What’s New in Gurobi 8.0
New Gurobi team members in 2018...

Miranda Gould
Licensing Admin

KJ Gundersen
VP of Finance

Robert Luce
Developer

Duke Perrucci
Chief Revenue Officer

Mark Reynaud
US Central Sales

Pano Santos
Sr. Technical Content Manager

Isela Warner
Director of Marketing

Yuriy Zinchenko
Support Engineer
What's New in 8.0?

• New features
  • Significant enhancements to Compute Server and Instant Cloud
  • Enhanced MATLAB and R interfaces
  • Support for MATLAB problem-based modeling
  • Partition heuristic
  • Multiple MIP starts
    • Support for .NET Core 2.0

• Performance improvements
Enhanced MATLAB and R interfaces
Completing the MATLAB and R Interfaces

• Previously missing features:
  • Multi-objective optimization
  • Solution pools
  • General constraints
  • Several other (smaller) features

• Now available in MATLAB and R

• Also now available:
  • Full set of Gurobi examples for MATLAB and R
MATLAB Problem-Based Modeling

• Problem-based modeling: new feature in MATLAB 2017b Optimization Toolbox
  • A more structured approach to building an optimization model

```matlab
x = optimvar('x', 'Type','integer','LowerBound',0,'UpperBound',1);
y = optimvar('y', 'Type','integer','LowerBound',0,'UpperBound',1);
z = optimvar('z', 'Type','integer','LowerBound',0,'UpperBound',1);
prob = optimproblem('ObjectiveSense','maximize');
prob.Objective = x + y + 2 * z;
prob.Constraints.cons1 = x + 2 * y + 3 * z <= 4;
prob.Constraints.cons2 = x + y >= 1;
sol = solve(prob, options)
```

• You can now call Gurobi through the MATLAB `solve()` function
Partition Heuristic
User-Specified Local Improvement Heuristic

• RINS is our most effective heuristic
• It is a sub-MIP heuristic:
  • Fix a subset of the variables to incumbent values
  • Solve the resulting MIP (recursively)
    • Reoptimizes over just that portion of the problem
• Sub-MIP heuristics extremely effective in general
• How to choose the sub-problem to reoptimize?
  • RINS chooses automatically
  • New feature allows user to make the choice
    • Example sub-problems:
      • All decisions related to a single time period
      • All decisions related to a single machine
      • All decisions related to physical sub-regions (e.g., Western US, Eastern US, etc.)
Multiple MIP Starts
Multiple MIP Starts

- User can now provide multiple MIP starts
- Useful when you have multiple partial solutions
  - MIP solver will try to complete them, and will store the ones it finds
- For distributed MIP, MIP starts will be evaluated on different machines
.NET Core 2.0
.NET Core 2.0

- Gurobi users can now write .NET programs that run on Windows, Linux, and Mac
  - Using new Microsoft .NET Core 2.0
Compute Server and Instant Cloud
• **Standard Communication Protocol** – Support for HTTP and HTTPS has been added for improved security, ease of deployment, and robustness.

• **Clustering** – New clustering that allows nodes to be added to and removed from clusters dynamically.
Compute Server Monitoring and Management

- **Commands** – A new command line tool can now monitor and manage your cluster.
  - Users can display the status of the cluster, list running and recently processed jobs, and access job logs and parameters.
  - An administrator can list the licenses, abort jobs, and manage the nodes of the cluster.
- **REST API** – A new cluster REST API allows you to programmatically monitor and manage your cluster.

```
$ grbcluster --server=server1 nodes
ADDRESS STATUS TYPE LICENSE #Q #R JL IDLE %MEM %CPU
server1 ALIVE COMPUTE VALID 0 1 2 0m 66.02 99.04
server2 ALIVE COMPUTE VALID 0 0 2 1m 50.19 2.42

$ grbcluster --server=server1 recent
JOBID  ADDRESS STATUS STIME USER OPT API
8a8cd65e server1 COMPLETED 2018-04-09 10:44:54 user1 OPTIMAL Python
ecfabe5eb server2 COMPLETED 2018-03-29 12:05:25 user1 INFEASIBLE Java
fe60d00b server1 IDLETIMEOUT 2018-03-29 12:01:35 user1 OPTIMAL Python
977eb86d server2 DISCONNECTED 2018-03-28 11:51:49 user1
```
• **HTTPS** – Support for HTTPS has been added for better security, robustness, and easier integration.
Improved Machine and Pool Management

- **Pool Scaling** – New pool scaling allows the user to dynamically add or remove machines to and from pools.
- **Machine Metrics** – New machine metrics display CPU and memory usage on each machine.
New Job Monitoring

- **Job List** – A new job list now displays the jobs queued, running, and recently processed.
- **Job Dashboard** – A new dashboard displays detailed job information, such as status, log, parameters, and metrics, in one place.
- **Job History** – A new archive functionality allows users to store job logs and statuses from the past three months.
Gurobi 8.0 Performance Improvements
Two Kinds of Benchmarks

- Internal benchmarks
  - Most important: compare Gurobi version-over-version
  - Based on internal library of 4538 models

- External, competitive benchmarks
  - Conducted by Hans Mittelmann, Arizona State University
    - [http://plato.asu.edu/bench.html](http://plato.asu.edu/bench.html)
  - For MIP largely based upon MIPLIB 2010
Internal Benchmarks
### Gurobi MIP Library

The Gurobi MIP Library contains a total of 4538 models, ranging from small to very large in size. The graph illustrates the relationship between the number of rows (M) and columns (N) for these models.

