Estimating Objective Boundaries for Constraint Optimization Problems

Helge Spieker, Arnaud Gotlieb

Certus SFI / Simula Research Laboratory

NordConsNet Workshop 29.05.2018

Certus

Centre for Software Verification & Validation

helge@simula.no

simula

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

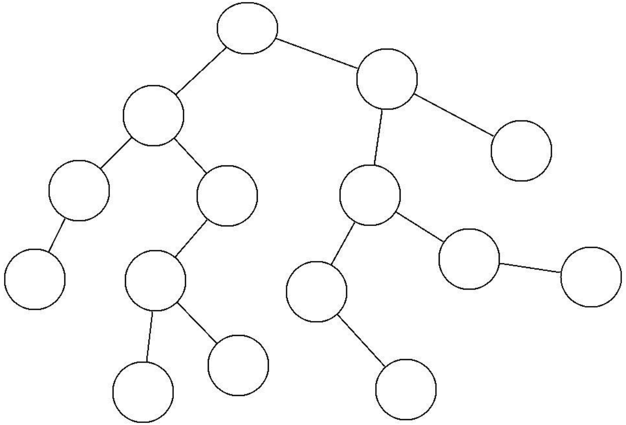
ABB Robotics, Norway

Cancer Registry of Norway

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

```
% Inputs
int: n;
array[1..n] of int: limits;
int: globallimit = n*n;

% Decision variables
array[1..n] of var 0..globallimit: x;

% Constraints
constraint alldifferent(x);
constraint increasing(x);
constraint forall(i in index_set(limits))(x[i] > limits[i]);

% Optimization
var int: objective = sum(x);
solve minimize objective;
```

Constraint Optimization Problem

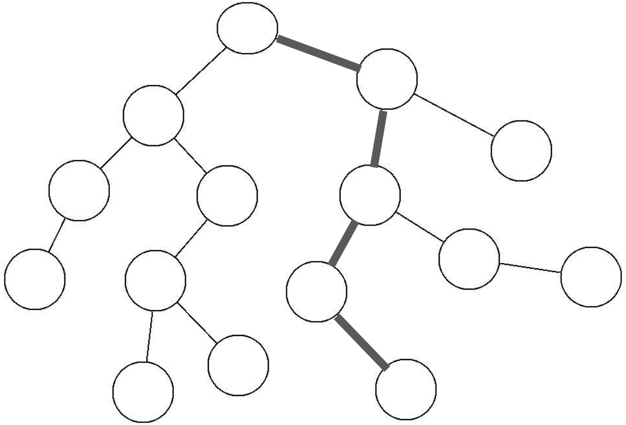


Search Tree

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

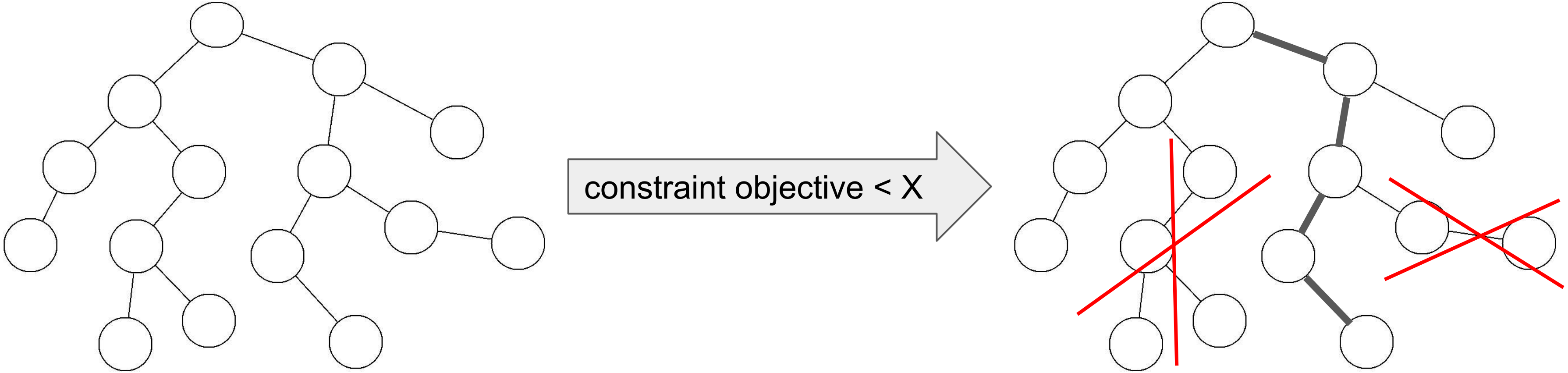
```
% Inputs
int: n;
array[1..n] of int: limits;
int: globallimit = n*n;

% Decision variables
array[1..n] of var 0..globallimit: x;

% Constraints
constraint alldifferent(x);
constraint increasing(x);
constraint forall(i in index_set(limits))(x[i] > limits[i]);

% Optimization
var int: objective = sum(x);
solve minimize objective;
```

