An Introduction to ODH|CPLEX

Alkis Vazacopoulos
Robert Ashford
Optimization Direct Inc.

April 2018

OPTIMIZATION
DIRECT

Business Partner

Summary

- New features
- Challenges of Large Scale Optimization
- The ODHeuristics approach
  - ODHeuristics Engine
  - ODH|CPLEX Optimizer
- Scheduling, supply chain, telecomms and benchmark examples

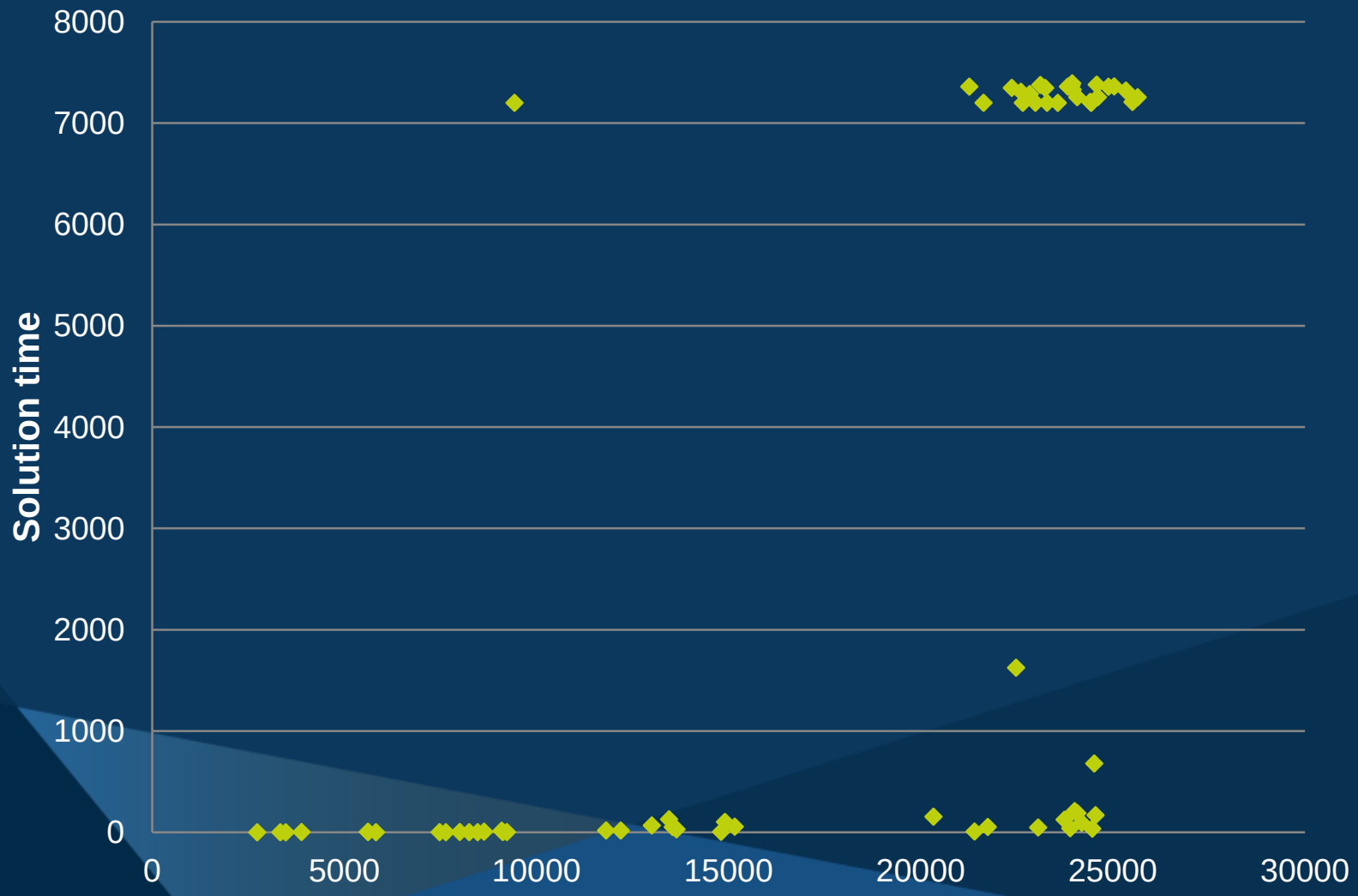
Whats's New

- Python support (2.7, 3.5 and 3.6)
- Index Key call-back
  - flexible user-specified decomposition
- Periodic RINS heuristic
  - Effective on *some* models
- Support for CPLEX 12.8

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

Solution time vs Size



OPTIMIZATION DIRECT

Business Partner

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, Java and Python
  - 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
  - Using user call-back; or
  - Automatically inferred from matrix structure
- Deterministic or Opportunistic

Examples: Large Scale Scheduling, Supply Chain and Telecomms Models

Model	entities	rows	cols	integers
Easy	314	299288	57804	57804
Mixed	89177	553715	496455	153183
Medium	314	389560	94200	94200
Difficult	406	371964	149132	149132
Large	302965	2836736	4892396	1827140
Phase_1	283	421650	155336	154828
Huge	27000	2577916	12944400	12944400

