Modelling for Constraint Programming

Barbara Smith

4. Combining Viewpoints, Modelling Advice
Which Viewpoint to Choose?

- Sometimes one viewpoint is clearly better, e.g. if we can’t express the constraints in one.
- But different perspectives often allow different expression of the constraints and different implied constraints.
  - can be hard to decide which is better.
- We don’t need to choose one viewpoint – we can use two (or more) at once.
- We need *channelling constraints* to link the variables.
Combining Viewpoints: Permutation Problems

- Dual viewpoints of a permutation problem with variables $x_1, x_2, \ldots, x_n$ and $d_1, d_2, \ldots, d_n$
- Combine them using the channelling constraints $(x_i = j) \equiv (d_j = i)$
- The channelling constraints are sufficient to ensure that $x_1, x_2, \ldots, x_n$ are all different
  - but might be beneficial to have a specific allDifferent constraint as well and enforce GAC
Combining Viewpoints: Integer & Set Variables

- In a nurse rostering problem, we can allocate shifts to nurses or nurses to shifts
- First viewpoint:
  - an integer variable $n_{ij}$ for each nurse $i$ and day $j$
  - its value is the shift that nurse $i$ works on day $j$
- Second viewpoint:
  - a set variable $S_{kj}$ for each shift $k$ and day $j$
  - its value is the set of nurses that work shift $k$ on day $j$
- Channelling constraints: $(n_{ij} = k) \equiv (i \in S_{kj})$
- Constraints on nurse availability are stated in the first viewpoint; constraints on work requirements in the 2nd (e.g. no. of nurses required for each shift)
The Golomb Ruler Problem

- A Golomb ruler with \( m \) marks consists of:
  - a set of \( m \) integers \( 0 = x_1 < x_2 < \ldots < x_m \)
  - the \( m (m - 1)/2 \) differences \( x_j - x_i \) are all different
  - Objective: find a ruler with minimum length \( x_m \)

- First viewpoint: variables \( x_1, x_2, \ldots, x_m \)
  - \( x_j - x_i \neq x_l - x_k \) for all distinct pairs
  - \( x_1 < x_2 < \ldots < x_m \)

- Second viewpoint: variables \( d_{ij}, 1 \leq i < j \leq m \)
  - allDifferent\( (d_{11}, d_{12}, \ldots, d_{m-1,m}) \)
  - \( d_{ik} = d_{ij} + d_{jk} \) for \( 1 \leq i < j < k \leq m \)
  - Channelling constraints: \( d_{ij} = x_j - x_i \)
  - not always binary
Constraints in Combined Viewpoints

- It is safe to combine two complete models of a problem, with channelling constraints.
- But often unnecessary and inefficient.
- If some constraints are more easily expressed in one viewpoint, we don’t need them in both.
  - e.g. nurse rostering.
  - Constraints on nurse availability are stated in the ‘nurse’ viewpoint.
  - Constraints on work requirements (e.g. no. of nurses required for each shift) are stated in the ‘shift’ viewpoint.
- Or if they propagate better in one viewpoint.
  - e.g. $x_j - x_i \neq x_l - x_k$ v. allDifferent($d_{11}, d_{12}, ..., d_{m-1,m}$) in the Golomb ruler problem.
Choosing the Search Variables

- We need to choose a set of variables such that an assignment to each one, satisfying the constraints, is a complete solution to the problem.
- Assume we pass the search variables to the search algorithm in a list or array:
  - the order defines a static variable ordering
    - though we can still use a dynamic ordering
Search Variables

- When a model combines two (or more) viewpoints of a problem, which variables should drive the search?
- Assigning values to either set of variables would be sufficient to solve the problem
  - even if we did not express the problem constraints on those variables
  - the channelling constraints ensure that we can assign values to one set of variables but define the constraints in the other viewpoint, if we want
Search Variables – Permutation Problems

- We can use both sets of variables as search variables
  - e.g. use a dynamic variable order e.g. variable with smallest domain in either viewpoint
    - combines variable and value ordering: dual variable with smallest domain corresponds to the value occurring in fewest domains (in the other viewpoint)
Search Variables – Golomb Ruler

- Search strategy assigns values to $x_1, x_2, \ldots, x_m$ in that order
- NB there are only $m$ of these variables and $m(m-1)/2$ difference variables
  - heuristic: prefer the smaller set of search variables
SONET Problem: Viewpoints

- Whether a given node is on a given ring:
  - \( x_{ij} = 1 \) if node \( i \) is on ring \( j \)
- Which ring(s) each node is on:
  - \( N_i \) = set of rings node \( i \) is on
- Which nodes are on each ring
  - \( R_j \) = set of nodes on ring \( j \)
- In principle, any of these viewpoints could be the basis of a complete CSP model
  - channelling constraints \( (x_{ij} = 1) \equiv (i \in R_j) \equiv (j \in N_i) \)
- There are also auxiliary variables
  - \( n_i \) = the number of rings each node is on ( = \( |N_i| \))
Possible Choices

- Use just one set of variables, e.g. $x_{ij}$ – the others are just for constraint propagation
- Use two (or more) sets of variables (of the same type) e.g. $R_{j}, N_{i}$
  - interleave them in a sensible (static) order
  - or use a dynamic ordering applied to both sets of variables
- Use an incomplete set of variables first, to reduce the search space before assigning a complete set
  - e.g. decide how many rings each node is on (search variables $n_{i}$) and then which rings each node is on ($x_{ij}$)
    - another strategy adds assigning the objective variable first – see earlier
Automating Modelling

- There are lots of choices to make in modelling a problem as a CSP
  - difficult even with experience
- Can it be automated?
  - some initial steps so far
    - e.g. systems that propose models given a high-level specification
    - descriptions of common patterns in modelling
Advice from the Folklore

• **Reduce the number of variables**
  
  • if we only use one viewpoint:
    
    • a model which needs fewer variables to describe the solutions to the problem is likely to be a better model
    
    • e.g. an integer model is probably better than a Boolean model
  
  • But only if the variables allow the constraints to be expressed in a way that propagates well
    
    • artificially reducing the number of variables by inventing a single variable to replace a pair of variables will not give a better model
Advice from the Folklore 2

• Reduce the number of constraints
  • rewriting a set of constraints in a more compact form is likely to be beneficial, if the resulting constraints can propagate efficiently
    • e.g. combine constraints with the same scope
    • use a global constraint to replace a set of constraints
  • But simply conjoining constraints for the sake of reducing their number will not give a better model if the new constraints cannot propagate efficiently
More Advice

- **Add more variables**
  - auxiliary variables to allow constraints to be expressed
  - new viewpoints allowing a different perspective on the problem

- **Add more constraints**
  - implied constraints
  - channelling constraints to link new variables

- **Check empirically**
  - that a change does reduce run-time
Conclusion

• Aim for a rich model
  • multiple viewpoints
  • auxiliary variables
  • implied constraints

• Understand the problem as well as you can
  • build that insight into the model
  • the better you can understand a problem, the better you can solve it

THE END