Constraint Optimization Problem



Search Tree

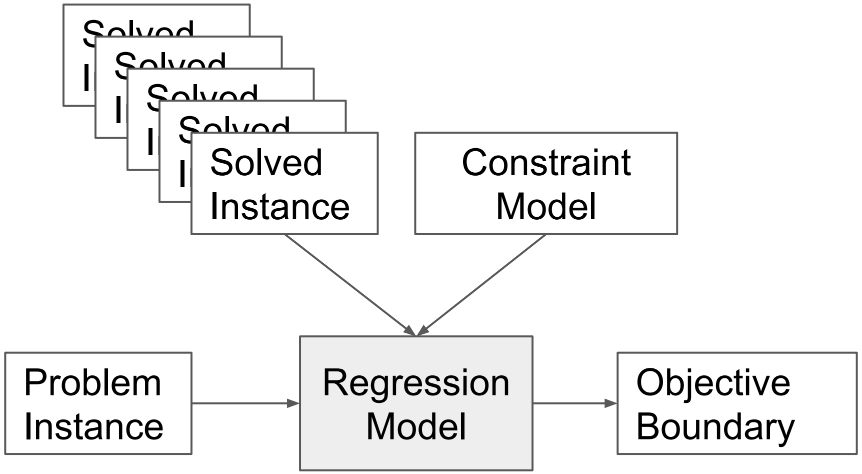
An objective boundary reduces the search space



