Visualization of Branch-and-bound Algorithms

Brady Hunsaker

University of Pittsburgh

2007 Nov 5
Collaborators

Osman Özaltın
Ph.D. Student
Lehigh University

Ted Ralphs

Reasons to monitor the progress of branch-and-bound
Current measures
Our approach
Visualization tools
Current and future work
Why monitor the progress of B&B algorithms?
Why monitor the progress of B&B algorithms?

- How good is the best solution so far?
Why monitor the progress of B&B algorithms?

- How good is the best solution so far?
- How much longer do we have to wait until the algorithm terminates?
- How likely is it that a better solution will be found, and how much better will it be?
- Should we change any algorithm strategies? (branching, node selection,...)
Why monitor the progress of B&B algorithms?

- How good is the best solution so far?
- How much longer do we have to wait until the algorithm terminates?
- How likely is it that a better solution will be found, and how much better will it be?
Why monitor the progress of B&B algorithms?

- How good is the best solution so far?
- How much longer do we have to wait until the algorithm terminates?
- How likely is it that a better solution will be found, and how much better will it be?
- Should we change any algorithm strategies? (branching, node selection,...)
How do we monitor the progress of B&B now?

- Most commercial and open-source solvers can report:
  - optimality gap
  - number of active nodes
  - some internal measures for guiding the algorithm

Each of these methods have some strengths and weaknesses.
How do we monitor the progress of B&B now?

Most commercial and open-source solvers can report:

- optimality gap
How do we monitor the progress of B&B now?

- Most commercial and open-source solvers can report:
  - optimality gap
  - number of active nodes
How do we monitor the progress of B&B now?

- Most commercial and open-source solvers can report:
  - optimality gap
  - number of active nodes
  - some internal measures for guiding the algorithm
How do we monitor the progress of B&B now?

- Most commercial and open-source solvers can report:
  - optimality gap
  - number of active nodes
  - some internal measures for guiding the algorithm
- Each of these methods have some strengths and weaknesses
Current methods: optimality gap

- **Strength**: guarantee on quality of solution
Current methods: optimality gap

- **Strength**: guarantee on quality of solution
- **Strength**: nonincreasing
Current methods: optimality gap

- **Strength:** guarantee on quality of solution
- **Strength:** nonincreasing
- **Weakness:** may remain constant for long periods, then drop suddenly
Current methods: number of active nodes

- **Strength**: some sense of "work remaining"

![Graph showing number of active nodes over time](image-url)
Current methods: number of active nodes

- **Strength**: some sense of “work remaining”
- **Weakness**: may go up and down
Current methods: number of active nodes

- **Strength:** some sense of "work remaining"
- **Weakness:** may go up and down
- **Weakness:** each active node counts equally
Prior work and inspiration

- VBCTool (Diehl, Jünger, Kupke, Leipart)
Prior work and inspiration

- VBCTool (Diehl, Jünger, Kupke, Leipart)
- Images on the TSP website (www.tsp.gatech.edu)
Prior work and inspiration

- VBCTool (Diehl, Jünger, Kupke, Leipart)
- Images on the TSP website (www.tsp.gatech.edu)
- XPRESS-MP
Prior work and inspiration

- VBCTool (Diehl, Jünger, Kupke, Leipart)
- Images on the TSP website (www.tsp.gatech.edu)
- XPRESS-MP
- Cornuéjols, Karamanov, Li (2006)
Key Observations

1. B&B algorithms generate lots of data during the solution procedure.
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes

2. Current methods use only a small amount of this data

3. Most prior work only considers one type of information at a time

4. Develop tools that consider as much data as possible!
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:

B&B algorithms generate lots of data during the solution procedure. For each node: Number of nodes.
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:
     - LP Bound

2. Current methods use only a small amount of this data

3. Most prior work only considers one type of information at a time

4. Develop tools that consider as much data as possible!
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:
     - LP Bound
     - Integer infeasibility
**Key Observations**

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:
     - LP Bound
     - integer infeasibility
     - history/position in tree (e.g. depth and parent)

2. Current methods use only a small amount of this data

3. Most prior work only considers one type of information at a time

4. Develop tools that consider as much data as possible!
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:
     - LP Bound
     - integer infeasibility
     - history/position in tree (e.g. depth and parent)

2. Current methods use only a small amount of this data
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:
     - LP Bound
     - integer infeasibility
     - history/position in tree (e.g. depth and parent)

