Forward Checking

• Idea: Reduce domain of unassigned variables based on assigned variables.
• Each time variable is instantiated, delete from domains of the uninstantiated variables all of those values that conflict with current variable assignment.
• Identify dead ends without having to try them via backtracking.

Comparison of CSP Algorithms

<table>
<thead>
<tr>
<th>Problem</th>
<th>BT</th>
<th>BT+MR</th>
<th>BT+FC</th>
<th>BT+FC+MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>(&gt;1,000K)</td>
<td>(&gt;1,000K)</td>
<td>2K</td>
<td>60</td>
</tr>
<tr>
<td>n-queens</td>
<td>(&gt;40,000K)</td>
<td>13,500K</td>
<td>(&gt;40,000K)</td>
<td>817K</td>
</tr>
</tbody>
</table>

Constraint Propagation (Arc Consistency)

Arc Consistency — state is arc-consistent, if every variable has some value that is consistent with each of its constraints (consider pairs of variables)

• Init: $Q$ is queue with all (directed) arcs $(X_i, X_j)$ in CSP
• WHILE $Q$ is not empty
  - $(X_i, X_j) = \text{remove}_\text{first}(Q)$
  - FOREACH $x \in \text{dom}(X_i)$
    * IF no $y \in \text{dom}(X_j)$ satisfies constraint $(X_i, X_j)$
      · THEN remove $x$ from $\text{dom}(X_i)$
    - IF $\text{dom}(X_i)$ changed
      · THEN add all arcs $(X_k, X_i) \notin Q$ to $Q$

Constraint Propagation (K-Consistency)

• K-Consistency generalizes arc-consistency (2-consistency).
• Consistency of groups of K variables.

Local Search for CSPs

Remarks

• Infinite discrete domains and continuous domains
• Exploiting special problem structure
• Dramatic recent progress in Constraint Satisfaction. Methods can now handle problems with 10,000 to 100,000 variables, and up to 1,000,000 constraints.