Full Search Tree

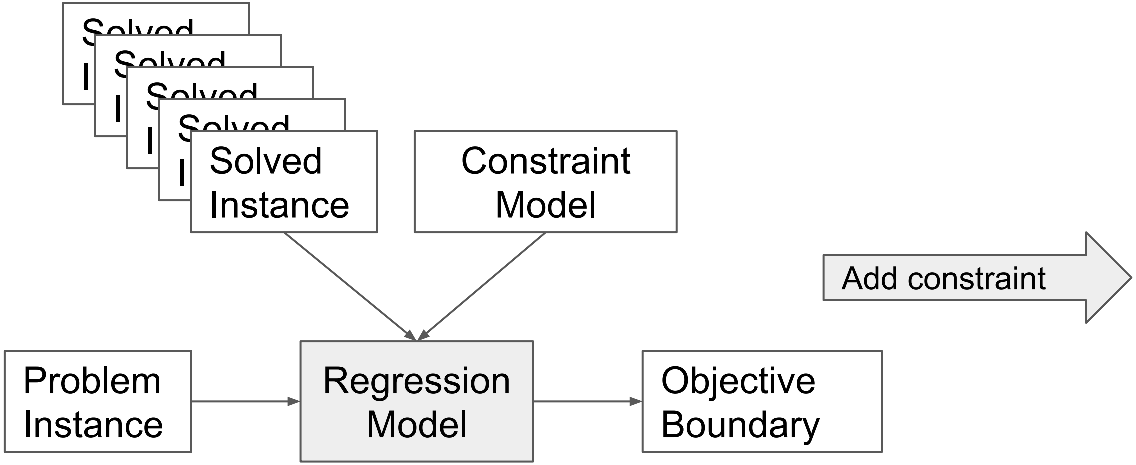
Pruned Search Tree

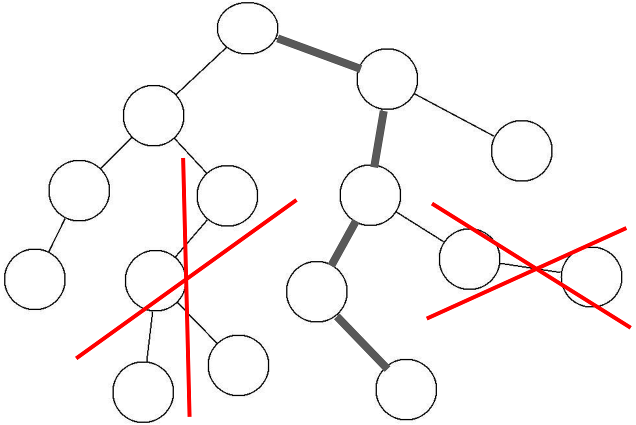
We train a regression model from solved instances to estimate objective boundaries.



simula

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

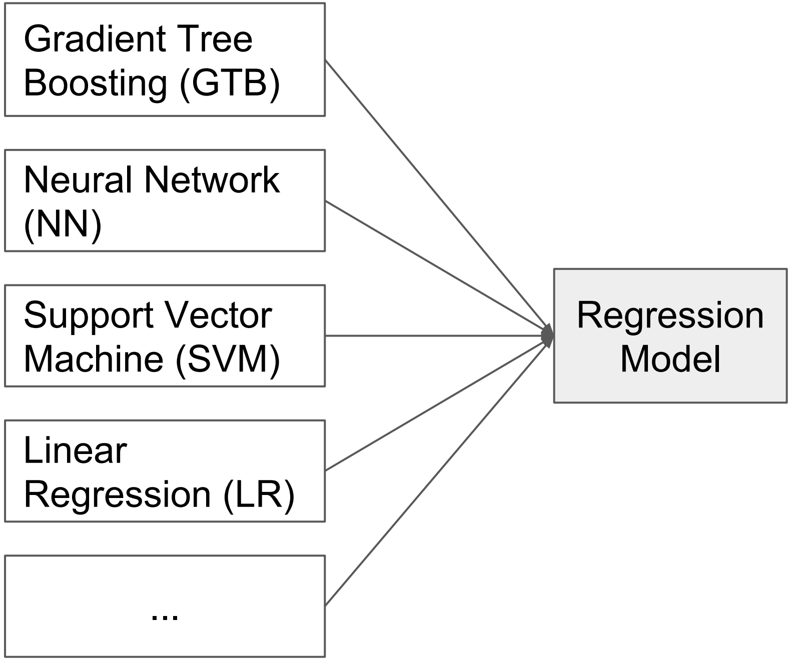




Search Tree with Prunings

simula

The regression model in boundary estimation can be any supervised machine learning model.



simula

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

Inputs		Targets
Instance	+	Objective Value
Instance	+	Objective Value
Instance	+	Objective Value
Instance	+	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
Amadini, R., Gabbrielli, M., & Mauro, J. (2014). An enhanced features extractor for a portfolio of constraint solvers. In Symposium on Applied Computing (pp. 1357–1359). https://doi.org/10.1145/2554850.2555114

