Artificial Intelligence Techniques for Automatic Reformulation and Solution of Structured Mathematical Models

Luis Perez Sanchez

perez@di. unipi . it

October 28, 2010

Some Motivations

- Increasingly large and complex models.

Some Motivations

- Increasingly large and complex models.
- Practitioners tend to generate natural formulations.

Some Motivations

- Increasingly large and complex models.
- Practitioners tend to generate natural formulations.
- Solution requires specific formulation that reveals certain structures: LP, MILP, DCP, MCF, MMCF, Path, KnapSacking, etc

Some Motivations

- Increasingly large and complex models.
- Practitioners tend to generate natural formulations.
- Solution requires specific formulation that reveals certain structures: LP, MILP, DCP, MCF, MMCF, Path, KnapSacking, etc
- Suitable formulations are created by few specialists.

Some Motivations

- Increasingly large and complex models.
- Practitioners tend to generate natural formulations.
- Solution requires specific formulation that reveals certain structures: LP, MILP, DCP, MCF, MMCF, Path, KnapSacking, etc
- Suitable formulations are created by few specialists.
- Due to biasing, the best formulation is often ignored.

Some Motivations

- Increasingly large and complex models.
- Practitioners tend to generate natural formulations.
- Solution requires specific formulation that reveals certain structures: LP, MILP, DCP, MCF, MMCF, Path, KnapSacking, etc
- Suitable formulations are created by few specialists.
- Due to biasing, the best formulation is often ignored.
- In the application of solution methods we have the same phenomenon.

Main Goal

Definition of an environment that allows

1. structured modeling,
2. $\langle$(re)formulation, solver, configuration$\rangle$ selection, and
3. structured solver application.