OPTIMIZATION DIRECT

Business Partner

ODH|CPLEX Results

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

	CPLEX 12.7.1			ODH|CPLEX Optimizer 3.2.0		
	4 threads (deterministic)			3 (CPLEX) + 1 (ODH) thrds (det)		
	Solution	Time	Gap	Solution	Time	Gap
Easy	96	2 hr 52’	0%	96	1 hr 13’	0%
Mixed	3159264	10’	1%	3156023	7’	1%
	3153147	8 hrs	0.05%	3153082	8 hrs	0.05%
Medium	1797	8 hrs	96%	126	8 hrs	39%
Difficult	none	8 hrs	∞	1564	8 hrs	79%
Large	1.23548e+7	8 hrs	8.1%	1.14820e+7	8 hrs	1%
Phase_1	none	8 hrs	∞	none	8 hrs	∞
Huge	601	8 hrs	77%	502	8 hrs	72%

OPTIMIZATION DIRECT

Business Partner IBM

ODH|CPLEX Results

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

	CPLEX 12.7.1			ODH|CPLEX Optimizer 3.2.0		
	12 threads (deterministic)			9 (CPLEX) + 3 (ODH) thrds (det)		
	Solution	Time	Gap	Solution	Time	Gap
Easy	96	3 h 19’	0%	96	4 hr 11’	0%
Mixed	3164927	13’ 1”	1%	3172067	7’ 20”	1%
	3151448	8 hrs	0%	3152986	8 hrs	0.04%
Medium	1760	8 hrs	96%	120	8 hrs	36%
Difficult	none	8 hrs	∞	854	8 hrs	62%
Large	1.414254e+8	8 hrs	92%	1.146618e+7	8 hrs	1%
Phase_1	none	8 hrs	∞	33879	8 hrs	10%
Huge	428	8 hrs	67%	411	8 hrs	66%

OPTIMIZATION DIRECT

Business Partner IBM

More Customer Models: Statistics

Model	rows	cols	nonzeros	binaries	integers
Rprs	126367	107788	431668	23652	0
Ship4_14	732311	719937	3384573	664678	0
Ship4_8	343629	336960	1538418	299656	0
20160824	138562	109964	552333	69000	0
20160920	108675	81488	415981	54000	0
822_p1	395435	543700	2235636	8695	7050
822_p2	395439	543700	2235640	8695	7050
824_p1	45840	70697	263418	10944	4119
Apso374-2	579250	509399	2457594	72202	0
Big_p1	866477	1176512	4788339	87721	1
Dsj-12Jan_p1	40474	44304	177842	9552	4807
Gom_p1	37224	65845	211514	6979	5909
IG_20161019	105704	79070	421342	52500	0

More Customer Models: Results

24 Threads	CPLEX 12.7.1		ODH|CPLEX 3.2.0	
Model	Gap	Time	Gap	Time
Rprs	0%	516	0%	273
Ship4_14	8.3%	7200	6.8%	7200
Ship4_8	5.3%	7200	5.3%	7200
20160824	92.8%	7200	32.7%	7200
20160920	22.2%	7200	9.9%	7200
822_p1	0%	479	0%	94
822_p2	0%	276	0%	113
824_p1	0%	9	0%	2
Apso374-2	21.9%	7200	0.5%	7200
Big_p1	0%	276	0%	350
Dsj-12Jan_p1	4.9%	7200	4.4%	7200
Gom_p1	0%	0.2	0%	0.4
IG_20161019	11.1%	7200	5.9%	7200

OPTIMIZATION DIRECT

Business Partner IBM

More Customer Models: Results

Summary

Tests run on 24 threads on 24 core Intel Xeon E5-2690v3 3Ghz under win64

2 hour time limit

	CPLEX 12.7.1		ODH|CPLEX 3.2.0	
	Gap	**Time**	**Gap**	**Time**
Average	12.8%	771	5.0%	579

OPTIMIZATION DIRECT

Business Partner IBM

ODH|CPLEX Benchmark Results
Mittelmann's 'Solvable' MIPLIB2010 subset

Intel Xeon Gold 6138, 40 cores, 256GB, 2.00GHz, Linux

Current solvable MIPLIB2010 subset less numerically unstable models

Tests run by Hans **NOT** Optimization Direct

40 Threads, 2 hour time limit

Look at relative performance models taking at least τ minutes

	CPLEX 12.8.0 takes at least τ			ODH|CPLEX takes at least τ		
τ	# models	s.g.m.	a.m.	# models	s.g.m.	a.m.
0	220	-1.6%	+7.4%	220	-1.6%	+7.4%
5	37	+19%	+12%	43	-10%	+4.7%
10	27	+21%	+13%	29	-17%	+4.1%

OPTIMIZATION DIRECT

Business Partner

ODH|CPLEX Benchmark Results
Mittelmann's 'Slightly Pathalogical' Models

"This benchmark is not giving a representative impression of the relative performance of the codes"

Intel Xeon X5680 (32GB, Linux, 64 bits, 2*6 cores)