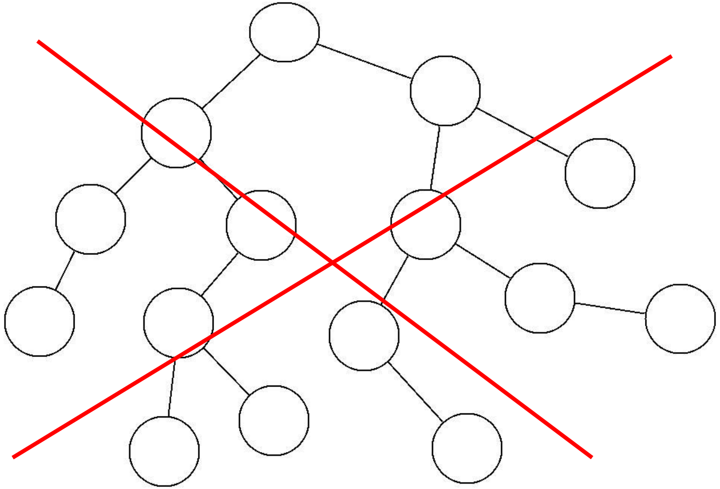
simula

Misestimations can render the problem unsatisfiable, therefore we need to take countermeasures.

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

Three counter-measures:

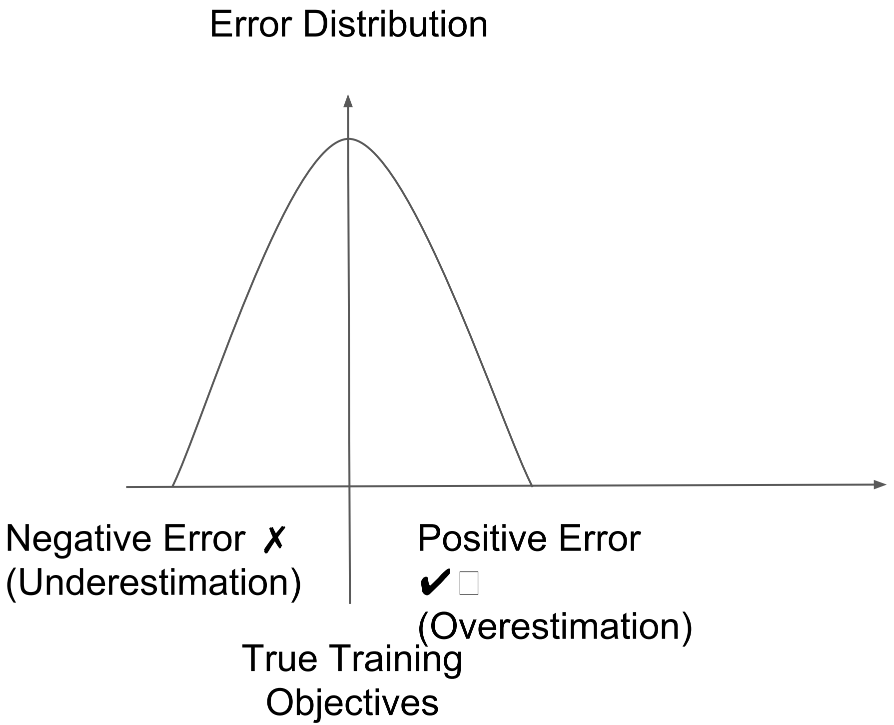
a) Label-shift
b) Asymmetric loss function
c) Restart with negated constraint



