Constraint Programming Methodology

Helmut Simonis

email: helmut.simonis@insight-centre.org
homepage: http://http://insight-centre.org/

Insight SFI Centre for Data Analytics
School of Computer Science and Information Technology
University College Cork
Ireland

ACP Winterschool 2024

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 at Insight the SFI Research Centre for Data Analytics at UCC, which is co-funded under the European Regional Development Fund.

A version of this material was developed as part of the ECLiPSe ELearning course: https://eclipseclp.org/ELearning/index.html. Support from Cisco Systems and the Silicon Valley Community Foundation is gratefully acknowledged.

What we want to introduce

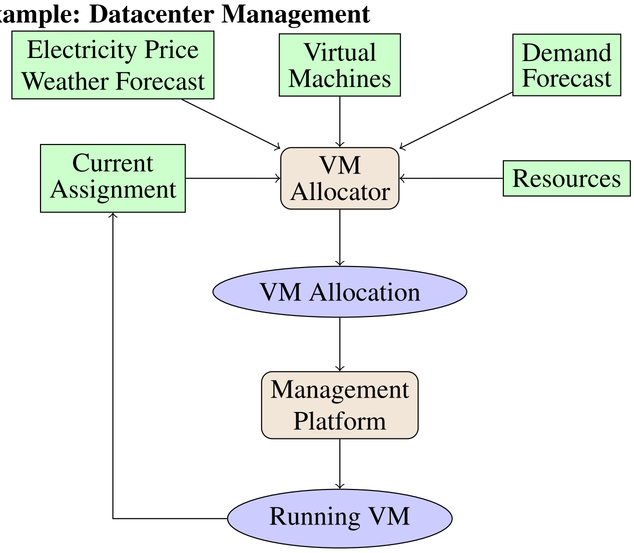
- How does a Constraint Model fit into a bigger system
- Interaction with stakeholders
- You define what the problem is
- 12 Steps to success

1 The Bigger Picture

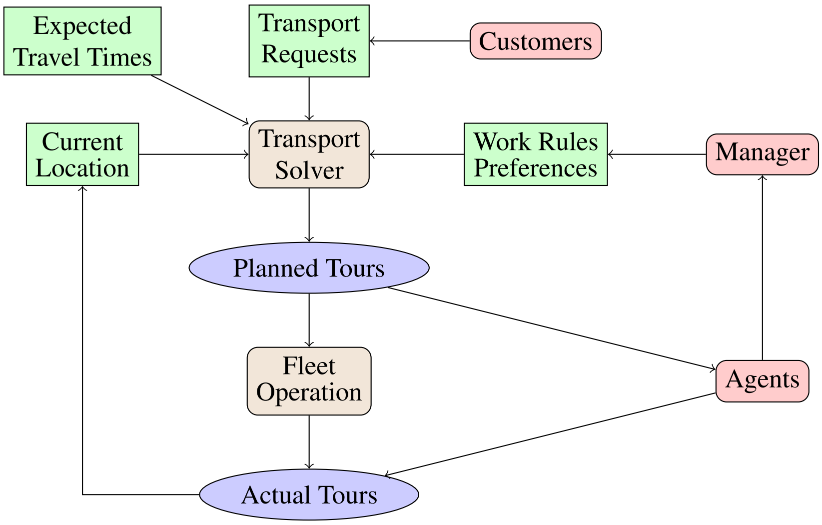
CP Model is Part of a Larger System

- Where do data come from?
- Control over data
- Generated plan
- Externalized representation of constraints
- Implementation of plan
- Feedback from previous runs