**Graph Details:**
- **X-axis:** Number of rows (M) ranging from 1 to 10,000,000.
- **Y-axis:** Number of columns (N) ranging from 1 to 1E+09.

**Key Observations:**
- Approximately 1% of the models have between 100,000 and 1,000,000 rows.
- About 0.5% of the models have between 10,000 and 100,000 columns.

**Legend:**
- Red lines indicate specific ranges: 1,000,000 rows and 100,000 columns.

---

**Copyright © 2018, Gurobi Optimization, LLC**
## Performance Improvements: 7.5 to 8.0

<table>
<thead>
<tr>
<th>Problem Class</th>
<th>&gt;1s</th>
<th></th>
<th></th>
<th>&gt;100s</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>Wins</td>
<td>Losses</td>
<td>Speedup</td>
<td>#</td>
<td>Wins</td>
</tr>
<tr>
<td>LP: concur.</td>
<td>482</td>
<td>66</td>
<td>56</td>
<td>1.01x</td>
<td>151</td>
<td>45</td>
</tr>
<tr>
<td>primal</td>
<td>487</td>
<td>79</td>
<td>62</td>
<td>1.07x</td>
<td>209</td>
<td>54</td>
</tr>
<tr>
<td>dual</td>
<td>466</td>
<td>110</td>
<td>61</td>
<td>1.12x</td>
<td>171</td>
<td>77</td>
</tr>
<tr>
<td>barrier</td>
<td>490</td>
<td>58</td>
<td>43</td>
<td>1.05x</td>
<td>165</td>
<td>37</td>
</tr>
<tr>
<td>MIP</td>
<td>2263</td>
<td>1068</td>
<td>634</td>
<td>1.16x</td>
<td>920</td>
<td>484</td>
</tr>
<tr>
<td>MIQP</td>
<td>177</td>
<td>71</td>
<td>46</td>
<td>1.15x</td>
<td>77</td>
<td>40</td>
</tr>
<tr>
<td>MIQCP</td>
<td>261</td>
<td>136</td>
<td>73</td>
<td>1.22x</td>
<td>68</td>
<td>38</td>
</tr>
</tbody>
</table>

- Gurobi 7.5 vs. 8.0: > 1.00x means that Gurobi 8.0 is faster than Gurobi 7.5
### Performance Improvements: 7.0 to 8.0

<table>
<thead>
<tr>
<th>Problem Class</th>
<th>&gt;1s</th>
<th></th>
<th></th>
<th></th>
<th>&gt;100s</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>Wins</td>
<td>Losses</td>
<td>Speedup</td>
<td>#</td>
<td>Wins</td>
<td>Losses</td>
<td>Speedup</td>
</tr>
<tr>
<td>LP: concur.</td>
<td>449</td>
<td>135</td>
<td>134</td>
<td>1.15x</td>
<td>146</td>
<td>72</td>
<td>44</td>
<td>1.46x</td>
</tr>
<tr>
<td>primal</td>
<td>459</td>
<td>138</td>
<td>133</td>
<td>1.24x</td>
<td>208</td>
<td>87</td>
<td>48</td>
<td>1.49x</td>
</tr>
<tr>
<td>dual</td>
<td>441</td>
<td>165</td>
<td>119</td>
<td>1.32x</td>
<td>171</td>
<td>96</td>
<td>43</td>
<td>1.82x</td>
</tr>
<tr>
<td>barrier</td>
<td>457</td>
<td>126</td>
<td>136</td>
<td>1.13x</td>
<td>157</td>
<td>68</td>
<td>50</td>
<td>1.44x</td>
</tr>
<tr>
<td>MIP</td>
<td>2276</td>
<td>1351</td>
<td>561</td>
<td>1.52x</td>
<td>980</td>
<td>669</td>
<td>225</td>
<td>2.10x</td>
</tr>
<tr>
<td>MIQP</td>
<td>121</td>
<td>84</td>
<td>24</td>
<td>3.76x</td>
<td>67</td>
<td>52</td>
<td>12</td>
<td>9.24x</td>
</tr>
<tr>
<td>MIQCP</td>
<td>244</td>
<td>105</td>
<td>89</td>
<td>1.20x</td>
<td>68</td>
<td>38</td>
<td>26</td>
<td>1.83x</td>
</tr>
</tbody>
</table>

- Gurobi 7.0 vs. 8.0: > 1.00x means that Gurobi 8.0 is faster than Gurobi 7.0
Continual Performance Improvements

Nearly a 2x average improvement per major release

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.40GHz
4 cores, 8 hyper-threads
32 GB RAM

Gurobi Version-to-Version Pairs

MIP test set has 3740 models:
- 229 discarded due to inconsistent answers
- 908 discarded that none of the versions can solve
- Speed-up measured on >100s bracket: 1227 models

Copyright © 2018, Gurobi Optimization, LLC