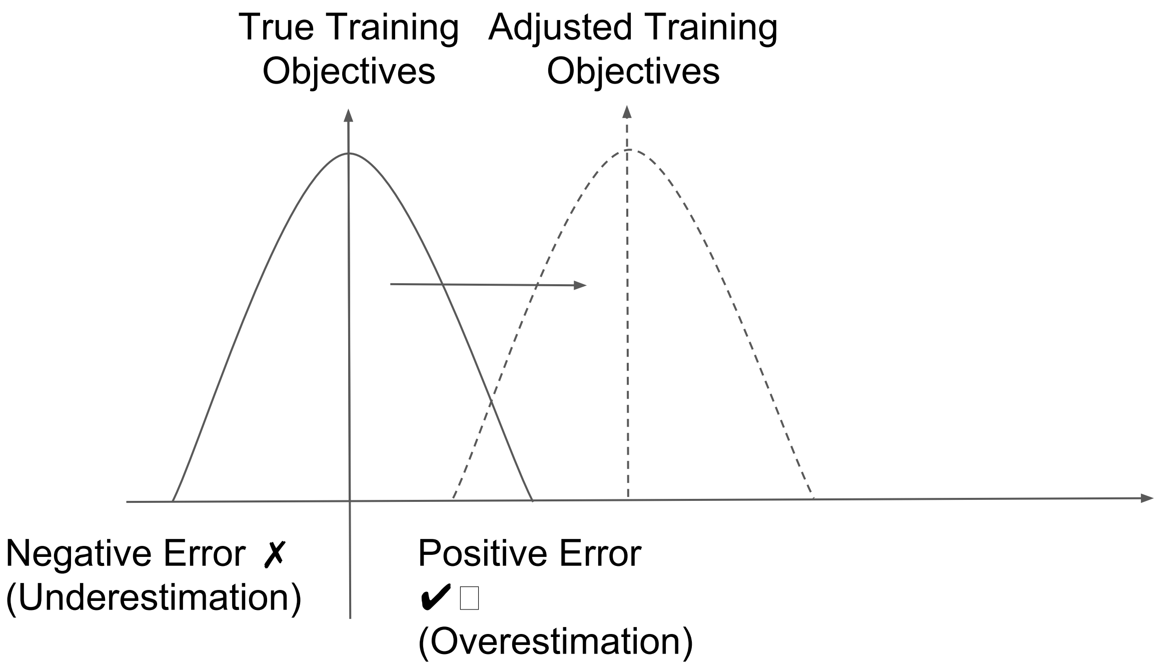
Unsatisfiable Problem

simula

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.



simula

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

$$y'_{true} = y_{true} + \lambda * (UpperObjDomain - y_{true})$$

y'_{true}: Adjusted objective value of training sample

y_{true}: Original objective value of training sample

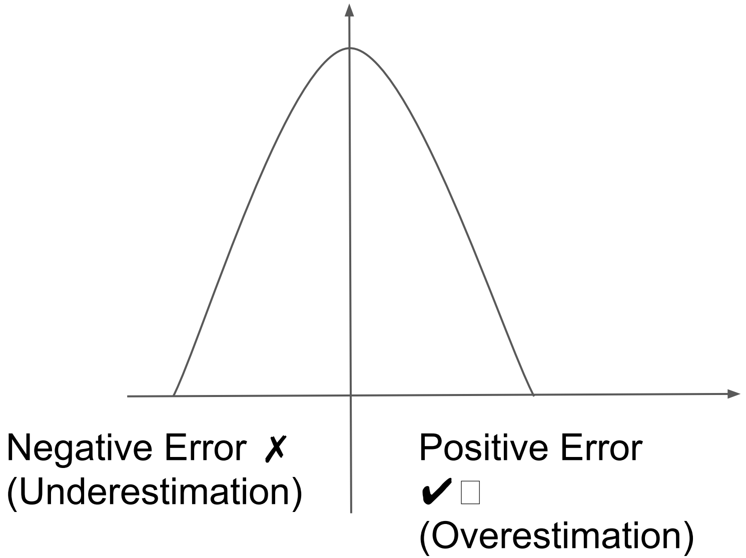
λ: Configuration parameter ($\lambda \in [0,1]$)

UpperObjDomain: Upper domain boundary of objective variable

simula

Standard model training uses symmetric loss and penalizes under- and overestimations equally.