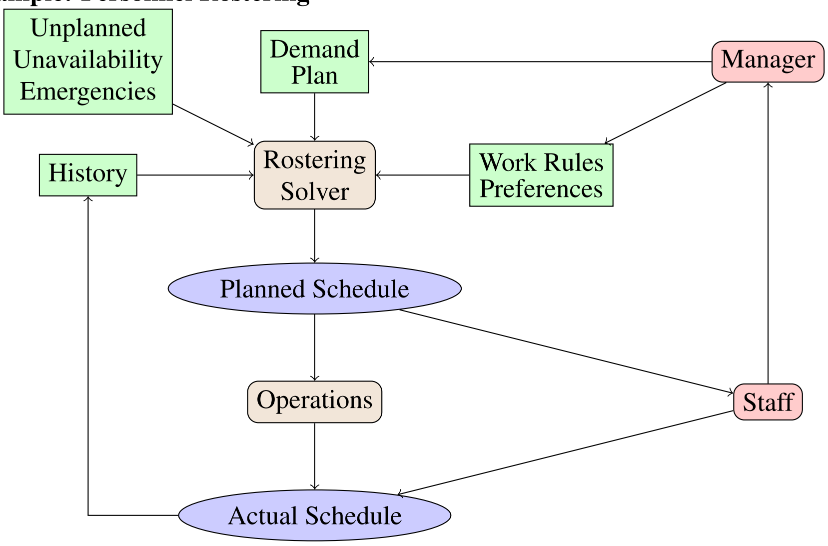
Example: Datacenter Management



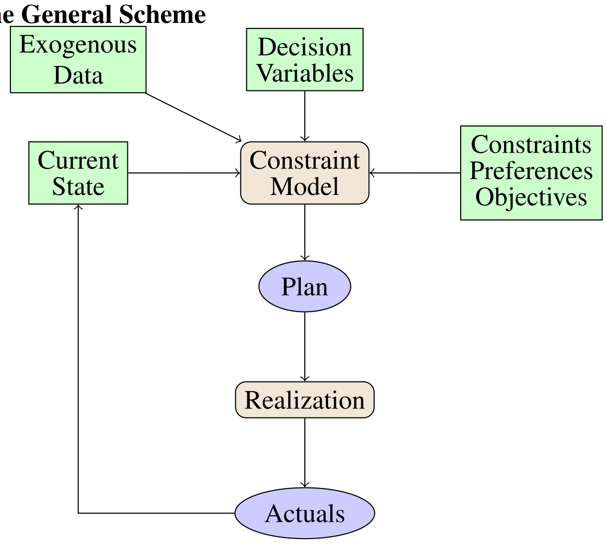
Example: Transport



Example: Personnel Rostering



The General Scheme



Key Questions: How? Who? When? Where?

- (How is work done?)
- Who performs work?
- When is it performed?
- (Where is it performed?)

2 12 Steps to Success

Step 1: Literature Review

- Which problem are other people solving?
- What are the favourite tecniques, why?
- What tools are used?
- Learn the domain specific language

Step 2: Description of Problem

- Clearly defined use cases, ranked by importance
- Textual description of problem before math
- Important: Involve all stakeholders
- Important: Identify champions and their benefits
- State what is outside the scope

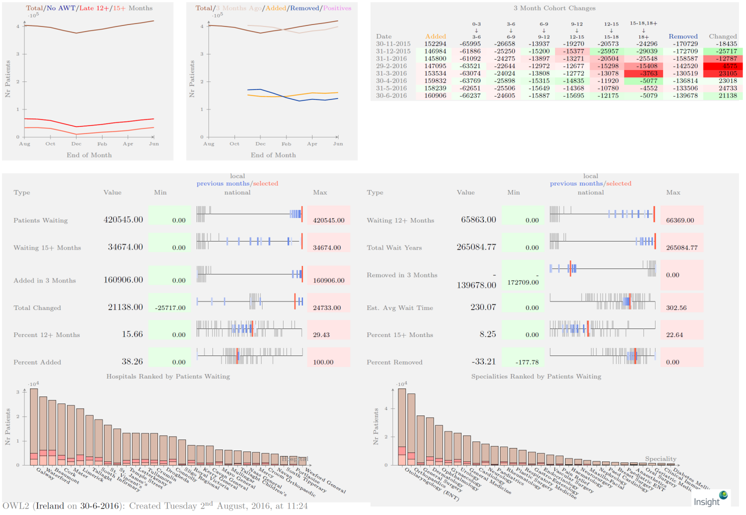
Step 3: Example Data and Solution

- Best is years of existing data and solutions
- Important: Set with manageable problem size
- Independent checker of solutions
- Possible: Trivial problem example for manual exploration