Tests run by Hans **NOT** Optimization Direct

12 Threads, 3 hour time limit

Look at relative performance models taking at least τ minutes

	CPLEX 12.8.0 takes at least τ			ODH|CPLEX takes at least τ		
τ	# models	s.g.m.	a.m.	# models	s.g.m.	a.m.
0	40	+12.6%	+7.5%	40	+12.6%	+7.5%
5	30	+15%	+7.5%	26	+1.9%	+6.1%
10	23	+13%	+7.8%	23	+0.6	+5.8%

OPTIMIZATION DIRECT

Business Partner IBM

ODH|CPLEX Performance

- Relatively better for hard-to-solve models
  - Not necessarily large
  - Can get benefit on small models
- Relatively better when run with more threads
  - Better to have at least 6 cores
  - Hardware performance bottleneck currently bus speed

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
www.optimizationdirect.com

OPTIMIZATION DIRECT

Business Partner

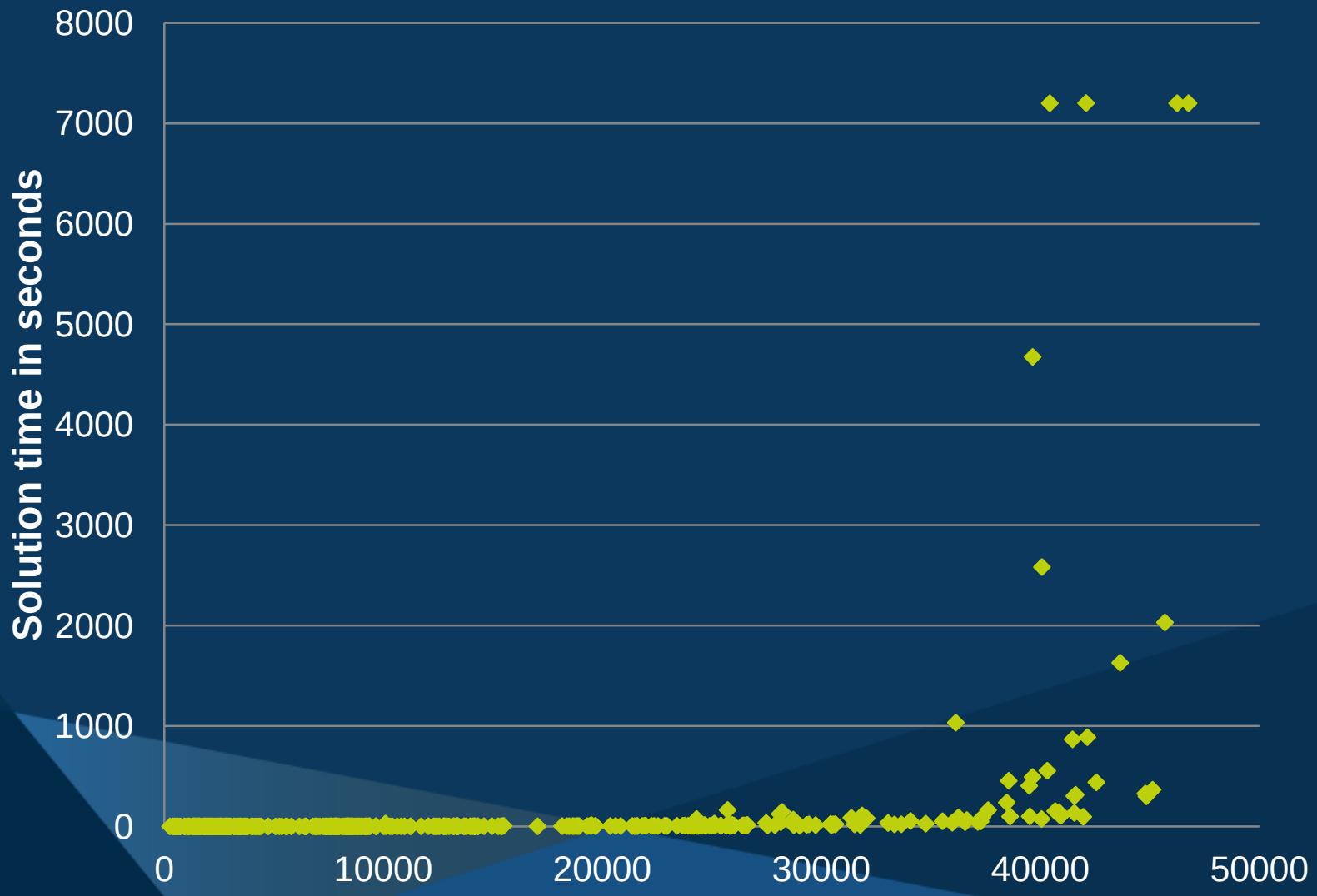
Why Is There a Problem?