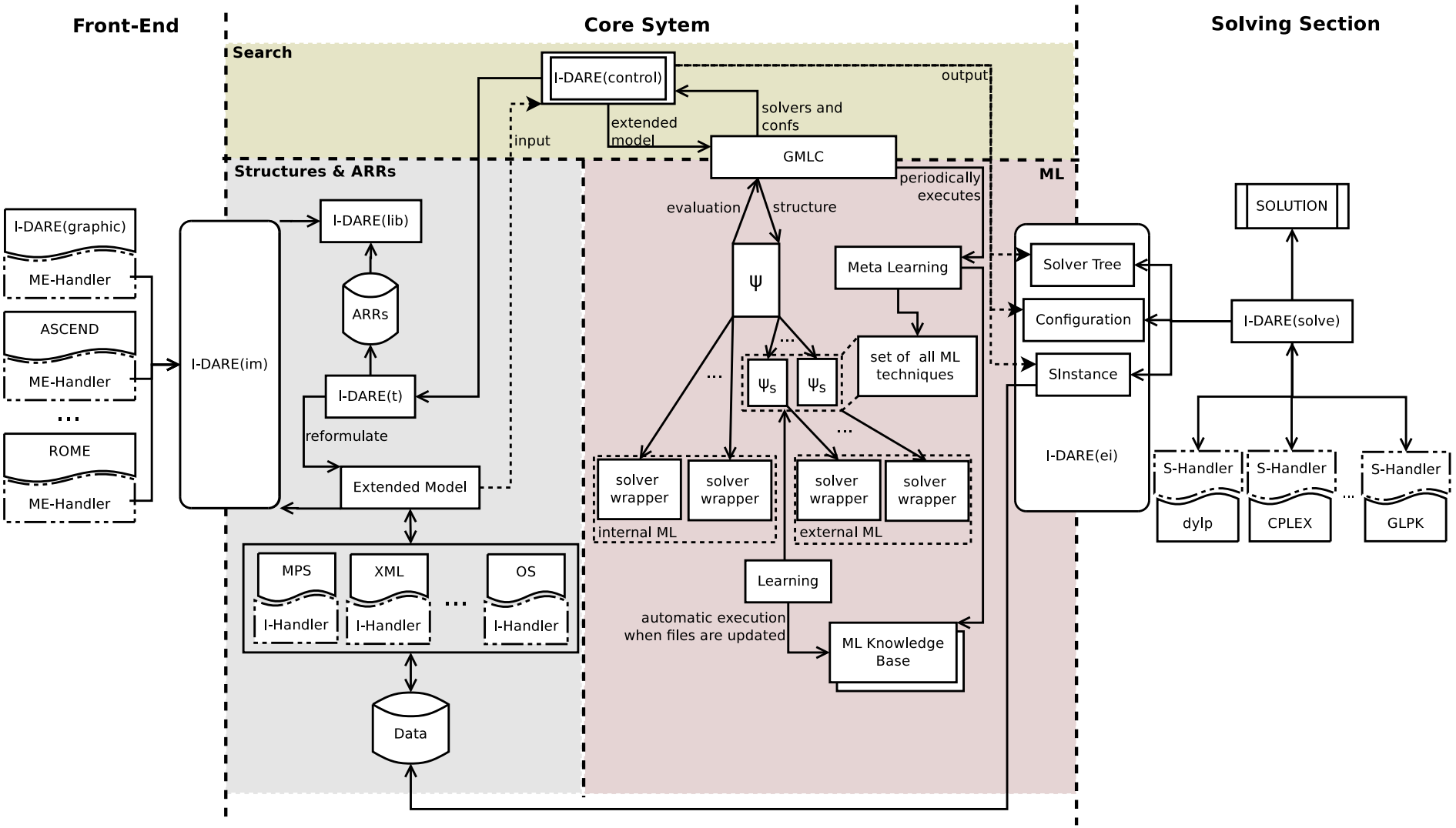
Main Goal

Definition of an environment that allows

1. structured modeling,
2. $\langle$(re)formulation, solver, configuration$\rangle$ selection, and
3. structured solver application.

I-DARE - Intelligence–Drive Automatic Reformulation Engine.

I-DARE


Front-End
Core Sytem
Solving Section
Search
I-DARE(control)
output
solvers and confs
extended model
input
GMLC
Structures & ARRs
periodically executes
ML
evaluation
structure
I-DARE(graphic)
ME-Handler
ASCEND
ME-Handler
ROME
ME-Handler
I-DARE(im)
I-DARE(lib)
ARRs
I-DARE(t)
reformulate
Extended Model
Ψ
Meta Learning
Ψs
Ψs
set of all ML techniques
solver wrapper
solver wrapper
internal ML
solver wrapper
solver wrapper
external ML
Solver Tree
Configuration
SInstance
I-DARE(ei)
SOLUTION
I-DARE(solve)
S-Handler
S-Handler
S-Handler
dylp
CPLEX
GLPK
MPS
XML
OS
I-Handler
I-Handler
I-Handler
Learning
automatic execution when files are updated
ML Knowledge Base
Data

Search Spaces

- Extended Model: Formulation + Enhanced Instance

Search Spaces

- Extended Model: Formulation + Enhanced Instance
  - All the model and data information can be queried.

Search Spaces

- Extended Model: Formulation + Enhanced Instance
  - All the model and data information can be queried.
  - Using $\Xi(F)$ we can move through reformulations (in case they were created by ARR$^{\Sigma}$ data information will be also accessible).

Search Spaces

- Extended Model: Formulation + Enhanced Instance
  - All the model and data information can be queried.
  - Using $\Xi(F)$ we can move through reformulations (in case they were created by ARR$^{\Sigma}$ data information will be also accessible).
- All solver information is accessible from $\mathcal{F}$LORA-*2* (registration, configuration and signatures), plus the solvers' tree structure.

Search Spaces

- Extended Model: Formulation + Enhanced Instance
  - All the model and data information can be queried.
  - Using $\Xi(F)$ we can move through reformulations (in case they were created by ARR$^{\Sigma}$ data information will be also accessible).
- All solver information is accessible from $\mathcal{F}$LORA-*2* (registration, configuration and signatures), plus the solvers' tree structure.
- Configurations, the used XML defines a well structured way to access the solver's configuration.
  - We can access the parameters and their domains.

Controlling the Search

- We have chosen to provide a general and abstract setting for performing the search.

Controlling the Search

- We have chosen to provide a general and abstract setting for performing the search.
- A general interface is proposed to define the control mechanism,

```
1  d_control [
2      process(d_InstanceWrapper) -> [d_InstanceWrapper, _term]
3  ].
```

- This interface receives the model+data of the problem.
- and returns an equivalent instance, plus the solver's tree and their configurations.

Controlling the Search

- We have chosen to provide a general and abstract setting for performing the search.
- A general interface is proposed to define the control mechanism,

```
1  d_control [
2      process(d_InstanceWrapper) -> [d_InstanceWrapper, _term]
3  ].
```

- This interface receives the model+data of the problem.
- and returns an equivalent instance, plus the solver's tree and their configurations.
- We focused on the subproblem of searching in the (Solver, Configuration) space.