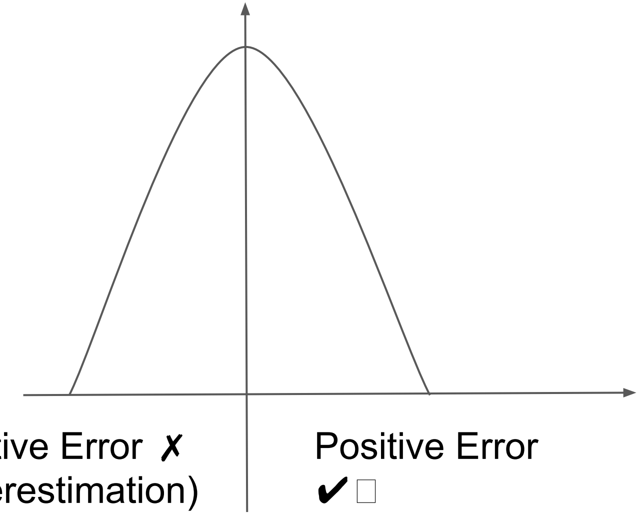
Symmetric Error Distribution



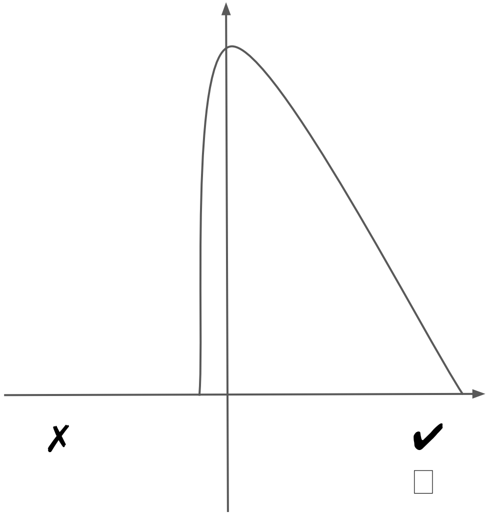
simula

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

Symmetric Error Distribution

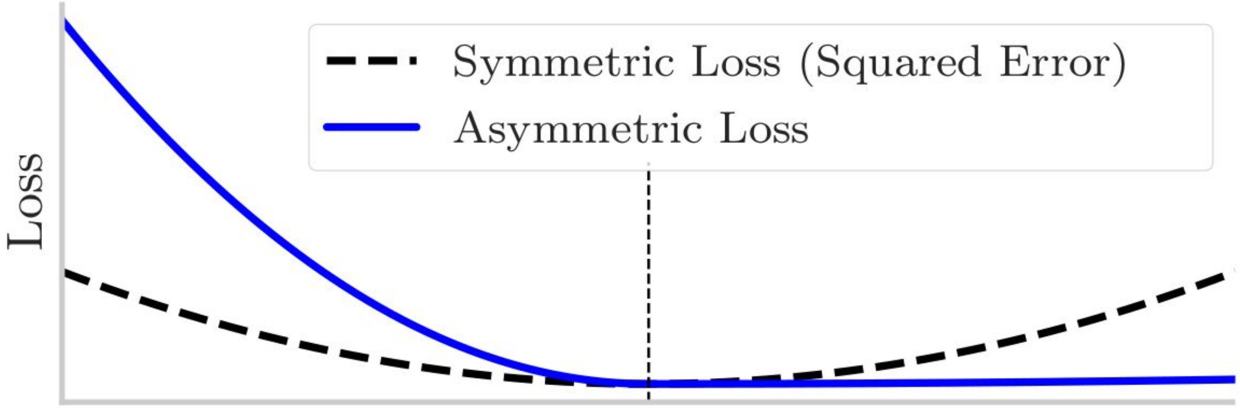


Asymmetric Error Distribution



simula

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



Negative Error ✗
(Underestimation)

Positive Error
✔
(Overestimation)

simula

If an unsatisfiable instance occurs at runtime, the solver can be restarted with a negated constraint.

constraint objective < X

✖ □ Unsatisfiable

Negate constraint

constraint objective >= X

✓ Satisfiable

Evaluation

We selected 7 COPs with the most instances from the MiniZinc benchmark repository.