- 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



OPTIMIZATION DIRECT

Business Partner

Parallelization and Hardware

- CPU clock speed is not improving
  - Power consumed (and heat generated) ~ speed^2
- 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 (~20MB/sec)

Further Examples: Large Scale Scheduling and Supply Chain Models

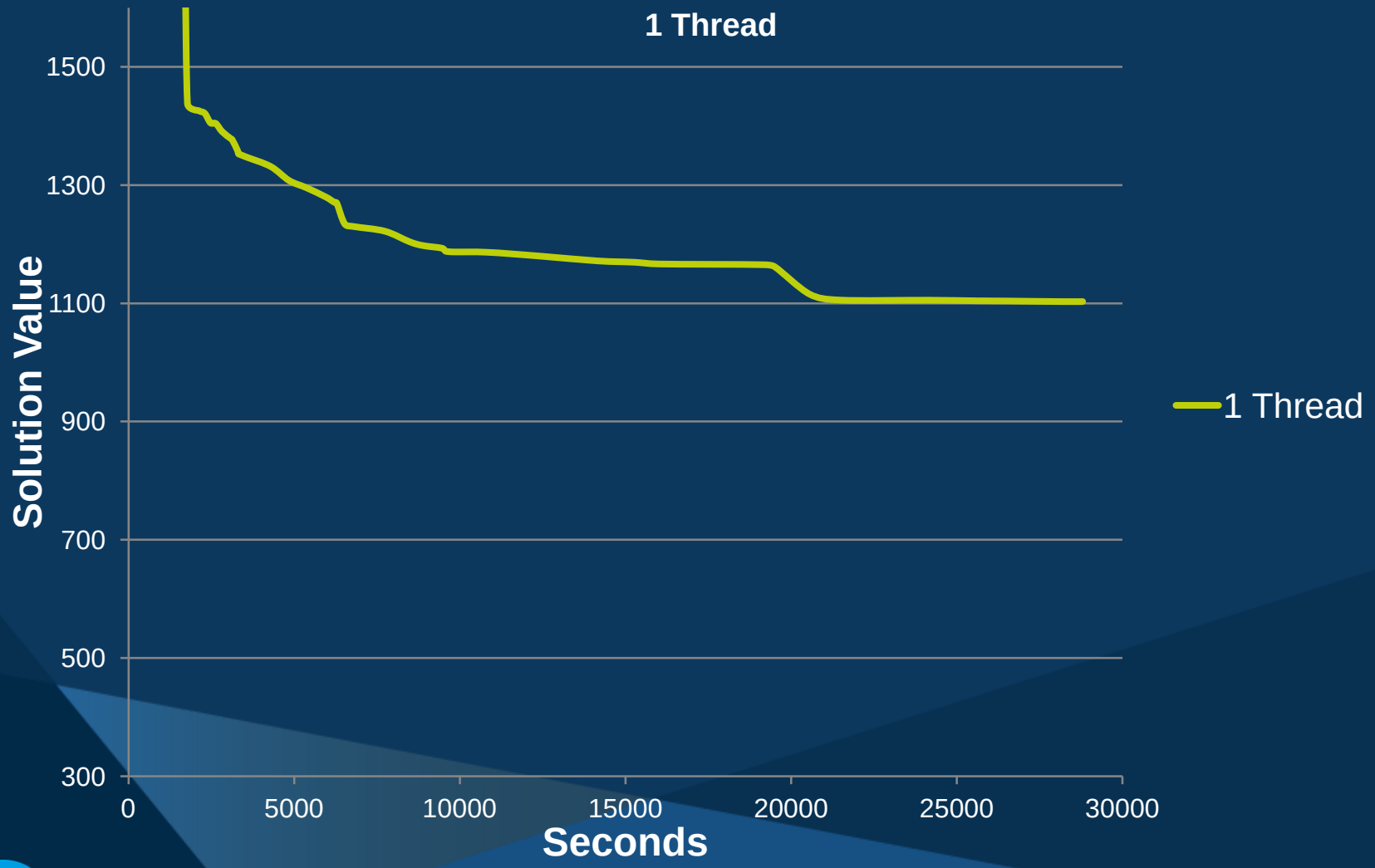
Model	entities	rows	cols	integers
Mixed	89177	553715	496455	153183
Phase1	283	421650	155336	154828

