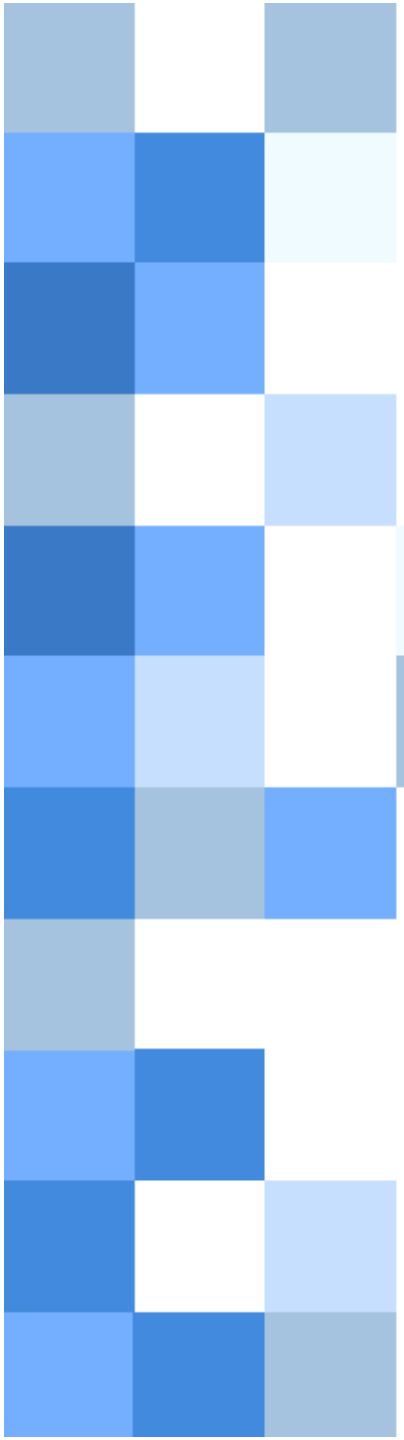


CP Next Challenge: Simplicity of Use

Jean-François Puget
ILOG

Motivation

An industry (aka evil) stand point

- **We monitor how our CP tools are used by our customers in industry**
  - **Many customers achieve very good performance with CP**
  - **But they find it difficult to learn, use and maintain**
  - **CP academic research offers little help here**
- **Math programming (MP) community seems more interested in usability**
  - **SAT community as well**

Overview

- Compare CP with MP
- CP is too complex for engineers in industry
  - How serious is this?
- Is CP academic research addressing the problem?
  - No, it even makes it worse!
- What can we do about it?
  - Interesting new research topics
- Conclusions

ILOG optimization products

Same R&D team (I am their boss )

- **Constraint Programming (CP) products**
  - **ILOG Solver and specialized extensions**
  - **ILOG OPL Studio**
- **Math Programming (MP) products**
  - **ILOG CPLEX**
  - **ILOG OPL Studio**
- **It is "easy" to compare them and learn from their differences**

MP CP comparison

Not that easy! Cultural difference

MP		CP
Programming • Planning	≠	**Programming** • Computer programming
Solution • Solution • Feasible solution	≠	**Solution** • Optimal solution • Solution
Decision variables	≠	**Variables**
X >=0 by default • No upper bound is OK	≠	**X in [a, b]** • a can be < 0 • a,b must be finite
Almost integer is OK • Floating point computations	≠	**Integrality is strict** • Integer computations

MP CP comparison

Many common points

- **A problem is described by**
  - **Variables**
  - **Constraints (linear for MP, general for CP)**
  - **An objective function**
- **Models look similar**
- **Tree search/ branch and bound for both**
  - **Inference at each node**
    - Bound strengthening / Constraint propagation
    - LP relaxation / Global constraints

Many hybrid combinations

- **Use LP in global constraints for CP**
- **Use MP on one part, CP on the other, in sequential order**
- **Use MP on one part, CP on another, concurrently**
  - **Dantzig Wolfe decomposition**
    - MP on the master problem
    - CP for generating variables (columns)
- **Linearize CP constraints**

What precedes is not relevant!

