toulbar2
An exact cost function network solver

Simon de Givry
INRAE, Toulouse

ROADEF'23, Rennes, 02-23-2023

Context

- Constraint Satisfaction Problems which are under/over constrained
  - Valued CSP (Schiex, Fargier, Verfaillie, IJCAI'1995) add preferences between solutions.

- Open-source branch-and-bound solver
  - *ToolBar (Toulouse & Barcelona)* (C code)

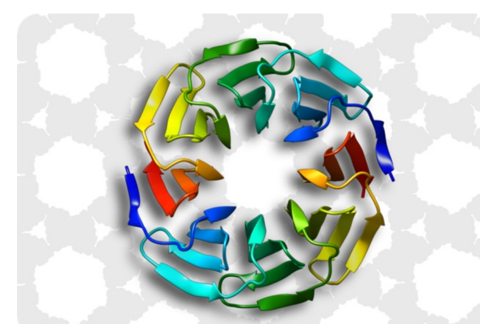
    (Schiex, Larrosa IJCAI'2003)
  - **Toulbar2** (C++ code)

    (Sanchez, Givry, Schiex, Constraints, 2008)

Collaborations

Research in optimization algorithms

Applied research in *protein design*



Optimization framework

X set of discrete variables,

Minimize $F(X)$

such that $C(X)$ is satisfied

Optimization framework

X set of discrete variables,

F set of cost functions, $f_S(X[S]) \in \mathbb{N}$

Minimize $\sum_{f_S \in F} f_S(X[S])$

such that $C(X)$ is satisfied

Optimization framework

X set of discrete variables,

F set of cost functions,

$\top$ infinite cost (forbidden assignment),

Minimize $\sum_{f_S \in F} f_S(X[S])$

Cost Function Network (CFN)

Optimization framework

X set of discrete variables,

F set of cost functions,

T infinite cost (forbidden assignment),

Minimize $\sum_{f_S \in F} f_S(X[S])$

NP-hard problem

X	f1(X)
a	0
c	1
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	1
	g	2
	t	0
c	a	1
	c	∞
	g	0
	t	2
g	a	2
	c	0
	g	∞
	t	1
t	a	0
	c	2
	g	1
	t	∞

+

Y	f3(Y)
a	0
c	1
g	1
t	1

Minimize f1(X) + f2(X,Y) + f3(Y)

Python3 interface

```
#pip install pytoulbar2

import pytoulbar2 as tb2

cfn = tb2.CFN()

cfn.AddVariable('X', ['a','c','g','t'])
cfn.AddVariable('Y', ['a','c','g','t'])

cfn.AddFunction(['X'], [0,1,1,1])
cfn.AddFunction(['Y'], [0,1,1,1])
cfn.AddFunction(['X', 'Y'], [cfn.Top,1,2,0,1,cfn.Top,0,2,2,0,cfn.Top,1,0,2,1,cfn.Top])

print(cfn.Solve(showSolutions=3))

#Output:
#New solution: 1 (0 backtracks, 0 nodes, depth 2, 0.003 seconds)
#X=a Y=t
#([0, 3], 1.0, 1)
```

Cost function in extension

X	Y	Z	f(X,Y,Z)
a	a	a	∞
	a	c	1
	a	g	2
	a	t	0
a	c	a	1
	c	c	1
	c	g	0
	c	t	2
a	g	a	2
	g	c	0
	g	g	2
	g	t	1
a	t	a	0
	t	c	2
	t	g	1
	t	t	0

… 4*4*4 = 64 assignments!

Cost function using a compact table

X1	X2	X3	X4	X5	X6	X7	f(X1,X2,X3,X4,X5, X6,X7)
g	a	t	t	a	c	a	1
Other tuples (i.e., default cost):							0

cfn.AddCompactFunction(['X1','X2','X3','X4','X5','X6','X7'], 0, [['g','a','t','t','a','c','a']], [1])

Here, it is a soft clause.