2. Current methods use only a small amount of this data

3. Most prior work only considers one type of information at a time
Key Observations

1. B&B algorithms generate lots of data during the solution procedure
   - Number of nodes
   - For each node:
     - LP Bound
     - Integer infeasibility
     - History/position in tree (e.g. depth and parent)

2. Current methods use only a small amount of this data

3. Most prior work only considers one type of information at a time

4. Develop tools that consider as much data as possible!
Approach

- Used open-source solvers GLPK, SYMPHONY, and CBC
Approach

- Used open-source solvers GLPK, SYMPHONY, and CBC
- Modified solver codes to output information when nodes are added and processed
Approach

- Used open-source solvers GLPK, SYMPHONY, and CBC
- Modified solver codes to output information when nodes are added and processed
- Wrote new code to create visual representations of the data by parsing the output file
Approach

- Used open-source solvers GLPK, SYMPHONY, and CBC
- Modified solver codes to output information when nodes are added and processed
- Wrote new code to create visual representations of the data by parsing the output file
  - Output file can be processed at any point during the solving process
Approach

- Used open-source solvers GLPK, SYMPHONY, and CBC
- Modified solver codes to output information when nodes are added and processed
- Wrote new code to create visual representations of the data by parsing the output file
  - Output file can be processed at any point during the solving process
  - Parsing is done with Perl; images are created with Gnuplot
Example of output from solver

```plaintext
# CBC
0.040003 heuristic -28.000000
2.692169 branched 0 -1 N -39.248099 16 0.169729
2.692169 pregnant 2 0 R -39.248063 14 105.991922
2.708170 pregnant 3 0 L -38.939929 6 0.105246
2.764173 pregnant 5 2 R -39.244862 12 49.115388
2.764173 branched 2 0 R -39.248063 14 105.991922
```
Visual Representations

- Visual representations:
  - Histogram of active node LP bounds
Visual Representations

- Visual representations:
  - Histogram of active node LP bounds
  - Scatter plot of active node LP bounds & integer infeasibility
Visual Representations

- Visual representations:
  - Histogram of active node LP bounds
  - Scatter plot of active node LP bounds & integer infeasibility
  - Incumbent node history in scatter plot
Visual Representations

- Visual representations:
  - Histogram of active node LP bounds
  - Scatter plot of active node LP bounds & integer infeasibility
  - Incumbent node history in scatter plot
  - B&B trees showing the LP bound of each node
Visualization tools: Histogram of active node LP bounds

- Horizontal axis is the LP bound
- Vertical axis is number of active nodes
- Green vertical line shows the current incumbent value and the blue one shows the overall LP bound
Example histogram series 1: l152lav (MIPLIB 2003)

histogram of objective values 000

number of active nodes

objective interval

Hunsaker (Univ of Pittsburgh)  Visualization  2007 Nov 5  14 / 29
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)

Histogram of objective values 002

- Objective interval
- Number of active nodes
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)

Histogram of objective values 008

Number of active nodes

Objective interval

Hunsaker (Univ of Pittsburgh)
Example histogram series 1: l152lav (MIPLIB 2003)

Histogram of objective values 009

- X-axis: Objective interval
- Y-axis: Number of active nodes

- Range: 4679.000 to 4727.000
- Bar heights indicating frequency of objective values within the interval.
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)

Histogram of objective values 013

Number of active nodes

Objective interval

Values: 4697.000 - 4722.000
Example histogram series 1: l152lav (MIPLIB 2003)

Histogram of objective values 014

Number of active nodes

Objective interval

4700.000 4722.000

1200

1000

800

600

400

200

0
Example histogram series 1: l152lav (MIPLIB 2003)
Example histogram series 1: l152lav (MIPLIB 2003)

histogram of objective values 016

number of active nodes

objective interval
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)

Histogram of objective values 001

Number of active nodes vs. objective interval

Histogram with bars ranging from 358.764 to 738.404.
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)

Histogram of objective values 004

- Objective values range from 358.764 to 587.163.
- The histogram shows the distribution of active nodes across different objective intervals.
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)

histogram of objective values 006

number of active nodes

objective interval

Hunsaker (Univ of Pittsburgh)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Example histogram series 2: swath (MIPLIB 2003)
Visualization tools: Scatter plot of active node LP bounds & integer infeasibility

- Horizontal axis is the integer infeasibility
- Vertical axis is the LP bound
- Green horizontal line is the current incumbent value
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Example scatter plot series 1: swath (MIPLIB 2003)
Patterns in integer infeasibility: SYMPHONY

Scatterplot: mod008, SYMPHONY, 40s

Graph showing the relationship between the sum of integer infeasibilities and the objective value.
Patterns in integer infeasibility: SYMPHONY

