A web-based solution platform for Multi-objective Integer Programs

Banu Lokman, Gökhan Ceyhan, Murat Köksalan

*Middle East Technical University, Department of Industrial Engineering, Ankara*

supported by
The Scientific and Technological Research Council of Turkey
(Project No: 215M844)

July 2017
OUTLINE

- Introduction
- The Algorithms
- A web-based solution platform for Multi-objective Integer Programs – under construction
- Conclusions and Future Work
"Max" \[ \left\{ z_1(x), z_2(x), ..., z_q(x) \right\} \]
subject to
\[ x \in X \]
where
\[ z_i : \text{the } i^{th} \text{ objective function} \]
\[ x \in \mathbb{Z}^l : \text{decision vector} \]
\[ X : \text{solution (decision) space} \]
\[ q : \text{the number of objectives} \]
A feasible solution $x \in X$ is called **efficient**, if there is **no other** $x' \in X$ such that

\[
\begin{align*}
  z_i(x) \leq z_i(x') & \quad \text{for all } i = 1, \ldots, q \\
  z_i(x) < z_i(x') & \quad \text{for at least one } i.
\end{align*}
\]

If $x$ is efficient, $z(x)$ is called **nondominated** point.

$x_2 \in X$ is said to **dominate** $x_1 \in X$ and $z(x_2)$ is said to dominate $z(x_1)$ if

\[
\begin{align*}
  z_i(x_1) \leq z_i(x_2) & \quad \text{for all } i = 1, \ldots, q \\
  z_i(x_1) < z_i(x_2) & \quad \text{for at least one } i.
\end{align*}
\]
Multi-objective Integer Programs (MOIPs) have many applications in real life:

- Budgeting Problems
- Network Design Problems
- Routing Problems
- Location and Hub-Location Problems
- Scheduling Problems

Finding nondominated points is typically hard.


X is discrete and “large”.

Grows fast with problem size
Studies to generate all nondominated points

- Tenfelde-Podehl (2003), Dhaenens et al. (2010) and Przybylski et al. (2010), propose methods based on two-phase method.
- Recently, more efficient algorithms developed by:

where the properties of the epsilon-constraint method are used, and the search region is decomposed.
Sylva and Crema (2004)

\[(P_{\lambda(n)})\]

\[
\text{Max } \sum_{i=1}^{q} \lambda_i z_i(x)
\]

subject to

\[
z_i(x) \geq (z_i^k(x) + 1) t_{ik} - M(1 - t_{ik}) \quad \forall i \quad \forall k
\]

\[
\sum_{i=1}^{q} t_{ik} \geq 1 \quad \forall k
\]

\[
t_{ik} \in \{0,1\} \quad k = 1,...,n \quad i = 1,2,...,q
\]

\[
x \in X
\]

n(q+1) additional constraints and nq binary variables to guarantee a new nondominated point different from any of the existing ones.
Lokman and Köksalan (2013)

\((P_{\lambda(n)})\)

\[\begin{align*}
\text{Max } & \ z_i(x) + \sum_{\substack{i=1 \atop i \neq m}}^{q} \rho_i z_i(x) \\
\text{subject to} & \\
\quad & z_i(x) \geq \left( z_i^k(x) + 1 \right) t_{ik} - M(1 - t_{ik}) \quad \forall i \neq m \quad \forall k \\
\quad & \sum_{\substack{i=1 \atop i \neq m}}^{q} t_{ik} \geq 1 \quad \forall k \\
\quad & t_{ik} \in \{0, 1\} \quad k = 1, \ldots, n \quad i = 1, 2, \ldots, q \quad i \neq m \\
\quad & x \in X
\end{align*}\]

\(n(q)\) additional constraints and \(n(q-1)\) binary variables to guarantee a new nondominated point different from any of the existing ones.
Lokman and Köksalan (2013)

- Partitions the feasible space into subspaces.
- Employs a search procedure.
- Solves a number of models by imposing bounds on the objectives rather than adding additional constraints or binary variables.
A web-based solution platform

A web-based solution platform to generate

- Nondominated points (all or regional)
- Representative nondominated set with desired quality level

nMOCO-S
All Nondominated points by Multi-Objective Combinatorial Optimization Solver
Generate efficient solutions of your multi-objective combinatorial problem. Upload your file, compute on the cloud, download results.

