Linear programming, Constraint programming, MIP: Finds optimal solution for small problems but doesn’t scale well

Local search: Finds solution with lower quality, but scales better.

**Constraint programming:**

Uses branch and prune. Finds a feasible solution (no optimization function)

Each constraint needs to:

* Check feasibility
* Prune the search space (limit the domain of some of the variables)

Each constraint is checked separately.

Constraint types:

* regular constraints:
* global constraints: like Lego building blocks. Handle multiple variables at once. Examples:
  + all\_different
  + table constraint: specify a subset of all var values combinations as being legal.
  + Lexicographical ordering
  + At most

Symmetry: we want to break it, to reduce the search space.

Many types. For example: Var symmetry, value symmetry.

We can add constraints that break the symmetry for ex. Impose lexicographical order

Redundant constraints: Don’t change the set of feasible solutions, but speedup search. Surrogate constraints – a linear combination of existing constraints – can provide a global view.

Dual modeling: model the problem in 2 ways and join them to 1 model.

Searching the search space: “first-fail” – search first where you are more likely to fail.

Searching in constraint programming :

* variable/value labeling: first choose variable, then assign value. (start with variables with the smallest domain, choose value with the least variable)
* value/variable labeling: first choose value, then assign it to a variable
* domain splitting
* focusing on the objective
* symmetry breaking during search
* randomization and restarts