Cost functions in intention

cfn.**AddAllDifferent**(*X*), cfn.AddGlobalFunction(*X*, ***'samong'***,params), cfn.AddGlobalFunction(*X*, ***'sgcc'***,params), cfn.AddGlobalFunction(*X*, ***'sregular'***, params), cfn.AddGlobalFunction(*X*, ***'sgrammar'***,params), ...

Linear constraints

- Pseudo-Boolean linear constraint $X_i \in \{0,1\}$

$$2*X_1+3*X_2+4*X_3+5*X_4 \geqslant 10$$

```
cfn.AddLinearConstraint([2,3,4,5],['X1','X2','X3','X4'],'>=',10)
```

- Linear constraint $X_i \in \{0,1,2\}$

$$2*X_1+3*X_2+4*X_3+5*X_4 \leqslant 10$$

```
cfn.AddLinearConstraint([2,3,4,5],['X1','X2','X3','X4'],'<=',10)
```

Integer coefficients only!

Generalized linear constraint

- Hamming distance constraint

$$(X_1=2)+(X_2=3)+(X_3=4)+(X_4=5)\leqslant 2$$

cfn.AddGeneralizedLinearConstraint([(‘X1’,2,1),(‘X2’,3,1),(‘X3’,4,1),(‘X4’,5,1)], ‘<=’, 2)

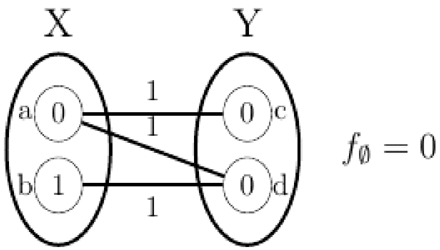
(propagation by a dedicated linear relaxation of Multiple-Choice Knapsack Problem)

- Used to translate many XCSP3 constraints

sum, count, nValues, cardinality, notAllEqual, cumulative

Toulbar2 XCSP3-core xml reader (20 constraints)

Exact solving methods



Branch-and-bound combined with equivalence preserving transformations

strenght

- FDAC (Schiex, Larrosa, IJCAI'2003)
- EDAC (Heras, Larrosa, Givry, Zytnicki, IJCAI'2005)
- VAC (Cooper, Givry, Sanchez, Schiex, Zytnicki, AAAI'2008) (Trösser, Givry, Katsirelos, CPAIOR'2020)
- VPWC (Montalbano, Givry, Katsirelos, Werner, CPAIOR'2023)
- OSAC (Cooper, Givry, Schiex, IJCAI'2007) ≈ *Linear Programming*
- EDmaxRPC (Nguyen, Bessiere, Givry, Schiex, Constraints 2017)
- *VSAC-SR* (Dlask, Werner Givry, CP'2021, Constraints 2023)

X	f1(X)
a	0
c	1
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	1
	g	2
	t	0
c	a	1
	c	∞
	g	0
	t	2
g	a	2
	c	0
	g	∞
	t	1
t	a	0
	c	2
	g	1
	t	∞

+

Y	f3(Y)
a	0
c	1
g	1
t	1

X	f1(X)
a	0
c	1
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	1
	g	2
	t	**1**
c	a	1
	c	∞
	g	0
	t	**3**
g	a	2
	c	0
	g	∞
	t	**2**
t	a	0
	c	2
	g	1
	t	∞

+

Y	f3(Y)
a	0
c	1
g	1
t	**0**

X	f1(X)
a	**1**
c	1
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	**0**
	g	**1**
	t	**0**
c	a	1
	c	∞
	g	0
	t	**3**
g	a	2
	c	0
	g	∞
	t	**2**
t	a	0
	c	2
	g	1
	t	∞

+

Y	f3(Y)
a	0
c	1
g	1
t	**0**

X	f1(X)
a	**1**
c	1
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	**0**
	g	**2**
	t	**0**
c	a	1
	c	∞
	g	**1**
	t	**3**
g	a	2
	c	0
	g	∞
	t	**2**
t	a	0
	c	2
	g	**2**
	t	∞