An industry stand point

- **What matters is the usefulness of the technology**
  - **Is CP (or MP) useful for my problem?**
  - **How can I assess that quickly?**
  - **Is a CP (or MP) code easy to tune, to maintain?**
  - **Is a CP (or MP) code robust ?**
    - Does performance depend too much on data?
- **When faced with these questions**
  - **CP doesn't provide good answers**
  - **MP seems much more appealing**
- **CP performance is not the issue per se**
  - **CP has numerous successes in industry**

Example 1

- **A big software company in SCM**
  - **They use ILOG Solver and ILOG Scheduler in their scheduling package**
  - **This is successful**
    - Performance is great
- **They will not launch new CP projects**
  - **CP is difficult to maintain**

Example 1 continued

True story

- **They want transfer maintenance to an India team trained 1 week on CP**
  - **Indian people are smart and cheap**
- **First question from India:**
  - **Why do you write in your code**
    IlcOr(x==a, x!=a)
    since this is a tautology?

- **This is a non deterministic choice point…**

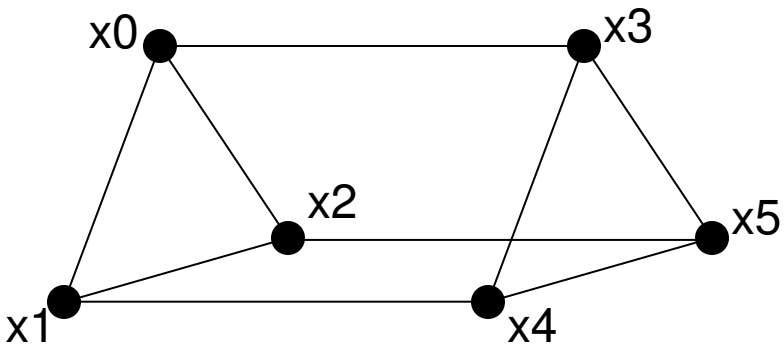
Example 2 : from another customer

True story

- What solution does the system outputs when there is no solution?
  - None
- I want a solution!
  - Nope
- Explain me why there is no solution!

Example 3: graceful graph

Graph coloring problem



m vertices and n edges. Color vertices and edges with numbers from 0 to n

- Colors of the vertices must be all different
  - all different (x1, x2, …, xm)
- Other constraints on edges
  - Color of edge (i, j) is abs(xi – xj)
  - Colors of edges must be all different

Example 3

Tighten the representation

- All different (abs(xi - xj)) for all edges (i,j)
- xi and xj are different, therefore

  abs(xi – xj) > 0
- Stating this makes the all different constraint tight
  - Much better propagation of the global constraint
  - Running times improve a lot
- This is straightforward, isn't it?

Example 4 : coin design problem

From [Wallace 03]

- **Select coin values so that any change can be given with a minimum amount of coins**
- **Mark said that a CP solution is easy to produce!**

- **Let's see if you're as clever as Mark…**

A model

Using OPL

```
range coin 1..6;
range change 1..100;
var int value[coin] in 1..100;
var int num[coin] in 1..100;
var int sel[change, coin] in 0..100;
minimize sum(i in coin) num[i] subject to {
   forall(i in coin, s in change)
      sel[s,i] <= num[i];
   forall(s in change)
      sum(i in coin) value[i]*sel[s,i] = s;
};
```

It runs forever !

To make it work

Write search!

```
search{
    generate(value);
    generate (num);
    forall(s in change) generate(sel[s]);
};
```

Straightforward isn't it?

It runs forever !

To make it work (continued)

The problem has symmetries

```
forall(i in coin: i <> 6) {
        value[i]*num[i] < value[i+1];
   };
```

It runs in .45 seconds on this laptop!

See, CP is really good on this problem!