Objective function

- An effective way of evaluating the quality of the elements of (Solver, Configuration) subspace is needed.

Objective function

- An effective way of evaluating the quality of the elements of (Solver, Configuration) subspace is needed.
- This "quality" measure will be the objective function of the search called ψ.

Objective function

- An effective way of evaluating the quality of the elements of (Solver, Configuration) subspace is needed.
- This "quality" measure will be the objective function of the search called ψ.
- ψ may measure the running time or accuracy.

Objective function

- An effective way of evaluating the quality of the elements of (Solver, Configuration) subspace is needed.
- This "quality" measure will be the objective function of the search called ψ.
- ψ may measure the running time or accuracy.
- An arithmetic or algorithmic description of ψ will hardly be available.

Objective function

- An effective way of evaluating the quality of the elements of (Solver, Configuration) subspace is needed.
- This "quality" measure will be the objective function of the search called ψ.
- ψ may measure the running time or accuracy.
- An arithmetic or algorithmic description of ψ will hardly be available.
- We propose the application of ML techniques to approximate it (based on known observations).

Objective function

- ML will need a description of the data points to be used, since the whole flat instance is too much (or enough) information.

Objective function

- ML will need a description of the data points to be used, since the whole flat instance is too much (or enough) information.
- Relevant "features" may change from problem class to problem class, even more, from solver to solver.

Objective function

- ML will need a description of the data points to be used, since the whole flat instance is too much (or enough) information.
- Relevant "features" may change from problem class to problem class, even more, from solver to solver.
  - for LP: number of variables, constraints, density of the constraint matrix.

Objective function

- ML will need a description of the data points to be used, since the whole flat instance is too much (or enough) information.
- Relevant "features" may change from problem class to problem class, even more, from solver to solver.
  - for LP: number of variables, constraints, density of the constraint matrix.
  - degeneracy of the vertex of the polyhedron affects Simplex but not Interior-Point.

Objective function

- ML will need a description of the data points to be used, since the whole flat instance is too much (or enough) information.
- Relevant "features" may change from problem class to problem class, even more, from solver to solver.
  - for LP: number of variables, constraints, density of the constraint matrix.
  - degeneracy of the vertex of the polyhedron affects Simplex but not Interior-Point.
- Each solver must supply the set of features relevant to itself.

Features

- A solver wrapper will provide with a set of features relevant to the solver and possible configurations.

```
1  d_solverWrapper [
2      solver          => d_solver,
3      retrieve(?EM, ?CT) => [_list, CT],
4      [internal]
5  ].
```

Features

- A solver wrapper will provide with a set of features relevant to the solver and possible configurations.

```
1  d_solverWrapper [
2      solver          => d_solver,
3      retrieve(?EM, ?CT) => [_list, CT],
4      [internal]
5  ].
```

- If it is internal the features will be a few (or one) representing the evaluation of ψ (with one configuration).

Features

- A solver wrapper will provide with a set of features relevant to the solver and possible configurations.

```
1  d_solverWrapper [
2      solver          => d_solver,
3      retrieve(?EM, ?CT) => [_list, CT],
4      [internal]
5  ].
```

- If it is internal the features will be a few (or one) representing the evaluation of ψ (with one configuration).
- If it is external it returns the features and configuration to be evaluated by a ML technique.

Features

- A solver wrapper will provide with a set of features relevant to the solver and possible configurations.

```
1  d_solverWrapper [
2      solver          => d_solver,
3      retrieve(?EM, ?CT) => [_list, CT],
4      [internal]
5  ].
```

- If it is internal the features will be a few (or one) representing the evaluation of ψ (with one configuration).
- If it is external it returns the features and configuration to be evaluated by a ML technique.
- There intermediate scenarios (also external) that may use sophisticated techniques to compute the features.

Features

- The solver wrapper is also parametrized using a configuration template, to constraint the usable configurations.

Features

- The solver wrapper is also parametrized using a configuration template, to constraint the usable configurations.
  - One may demand a certain accuracy,

Features

- The solver wrapper is also parametrized using a configuration template, to constraint the usable configurations.
  - One may demand a certain accuracy,
  - maximum CPU time,

Features

- The solver wrapper is also parametrized using a configuration template, to constraint the usable configurations.
  - One may demand a certain accuracy,
  - maximum CPU time,
  - architectural constraints,

Features

- The solver wrapper is also parametrized using a configuration template, to constraint the usable configurations.
  - One may demand a certain accuracy,
  - maximum CPU time,
  - architectural constraints,
  - price constraints.

