

DM826 – Spring 2011
Modeling and Solving Constrained Optimization Problems

Lecture 9
Search (cntd.)

Marco Chiarandini

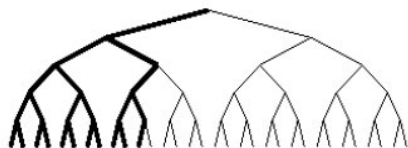
Department of Mathematics & Computer Science
University of Southern Denmark

1. Search

Randomization in Search Tree

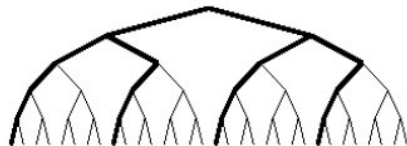
- Dynamical selection of solution components in construction or choice points in backtracking.
- Randomization of construction method or selection of choice points in backtracking while still maintaining the method complete
↳ *randomized systematic search.*
- Randomization can also be used in incomplete search

Bounded-backtrack search:



`bbs(10)`

Depth-bounded, then bounded-backtrack search:

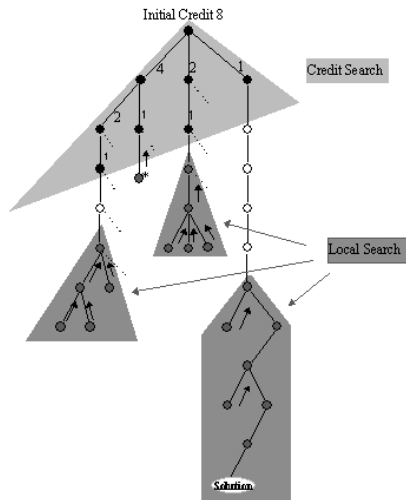


`dfs(2, bbs(0))`

http://4c.ucc.ie/~hsimonis/visualization/techniques/partial_search/main.htm

Credit-based search

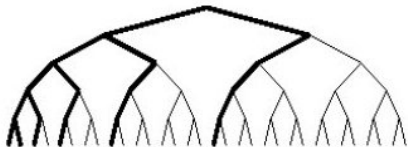
- Key idea: important decisions are at the top of the tree
- **Credit** = backtracking steps
- Credit distribution: one half at the best child the other divided among the other children.
- When credits run out follow deterministic best-search
- In addition: allow limited backtracking steps (eg, 5) at the bottom
- **Control parameters**: **initial credit**, **distribution** of credit among the children, **amount of local backtracking** at bottom.



Incomplete Search

Limited Discrepancy Search (LDS)

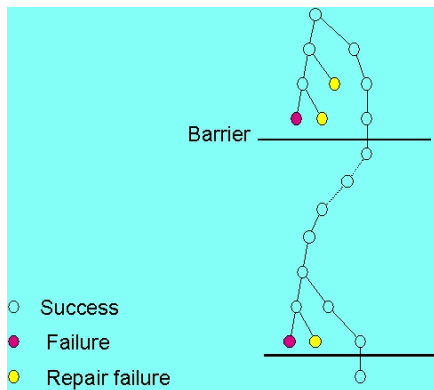
- Key observation that often the heuristic used in the search is nearly always correct with just a few exceptions.
- Explore the tree in increasing number of **discrepancies**, modifications from the heuristic choice.
- Eg: count one discrepancy if second best is chosen
count two discrepancies either if third best is chosen or twice the second best is chosen
- **Control parameter**: the **number of discrepancies**



Incomplete Search

Barrier Search

- Extension of LDS
- Key idea: we may encounter several, independent problems in our heuristic choice. Each of these problems can be overcome locally with a limited amount of backtracking.
- At each **barrier** start LDS-based backtracking



- Uses a complete-state formulation
- Initial state: a value assigned to each variable (randomly)
- Changes the value of one variable at a time
- Evaluation of a state:
number of constraints violated or variables to change (see soft constraints)
- Min-conflict heuristic [Minton et al., 1992]:
 - pick one variable involved in a constraint violation at random
 - assign to it the best value
- Run-time independent from problem size

Hoos H.H. and Tsang E. (2006). **Local Search Methods**, chap. 5. Elsevier.

Minton S., Johnston M., Philips A., and Laird P. (1992). **Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems.** *Artificial Intelligence*, 58(1-3), pp. 161–205.