MP for dummies (engineers)

Modeling is what matters

1. **Model your problem with**
   - **Decision variables (integer and continuous)**
   - **Linear constraints**
   - **Linear objective**
2. **Run**

It may output solutions without any tuning

MP for dummies (engineers)

Modeling can be tricky

- It may run forever
- In such case, change the model
  - Tighten it
  - Use Dantzig Wolfe decomposition
- This is sufficient for the vast majority of MP applications
  - There is no need to understand MP algorithms
- Difficult problems are still difficult
  - For these, one can use advanced techniques such as branch and price, specialized cut generation, etc.
  - This is not for dummies.

CP for dummies

Is modeling what matters?

1. **Model your problem with**
   - **Decision variables (integer or continuous)**
   - **Constraints**
   - **Some objective if you really insist**
2. **Run**

It runs forever !

Modeling can be tricky

- **Your model is not right**
  - **Use that fancy global constraint**
    - Yes, I know it is logically equivalent to the constraints you already stated
  - **Why didn't you wrote a specialized propagator for you problem?**
    - Why didn't you write a specialized CP language?
  - **You should use a dual model and channeling constraints**
    - Which dual model?
    - Try any you can think of!
- **These advices assume the user understands what constraint propagation can do and what it can't do**

CP for dummies

Modeling can be tricky, the rest too!

- **You're not using the software as documented!**
  - **You need to write a search code**
    - Non deterministic
    - Recursive
    - Side effect free
  - **Exactly the opposite of what you're used to code**
    - Deterministic
    - Iterative
    - With side effects
- **This advice assumes the user is a clever computer scientist**

CP for dummies

Modeling can be tricky, the rest too!

- **Your problem has symmetries!**
  - **State generators of the symmetry group then call GAP-XXX symmetry breaking constraint technique**
  - **State lex constraints**
    - Beware, they must use a compatible ordering
- **Your objective function does not propagate!**
  - **Replace sum by max**
    - It does not fit your business needs? Change your business!
  - **Change the search strategy so that good solutions are generated first**
    - I can't tell you how to

CP for dummies

Modeling can be tricky, the rest too!

- **Use dominance constraints (or conditional symmetry breaking)**
- **Use a hybrid approach with one of the following:**
  - **LP solver**
  - **Local search method**
  - **Both**
- **Try**
  - **Russian doll search,**
  - **LDS (ouch, this is patented)**
  - **Large Neighborhood Search**
- **…**

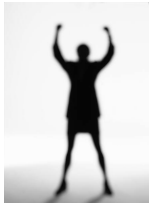
CP vs MP for dummies

- In order to be able to use CP, one must master modeling
  - **Same for MP.**
  - **There are many more possibilities with CP**
    - Too many of them
- One must also master
  - **Search**
  - **Constraint propagation**
- MP is useable even if you do not understand the internal algorithms

CP for dummies

Oxymoron

- CP is designed for clever users
  - Academics
  - Authors of CP systems
- Clever users that master all the complexity can achieve very good results
  - Our consultants are great
  - Academics are great



- Beginners cannot
  - However, they can achieve something with MP (CPLEX).

MP at work

- Modeling is what matters
- Modeling can be done independently from solvers
  - Several modeling languages: AMPL, GAMS, OPL, MOSEL, MPL, AIMMS
  - A standard file format: MPS
- This is good because business users not tied to a particular vendor
- Common set of benchmarks

MP research

- **Oriented towards algorithms**
  - **Input : an MPS file**
  - **Output : a solution + duality information (gap, reduced cost for LP)**
- **Improvements do not require new modeling features**
  - **CPLEX speedup is 1,000,000 in 10 years**
    - 1,000 comes from hardware
    - 1,000 comes form software

CP research

- **CP is oriented towards toolkits**
  - **One has to combine various pieces when trying to solve a problem**
    - See the famous CP for dummies series!
- **CP improvements are packaged into new modeling or search features**
  - **Global constraints**
  - **Search abstractions**
  - **Symmetry breaking techniques**