Scatterplot: mod008, SYMPHONY, 42s
Visualization tools: Incumbent node history in scatter plot

- Horizontal axis is the integer infeasibility
- Vertical axis is the LP bound
- Green line shows ancestors of the incumbent node
Example incumbent node history series 1: l152lav (MIPLIB 2003)
Example incumbent node history series 1: l152lav (MIPLIB 2003)
Example incumbent node history series 2: liu (MIPLIB 2003)
Visualization tools: B&B trees

- Vertical axis is the LP bound
- Nodes are horizontally positioned to make the pictures more readable
- Alternatively, horizontal positions may be fixed based on position in the tree
Visualization tools: B&B trees

- Node color legend:
  - green: branched
  - yellow: candidate or pregnant
  - red: fathomed
  - blue: infeasible
Example B&B trees

B&B tree (dataset2per8inv10.dat 3s )
Example B&B trees
Example B&B trees

B&B tree (dataset2per8inv10.dat 7s )
Example B&B trees

B&B tree (dataset2per8inv10.dat 11s)
Example B&B trees
Example B&B trees

B&B tree (dataset2per8inv10.dat 3s )

obj. value

-28 -30 -32 -34 -36 -38
Example B&B trees

B&B tree (dataset2per8inv10.dat 7s)
Example B&B trees
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)

![Tree Diagram](image-url)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 1: l152lav (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 2: liu (MIPLIB 2003)
Example B&B tree series 3

B&B tree (dataset2per8inv10.dat 3s )

obj. value

-28
-30
-32
-34
-36
-38
Example B&B tree series 3
Example B&B tree series 3

B&B tree (dataset2per8inv10.dat 7s)
Example B&B tree series 3

B&B tree (dataset2per8inv10.dat 11s)
Example B&B tree series 3
Example B&B tree series 3

B&B tree (dataset2per8inv10.dat 15s )

obj. value

-38

-36

-34

-32

-30

-28
Example B&B tree series 3
Example B&B tree series 3

B&B tree (dataset2per8inv10.dat 23s )

obj. value

-38

-36

-34

-32

-30

-28
Example B&B tree series 3
Example B&B tree series 3

B&B tree (dataset2per8in10.dat 31s )
Example B&B tree series 3

B&B tree (dataset2per8inv10.dat 31s)
Imagined uses

- Predict time to completion (through experienced user or automated)
Imagined uses

- Predict time to completion (through experienced user or automated)
- Quickly determine what makes an instance hard
Imagined uses

- Predict time to completion (through experienced user or automated)
- Quickly determine what makes an instance hard
- Identify key LP solutions that should be investigated
Summary and Current Efforts

Our code and graphs provide a variety of visual information for users of branch-and-bound algorithms.
Summary and Current Efforts

- Our code and graphs provide a variety of visual information for users of branch-and-bound algorithms.
- A paper has been submitted. A pre-print is available at Optimization Online.

Future work: estimate likelihood of better integer solutions
Can other information be extracted: recommended node selection strategy or cuts?
What will other researchers do with the tools?
Summary and Current Efforts

- Our code and graphs provide a variety of visual information for users of branch-and-bound algorithms.
- A paper has been submitted. A pre-print is available at Optimization Online.
- The code has been submitted to COIN-OR. At present, available at www.engr.pitt.edu/hunsaker/
Summary and Current Efforts

- Our code and graphs provide a variety of visual information for users of branch-and-bound algorithms.
- A paper has been submitted. A pre-print is available at Optimization Online.
- The code has been submitted to COIN-OR. At present, available at www.engr.pitt.edu/hunsaker/
- Future work: estimate likelihood of better integer solutions
Summary and Current Efforts

- Our code and graphs provide a variety of visual information for users of branch-and-bound algorithms.
- A paper has been submitted. A pre-print is available at Optimization Online.
- The code has been submitted to COIN-OR. At present, available at www.engr.pitt.edu/hunsaker/
- Future work: estimate likelihood of better integer solutions
- Can other information be extracted: recommended node selection strategy or cuts?
Summary and Current Efforts

- Our code and graphs provide a variety of visual information for users of branch-and-bound algorithms.
- A paper has been submitted. A pre-print is available at Optimization Online.
- The code has been submitted to COIN-OR. At present, available at www.engr.pitt.edu/hunsaker/
- Future work: estimate likelihood of better integer solutions
- Can other information be extracted: recommended node selection strategy or cuts?
- What will other researchers do with the tools?