Generic ML

- Using the features and CT several data points are created.

Generic ML

- Using the features and CT several data points are created.
- Different ML mechanism may be proposed (and even used simultaneously).

Generic ML

- Using the features and CT several data points are created.
- Different ML mechanism may be proposed (and even used simultaneously).
- A general interface is proposed to plug the MLs.

```
1  d_machineLearning [
2       evaluate(_list) => _list ,
3      => train(_list , _list)
4  ].
```

Generic ML

- Using the features and CT several data points are created.
- Different ML mechanism may be proposed (and even used simultaneously).
- A general interface is proposed to plug the MLs.

```
1  d_machineLearning [
2       evaluate(_list) => _list ,
3      => train(_list , _list)
4  ].
```

- Note that for nested structures, the SW has the possibility to access the SWs of the sub-blocks to decompose the task of prediction of ψ.

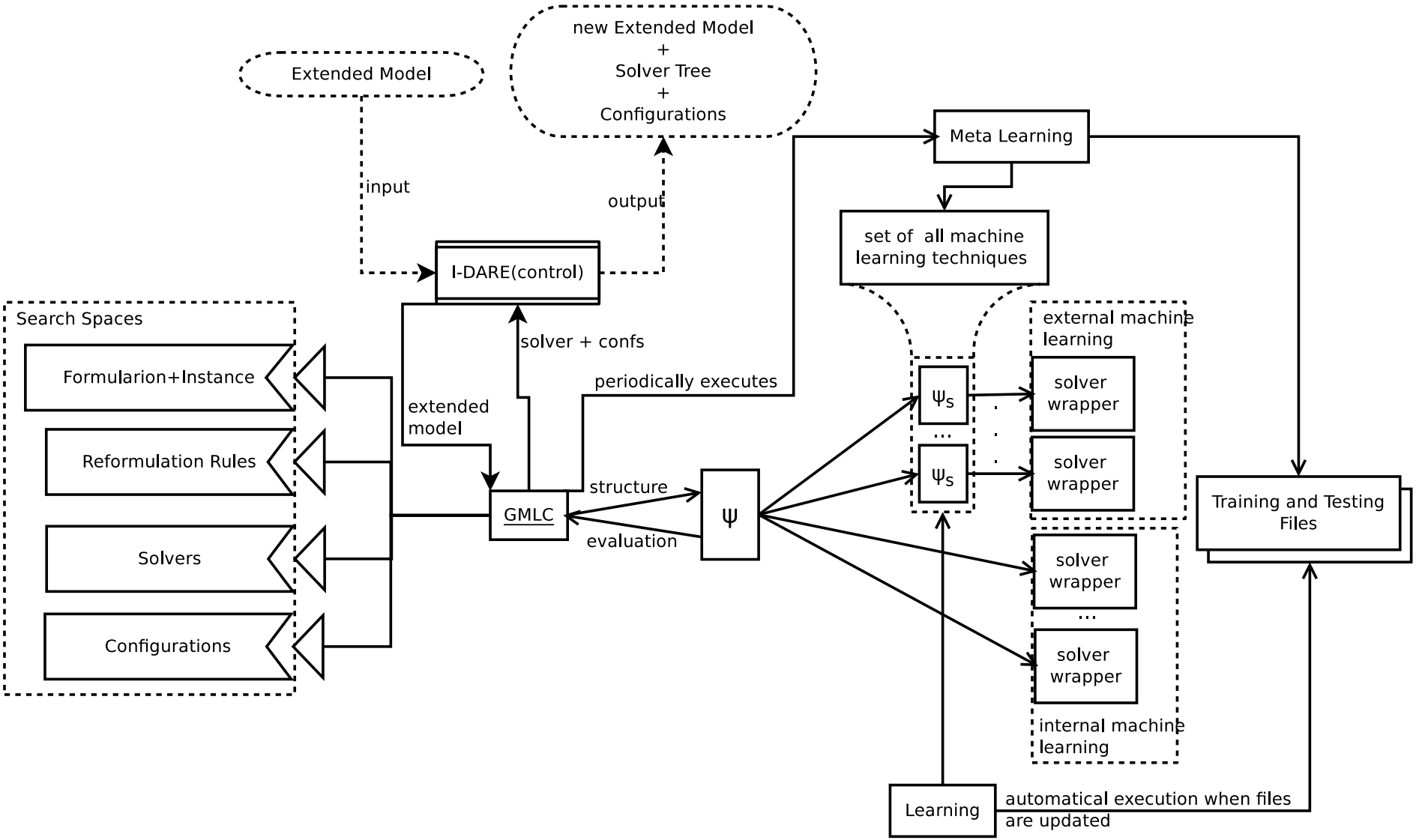
Generic ML

- Using the features and CT several data points are created.
- Different ML mechanism may be proposed (and even used simultaneously).
- A general interface is proposed to plug the MLs.

```
1  d_machineLearning [
2       evaluate(_list) => _list ,
3      => train(_list , _list)
4  ].
```