Try it!

Murat Köksalan
Banu Lokman
Gökhan Ceyhan
A web-based solution platform

www.onlinemoco.com/MOIP/

- In development phase.
- Applications (under construction):
  - **nMOCO-S**: Finds all nondominated points.
  - **rMOCO-S**: Finds a representative set of nondominated points
  - **iMOCO-S**: Integrates user preferences
  - **libMOCO-S**: Collects a variety of MOCO instances.
A web-based solution platform - nMOCO-S

- Generates all nondominated points for MOIPs.
- Based on Lokman and Köksalan (2013).
- **The idea:** The search region is partitioned and reduced progressively by removing the regions that are dominated by previously found nondominated points.
A web-based solution platform - nMOCO-S

The Approach - MOIP TREE

- **N-ary tree** structure
- **Tree height** = number of objectives - 2 = q-2
- **Number of child nodes of a node** = as many as the number of already generated non-dominated points = n
A web-based solution platform - nMOCO-S

The Approach - Demo: Tree creation and update

Let's create the initial tree with the following three nondominated points in the four dimensional criterion space.

<table>
<thead>
<tr>
<th>Points</th>
<th>z₁</th>
<th>z₂</th>
<th>z₃</th>
<th>z₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>402</td>
<td>469</td>
<td>521</td>
<td>324</td>
</tr>
<tr>
<td>2</td>
<td>420</td>
<td>393</td>
<td>508</td>
<td>452</td>
</tr>
<tr>
<td>3</td>
<td>318</td>
<td>477</td>
<td>487</td>
<td>521</td>
</tr>
</tbody>
</table>
At leaf nodes: compute the bounds at criterion \((q-1)\) and optimize \(q^{th}\) obj. function.
A web-based solution platform - nMOCO-S

The Approach - Demo: Tree creation and update

<table>
<thead>
<tr>
<th>Points</th>
<th>$z_1$</th>
<th>$z_2$</th>
<th>$z_3$</th>
<th>$z_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>402</td>
<td>469</td>
<td>521</td>
<td>324</td>
</tr>
<tr>
<td>2</td>
<td>420</td>
<td>393</td>
<td>508</td>
<td>452</td>
</tr>
<tr>
<td>3</td>
<td>318</td>
<td>477</td>
<td>487</td>
<td>521</td>
</tr>
</tbody>
</table>
A web-based solution platform - nMOCO-S

The Approach - Demo: Search

Left node space covers right one.
Use this info not to search the space of the right node.

new nondominated point

k=4
(460, 438, 486, 218)
A web-based solution platform - nMOCO-S

The Approach - Demo: Tree update
A web-based solution platform - rMOCO-S

200-item MOKP with 27,260 nondominated points
A web-based solution platform - rMOCO-S

rMOCO-S

Representative nondominated points by Multi-objective Combinatorial Optimization Solver

Represent the objective space of your problem with a few points in an efficient manner.

Learn more.
A web-based solution platform - rMOCO-S
A web-based solution platform - rMOCOS

- Representative nondominated sets are found for MOIPs.
  - SBA is designed to continue generating new points until the desired coverage gap value is satisfied.
  - TDA guarantees to achieve a desired coverage gap value stated by the DM at the outset.
  - SPA is designed to consider the whole solution space and allocate the representative points accordingly if the number of representative points is known in advance.
- A desired level of quality is guaranteed in representing the nondominated frontier.
- The methods are computationally efficient based on extensive computational tests.
- New approaches based on the distribution properties.
A web-based solution platform - rMOCOS

Sylva and Crema (2007)

Max \( \alpha + \epsilon(z_1(x) + z_2(x)) \)

s.t.

