Constraint Programming

Course Notes 2

Nadjib Lazaar
University of Paris-Saclay
lazaar@lisn.fr

0.1 Optimization in CP

Optimization in CP extends the standard CSP framework by introducing an **objective function** f that must be optimized, alongside constraints that must be satisfied. To achieve this, an auxiliary variable c is added and constrained as $c = f(X)$, where X represents the variables of the CSP. This is called the **objective constraint**. The goal is to find a solution minimizing f. This is typically done by iteratively solving a sequence of satisfaction problems:

- Find an initial solution s satisfying all constraints.
- Add a constraint $c < f(s)$ to exclude solutions not better than s.
- Repeat the process until no solution is found. The last solution is optimal.

Constraint Propagation: To improve efficiency, constraint propagation techniques are applied to prune infeasible values of c.

0.1.1 Branch-and-Bound Algorithm

The Branch-and-Bound (B&B) algorithm combines backtracking search with pruning using bounds on the objective function.

Algorithm 1: Branch-and-Bound (B&B)

Input: $\langle X, D, C, f \rangle$: Optimization CSP
Output: Optimal solution s^* minimizing f, or **null** if none exists.

```
bestSolution ← null;
bestCost ← +∞;
function B&B(I : partial assignment);
if I is complete then
    cost ← f(I);
    if cost < bestCost then
        bestCost ← cost;
        bestSolution ← I;
    return;
Select a variable x_i ∉ I;
foreach v ∈ D(x_i) do
    I′ ← I ∪ {x_i = v};
    if f(I′) < bestCost and I′ satisfies C then
        B&B(I′);
return bestSolution;
```

Advantages: B&B systematically explores the search space, pruning sub-optimal branches, and guarantees optimality if a solution exists.

Applications: Widely used in scheduling, planning, and resource allocation where both feasibility and optimality are crucial.

1 Global Constraints in CP

Global constraints are one of the most powerful tools in the realm of CP. They encapsulate recurring patterns of constraints into higher-level abstractions, making problem modeling more expressive and computationally efficient [Beldiceanu et al., 2007].

Definition of Global Constraints

A **global constraint** is a constraint defined over an arbitrary set of variables that captures a specific property or pattern, often recurring across multiple problem domains.

Alldifferent Global Constraint Example

The `alldifferent` constraint ensures that all variables in its scope take distinct values. By aggregating several low-level constraints into a single global constraint, it is possible to leverage specialized algorithms for propagation and consistency enforcement.

1.1 Strength of Global Constraints

The power of global constraints lies in their ability to:

- **Capture High-Level Properties:** A global constraint expresses a complex property in a single, declarative statement, reducing modeling effort and increasing readability.
- **Enhance Propagation:** By using advanced filtering algorithms, global constraints can achieve higher levels of domain reduction, thus decreasing the search space.
- **Encourage Reuse:** Commonly used global constraints can be applied across various problems and domains, fostering modularity.

1.2 Decomposability and Complexity

Once a property is captured by a global constraint, it is essential to analyze it in terms of decomposition and complexity:

- **Arc-Consistency (AC) Decomposability:** A global constraint is said to be *AC-decomposable* if it can be rewritten as an equivalent set of simpler constraints (e.g., binary constraints) while maintaining arc-consistency. For instance, the Berge-acyclic property of the underlying graph structure is a key criterion for checking AC-decomposability.
- **Computational Complexity:** Many global constraints are inherently *NP-hard*, meaning their filtering algorithms are computationally expensive. When feasible, practical approximations or incomplete propagators are employed to strike a balance between efficiency and pruning power.

1.3 Propagation and Filtering Algorithms

A hallmark of global constraints is the ability to define and implement different propagators to enforce consistency levels:

- **Complete Propagators:** These achieve the strongest possible domain reductions while respecting the semantics of the constraint. However, they may be computationally expensive.
- **Partial Propagators:** These perform limited domain pruning to maintain efficiency, trading completeness for runtime performance.

For example, the `alldifferent` constraint can use algorithms based on graph matching (e.g., Hopcroft-Karp) to achieve arc-consistency. Similarly, the `sum` constraint can employ flow-based algorithms for effective propagation.

1.4 Examples of Global Constraints

- `alldifferent`: Ensures all variables in its scope have distinct values. Applications include scheduling and assignment problems.
- `sum`: Enforces a linear equation among variables, useful in resource allocation.
- `global_cardinality`: Extends `alldifferent` by bounding the frequency of each value in the domain.
- `cumulative`: Used in scheduling to ensure resource usage does not exceed a given capacity.
- `regular`: Validates sequences of variables against a finite automaton, often used in string problems or routing.

1.5 Importance in Practice

The use of global constraints allows practitioners to focus on problem modeling while relying on optimized propagators to handle the computational complexity. Their role in CP is indispensable, as they:

- Provide modular, reusable building blocks for constraint models.
- Leverage domain-specific knowledge through tailored propagation techniques.
- Reduce the gap between problem formulation and efficient resolution.

1.6 Research Directions

Active research in global constraints continues to explore:

- **New Propagators:** Developing efficient algorithms for emerging global constraints.
- **Hybrid Techniques:** Combining propagation with other solving paradigms, such as linear programming or SAT.
- **Scalability:** Ensuring propagation algorithms remain effective on large-scale instances.

Global constraints epitomize the declarative nature of CP, enabling powerful abstraction while offering practical efficiency. Their continued development remains a cornerstone of advancing CP theory and practice.

References

[Beldiceanu et al., 2007] Beldiceanu, N., Carlsson, M., Demassey, S., and Petit, T. (2007). Global constraint catalogue: Past, present and future. *Constraints*, 12(1):21–62.