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%

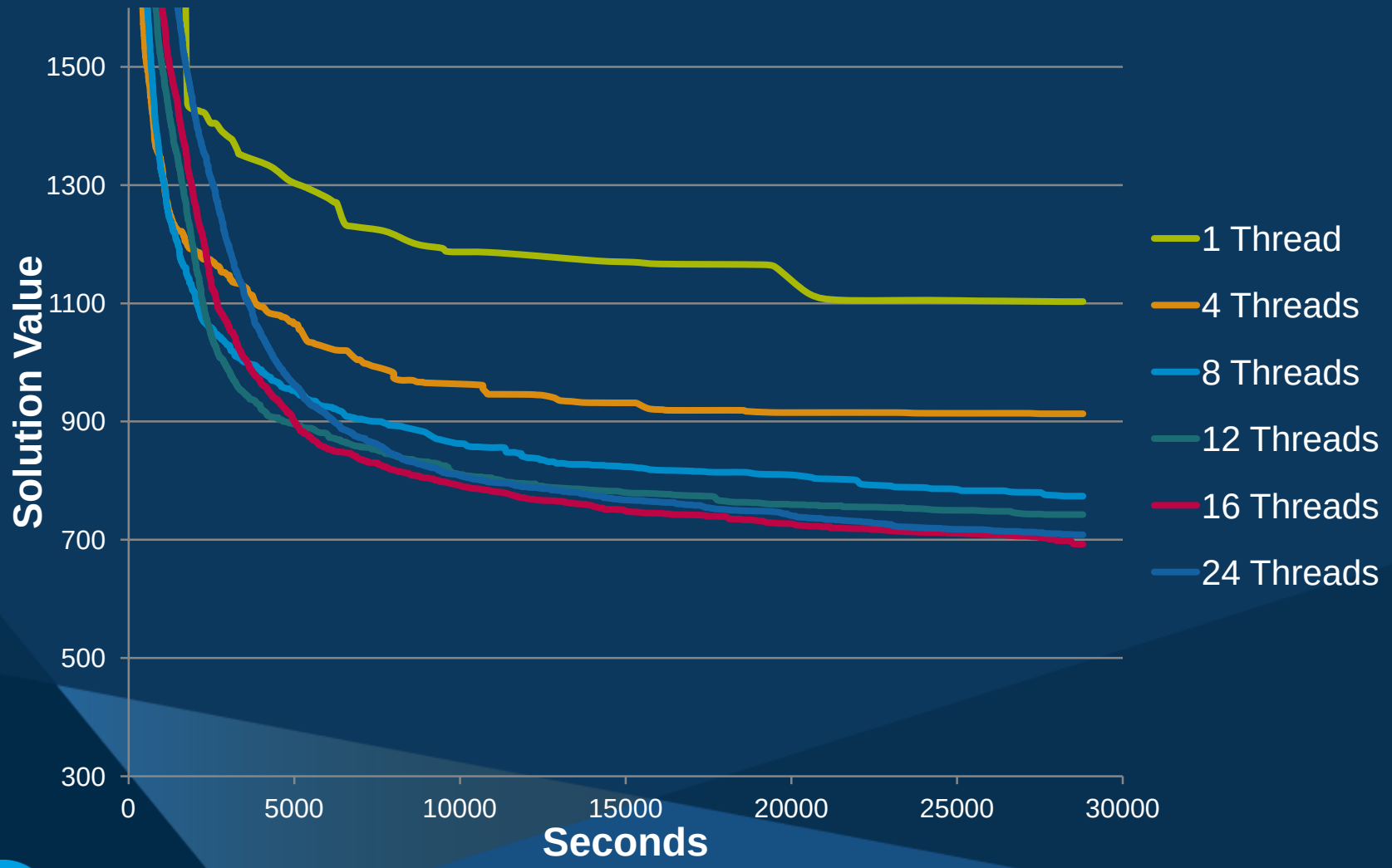
Difficult Model Heuristic Behavior



OPTIMIZATION DIRECT

Business Partner IBM

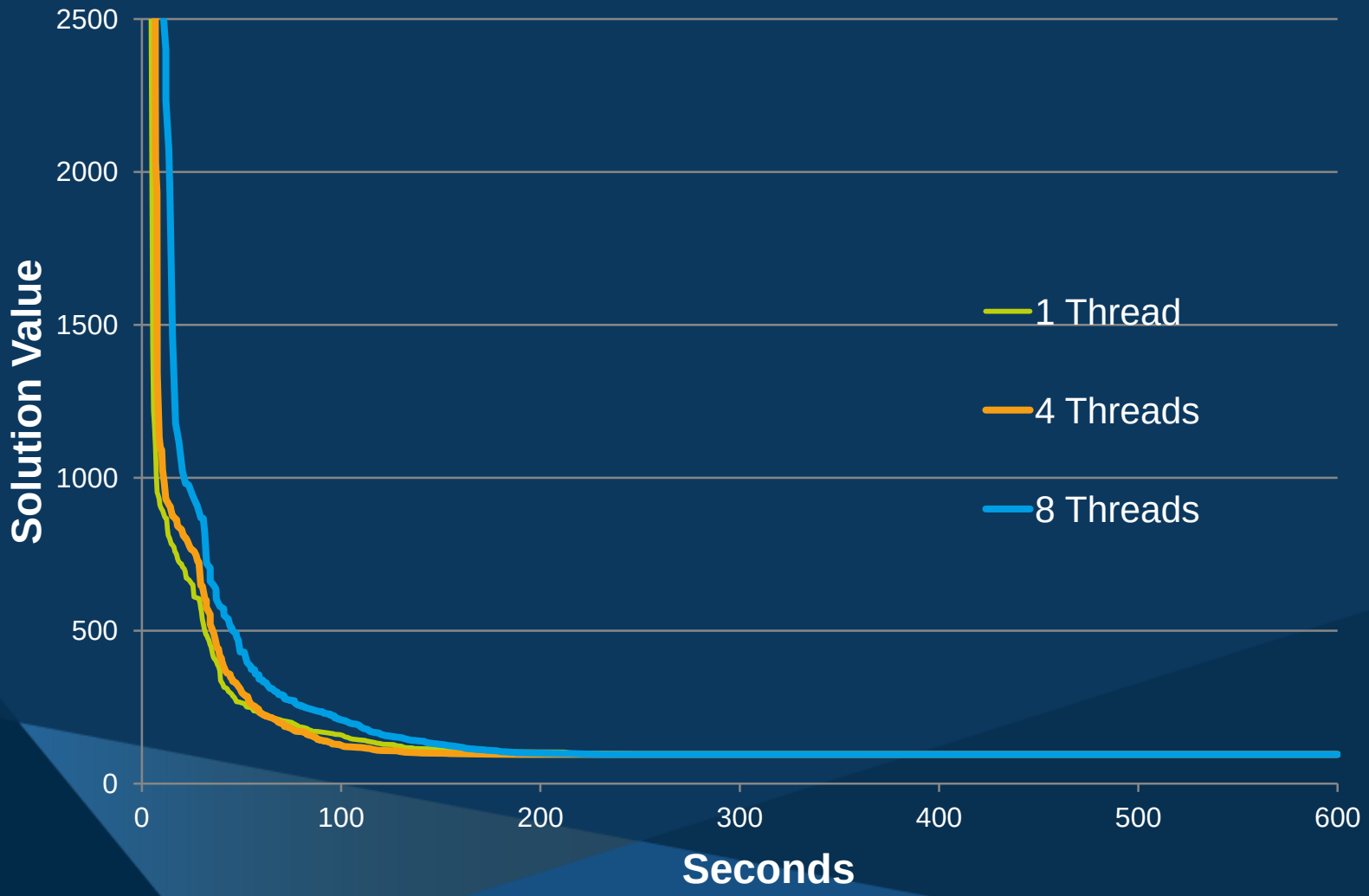
Difficult Model Heuristic Behavior



OPTIMIZATION DIRECT

Business Partner

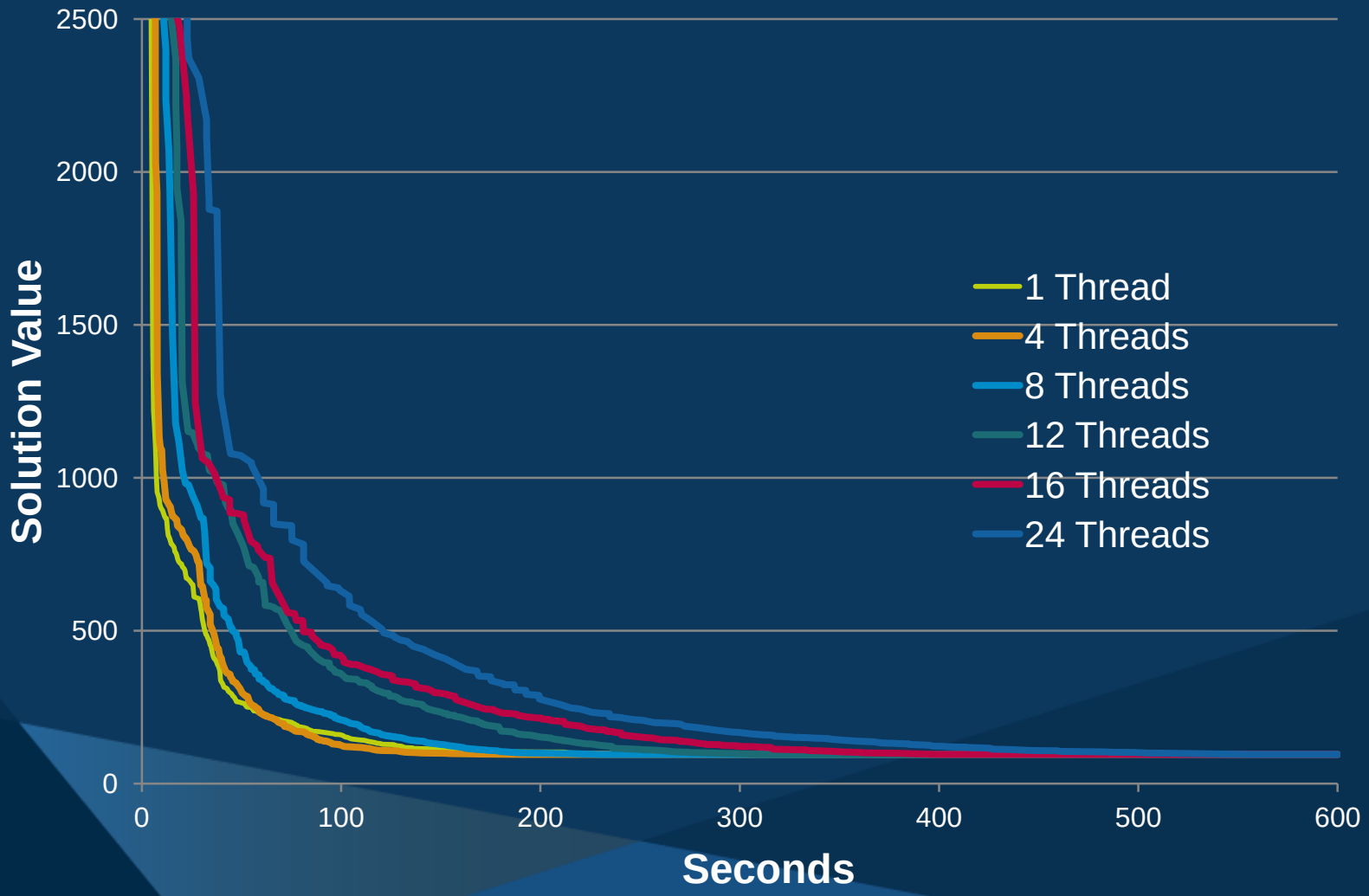
Easy Model Heuristic Behavior



OPTIMIZATION DIRECT

Business Partner

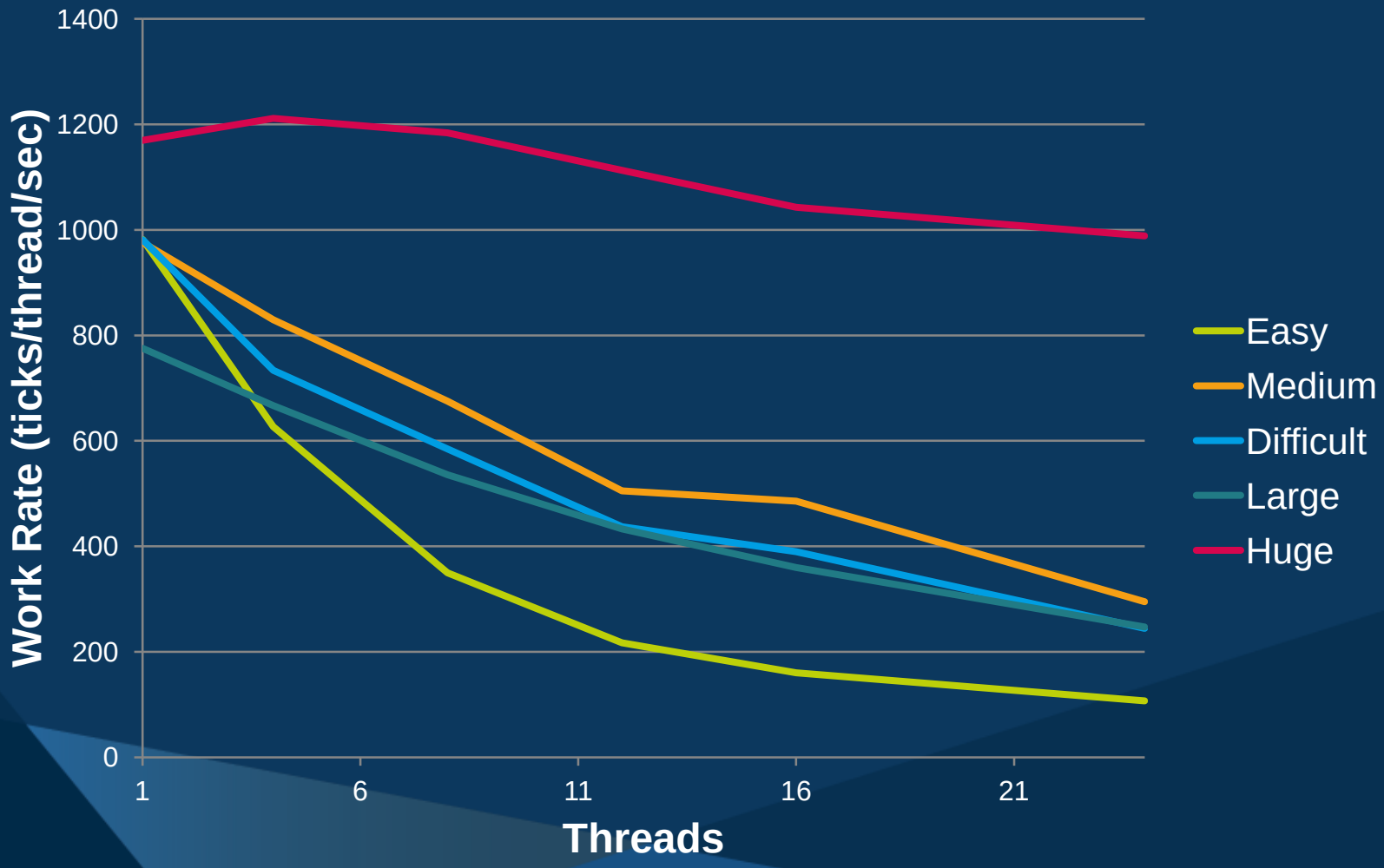
Easy Model Heuristic Behavior



OPTIMIZATION DIRECT

Business Partner

Work Rate



OPTIMIZATION DIRECT

Business Partner

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

OPTIMIZATION DIRECT

Business Partner

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)

OPTIMIZATION DIRECT

Business Partner IBM

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 that 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

- Usually very beneficial on hard user models
  - Especially scheduling
- Can be of benefit in general
  - Relatively better on harder-to-solve models
  - Relatively better with more cores (>4)
- Tested on customer models
- Tested on subsets of MIPLIB2010 used by Mittelmann

ODH|CPLEX Customer Models: Statistics

Model	Rows	Cols	Nonzeros	Binaries	Integers	SOSs
abaco	181170	306165	983149	5566	542	
allocam	1709	22888	63965	22822		
b19374	182588	104038	1642666	100601		
bp_case3	126128	269091	907017	9214		
case299	169807	142764	2439949	9415		
coal0	10977	11924	83999		2842	
cstscm2	186000	235715	758216	4947	609	
dpa	65417	126402	573546	59800		
exp01	1602710	1574843	7437368	1464408		
exp02	2628020	2617643	12652328	2507208		
fra2	3340550	160413	14930048	157488	1050	
fra21	31225	182113	430866	157488	1050	
fra22	31592	184104	435907	159112	1132	