Problem	**Number of Instances**
MRCPSP	11182
RCPSP	2904
2D Bin Packing	500
Cutting Stock	121
Jobshop	74
VRP	74
Open Stacks	50

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

simula

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

simula

Boundary estimation with neural networks prunes the objective domain up to 73%.

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

simula

Boundary estimation with neural networks prunes the objective domain up to 73%.

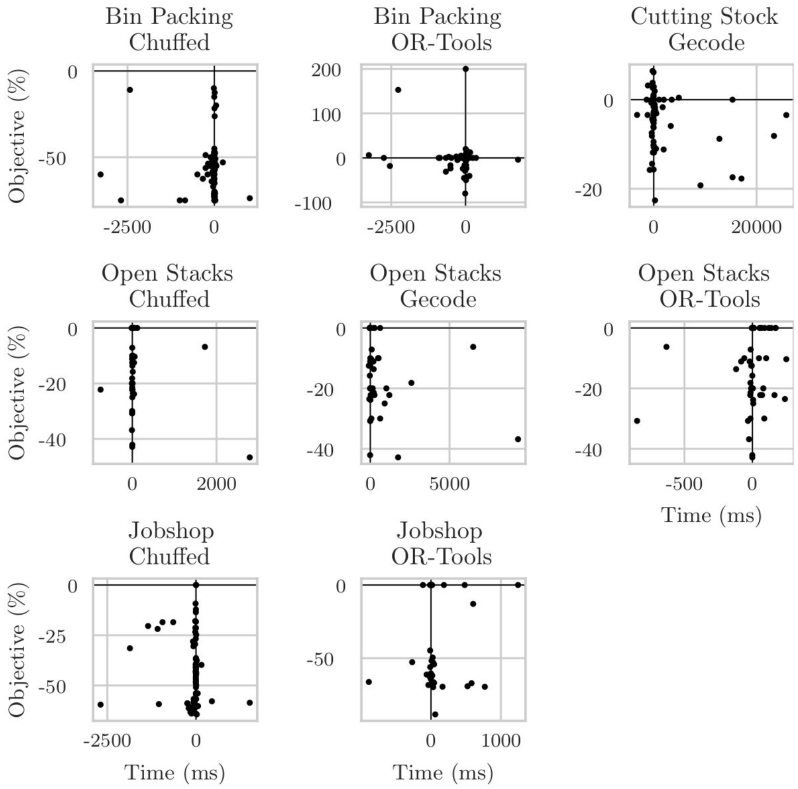
	GTB_a		GTB_s		LR		NN_a		NN_s		SVM	
	Gap	Prn	Gap	Prn	Gap	Prn	Gap	Prn	Gap	Prn	Gap	Prn
Bin Packing	5.0	42	4.5	45	6.0	29	**3.5**	**53**	3.5	52	5.0	46
Cutting Stock	1.9	7	1.2	12	1.2	8	1.5	6	**1.1**	**16**	3.7	2
Jobshop	2.8	63	3.4	55	4.6	36	**2.4**	**69**	2.8	63	3.4	49^{**}
MRCPSP	2.8	45	2.7	47	3.6	31	**2.2**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

simula

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.



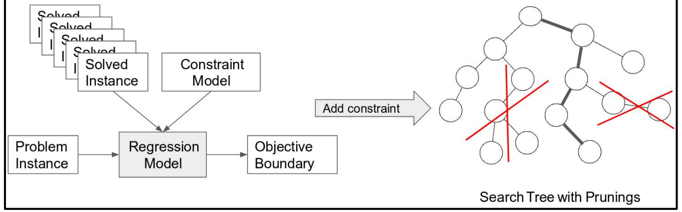
simula

We identified challenges and limitations, that need to be addressed in future work.