- Note that for nested structures, the SW has the possibility to access the SWs of the sub-blocks to decompose the task of prediction of ψ.
- ML will provide with sufficient tools to handle the search in the (Solver, Configuration) subspace.

The overall process



Experiments – MCF

Number of graphs: 144

Solver	nbest	b/d ratio	% best
RelaxIV	80	0.77 (0.22)	70.8
MCFSimplex	34	0.86 (0.17)	29.2

Experiments – MCF

Number of graphs: 144

Solver	nbest	b/d ratio	% best
RelaxIV	80	0.77 (0.22)	70.8
MCFSimplex	34	0.86 (0.17)	29.2

Initial set of features common to both solvers (Nodes, Arc, Degree, Costs, Paths, etc).

Solver	e. err	b/ml ratio	ml/d ratio	opt. features
RelaxIV	0.542 (0.871)	0.770 (0.205)	0.966 (0.305)	[0, 1, 4, 8, 12, 16, 24]
MCFSimplex	0.500 (0.723)	0.855 (0.182)	0.919 (0.158)	[0, 1, 18, 21, 23]

Experiments – MCF

Number of graphs: 144

Solver	nbest	b/d ratio	% best
RelaxIV	80	0.77 (0.22)	70.8
MCFSimplex	34	0.86 (0.17)	29.2

Initial set of features common to both solvers (Nodes, Arc, Degree, Costs, Paths, etc).

Solver	e. err	b/ml ratio	ml/d ratio	opt. features
RelaxIV	0.542 (0.871)	0.770 (0.205)	0.966 (0.305)	[0, 1, 4, 8, 12, 16, 24]
MCFSimplex	0.500 (0.723)	0.855 (0.182)	0.919 (0.158)	[0, 1, 18, 21, 23]

s. err	b/ml ratio	ml/dR ratio	ml/dS ratio
0.145	0.787 (0.208)	0.933 (0.342)	0.791 (0.450)

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances
- Unified solver interface

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances
- Unified solver interface
- Reformulation system based on input-output mappings.

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances
- Unified solver interface
- Reformulation system based on input-output mappings.
- A General Control mechanism

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances
- Unified solver interface
- Reformulation system based on input-output mappings.
- A General Control mechanism
  - selection of solver + configuration can be done automatically,

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances
- Unified solver interface
- Reformulation system based on input-output mappings.
- A General Control mechanism
  - selection of solver + configuration can be done automatically,
  - selection criteria may include practical constraints (like architecture, price, etc),

Conclusions

- This work defines the grounds of a system to harness a vast body of knowledge about reformulation and algorithms that are best for certain problems.
- This required defining an appropriate general concept of *structured formulation*.
- Unified way of representing the instances
- Unified solver interface
- Reformulation system based on input-output mappings.
- A General Control mechanism
  - selection of solver + configuration can be done automatically,
  - selection criteria may include practical constraints (like architecture, price, etc),
  - we may use potentially any search mechanism to move in the reformulation space

Future Work

- Augment the structure library.

Future Work

- Augment the structure library.
- Increase the amount of solvers.

Future Work

- Augment the structure library.
- Increase the amount of solvers.
- Create more ARRs.

Future Work

- Augment the structure library.
- Increase the amount of solvers.
- Create more ARRs.
- Study the effective deployment of the system, to create a dynamic market of problem solution.

Future Work

- Augment the structure library.
- Increase the amount of solvers.
- Create more ARRs.
- Study the effective deployment of the system, to create a dynamic market of problem solution.
- Study other ML techniques and their parametrization.

Future Work

- Augment the structure library.
- Increase the amount of solvers.
- Create more ARRs.
- Study the effective deployment of the system, to create a dynamic market of problem solution.
- Study other ML techniques and their parametrization.
- Application of different search strategies to deal with the reformulation space.