

Homework 4

Implementing Advanced Backtrack-Search Mechanisms

Assigned: Wednesday, October 19, 2005

Due: Tuesday, November 1st, 2005

Total value: 120 points. Penalty of 20 points for lack clarity and documentation in code. Bonus of 35 points for additional work and constructive feedback.

Notes: This homework must be done individually. *If you receive help from anyone, you must clearly acknowledge it.* Always acknowledge sources of information (URL, book, class notes, etc.). Please inform instructor quickly about typos or other errors.

The goal of this exercise is to implement 3 advanced backtrack search mechanisms and test it on the test cases of Homework 2.

- Implementing FC. 30 points
- Implementing CBJ. 30 points
- Implementing the hybrid of the above two mechanisms: FC-CBJ. 30 points
- Reporting the results obtained from solving the four examples of Homework 2. 30 points
- Your impressions. **Bonus:** 5 points

General indications:

- *Please make sure that you keep your code and protect your files.* Your name, date, and course number must appear in each file of code that you submit.
- All programs must be compiled, run and tested on `cse.unl.edu`. Programs that do not run correctly in this environment will not be accepted.
- You must submit a README file so that we know to run and test your code.

1 Advanced BT search mechanisms

1. Taking the ordering heuristic (i.e., LD, degree, and ddr) and the ordering strategy (i.e., static and dynamic) as parameters, implement the partial look-ahead technical FC.
2. Taking the ordering heuristic (i.e., LD, degree, and ddr) as parameters, and using a static ordering of the variables, implement the mechanism for reducing the backtracking effort ‘conflict-directed backtracking.’

Bonus (10 points): Implement dynamic variable ordering with CBJ.

3. Taking the ordering heuristic (i.e., LD, degree, and ddr) as parameters, and using a static ordering of the variables, implement the hybrid of the above two mechanisms FC-CBJ.

Bonus (10 points): Implement dynamic variable ordering with FC-CBJ.

2 Performance comparison

For the following working conditions:

- For one variable ordering heuristic (i.e., choosing one from LD, degree, and ddr),
- under the ordering strategy implemented (i.e., static and/or dynamic).
- searching for *one* solution,

Bonus (10 points): searching for *all* solutions for all 3 strategies, for both static and dynamic ordering,

Conduct the following tasks:

- Evaluate the the four search strategies implemented so far (i.e., the above 3 plus BT of Homework 3) in terms of #CC, #NV, and CPU time,
- compare their performance,
- and conclude with your observations.

Hint: verify Prosser’s conclusions and that of Kondrak and van Beek’s. Do these conclusions hold under dynamic orderings?

3 Your impressions

Tell us whether you find the set of Homework 2, Homework 3, and Homework 4 useful or not and how they can possibly be improved.