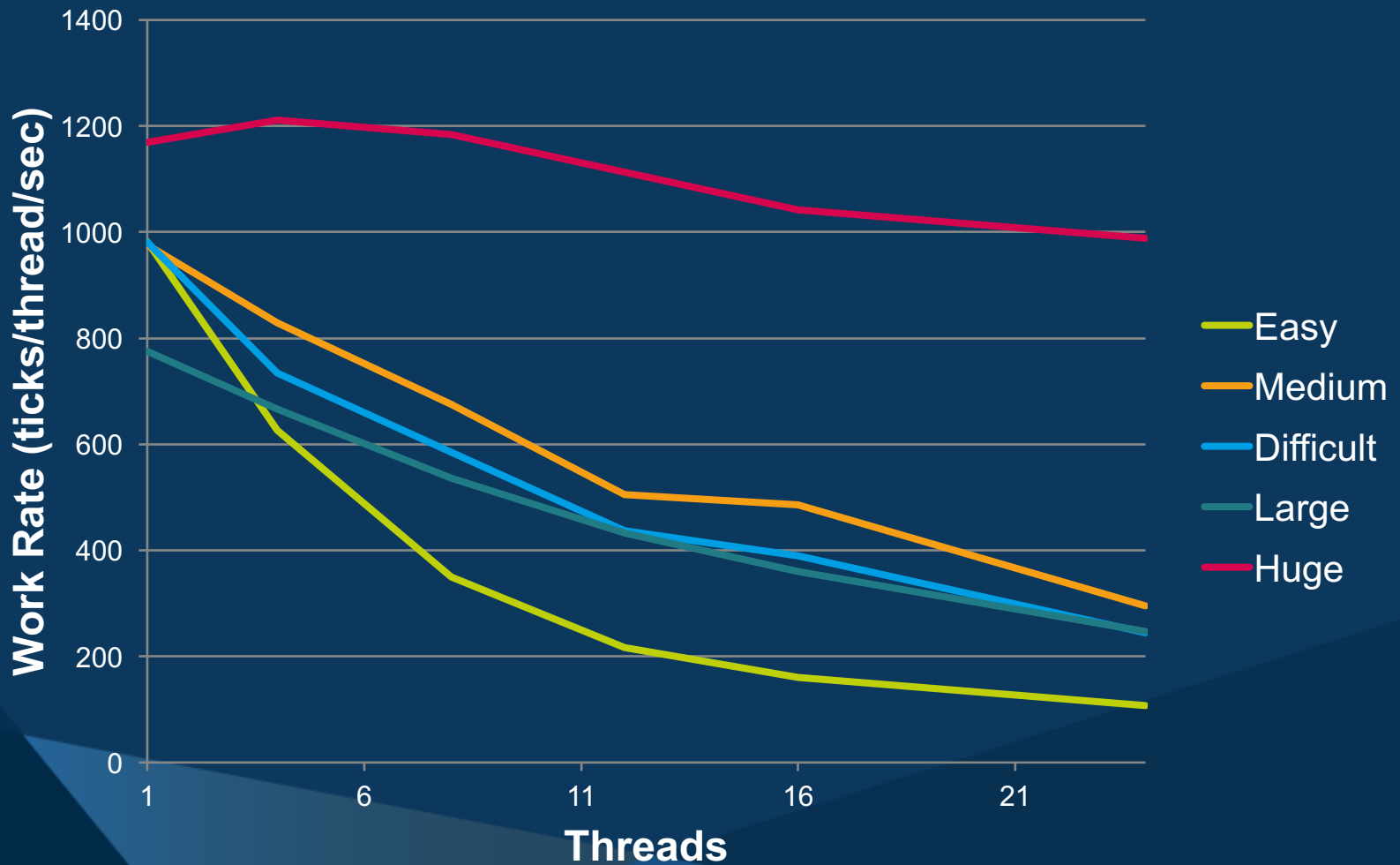




# Easy Model Heuristic Behavior



# Work Rate



# ODH-CPLEX Optimizer

- ODH Engine run underneath CPLEX
- Co-running continuously in separate (sets of) threads
- CPLEX initiates run of ODH
- Either can terminate and stop the other
  - time limit, gap, etc.
- Information exchanged between CPLEX and ODH Engine
- Deterministic or opportunistic modes

# ODH-CPLEX Design

- Presented as a subroutine library like CPLEX itself
- All CPLEX API calls are available including
  - Call-backs
  - Parameter interrogation and setting
- Just a different optimize call
- Behaves like CPLEX if ODH doesn't find anything useful (as much as technically possible)
- Supports ODH engine controls and call-back
- Supports C++/Concert and Java
- Supports Windows and Linux (64 bit only)

# ODH-CPLEX Optimizer Advantages

- Potentially faster than either alone
- Good solutions from ODH accelerate CPLEX, e.g:
  - reduced cost fixing
  - **tree pruning**
  - helps CPLEX heuristics
    - saves effort
    - provides better starting information (e.g. RINS)
- Good solutions from CPLEX accelerate ODH Engine
  - focus ODH's attention on different parts of model
- Use more cores than ODHeuristics stand-alone