+

Y	f3(Y)
a	0
c	1
g	**0**
t	**0**

X	f1(X)
a	**1**
c	**2**
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	**0**
	g	**2**
	t	**0**
c	a	**0**
	c	∞
	g	**0**
	t	**2**
g	a	2
	c	0
	g	∞
	t	**2**
t	a	0
	c	2
	g	**2**
	t	∞

+

Y	f3(Y)
a	0
c	1
g	**0**
t	**0**

X	f1(X)
a	**1**
c	**2**
g	1
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	**1**
	g	**2**
	t	**0**
c	a	**0**
	c	∞
	g	**0**
	t	**2**
g	a	2
	c	**1**
	g	∞
	t	**2**
t	a	0
	c	3
	g	**2**
	t	∞

+

Y	f3(Y)
a	0
c	**0**
g	**0**
t	**0**

X	f1(X)
a	**1**
c	**2**
g	**2**
t	1

+

X	Y	f2(X,Y)
a	a	∞
	c	**1**
	g	**2**
	t	**0**
c	a	**0**
	c	∞
	g	**0**
	t	**2**
g	a	**1**
	c	**0**
	g	∞
	t	**1**
t	a	0
	c	3
	g	**2**
	t	∞

+

Y	f3(Y)
a	0
c	**0**
g	**0**
t	**0**

X	f1(X)
a	**0**
c	**1**
g	**1**
t	**0**

$+$

X	Y	f2(X,Y)
a	a	∞
	c	**1**
	g	**2**
	t	**0**
c	a	**0**
	c	∞
	g	**0**
	t	**2**
g	a	**1**
	c	**0**
	g	∞
	t	**1**
t	a	0
	c	3
	g	**2**
	t	∞

$+$

Y	f3(Y)
a	0
c	**0**
g	**0**
t	**0**

$$f_{\varnothing}=1$$

$f_{\varnothing}$ **is the current problem lower bound !**

f_{X_i} **used by value and variable heuristics**

$T = 2$

X	f(X)
a	**0**
c	**∞**
g	**∞**
t	**0**

+

X	Y	g(X,Y)
a	a	∞
	c	**∞**
	g	**∞**
	t	**0**
c	a	**0**
	c	∞
	g	**0**
	t	**∞**
g	a	**∞**
	c	**0**
	g	∞
	t	**∞**
t	a	0
	c	**∞**
	g	**∞**
	t	∞

+

Y	f(Y)
a	0
c	**0**
g	**0**
t	**0**

$f_{\varnothing} = 1$

T is decreasing at each new solution found!

More value pruning!!

$T = 2$

X	f(X)
a	**0**
c	**∞**
g	**∞**
t	**0**

+

X	Y	g(X,Y)
a	a	∞
	c	**∞**
	g	**∞**
	t	**0**
c	a	**0**
	c	∞
	g	**∞**
	t	**∞**
g	a	**∞**
	c	**∞**
	g	∞
	t	**∞**
t	a	0
	c	**∞**
	g	**∞**
	t	∞