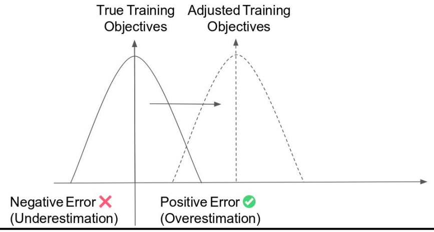
- No correct estimation guarantees -> Counter-measures to reduce risk
- Limited number of instances for most problems
- Potentially limited representational power through general features
- Which problems benefit from boundary constraints? Propagation possible?

Boundary estimation can improve the solving process from historical information.

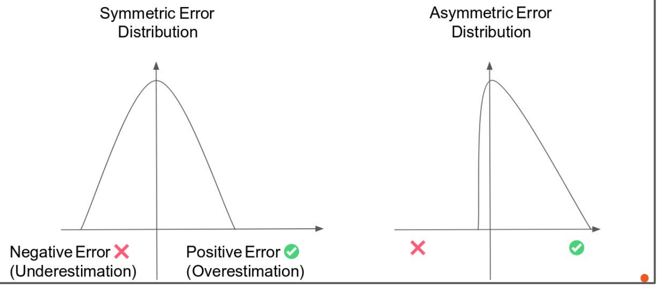
Objective boundaries allow to discard parts of the search tree and help to solve the problem.



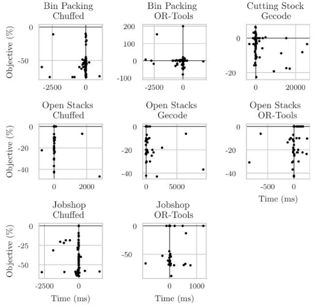
Label shift moves the labels of training samples away from true label to allow larger errors.



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



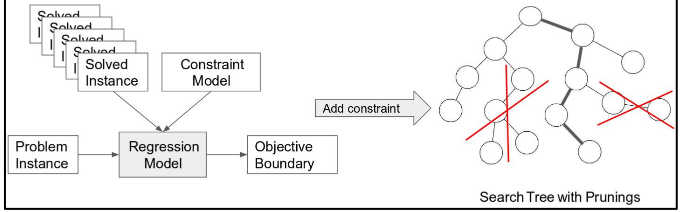
Boundary estimation can result in better first solutions in similar time.



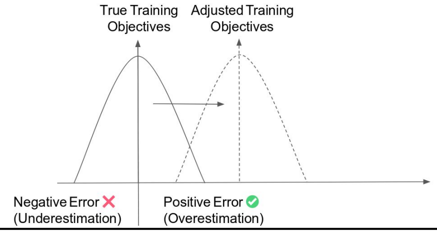
simula

Boundary estimation can improve the solving process from historical information.

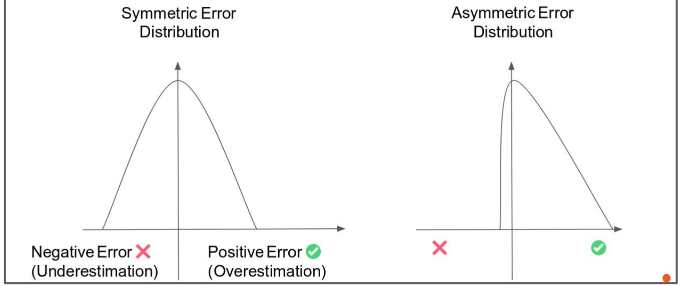
Objective boundaries allow to discard parts of the search tree and help to solve the problem.



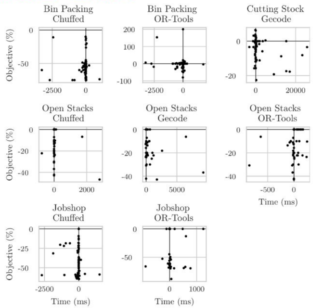
Label shift moves the labels of training samples away from true label to allow larger errors.



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



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



Thank you! Questions?