\[
\begin{align*}
z_1(x) &\geq z_1^1(x)y_1^1 + 1 + \alpha - M(1 - y_1^1) \\
z_2(x) &\geq z_2^1(xx)y_2^1 + 1 + \alpha - M(1 - y_2^1) \\
y_1^1 + y_2^1 & = 1 \\
z_1(x) &\geq z_1^2(x)y_1^2 + 1 + \alpha - M(1 - y_1^2) \\
z_2(x) &\geq z_2^2(x)y_2^2 + 1 + \alpha - M(1 - y_2^2) \\
y_1^2 + y_2^2 & = 1 \\
x &\in X \\
y_i^j &\in \{0, 1\} \quad i = 1, 2, \quad j = 1, 2
\end{align*}
\]
Decomposition to nondominated subspaces:

\[ S^j = (lb_1, lb_2) \]

Max \[ \alpha^j + \epsilon(z_1(x) + z_2(x)) \]

s.t.

\[ z_1(x) \geq lb_1 + \alpha^j \]
\[ z_2(x) \geq lb_2 + \alpha^j \]
\[ x \in X \]

\[ \alpha^* = \max_{j=1,2,3} \{ \alpha^j \} \]
Territory Defining Algorithm (TDA)

- SBA reduces the computational effort compared to SC substantially.
- SBA controls the coverage gap value and stopped when the DM is satisfied.
- If a desired coverage gap value, $\Delta$, is given at the outset, TDA
  - avoid generating points close to other points,
  - utilize the desired coverage gap information actively throughout the algorithm
  - constructs territories around the previously generated points that are inadmissible for the new point
  - keep searching different subspaces until finding a nondominated point.
A web-based solution platform - rMOCOS

Surface Projection Algorithm (SPA)

Approximate the nondominated frontier fitting Lp surface using the methodology of Koksalan and Lokman (2009).
Select a representative set of hypothetical points on the fitted surface.
(first discretize the surface and find the optimal subset.) For each hypothetical point, find the representative nondominated point at minimum Tchebycheff distance.

(c) Optimal subset of hypothetical points, $\alpha = 0.16$.

(d) The representative subset $R$, $\alpha_R = 0.18$. 

A web-based solution platform - rMOCOS

Surface Projection Algorithm (SPA)
We implement these algorithms as an online tool that allows the users to generate:

- a representative set satisfying a desired level of accuracy
- all nondominated points.

The tool provides the output in terms of the objective function values of the generated nondominated points as well as in terms of the decision variables corresponding to the desired nondominated points.

We also maintain a digital library that contains a collection of MOIPs and make their inputs and outputs available to researchers.
A web-based solution platform

Use Case:
A web-based solution platform - Demo

nMOCO-S: A MOKP with 3 objectives, 3 knapsacks, 25 items

Step 1. Create a text file following the format given in the guide.

For any problem for which an input data format is not available, user can input “.lp” file of the single objective problem.
A web-based solution platform - Demo

Step 2. Upload the problem.

Applications → nMOCO-S
Step 3. Query the status of the problem with the user mail and given jobID.

<table>
<thead>
<tr>
<th>Job Id</th>
<th>Issuer</th>
<th>Job Creation Time</th>
<th>Job Status</th>
<th>Job Completion Time</th>
<th>Processing Time (secs)</th>
<th>Job Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td><a href="mailto:test-user@onlinemoco.com">test-user@onlinemoco.com</a></td>
<td>2017-07-11 21:20:07.116</td>
<td>FINISHED_SUCCESS</td>
<td>2017-07-11 21:21:05.276</td>
<td>11.0</td>
<td>result.txt</td>
</tr>
</tbody>
</table>