+

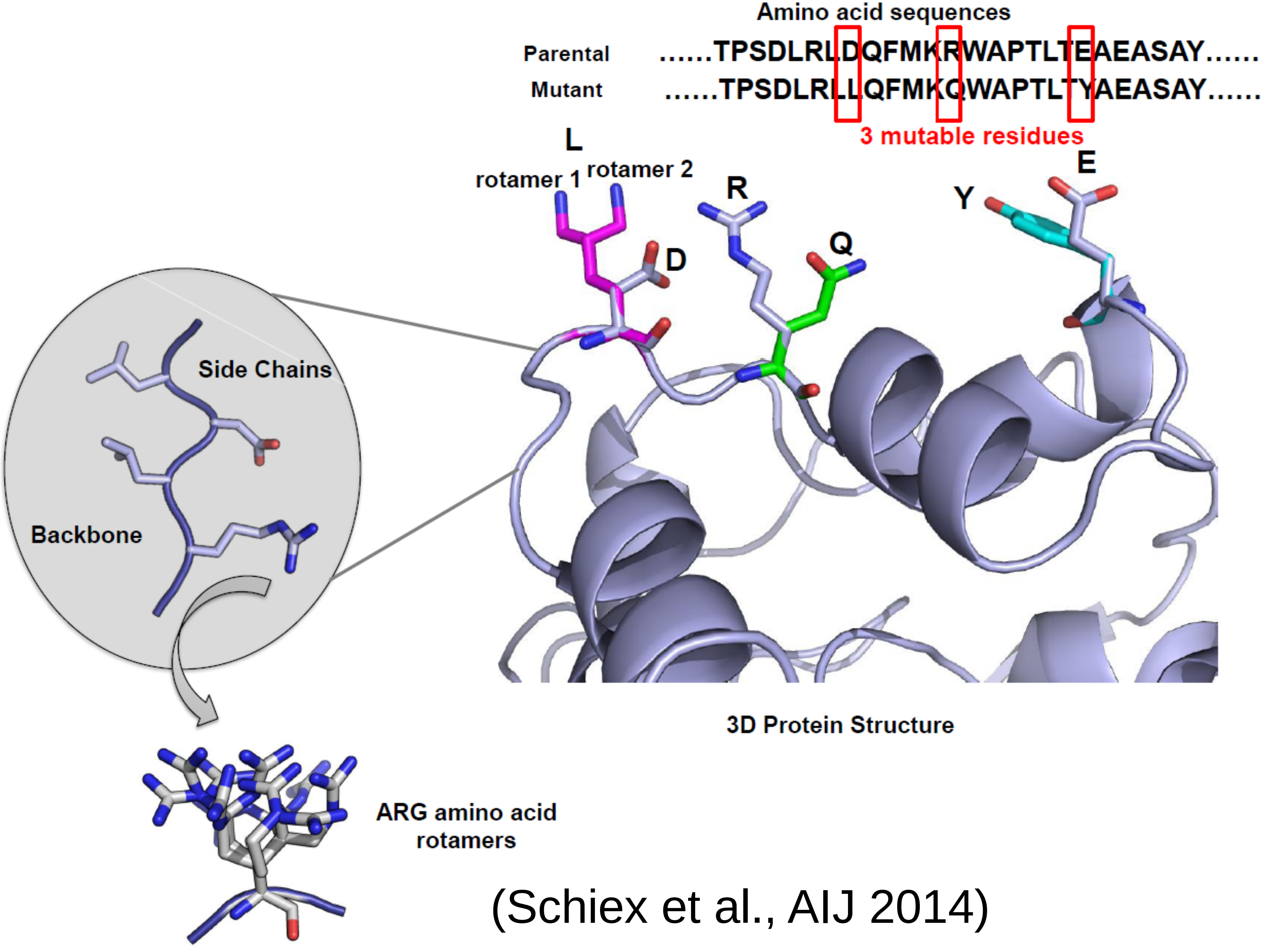
Y	f(Y)
a	0
c	**0**
g	**0**
t	**0**

$f_{\varnothing} = 1$

T is decreasing at each new solution found!

More value pruning!!

Protein design



(Schiex et al., AIJ 2014)

