The Certus Centre (www.certus-sfi.no)

- Centre for research-based innovation (Norwegian SFI)
- Hosted by Simula Research Laboratory in Oslo
- Dedicated to Software Validation & Verification
- Industrial collaborations and public partners
- Expertise:
  - Intelligent testing through artificial intelligence techniques
    (Constraint Programming, Machine Learning)

Kongsberg Maritime  Cisco Systems Norway  ABB Robotics, Norway  Cancer Registry of Norway
Constraint Optimization solvers transform an optimization problem into a search tree.

```plaintext
1 % Inputs
2 int: n;
3 array[1..n] of int: limits;
4 int: globalimit = n*n;
5
6 % Decision variables
7 array[1..n] of var 0..globalimit: x;
8
9 % Constraints
10 constraint alldifferent(x);
11 constraint increasing(x);
12 constraint forall(i in index_set(limits))(x[i] > limits[i]);
13
14 % Optimization
15 var int: objective = sum(x);
16 solve minimize objective;
```

Constraint Optimization Problem

Search Tree
Constraint Optimization solvers transform an optimization problem into a search tree.

```plaintext
1 % Inputs
2 int: n;
3 array[1..n] of int: limits;
4 int: globallimit = n*n;
5
6 % Decision variables
7 array[1..n] of var 0..globallimit: x;
8
9 % Constraints
10 constraint alldifferent(x);
11 constraint increasing(x);
12 constraint forall (i in index_set(limits))(x[i] > limits[i]);
13
14 % Optimization
15 var int: objective = sum(x);
16 solve minimize objective;
```

Constraint Optimization Problem

Search Tree
An objective boundary reduces the search space

Full Search Tree

constraint objective < X

Pruned Search Tree
We train a regression model from solved instances to estimate objective boundaries.
Objective boundaries allow to discard parts of the search tree and help to solve the problem.
The regression model in boundary estimation can be any supervised machine learning model.

- Gradient Tree Boosting (GTB)
- Neural Network (NN)
- Support Vector Machine (SVM)
- Linear Regression (LR)
- ...

(simula)
A supervised model is trained by providing a set training samples consisting of inputs and targets.

Inputs
- Instance
- Instance
- Instance
- Instance

Targets
- Objective Value
- Objective Value
- Objective Value
- Objective Value

Minimize Gap between Estimated Value and Target
Each problem instance is represented by a fixed size feature vector.

9 instance features statistically describe the parameters of each data structure:

- Number of elements, Minimum, Maximum, Mean, Median, Std. Deviation, Interquartile range, Skewness, Kurtosis

95 model features describe the constraint model, broadly categorized in:

- Variables, Domains, Constraints, Global Constraints, Objective Variable

The model features follow
Misestimations can render the problem unsatisfiable, therefore we need to take countermeasures.

If the estimated objective is too low, all solutions are excluded and the problem instance is unsatisfiable.

Three counter-measures:

a) Label-shift
b) Asymmetric loss function
c) Restart with negated constraint
Label shift moves the labels of training samples away from true label to allow larger errors.
Label shift moves the labels of training samples away from true label to allow larger errors.

- Negative Error (Underestimation)
- Positive Error (Overestimation)
Label shift moves the labels of training samples away from true label to allow larger errors.

\[ y'_{\text{true}} = y_{\text{true}} + \lambda \times (\text{UpperObjDomain} - y_{\text{true}}) \]

\( y'_{\text{true}} \): Adjusted objective value of training sample

\( y_{\text{true}} \): Original objective value of training sample

\( \lambda \): Configuration parameter (\( \lambda \in [0,1] \))

\( \text{UpperObjDomain} \): Upper domain boundary of objective variable
Standard model training uses symmetric loss and penalizes under- and overestimations equally.
An asymmetric loss functions steers the model towards only over- or underestimations.

Symmetric Error Distribution

Asymmetric Error Distribution

Negative Error $\times$ (Underestimation)

Positive Error ✔ (Overestimation)
An asymmetric loss functions steers the model towards only over- or underestimations.
If an unsatisfiable instance occurs at runtime, the solver can be restarted with a negated constraint.

- **Unsatisfiable**
  - constraint objective $< X$
  - Negate constraint
  - constraint objective $\geq X$

- **Satisfiable**
Evaluation
We selected 7 COPs with the most instances from the MiniZinc benchmark repository.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRCPSP</td>
<td>11182</td>
</tr>
<tr>
<td>RCPSP</td>
<td>2904</td>
</tr>
<tr>
<td>2D Bin Packing</td>
<td>500</td>
</tr>
<tr>
<td>Cutting Stock</td>
<td>121</td>
</tr>
<tr>
<td>Jobshop</td>
<td>74</td>
</tr>
<tr>
<td>VRP</td>
<td>74</td>
</tr>
<tr>
<td>Open Stacks</td>
<td>50</td>
</tr>
</tbody>
</table>

Available at https://github.com/MiniZinc/minizinc-benchmarks
The regression model in boundary estimation can be any supervised machine learning model.

- Gradient Tree Boosting (GTB) with symmetric and asymmetric loss
- Neural Networks (NN) with symmetric and asymmetric loss
- Support Vector Machines (SVM) with symmetric loss
- Linear Regression (LR) with symmetric loss
First, we compare the performance of the models to estimate the objective value.

- Repeated 10-fold cross-validation
- Train regression model per problem on 9 folds and test on remaining fold
- Average results over multiple repetitions with different random initialization
- Evaluation metrics

  Gap: Objective Boundary compared to true objective (for model comparison)

  Pruned (Prn): Reduction in objective domain size
Boundary estimation with neural networks prunes the objective domain up to 73%.