- # of nondominated points
- # of models solved
- elapsed time

nondominated points

```plaintext
182
410
12.783
636 646 959
653 558 941
645 570 936
640 583 933
...```
A web-based solution platform – Visualization Tools

Parzen Windows* (Ozarık, Koksalan and Lokman)

MOKP with 5652 nondominated points

A web-based solution platform – Visualization Tools

Parzen Windows* (Ozarık, Koksalan and Lokman)

MOAP with 6573 nondominated points

MOKP with 6500 nondominated points
A web-based decision support system for MOIPs generating:

- Nondominated set (all or regional)
- Representative sets based on a variety of algorithms.

Available to academic researchers.

A digital library.
FUTURE WORK

- New algorithms that consider
  - The distribution and density
  - Shape of the frontier
  - Preferences of the DM.
  - New quality measures.
Thank you...

Questions & Comments
References


EXACT ALGORITHMS  *(Lokman & Köksalan, JOGO’13)*

Comparison of Algorithms on MOKP for \( q = 3 \)

<table>
<thead>
<tr>
<th>Number of items</th>
<th>Problem</th>
<th>Number of nondominated points (N)</th>
<th>Solution time (CPU time in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sylva and Crema</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>35</td>
<td>14.24</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>43</td>
<td>38.47</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>61</td>
<td>102.40</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>67</td>
<td>121.82</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>77</td>
<td>259.51</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>57</td>
<td>118.13</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>76</td>
<td>314.61</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>103</td>
<td>818.26</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>108</td>
<td>2,043.33</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>132</td>
<td>5,291.38</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>157</td>
<td>5,285.43</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>163</td>
<td>5,253.49</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>168</td>
<td>12,406.04</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>182</td>
<td>14,740.24</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>470</td>
<td>Could not be solved in 15h</td>
</tr>
</tbody>
</table>
**EXACT ALGORITHMS** *(Lokman & Köksalan, JOGO’18)*

Comparison of Algorithms on MOKP for $q = 3$

<table>
<thead>
<tr>
<th>Number of items</th>
<th>Problem</th>
<th>Number of nondominated points (N)</th>
<th>Solution time (CPU time in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 1</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>280</td>
<td>4,823.95</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>356</td>
<td>5,173.84</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>519</td>
<td>12,082.91</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>784</td>
<td>33,699.41</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>912</td>
<td>35,557.58</td>
</tr>
</tbody>
</table>
Performance of Algorithm 2 (q=3)

- For q=3, the number of models solved, MS, to find all N nondominated points will be in the interval:
  \[ N + 1 \leq MS \leq (N + 1)(N + 2)/2. \]

- We observe that \( MS/N \) is in the interval \([1.80, 2.36]\) with an average of 2.13.

- That is, we roughly solve only 2 models for each nondominated point on average.

- This indicates the importance of the information obtained from the archives of Algorithm 2.

- \( MS \ll (N + 1)(N + 2)/2 \) especially for large N values.
Performance of Algorithm 2 (q=4)

- MS/N is again not sensitive to the value of N where the ratio is within the interval [5.67, 10.14] with an average value of 8.53.
- The value of MS/N increases with the number of objectives.

\[ N \leq MS \leq \sum_{n=0}^{N} \frac{(n+1)(n+2)}{2} \]

- In our experiments: \( MS \approx 9N \) \( (MS << \sum_{n=0}^{N} \frac{(n+1)(n+2)}{2}) \)
Performance of Algorithm 2

- Still may not be practical for large problems.
