Fast Solutions to CSPs

Presented by: Robert Effinger
              Dan Lovell

Presented To: 16.412J Cognitive Robotics
              MIT

References: “Dynamic Backtracking”
            Matthew L. Ginsberg, CIRL, University of Oregon
            Journal of Artificial Intelligence Research 1 (1993)
            p. 25-46

            “Hybrid algorithms for the constraint satisfaction problem”
            Prosser, P. Computational Intelligence 9 (1993), 268-299.

April 5, 2004
Motivation

- Mobile robot planning
- Resource scheduling
- Laying out a silicon chip
- Interpreting a visual image
- Manufacturing processes
- Design of airline timetable
- Radio frequency planning
Quick Definition of a CSP

Constraint Satisfaction Problem (I, V, C)

- I, a set of variables
- Vi, a set of possible values for each variable in I.
- C, a set of Cij constraints, each a binary relation

\[ C = \{C1,1 \ldots C1,n \ C2,1 \ldots C2,n \ldots Cn,n\} \]

A Solution is found when each variable I is assigned a value from its domain Vi and the set of all Constraints \{C\} is satisfied.
How our two talks fit together

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking

Six base styles of CSP search

Go Backwards

(Bobby)

More informed Styles

Go Forwards

(Dan)

Different styles

(1970’s)

(80’s and 90’s)

Hybrid Algorithms

(Dan)

Generally Faster

Chronological Backtracking

Backjumping

Conflict-Directed Backjumping

Dynamic Backtracking

Backmarking

Forward Checking
Dynamic Backtracking

and a review of: Chronological Backtracking and Conflict-Directed Backjumping

Advanced Lecture Topic: Fast Solutions to CSPs

Presented by: Robert Effinger

Presented To: 16.412J Cognitive Robotics
MIT

Reference: “Dynamic Backtracking”
Matthew L. Ginsberg, CIRL, University of Oregon

April 5, 2004
Overview

• Definition of a CSP

• Simple Map Coloring Example
  – Representing a CSP as a Search Tree
  – Introduce the Example Problem

• Compare Three Backtracking Algorithms
  – Chronological Backtracking
  – Conflict-Directed Backjumping
  – Dynamic Backtracking

• Summary of Dynamic Backtracking
  – Pros and Cons

April 5, 2004
Quick Definition of a CSP

Constraint Satisfaction Problem \( (I, V, C) \)

- \( I \), a set of variables
- \( V_i \), a set of possible values for each variable in \( I \).
- \( C \), a set of \( C_{ij} \) constraints, each a binary relation
  \[ C = \{C_{1,1} \ldots C_{1,n} \ C_{2,1} \ldots C_{2,n} \ldots C_{n,n}\} \]

A Solution is found when each variable \( I \) is assigned a value from it’s domain \( V_i \) and the set of all Constraints \{C\} is satisfied.

\( I = \{A, B, C, D, E\} \)

\( V_i = \{\text{red, yellow, blue}\} \)

\( C_{ij} = (\text{no neighbor can be the same color}) \)
Simple Example Problem
Search Tree Representation of a CSP

Simple Map Coloring Example

- Variables are assigned values according to an instantiation order
- The search tree grows downward as until each variable is assigned a value from its domain.
- Dynamic Backtracking allows a dynamic instantiation order

Instantiation Order

1.)  
2.)  
3.)  
4.)  
5.)  

Search Tree
Changing the Color Ordering to Create an Interesting Example Problem

- Pushes the first feasible solution further into the search tree
- Still covers all possible permutations of value assignments to variables
- Still a valid CSP

Instantiation Order

1.) [Color assignments]
2.) [Color assignments]
3.) [Color assignments]
4.) [Color assignments]
5.) [Color assignments]
Compare Three Backtracking Algorithms

1.) Chronological Backtracking
2.) Conflict-Directed Backjumping
3.) Dynamic Backtracking
Sneak Preview of the Solution

Solve map example using:

1.) Chronological Backtracking
2.) Conflict-Directed Backjumping
3.) Dynamic Backtracking

Note:
This is what the solution will look like each time.
We will compare the # of nodes expanded (i.e. regions colored) until the first solution is found.
1.) Chronological Backtracking

**Chronological_Backtrack()**

1.) Set P = \{null\} (P is the partial solution to the CSP)
   Set Vi = \{1\} (start with first variable in instantiation order)

2.) If P = solution, return Success. If Vi = 0 return Failure
   Else if P = Consistent,
   set (Vi) to the next variable in instantiation order and assign it’s next domain color (c).
   Else if P = Inconsistent, remove (c) from domain of (Vi) and continue

3.) While domain of (Vi) is not empty, choose the next domain color (c) and return to step 2.

4.) If domain of (Vi) is empty (i.e. out of colors to try for (Vi)
   Remove (Vi) from P, set Vi = Vi – 1, and return to step 3.
1.) Chronological Backtracking

Notes:
Helpful notes will go here

<table>
<thead>
<tr>
<th>Instantiation Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) A</td>
</tr>
<tr>
<td>2.) B</td>
</tr>
<tr>
<td>3.) C</td>
</tr>
<tr>
<td>4.) D</td>
</tr>
<tr>
<td>5.) E</td>
</tr>
</tbody>
</table>

# of Nodes Expanded
1
1.) Chronological Backtracking

Notes:
Helpful notes will go here

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

# of Nodes Expanded
2
1.) Chronological Backtracking

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded

3
1.) Chronological Backtracking

Notes:

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

# of Nodes Expanded

4
1.) Chronological Backtracking

Instantiation Order:

1.) Chronological Backtracking

Notes:

1.)  A
2.)  B
3.)  C
4.)  D
5.)  E

# of Nodes Expanded 5
1.) Chronological Backtracking

Notes:

Instantiation Order :

1.)  A

2.)  B

3.)  C

4.)  D

5.)  E

# of Nodes Expanded

6
1.) Chronological Backtracking

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded

7
1.) Chronological Backtracking

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded: 8
1.) Chronological Backtracking

Notes:

<table>
<thead>
<tr>
<th>Instantiation Order</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.)</td>
<td>A</td>
</tr>
<tr>
<td>2.)</td>
<td>B</td>
</tr>
<tr>
<td>3.)</td>
<td>C</td>
</tr>
<tr>
<td>4.)</td>
<td>D</td>
</tr>
<tr>
<td>5.)</td>
<td>E</td>
</tr>
</tbody>
</table>

# of Nodes Expanded: 8
1. Chronological Backtracking

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded: 8
1.) Chronological Backtracking

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded

9
1.) Chronological Backtracking

Instantiation Order:

1.) $\rightarrow$ A
2.) $\rightarrow$ B
3.) $\rightarrow$ C
4.) $\rightarrow$ D
5.) $\rightarrow$ E

Notes:

# of Nodes Expanded: 9
1.) Chronological Backtracking

Notes:

<table>
<thead>
<tr>
<th>Instantiation Order</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.)</td>
<td>A</td>
</tr>
<tr>
<td>2.)</td>
<td>B</td>
</tr>
<tr>
<td>3.)</td>
<td>C</td>
</tr>
<tr>
<td>4.)</td>
<td>D</td>
</tr>
<tr>
<td>5.)</td>
<td>E</td>
</tr>
</tbody>
</table>

# of Nodes Expanded
10
1.) Chronological Backtracking

Notes:

Chronological backtracking doesn’t notice this is the same subtree, and still searches it.