<table>
<thead>
<tr>
<th></th>
<th>GTB\textsubscript{a}</th>
<th>GTB\textsubscript{s}</th>
<th>LR</th>
<th>\textit{NN}\textsubscript{a}</th>
<th>\textit{NN}\textsubscript{s}</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
</tr>
<tr>
<td>Bin Packing</td>
<td>5.0</td>
<td>42</td>
<td>4.5</td>
<td>45</td>
<td>6.0</td>
<td>29</td>
</tr>
<tr>
<td>Cutting Stock</td>
<td>1.9</td>
<td>7</td>
<td>1.2</td>
<td>12</td>
<td>1.2</td>
<td>8</td>
</tr>
<tr>
<td>Jobshop</td>
<td>2.8</td>
<td>63</td>
<td>3.4</td>
<td>55</td>
<td>4.6</td>
<td>36</td>
</tr>
<tr>
<td>MRCPS  P</td>
<td>2.8</td>
<td>45</td>
<td>2.7</td>
<td>47</td>
<td>3.6</td>
<td>31</td>
</tr>
<tr>
<td>Open Stacks</td>
<td>1.8</td>
<td>35*</td>
<td>1.8</td>
<td>35*</td>
<td>2.3</td>
<td>21*</td>
</tr>
<tr>
<td>RCPSP</td>
<td>4.2</td>
<td>21</td>
<td>2.4</td>
<td>41</td>
<td>3.0</td>
<td>27</td>
</tr>
<tr>
<td>VRP</td>
<td>39.8*</td>
<td>67</td>
<td>51.9**</td>
<td>59</td>
<td>81.2***</td>
<td>39</td>
</tr>
</tbody>
</table>
Boundary estimation with neural networks prunes the objective domain up to 73%.

<table>
<thead>
<tr>
<th></th>
<th>GTB_α</th>
<th></th>
<th>GTB_s</th>
<th></th>
<th>LR</th>
<th></th>
<th>NN_α</th>
<th></th>
<th>NN_s</th>
<th></th>
<th>SVM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
<td>Gap</td>
<td>Prn</td>
</tr>
<tr>
<td>Bin Packing</td>
<td>5.0</td>
<td>42</td>
<td>4.5</td>
<td>45</td>
<td>6.0</td>
<td>29</td>
<td>3.5</td>
<td>53</td>
<td>3.5</td>
<td>52</td>
<td>5.0</td>
<td>46</td>
</tr>
<tr>
<td>Cutting Stock</td>
<td>1.9</td>
<td>7</td>
<td>1.2</td>
<td>12</td>
<td>1.2</td>
<td>8</td>
<td>1.5</td>
<td>6</td>
<td>1.1</td>
<td>16</td>
<td>3.7</td>
<td>2</td>
</tr>
<tr>
<td>Jobshop</td>
<td>2.8</td>
<td>63</td>
<td>3.4</td>
<td>55</td>
<td>4.6</td>
<td>36</td>
<td>2.4</td>
<td>69</td>
<td>2.8</td>
<td>63</td>
<td>3.4</td>
<td>49</td>
</tr>
<tr>
<td>MRCPSP</td>
<td>2.8</td>
<td>45</td>
<td>2.7</td>
<td>47</td>
<td>3.6</td>
<td>31</td>
<td>2.2</td>
<td>57</td>
<td>2.3</td>
<td>55</td>
<td>2.7</td>
<td>47</td>
</tr>
<tr>
<td>Open Stacks</td>
<td>1.8</td>
<td>35*</td>
<td>1.8</td>
<td>35*</td>
<td>2.3</td>
<td>21*</td>
<td>1.8</td>
<td>33*</td>
<td>1.6</td>
<td>38*</td>
<td>2.0</td>
<td>32*</td>
</tr>
<tr>
<td>RCPSP</td>
<td>4.2</td>
<td>21</td>
<td>2.4</td>
<td>41</td>
<td>3.0</td>
<td>27</td>
<td>2.0</td>
<td>47</td>
<td>2.0</td>
<td>49</td>
<td>2.9</td>
<td>34</td>
</tr>
<tr>
<td>VRP</td>
<td>39.8*</td>
<td>67</td>
<td>51.9**</td>
<td>59</td>
<td>81.2***</td>
<td>39</td>
<td>31.7*</td>
<td>73</td>
<td>41.0*</td>
<td>68</td>
<td>52.0**</td>
<td>56**</td>
</tr>
</tbody>
</table>

*denotes a significant improvement, **denotes a very significant improvement.
Second, we measure the effect an estimated objective boundary has on the solver performance.

- 100 instances per problem
- Quality of first solution and time to find it
- Solvers used
  - Chuffed
  - Gecode 6.0
  - Google OR-Tools 6.7
- Default search heuristic of model or solver
Boundary estimation can result in better first solutions in similar time.
We identified challenges and limitations, that need to be addressed in future work.

- No correct estimation guarantees -> Counter-measures to reduce risk
- Limited number of instances for most problems
- Potentially limited representational power through general features
- Which problems benefit from boundary constraints? Propagation possible?
Boundary estimation can improve the solving process from historical information.

Objective boundaries allow to discard parts of the search tree and help to solve the problem.

Label shift moves the labels of training samples away from true label to allow larger errors.

An asymmetric loss functions steers the model towards only over- or underestimations.

Boundary estimation can result in better first solutions in similar time.
Boundary estimation can improve the solving process from historical information.

Objective boundaries allow to discard parts of the search tree and help to solve the problem.

An asymmetric loss functions steers the model towards only over- or underestimations.

Label shift moves the labels of training samples away from true label to allow larger errors.

Boundary estimation can result in better first solutions in similar time.

Thank you! Questions?