- Number of nondominated points may be prohibitive.
- Can be used to test performances of heuristics.

e.g. 51.28 hours to solve a MOKP with 200 items and 3 objectives (27,260 nondominated points)
The algorithm of Sylva and Crema (SC)

Max $F = \alpha + \epsilon \sum_{i=1}^{m} w_i z_i$

s.t.

$z_i \geq y_i^j \gamma_{ji} + \alpha - (M_i + U)(1 - \gamma_{ji}) \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$

$\sum_{i=1}^{m} \gamma_{ji} = 1 \quad j = 1, \ldots, n$

$\alpha \geq 0$

$\gamma_{ji} \in \{0, 1\} \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$

$z = z(x)$

$x \in X$
A Subspace-based Approach (SBA)

Solution time comparison of SBA and SC on 3-objective, 50-item knapsack problems (in secs)*

| $|R|$ | SBA       | SC       |
|-----|-----------|-----------|
| 5   | 1.40      | 1.01      | 0.47      | 0.03      |
| 50  | 36.83     | 10.51     | 81.43     | 27.10     |
| 100 | 86.37     | 34.77     | 3863.25   | 1989.54   |
| 120 | **103.54**| **44.75** | **15291.65**| **8096.38**|

*Based on 10 problems per cell and there are an average of 417 nondominated points in total.
A Subspace-based Approach (SBA)

| Problem   | $|R|$ | Coverage gap | Sol. time (sec) | Models solved |
|-----------|-----|--------------|-----------------|---------------|
| KP-25     | 5   | 0.20 | 0.04    | 0.64 | 0.17 | 16.60 | 1.84 |
| ($\bar{N} = 56$) | 25  | 0.05 | 0.02    | 7.4  | 1.78 | 164.88 | 29.51 |
|           | 50  | 0.02 | 0.01    | 15.87| 1.31 | 356.50 | 32.50 |
| KP-50     | 5   | 0.22 | 0.03    | 1.4  | 1.01 | 18.40 | 1.07 |
| ($\bar{N} = 417$) | 25  | 0.08 | 0.02    | 12.74| 2.56 | 224.10 | 34.00 |
|           | 50  | 0.04 | 0.01    | 36.83| 10.51| 639.80 | 161.49 |
| KP-100    | 5   | 0.24 | 0.02    | 1.24 | 0.51 | 18.60 | 1.43 |
| ($\bar{N} = 3289$) | 25  | 0.10 | 0.01    | 20.53| 2.31 | 280.60 | 21.26 |
|           | 50  | 0.05 | 0.00    | 76.75| 9.50 | 1013.20 | 97.46 |
|           | 100 | 0.00 | 0.00    | **227.33** | 29.56 | 2889.60 | 239.35 |
### A Subspace-based Approach (SBA)

| Problem  | $|R|$ | Coverage gap | Sol. time (sec) | Models solved |
|----------|------|--------------|-----------------|---------------|
| AP-10    | 5    | 0.22 | 0.05  | 0.93 | 0.46   | 17.50 | 2.37   |
|          | 25   | 0.07 | 0.02  | 11.23| 2.44   | 212.10| 31.66  |
|          | 50   | 0.04 | 0.02  | 24.88| 7.15   | 446.80| 117.07 |
|          | 100  | 0.02 | 0.01  | 48.81| 18.21  | 857.25| 274.26 |
|          | 5    | 0.23 | 0.03  | 1.61 | 0.55   | 18.40 | 1.71   |
| AP-20    | 25   | 0.10 | 0.02  | 24.96| 4.24   | 260.60| 38.65  |
|          | 50   | 0.06 | 0.01  | 82.88| 15.92  | 829.70| 139.71 |
|          | 100  | 0.04 | 0.01  | 219.94| 46.59  | 2116.50| 386.92 |
|          | 5    | 0.27 | 0.04  | 2.13 | 0.37   | 17.90 | 1.79   |
| AP-30    | 25   | 0.12 | 0.02  | 46.04| 7.74   | 300.00| 34.65  |
|          | 50   | 0.07 | 0.01  | 169.99| 38.71  | 1034.10| 164.17 |
|          | 100  | 0.04 | 0.01  | 546.14| 140.82 | 3175.60| 544.03 |
## Territory Defining Algorithm (TDA)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td>S</td>
<td>S</td>
<td>M</td>
<td>S</td>
<td>M</td>
<td>S</td>
</tr>
<tr>
<td>KP-25</td>
<td>$\Delta_5$</td>
<td>6.20</td>
<td>1.32</td>
<td>84.66</td>
<td>16.62</td>
<td>117.80</td>
<td>70.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{25}$</td>
<td>24.75</td>
<td>1.83</td>
<td>45.09</td>
<td>9.03</td>
<td>50.29</td>
<td>13.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{50}$</td>
<td>47.83</td>
<td>3.43</td>
<td>43.22</td>
<td>5.55</td>
<td>55.85</td>
<td>7.79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KP-50</td>
<td>$\Delta_5$</td>
<td>6.60</td>
<td>1.51</td>
<td>74.36</td>
<td>12.16</td>
<td>60.09</td>
<td>39.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{25}$</td>
<td>22.60</td>
<td>4.55</td>
<td>28.63</td>
<td>9.06</td>
<td>23.55</td>
<td>7.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{50}$</td>
<td>47.70</td>
<td>3.65</td>
<td>23.55</td>
<td>5.50</td>
<td>21.36</td>
<td>7.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{100}$</td>