Protein design

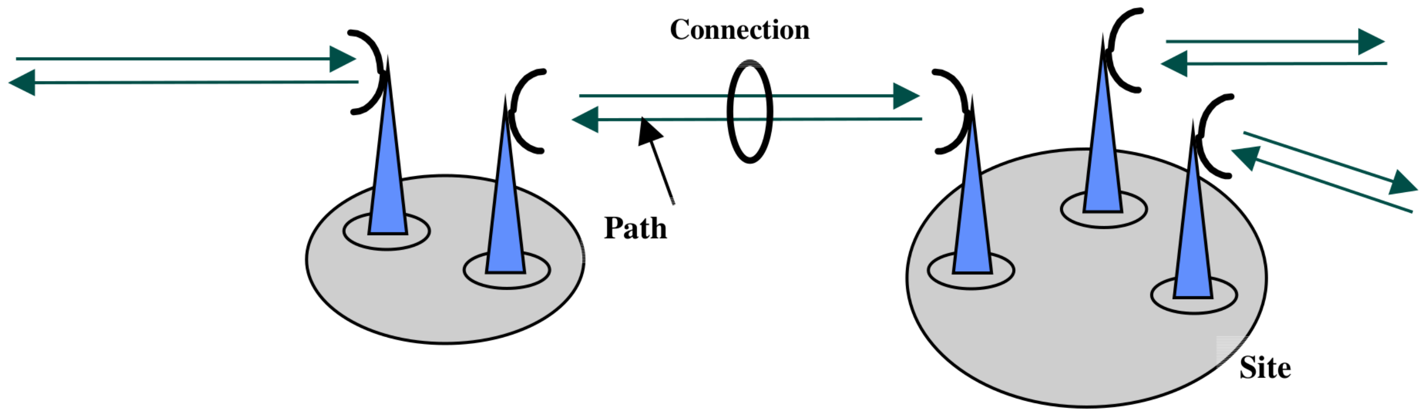
$$E = E_{\varnothing} + \sum_i E(i_r) + \sum_i \sum_{j>i} E(i_r, j_s)$$

	CPLEX 20.1	toulbar2 1.1.1
1BK2.matrix.24p.17aa	42.84 sec (0 node) Reduced MIP has 16,090 rows 284,424 columns 584,143 nonzeros 805 binaries	0.37 sec (38 nodes) |X|=24 max(|D|)=182 |F|=300

(Intel Xeon 2.5GHz with 256GB RAM)

Frequency Assignment Problem with Polarization

Challenge ROADEF 2001



The problem is to assign a pair (frequency f, polarization p) to every path.

$$1 \leq f \leq \sim 151 \quad p \in \{-1, 1\}$$

Constraints are:

(I) two paths must use equal or different frequencies
(II) the absolute difference between two frequencies should exactly be equal or different to a given number
(III) two paths must use equal or different polarizations
(IV) the absolute difference between two frequencies should be greater at a relaxation level *l* (0 to 10) than a given number *g_l* (resp. *d_l*) if polarization are equal (resp. different, usually *g_l* > *d_l*)

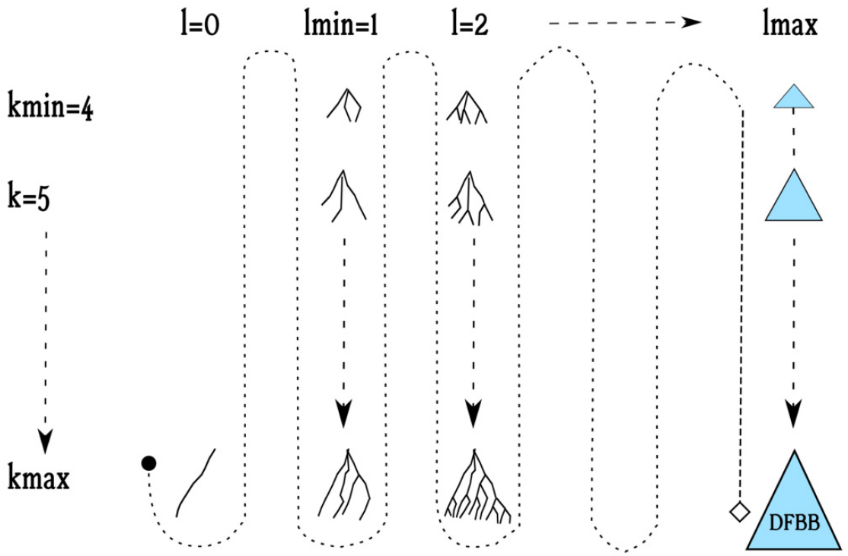
Find a feasible assignment with the smallest relaxation level *l*
and which minimizes the number of violations of (IV) at lower levels

Frequency Assignment Problem with Polarization

- Combine frequency with polarization into a single variable
  - Better propagation (only binary cost functions)