# ODH-CPLEX Optimizer Challenges

- Synchronization overheads in deterministic mode
  - For CPLEX up to 20% but usually less than 5%
- CPLEX solution paths are changed by just the presence of communication channel with ODH (even if no information is exchanged)
- Needs more memory than CPLEX or ODH alone
  - Twice as much
- Needs more processors/cores
  - 8+ cores to run well, 24+ cores ideal
  - Hyperthreaded (SMT) core splitting not a good idea
- Increases demand on memory bus
  - But less than if all threads were committed to ODH alone

# ODH-CPLEX Results

- Similar to CPLEX alone on MIPLIB2010
  - Tested on subsets used by Mittelmann
  - 4 threads and 24 threads
  - Resources spent on ODH engine nearly compensate for loss to main CPLEX solver
    - threads and memory bus contentions
  - Comparisons done with same number of threads in total
- Usually very beneficial on hard user models

# Further Examples: Large Scale Scheduling and Supply Chain Models

<b>Model</b>	<b>entities</b>	<b>rows</b>	<b>cols</b>	<b>integers</b>
Mixed	89177	553715	496455	153183
Phase1	283	421650	155336	154828



# ODH-CPLEX Results

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

	CPLEX 12.7.0			ODH-CPLEX Optimizer 3.0.2		
	4 threads (deterministic)			3 (CPLEX) + 1 (ODH) thrds (det)		
	Solution	Time	Gap	Solution	Time	Gap
Easy	96	14132"	0%	96	12973"	0%
Mixed	3158738	1385"	1%	3156936	463"	1%
	3153046	8 hrs	0.05%	3153036	8 hrs	0.05%
Medium	none	8 hrs	$\infty$	134	8 hrs	43%
Difficult	none	8 hrs	$\infty$	1185	8 hrs	73%
Large	1.23548e+7	8 hrs	8.1%	1.14764e+7	8 hrs	1%
Phase_1	none	8 hrs	$\infty$	51491	8 hrs	41%
Huge	523	8 hrs	73%	404	8 hrs	65%

# ODH-CPLEX Results

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

	CPLEX 12.7.0			ODH-CPLEX Optimizer 3.0.2		
	24 threads (deterministic)			18 (CPLEX) + 6 (ODH) thrds (det)		
	Solution	Time	Gap	Solution	Time	Gap
Easy	96	8720"	0%	96	4 hrs	17%
Mixed	3164927	781"	1%	3171650	483"	1%
	3151448	8 hrs	0%	3153247	4 hrs	0.06%
Medium	1774	8 hrs	96%	121	4 hrs	36%
Difficult	none	8 hrs	$\infty$	1050	4 hrs	69%
Large	1.23548e+7	8 hrs	8.05%	1.14857e+7	4 hrs	1%
Phase_1	none	8 hrs	$\infty$	32635	4 hrs	6.7%
Huge	428	8 hrs	67%	394	4 hrs	65%

# ODH-CPLEX Results: Recent User Models

<b>Model</b>	<b>rows</b>	<b>cols</b>	<b>nonzeros</b>	<b>binaries</b>	<b>integers</b>
Alabama	126367	107788	431668	23652	0
Alaska	732311	719937	3384573	664678	0
Arizona	343629	336960	1538418	299656	0
Arkansas	813225	675862	2953834	25518	14466
California	315763	153037	2537381	82723	0
Colorado	138562	109964	552333	69000	0
Connecticut	108675	81488	415981	54000	0
Delaware	395435	543700	2235636	8695	7050
Florida	395439	543700	2235640	8695	7050
Georgia	45840	70697	263418	10944	4119
Hawaii	579250	509399	2457594	72202	0
Idaho	866477	1176512	4788339	87721	1
Illinois	40474	44304	177842	9552	4807
Indiana	37224	65845	211514	6979	5909
Iowa	105704	79070	421342	52500	0

# ODH-CPLEX Results: Recent User Models

Model	CPLEX		ODH-CPLEX	
	Time	Gap	Time	Gap
Alabama	678.07	0%	229.78	0%
Alaska	7202.58	8.52	7203.63	6.85%
Arizona	7201.28	6.52%	7201.21	5.25%
Arkansas	7243.64	89.52%	7218.71	7.86%
California	7201.32	inf	7201.81	inf
Colorado	7200.20	93.65%	7200.23	26.79%
Connecticut	7200.68	26.76%	7202.22	14.70%
Delaware	571.82	0%	521.79	0%
Florida	238.25	0%	73.52	0%
Georgia	7.19	0%	4.35	0%
Hawaii	7200.55	21.87%	7216.42	0.49%
Idaho	267.10	0%	397.29	0%
Illinois	7222.21	4.57%	7214.97	6.23%
Indiana	0.20	0%	0.35	0%
Iowa	7200.68	9.86%	7201.20	5.82%

# Conclusions

- Customers now want to solve larger and large 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-CPLEX 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-CPLEX 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](mailto:rwa@optimizationdirect.com)

[www.optimizationdirect.com](http://www.optimizationdirect.com)