<td>90.40</td>
<td>5.87</td>
<td>23.90</td>
<td>8.98</td>
<td>25.34</td>
<td>13.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KP-100</td>
<td>$\Delta_5$</td>
<td>7.50</td>
<td>0.97</td>
<td>81.21</td>
<td>13.15</td>
<td>79.53</td>
<td>28.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{25}$</td>
<td>20.40</td>
<td>4.01</td>
<td>18.72</td>
<td>2.76</td>
<td>12.66</td>
<td>3.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{50}$</td>
<td>53.00</td>
<td>11.22</td>
<td>15.36</td>
<td>2.17</td>
<td>10.78</td>
<td>2.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\Delta_{100}$</td>
<td>97.80</td>
<td>10.12</td>
<td>10.57</td>
<td>1.23</td>
<td>8.13</td>
<td>1.47</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Models solved and solution time values show the number of models solved and solution times of TDA as percentages of those values of SBA, respectively.
### Territory Defining Algorithm (TDA)

| Problem | Threshold | $|R|$ | Models solved (%) | Solution time (%) |
|---------|-----------|------|-------------------|-------------------|
| AP-10   | $\Delta_5$ | 6.20 | 1.23              | 76.98             | 14.49             |
|         | $\Delta_{25}$ | 22.50 | 2.88              | 31.99             | 7.79              |
|         | $\Delta_{50}$ | 41.30 | 3.74              | 31.05             | 7.76              |
|         | $\Delta_{100}$ | 83.88 | 4.85              | 34.04             | 9.46              |
|         | $\Delta_5$ | 7.20 | 1.40              | 82.73             | 26.38             |
|         | $\Delta_{25}$ | 18.50 | 3.47              | 20.04             | 8.39              |
|         | $\Delta_{50}$ | 41.90 | 5.11              | 16.76             | 3.43              |
|         | $\Delta_{100}$ | 88.50 | 18.90             | 14.82             | 4.11              |
| AP-20   | $\Delta_5$ | 7.00 | 1.41              | 74.43             | 18.74             |
|         | $\Delta_{25}$ | 18.10 | 4.18              | 15.43             | 3.08              |
|         | $\Delta_{50}$ | 41.00 | 4.92              | 12.07             | 2.95              |
|         | $\Delta_{100}$ | 87.60 | 11.05             | 9.17              | 2.50              |
| AP-30   | $\Delta_{100}$ | 87.60 | 11.05             | 9.17              | 2.50              |

Models solved and solution time values show the number of models solved and solution times of TDA as percentages of those values of SBA, respectively.
A web-based solution platform - nMOCO-S

The Approach - Tree node data

- Node: \( k \)
- Depth of node: \( i \)
- Branching point: \( z^k \)
- Branching criterion: \( z^{k_i} \)
- Parent node: \( m \) with bound vector \( b^m \)
- Bounds: \( (b_1^m, b_2^m, \ldots, b_{i-1}^m, b_i^k, \ldots) \)
- All siblings at depth \( i \) have first \((i-1)\) bounds in common. Right sibling has a tighter bound than the one at left at criterion \( i \).
- No bound set for criteria after index \( i \).
- Leaf nodes have bounds at each criterion except the last one and completely defines the search region.
A web-based solution platform - nMOCO-S

The Approach - Tree node elements

- **Node**: defines the search space for an additional nondominated point
- **Branching point**: which nondominated point to be used to define a bound for the search space.
- **Branching criterion**: the criterion whose index is equal to the depth of the node
- **Parent node**: search space that contains the search space of the current node and its siblings
- **Bounds**: set by the branching criterion values of branching points of the current node and its parents
A web-based solution platform

www.onlinemoco.com/MOIP/

- In development phase.

- Applications:
  - nMOCO-S: Finds all nondominated points.
  - rMOCO-S: Finds a representative set of nondominated points
  - iMOCO-S: Integrates user preferences
  - libMOCO-S: Collects a variety of MOCO instances.

- Client side technologies: html5, javascript, jQuery
- Server side technologies: java servlet
- Web server: Apache Tomcat
- Database: Apache Derby and JDBC