- Use Parallel Variable Neighborhood Search (Ouali et al., AIJ 2020)

- See our CP’2020 tutorial



Frequency Assignment Problem with Polarization

Small instances	k	ROADEF'2001 Best Results	toulbar2	(time to best solution)
fapp01_0200	4	35,758,830	**35,758,826**	(1.333s)
fapp02_0250	2	40,370,567	**40,370,566**	(16.689s)
fapp03_0300	7	294,380,136	**294,380,135**	(26.155s)
fapp04_0300 *|X|=300* *max(|D|)=270* *|F|=2100*	1	24,648,190	**24,616,970**	(3.282s) *(optimality proof in 27 seconds!)*
fapp05_0350 *|X|=350* *max(|D|)=270* *|F|=2839*	11	**521,348,004**	521,348,429	(53.496s) *(521,348,004 in 89.788 seconds)*
test01_0150	4	18,292,591	**18,292,585**	(1.691s)

ROADEF competitors: 1 hour on Intel Pentium III 500MHz 128MB
Toulbar2: 1 minute on 21 cores of Intel Xeon 2.5GHz 256GB

Grid operation-based outage maintenance planning

Challenge ROADEF 2020

Find an optimal planning regarding a risk-based objective

$$18 \leqslant |I| \leqslant 706$$

Planning horizon within a year (day or week discretization)

$$53 \leqslant H \leqslant 365$$

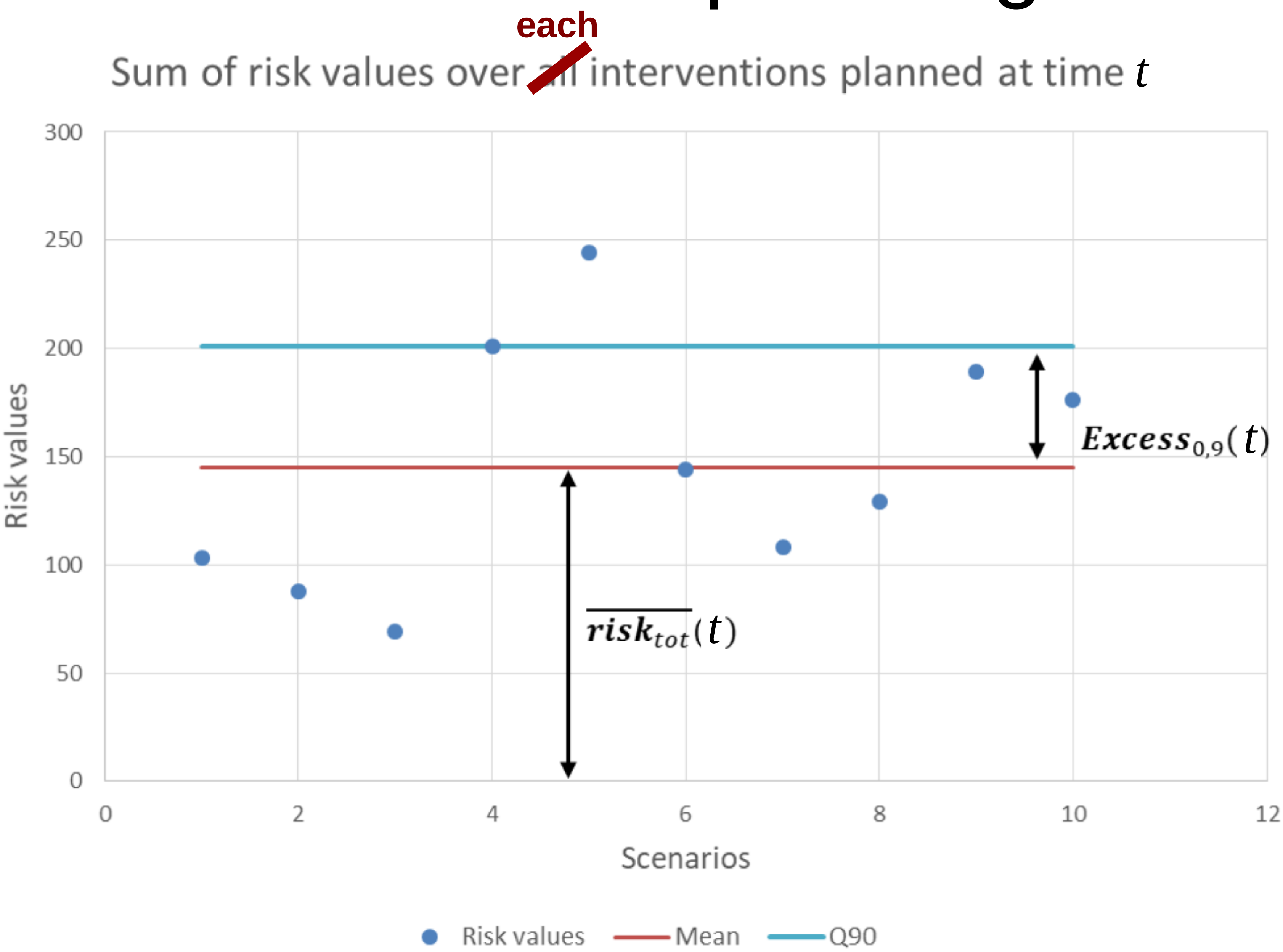
Grid operation-based outage maintenance planning

