

# ODH|CPLEX

## Recent Developments and Results

Alkis Vazacopoulos

Robert Ashford

Optimization Direct Inc.

April 2019

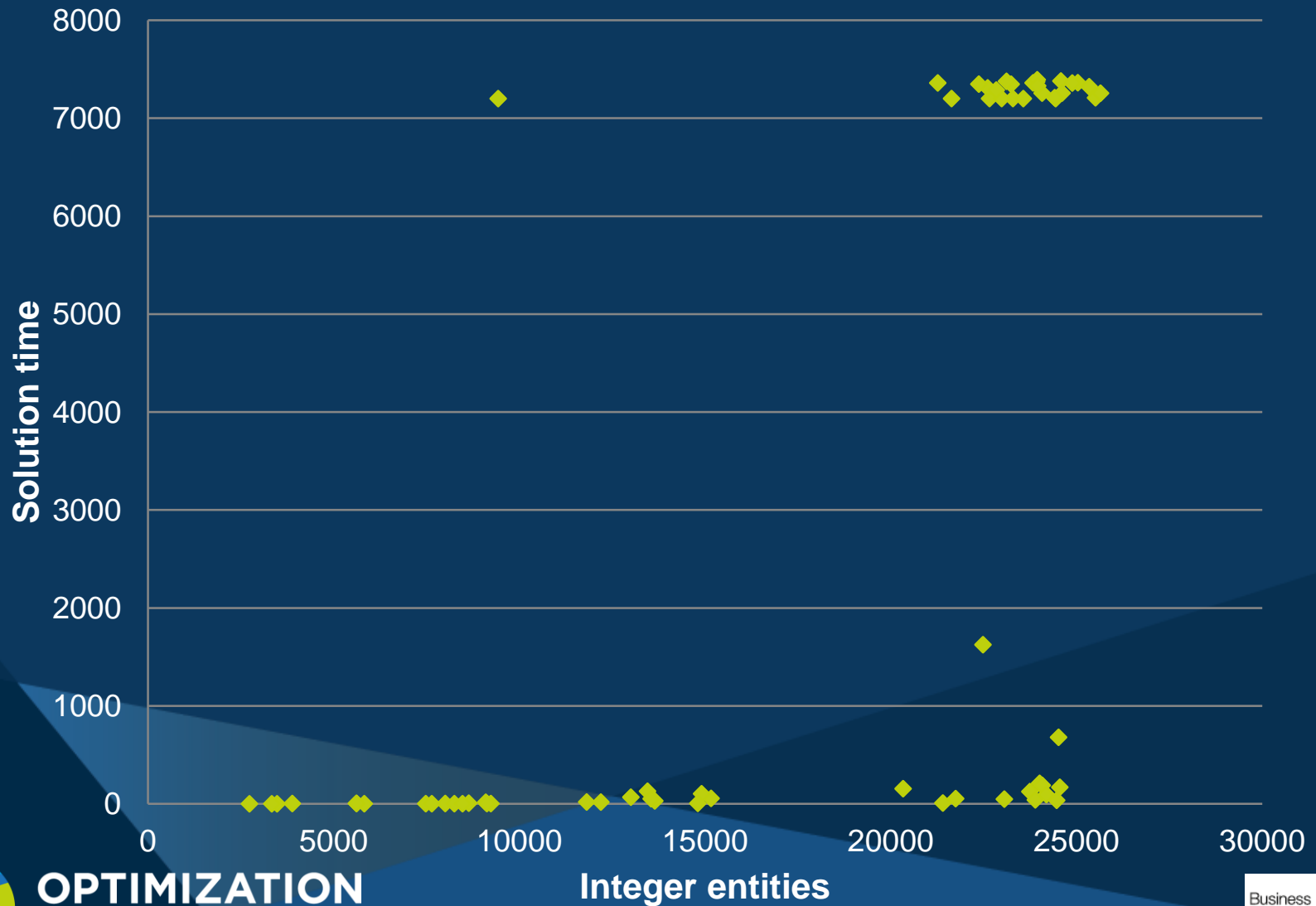
# Summary

- Challenges of Large Scale Optimization
- The ODHeuristics approach
  - ODHeuristics Engine
  - ODH|CPLEX Optimizer
- New features in release 4
- Scheduling, supply chain, telecomms and benchmark examples