Step 4: Visualizing Results

- Develop before solver
- Visualize existing solutions
- Perhaps end-users have existing visualizations
- Generic views for classes of problems (global constraints)
- Helps to understand poor performance
- Also: Compute KPI (Key Performance Indicators)

Example Dashboard: Patient Waitlists



Step 5: Implement Core Model

- One constraint type at a time
- Start with finding feasible, good solutions
- Find lower/upper bounds to estimate solution quality
- Start with basic search strategy

Step 6: Feedback from Domain Experts

- Are you solving the right problem?
- Are all stakeholders happy with solution and objectives?

Step 7: Study Obvious Alternatives

- If you have time (PhD students have time)
- Is there a straight-forward MIP/SAT model of problem?
- Do they work on small scale, large scale problems?
- Can you come up with good, feasible heuristics?

Step 8: Solve Integration Issues

- Not too early, time wasted if solver does not work
- Not too late, without integration solver is just a demonstrator

Step 9: Performance Engineering

- Problem specific search strategies
- Parallelism
- Improved model
- Improved solver
- Parameter tuning

Step 10: Fight Feature Creep

- If it works, then people want more
- Not in first release
- Implement core use case, nothing more
- Exception: Allow all constraints to be optional

Step 11: Improve Stability

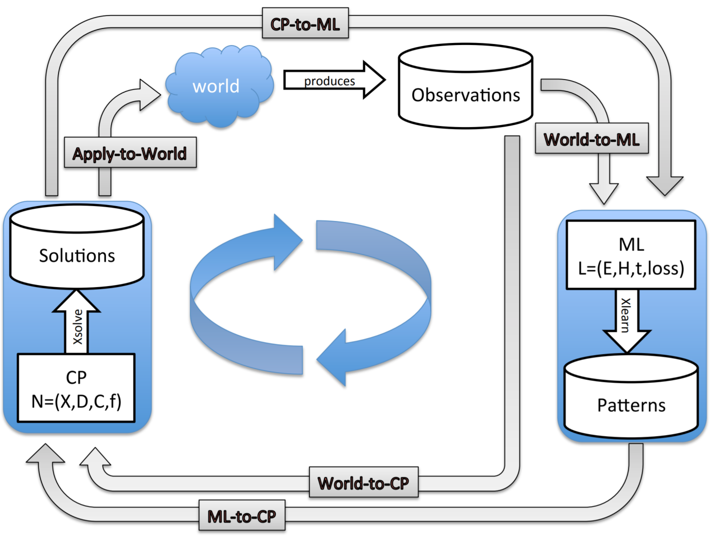
- What is really required?
- Remove experimental code
- How far can we push model?
- Code review against model description
  - Allow changes in description

Step 12: Tell the World

- Dozens of good CP applications hidden from view
- Consider writing application paper
  - Involve end-users and stakeholders
  - Describe problem from their perspective
  - If possible, publish data set and constraint description
- Submit instances to solver competitions
  - Other people will work on improved performance of your model