CFN model

- $|I|$ variables (start time of each intervention)
  with domain size H

- Two approximations (without solution quality guarantee)
  - Resource constraint (using generalized linear constraints)
    - Integer approximation of continuous weights (r multiplier)
  - Risk-based objective decomposed into $|I|$ cost functions
    for every intervention

Grid operation-based outage maintenance planning



Grid operation-based outage maintenance planning

Instance	Best Results 15 min	toulbar2 15min *r=10*	Best Results 90 min	toulbar2 1hour *r=100*
A_01(|I|=181,H=90)	1,767.82	1,770.15	1,767.82	1,769.50
A_02(|I|=89,H=90)	4,671.38	4,732.05 *(9.870s)*	4,671.38	4,732.02 *(9.132s)*
A_03(|I|=91,H=90)	848.18	848.18 *(5.060s)*	848.18	848.18 *(4.325s)*
A_04(|I|=706,H=365)	2,085.88	-	2,085.88	-
A_05(|I|=180,H=182)	635.22	651.49 *(50.63s)*	635.22	651.24 *(49.59s)*
A_06(|I|=180,H=182)	590.62	-	590.62	640.69
A_07(|I|=36,H=17)	2,272.78	2,280.16 *(0.103s)*	2,272.78	-
A_08(|I|=18,H=17)	744.29	750.03 *(0.269s)*	744.29	750.03 *(0.252s)*
A_09(|I|=18,H=17)	1,507.28	1,507.32 *(0.063s)*	1,507.28	1,507.32 *(0.061s)*
A_10(|I|=108,H=53)	2,994.85	3,022.76 *(1.452s)*	2,994.85	3,022.76 *(1.623s)*
A_11(|I|=54,H=53)	495.26	504.95 *(2.544s)*	495.26	504.95 *(2.452s)*
A_12(|I|=54,H=53)	789.63	828.88 *(0.668s)*	789.63	793.04 *(1.289s)*
A_13(|I|=179,H=90)	1,998.66	2,009.79	1,998.66	2,009.66
A_14(|I|=108,H=53)	2,264.12	2,302.56 *(90.13s)*	2,264.12	2,295.65 *(1,215s)*
A_15(|I|=108,H=53)	2,268.57	2,310.25 *(113.4s)*	2,268.57	2,305.29 *(1,036s)*
Distance to best results (except A04,A06,A07):		*1.39%*		*0.96%*

XCSP 2022 Competition Results

solver	score	optimum	best bound
Mistral	**93.00**	34	**99**
toulbar2	86.00	**51**	87
miniRBO	74.50	41	78
Sat4j-both	58.50	39	60
Sat4j-rs	43.00	33	46
Glasgow	31.50	21	34

Mini COP Track (158 instances)

(subset of constraints: allDifferent, Extension, Intension, sum, element)

Score computation:
1 pt. (optimum found or single best solution), 0.5pt. (best sol. Exaquo), else 0 pt.
Time limit: 40 min. on single core 3.5GHz ; space limit: 64 GB.