# 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



# ODHeuristics: What Is It?

- Tools for
  - handling large and/or difficult MIPs
  - exploits parallel hardware
    - typical server/workstation 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

# What's New in Release 4.24

- Performance enhancements
  - **70% FASTER** than 4.04 – up to 40% gain from CPLEX 12.90
  - Improved synchronization
    - Now works well under SMT e.g. 8 threads on 4 core Intel i7
  - Smarter sub-model management
  - Reduced memory fragmentation
- CPLEX 12.90 support
- Improved SOS support
- MATLAB support
- Python support (2.7, 3.5, 3.6 and 3.7)
- Flexible user-specified decomposition via call-back



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

<b>Model</b>	<b>Application</b>	<b>entities</b>	<b>rows</b>	<b>cols</b>	<b>integers</b>
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
Huge	Telecomms	27000	2577916	12944400	12944400

# ODH|CPLEX Results

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

## ODH|CPLEX Optimizer 4.24

## CPLEX 12.90

6 (CPLEX) + 2 (ODH) thrds (det)

8 threads (deterministic)

	Solution	Time	Gap	Solution	Time	Gap
Easy	96	1 hr 20'	0%	96	59'	0%
Mixed	3156023	10'	1%	3168351	36'	1%
	3153056	8 hrs	0.03%	3153320	8 hrs	0.05%
Medium	127	8 hrs	39%	1715	8 hrs	96%
Difficult	1131	8 hrs	71%	8hrs		inf
Large	11464086	8hrs	1%	13639300	8hrs	17%
Huge	451	8hrs	69%	542	8hrs	74%

# ODH|CPLEX Results

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

## ODH|CPLEX Optimizer 4.24

## CPLEX 12.90

9 (CPLEX) + 3 (ODH) thrds (det)

12 threads (deterministic)

	Solution	Time	Gap	Solution	Time	Gap
Easy	96	3hr 2'	0%	96	3hr 37'	0%
Mixed	3175091	12'	1%	3154283	32'	1%
	3153015	8hrs	0%	3153054	8hrs	0%
Medium	121	8hrs	36%	2155	8hrs	96%
Difficult	1022.88579	8hrs	68%		8hrs	inf
Large	11458053	8hrs	0.7%	12399714	8hrs	8.4%
Huge	419	8hrs	67%	455	8hrs	69%

# More Customer Models: Statistics

Name	Source/Application	rows	cols	nonz	binaries	ints	SOSs
Rprs	Modelling System Vendor	126367	107788	431668	23652		
Ship4_14	Modelling System Vendor	732311	719937	3384573	664678		
Ship4_8	Modelling System Vendor	343629	336960	1538418	299656		
20160824	Modelling Consultant	138562	109964	552333	69000		
20160920	Modelling Consultant	108675	81488	415981	54000		
822_p1	Modelling Consultant	395435	543700	2235636	8695	7050	
822_p2	Modelling Consultant	395439	543700	2235640	8695	7050	
824_p1	Modelling Consultant	45840	70697	263418	10944	4119	
Apso374-2	Modelling Consultant	579250	509399	2457594	72202		
Big_p1	Modelling Consultant	866477	1176512	4788339	87721	1	
Dsj-12Jan_p1	Modelling Consultant	40474	44304	177842	9552	4807	
IG_20161019	Modelling Consultant	105704	79070	421342	52500		

# More Customer Models: Statistics

Name	Source/Application	rows	cols	nonz	binaries	ints	SOSs
JDI	Forestry Mngmt	1218099	3788507	22142180	693064		
JDI_Aug1	Forestry Mngmt	884565	3328399	15406793	73997		
Ilim_Model	Forestry Mngmt	6383910	8072885	37786872	5017849	8329	
pulp	Field Installation Mngmt	33464	118451	1579189	118450		
M297C	Healthcare	20496	1625	33244512		1620	
M305B	Healthcare	20496	1625	33244512		1620	
Shim	Healthcare	20496	1625	33244512		1620	
M297C_mat	Healthcare	20496	1625	33244512		1620	
ict1	Modelling System Vendor	497290	1073230	3330837	8691		7642
F880ba86	Wireless Ind Strat Mngmt	745732	30332	1518188	30332		
P_201x201	Electricity Generation	21221	10404	81720	9999	203	
P_203x203	Electricity Generation	86134	41616	331967	40803	407	