3 Constraints in an Uncertain World

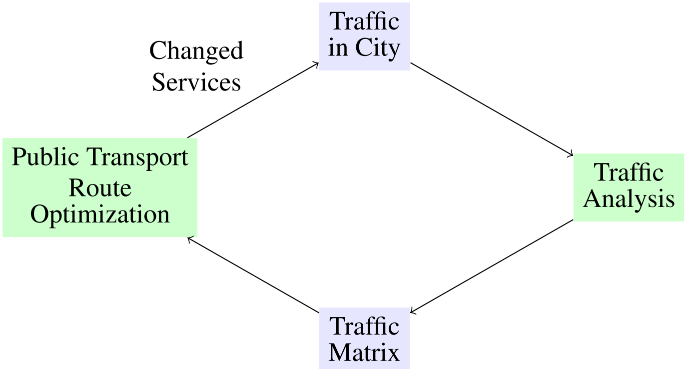
The ICON Loop



A Blueprint for Interaction

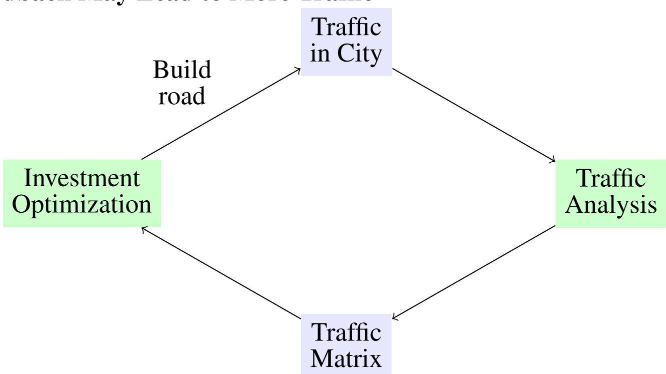
- Developed in the European ICON project
- Partners KU Leuven, Montpellier, Pisa, UCC
- Ways of combining Machine Learning with CP

Example: Intra-City Transport



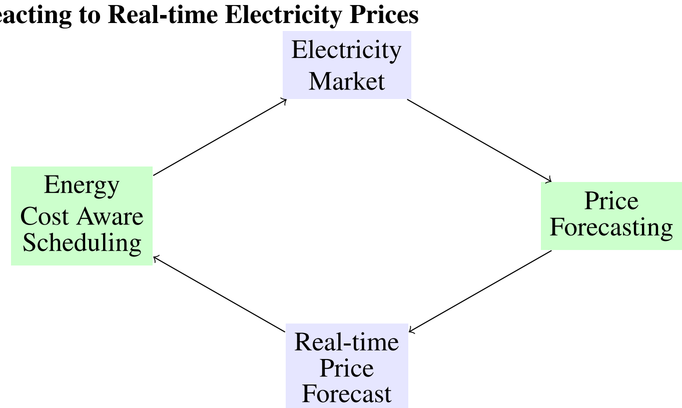
- Optimization only as good as data feeding into it

Feedback May Lead to More Traffic



- The “world” reacts to changes
- That may be difficult to predict

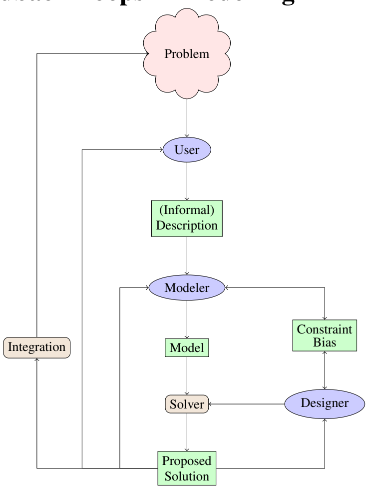
Reacting to Real-time Electricity Prices



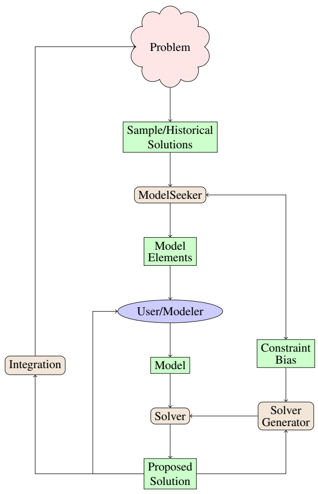
- Good way to optimize cost individually
- May lead to oscillation if everybody does it

4 There is no "The Model"

Feedback Loops in Modelling



The Future: Automated Modelling



5 Conclusion

Points to Remember

- A CP application is part of a larger system
- CP Model is rarely cause of project failure
  - No clear champion
  - No clear use case
  - Data not available/data quality
- Every problem is different, you decide what to model
- Understand the interaction between tools and problem