XCSP 2022 Competition Results

solver	score	optimum	best bound
Picat	**137.00**	**121**	139
CoSoCo	122.00	66	**144**
Mistral	95.50	51	118
toulbar2	91.00	73	100
RBO	89.50	55	104
Choco	83.00	59	97
Sat4j-both	56.00	47	62
Sat4j-rs	48.00	38	57

Full COP Track (250 instances)

solver	score	optimum	best bound
Choco	**184.50**	71	**191**
toulbar2	118.50	**80**	119

Parallel COP Track (250 instances, using 4 cores)

Toulbar2 platform

https://forgemia.inra.fr/thomas.schiex/cost-function-library (+18.281 instances)

http://genoweb.toulouse.inra.fr/~degivry/evalgm (3.026 instances)

https://miat.inrae.fr/toulbar2/ (main site)
https://toulbar2.github.io/toulbar2 (documentation, tutorials, examples)

Debian and ubuntu packages v1.1.1

Github latest source and release v1.2.0 (linux, macos, win exe)
Q&A User discussions

Python interface v1.2.0 (pip install pytoulbar2)

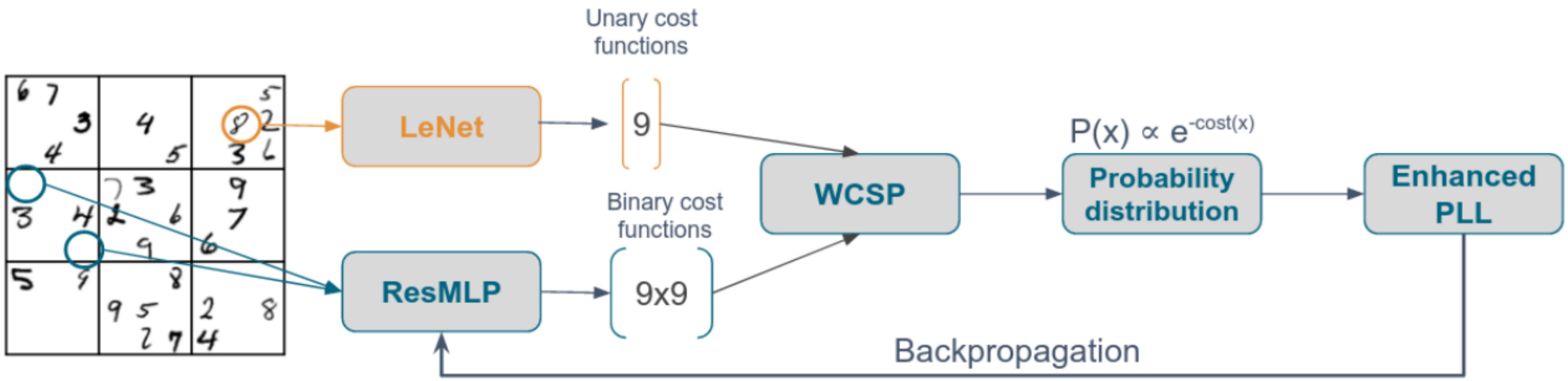
toulbar2 read many input formats: cfn, wcsp, wcnf, uai, qpbo, opb, xml (xcsp3)

Current work on **bilevel discrete optimization** (see another ROADEF'23 talk), **multicriteria**, and CFN **learning** from data [Defresne et al, AAAI-23 workshop]

Application on natural-input problems

Visual sudoku

> Learn to play Sudoku & to recognize digit



- Solver able to correct digit mis-classification

SATNet	Theoretical (no corrections)	Ours
63.2 %	74.2%	**94.1** $\pm$ 0.8%

[Defresne et al, AAAI-23 CPML workshop]