ODH|CPLEX Customer Models: Statistics

Model	Rows	Cols	Nonzeros	Binaries	Integers	SOSs
fra23	31569	183357	434422	158365	1132	
fra2r	31230	182113	432486	157488	1050	
gcbp	10569	8311	74428	897	210	
genmod3	38978	41856	99826	16368		
harvest1	795	17419	418553	17171		
kat1	497290	1073230	3330837	8691		7642
kat2	679266	1538689	5076733	15628		13404
maletraits	35049	951	76080	890		
mbm	22100	30100	450980	30000		
mem_cg	35971	72578	192419	31972		
nuc	1282	3729	14699	3729		
nvsp01	249368	732735	3293146	730892	1634	
para1	1167600	1130706	4381976	14319		

OPTIMIZATION DIRECT

Business Partner IBM

ODH|CPLEX Customer Models: Statistics

Model	Rows	Cols	Nonzeros	Binaries	Integers	SOSs
pgom	76225	265681	954365	9184	1136	
prakcr	10127	31278	237321	30888		
producao	9410	7488	25263	6675		
radiumfoam	76349	49322	476729	32714		
rjune06	243857	362099	1191709	60814		
scd	349601	333122	1538512	1560		
sp3	813225	675862	2953834	25518	14466	
spp1	442204	444054	1640953	132731		
uni01	216288	303133	930668	6007	573	
utopia	970500	957951	3817642	470		
woods	315763	153037	2537381	82723		

OPTIMIZATION DIRECT

Business Partner

ODH|CPLEX Customer Models: Statistics

Summary

Set of 37 hard models chosen by customer

ODH|CPLEX not specifically designed or tuned for them

Model	Rows	Cols	Nonzeros	Binaries	Integers	SOSs
Max	3340550	2617643	14930048	2507208	14466	13404
Min	795	951	14699	470	210	7642
Average	402230	392889	2105946	173160	2110	10523

OPTIMIZATION DIRECT

Business Partner

ODH|CPLEX Customer Models: Results

	CPLEX 12.7.0		ODH|CPLEX 3.1.8	
Model	**Gap**	**Time**	**Gap**	**Time**
abaco	0.00%	4988	0.00%	3103
allocam	0.00%	6307	0.00%	8026
b19374	0.00%	452	0.05%	10800
bp_case3	0.56%	10800	0.60%	10800
case299	2.75%	10800	2.05%	10800
coal0	23.85%	10800	22.50%	10800
cstscm2	0.06%	10800	0.06%	10800
dpa	0.02%	10800	0.02%	10800
exp01	1.18%	10800	1.57%	10800
exp02	0.06%	10800	0.07%	10800
fra11	0.19%	10800	0.19%	10800
fra21	0.01%	10800	0.00%	344
fra22	0.07%	10800	0.08%	10800

OPTIMIZATION DIRECT

Business Partner

ODH|CPLEX Customer Models: Results

	CPLEX 12.7.0		ODH|CPLEX 3.1.8	
Model	**Gap**	**Time**	**Gap**	**Time**
fra23	1.20%	10800	1.70%	10800
fra2r	0.33%	10800	0.44%	10800
gcbp	68.64%	10800	29.62%	10800
genmod3	∞	10800	∞	10800
harvest1	69.59%	10800	46.46%	10800
kat1	51.52%	10800	22.44%	10800
kat2	89.52%	10800	21.14%	10800
maletraits	4.75%	10800	4.75%	10800
mbm	1.94%	10800	1.44%	10800
mem_cg	11.91%	10800	9.86%	10800
nuc	13.35%	10800	8.64%	10800
nvsp01	0.02%	10800	0.02%	10800
para1	2.03%	10800	2.16%	10800

ODH|CPLEX Customer Models: Results

	CPLEX 12.7.0		ODH|CPLEX 3.1.8	
Model	**Gap**	**Time**	**Gap**	**Time**
pgom	0.60%	10800	0.60%	10800
prakcr	0.00%	4439	0.00%	2327
producao	11.27%	10800	6.31%	10800
radiumfoam	0.02%	10800	0.03%	10800
rjune06	0.07%	10800	0.55%	10800
scd	9.81%	10800	7.16%	10800
sp3	10.05%	10800	3.18%	10800
spp1	0.21%	10800	0.23%	10800
uni01	0.00%	2543	0.00%	3198
uria1	0.21%	10800	0.25%	10800
utopia	0.84%	10800	0.72%	10800
woods	∞	10800	∞	10800

OPTIMIZATION DIRECT

Business Partner

ODH|CPLEX Customer Models: Results

Summary

Tests run by customer on 12 threads on 6 core Intel Xeon win64

3 hour time limit

	CPLEX 12.7.0		ODH|CPLEX 3.1.8	
	Gap	**Time**	**Gap**	**Time**
Average	10.46%	9026	5.41%	8809

OPTIMIZATION DIRECT

Business Partner IBM

ODH|CPLEX Developments Underway

- Support for CPLEX 12.8
  - Won’t harm CPLEX search strategies by increasing tick counts
- New automatic matrix decomposition method
  - Key to effective use of the technology if users do not want to specify decomposition
- Direct Python support
  - Python users currently use C API