Example: symmetry breaking

Different approaches for similar issue

- **In MP (Margot)**
  - **Use graph automorphism software (NAUTY) to automatically compute the symmetries of the problem**
- **In SAT (Aloul et al)**
  - **Use graph automorphism software (SAUCY) to automatically compute the symmetries of the problem**

- **In CP**
  - **Symmetries are assumed to be given as input**
  - **(Mc Donald 03)(Kelsey et al 03) : Use new modeling constructs to express symmetries !**

CP academic research is wrong

From an industry stand point

- The more successful CP research, the richer CP becomes
  - Selecting the right set of CP constructs for a given problem is becoming harder an harder
  - This makes the life of engineers worse and worse!
  - Granted, it also expands the set of problems solvable with CP. This is not relevant for engineers.
- The more successful CP academic research is, the less usable CP is !

What can we do?

- Fire academics?
  - No, see later why!
- Let's not bother with engineers?
  - No, because industrial successes motivates funding agencies
- Learn from MP community
  - Yes!
  - Note that we can learn from SAT community as well

Learn from MP

Modeling is what matters most

- **Standard for expressing CP models**
  - **Standard file format**
- **Existing CP books are about**
  - **CP algorithms**
    - Propagation
    - Search
  - **CP language design**
- **They are useful for CP system design**
  - **Not for using a CP system to solve a given problem**
- **No book on modeling per se**
  - **MP has some books [Williams]**

Learn from MP

Think algorithms

- **Create executables**
  - **Input is a problem, output is a solution**
  - **Running time is what matters!**
    - Not node count, nor number of constraints checks
- **Set up challenges at CP conferences**
  - **Entries are executables that takes as input problem instances, and try to solve them**
  - **Similar challenges exits in combinatorial optimization (DIMACS), data mining (KDD), model checking, etc.**

Learn from MP

Think about optimization

- CP is not geared towards optimization
  - Almost no cost aware global constraints
  - Search goals do not use the objective function
  - I am not sure preferences and soft constraints are the right answer
- CP conference is never co located with an OR conference…
  - CPAIOR conference is a better place for optimization
- Add optimization problems to CSPLIB

Learn from MP

Search code should come for free

- Improve “out of the box” performance
- Move away from DFS
  - CPLEX is DFS most of the time, but not always
- Randomization
  - Random restarts
  - Randomize variable and value selection

Learn from MP

Search code should come for free

- **Learn during search**
  - **No goods, symmetry breaking**
  - **Learn which decisions have an impact on search tree size [Refalo 04]**
- **Develop generic combination of local search and propagation**
  - **[Perron 04]**
- **It does not matter if these techniques are not as good as ad hoc CP codes!**

Learn from MP

Reformulate before search

- Called “presolve” in MP
- For instance
  - From X ≠ Y, Y ≠ Z, Z ≠ X add all_different(X,Y,Z)
- The point is that CP improvements should not require model reformulation by users
  - We can’ say that CP is 1,000,000 faster than 10 years ago
  - Although this is probably true
- The speedup obtained with reformulation should not be offset by the time needed by presolve!
  - Running time is what matters!

Explanations

- CPLEX provides for sensitivity analysis and explanations
- Explanations are important when the problem is over constrained

Conclusion

- CP can solve complex problems, with good performance
- BUT, CP is difficult to learn use and maintain for engineers
  - Not because they are dumb
- CP academic research is making this worse every year
- MP provides good out of the box performance
  - SAT too

What academics could do

Some of this is already happening

- CP academic research should look at new topics
  - Standard file formats / modeling languages
  - Search
  - Explanations
  - Optimization
  - Modeling practice, Books
  - Presolve
  - Challenges
- Running time is what matters
  - Not node count

If we (CP community) don't do this…

- CP will stay …
  - … as an academic research topic
  - … embedded in industry packages
- Generic CP systems will disappear
  - Not ILOG
- Funding will disappear
- Academics will disappear