# More Customer Models: Results

8 Threads	ODH CPLEX 4.24		CPLEX12.90	
Model	Time	Gap	Time	Gap
Rprs	214	0.0%	607	0.0%
Ship4_14	7201	7.0%	7201	7.3%
Ship4_8	7201	4.6%	7201	5.3%
20160824	7200	37.2%	7200	77.7%
20160920	7200	11.4%	7200	19.7%
822_p1	115	0.0%	314	0.0%
822_p2	86	0.0%	178	0.0%
824_p1	2	0.0%	14	0.0%
Apso374-2	7201	0.5%	7200	0.7%
Big_p1	198	0.0%	215	0.0%
Dsj-12Jan_p1	7204	5.7%	7205	4.7%
IG_20161019	0	0.0%	0	0.0%

# More Customer Models: Results

8 Threads	ODH CPLEX 4.24		CPLEX12.80	
Model	Time	Gap	Time	Gap
JDI	7203	1.4%	7201	1.5%
JDI_Aug1	7203	15.5%	7203	inf
Ilim_Model	548	0.0%	228	0%
pulp	251	0.0%	525	0%
M297C	7204	14.6%	7203	37.5%
M305B	7202	18.8%	7202	36.6%
Shim	7203	15.7%	7203	36.1%
M297C_mat	7204	27.8%	7203	47.3%
ict1	3317	0.0%	3743	0.0%
f880ba86-b266	7200	6.6%	7200	30.9%
P_201x201	7201	0.1%	7200	0.1%
P_203x203	7200	0.3%	7200	7.1%

# More Customer Models: Results

24 Threads	ODH CPLEX 4.24		CPLEX12.90	
	Time	Gap	Time	Gap
Model				
Rprs	406	0.0%	449	0.0%
Ship4_14	7203	6.5%	7204	7.7%
Ship4_8	7202	4.8%	7202	5.4%
20160824	7202	14.2%	7200	93.2%
20160920	7201	8.0%	7201	14.2%
822_p1	76	0.0%	366	0.0%
822_p2	112	0.0%	220	0.0%
824_p1	2	0.0%	13	0.0%
Apso374-2	7203	0.4%	7244	0.8%
Big_p1	359	0.0%	177	0.0%
Dsj-12Jan_p1	7202	8.0%	7224	3.7%
IG_20161019	0	0.0%	0	0.0%



# More Customer Models: Results

24 Threads Model	ODH CPLEX 4.24		CPLEX12.90	
	Time	Gap	Time	Gap
JDI	7210	1.6%	7203	1.6%
JDI_Aug1	7209	8.8%	7207	inf
Ilim_Model	424	0.0%	358	0.0%
pulp	240	0.0%	745	0.0%
M297C	7210	14.7%	7207	46.8%
M305B	7213	20.6%	7209	48.7%
Shim	7209	21.6%	7210	49.2%
M297C_mat	7210	12.6%	7206	43.1%
ict1	3861	0.0%	3289	0.0%
f880ba86-b266	7201	6.6%	7200	19.3%
P_201x201	7216	0.1%	7203	0.1%
P_203x203	7202	0.2%	7201	6.9%

# More Customer Models: Results

	Avg 8 Threads		Avg 24 Threads	
	Time	Gap	Time	Gap
ODH CPLEX 4.04	4833	11.7%	4811	9.1%
CPLEX 12.80	4911	23.4%	4942	18.3%
ODH CPLEX 4.24	4798	6.9%	4830	5.3%
CPLEX 12.90	4842	16.8%	4837	17.9%
ODH CPLEX 4.04/4.24		<b>1.7</b>		<b>1.7</b>
CPLEX 12.80/12.90		<b>1.4</b>		<b>1.0</b>

# 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
  - Effective with 8 threads on 4 core Intel i-7 using SMT
  - 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](mailto:rwa@optimizationdirect.com)

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