(a.k.a. thrashing)

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

# of Nodes Expanded

15
1.) Chronological Backtracking

Notes:

<table>
<thead>
<tr>
<th>Instantiation Order:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) A</td>
</tr>
<tr>
<td>2.) B</td>
</tr>
<tr>
<td>3.) C</td>
</tr>
<tr>
<td>4.) D</td>
</tr>
<tr>
<td>5.) E</td>
</tr>
</tbody>
</table>

# of Nodes Expanded: 16
1.) Chronological Backtracking

Notes:

<table>
<thead>
<tr>
<th>Instantiation Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) B</td>
</tr>
<tr>
<td>2.) A</td>
</tr>
<tr>
<td>3.) C</td>
</tr>
<tr>
<td>4.) D</td>
</tr>
<tr>
<td>5.) E</td>
</tr>
</tbody>
</table>

# of Nodes Expanded 16
1.) Chronological Backtracking

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded

16
1.) Chronological Backtracking

Notes:

1.) Chronological Backtracking

Instantiation Order:

1.) ➔ A
2.) ➔ B
3.) ➔ C
4.) ➔ D
5.) ➔ E

# of Nodes Expanded

16
1.) Chronological Backtracking

Instantiation Order:
1.) A
2.) B
3.) C
4.) D
5.) E

Cost = 16

# of Nodes Expanded
32

Notes:
Search is repeating the same problem.
No solution is possible.
1.) Chronological Backtracking

<table>
<thead>
<tr>
<th>Instantiation Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) ( \rightarrow ) A</td>
</tr>
<tr>
<td>2.) ( \rightarrow ) B</td>
</tr>
<tr>
<td>3.) ( \rightarrow ) C</td>
</tr>
<tr>
<td>4.) ( \rightarrow ) D</td>
</tr>
<tr>
<td>5.) ( \rightarrow ) E</td>
</tr>
</tbody>
</table>

Notes:
Now we’re getting somewhere

# of Nodes Expanded 33
1.) Chronological Backtracking

Notes:
Now we’re getting somewhere

<table>
<thead>
<tr>
<th>Instantiation Order</th>
<th># of Nodes Expanded</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) A</td>
<td>34</td>
</tr>
<tr>
<td>2.) B</td>
<td></td>
</tr>
<tr>
<td>3.) C</td>
<td></td>
</tr>
<tr>
<td>4.) D</td>
<td></td>
</tr>
<tr>
<td>5.) E</td>
<td></td>
</tr>
</tbody>
</table>
1.) Chronological Backtracking

Notes:

Now we’re getting somewhere

Instantiation Order:

1.) \[ \rightarrow \text{A} \]
2.) \[ \rightarrow \text{B} \]
3.) \[ \rightarrow \text{C} \]
4.) \[ \rightarrow \text{D} \]
5.) \[ \rightarrow \text{E} \]

# of Nodes Expanded

35
1.) Chronological Backtracking

Instantiation Order:

1.) \( A \)
2.) \( B \)
3.) \( C \)
4.) \( D \)
5.) \( E \)

Notes:

Now we’re getting somewhere

# of Nodes Expanded: 36
1.) Chronological Backtracking

Notes:

Solution Found!!

# of Nodes Checked

= 37

Instantiation Order :

1.)  A
2.)  B
3.)  C
4.)  D
5.)  E

# of Nodes Expanded

37
2.) Conflict-Directed Backjumping
2.) Conflict-Directed Backjumping

Conflict-Directed Backjumping()
1.) Set P = \{null\} and Set E = \{null\}
2.) If P = solution, return Success. If Vi = 0 return Failure.
   Else if P = Consistent,
     set next as (Vi) and assign next color (c), and goto step 2.
   Else if P = Inconsistent,
     remove (c) from domain of (Vi) and continue
3.) While domain of (Vi) is not empty
   choose the next domain color (c) and goto step 2.
4.) If domain of (Vi) is empty
   add (Vi, c) to Ei
   set Vi = most recent variable in Ei, and un-assign any
   variable later in the instantiation order
   remove any (Ej) involving (Vi), return to step 3.

Instantiation Order:
1.) $\rightarrow$ A
2.) $\rightarrow$ B
3.) $\rightarrow$ C
4.) $\rightarrow$ D
5.) $\rightarrow$ E

(E) Eliminating Explanations

(A database of Conflicts)

there are many ways to store these conflicts
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \( \rightarrow \) A
2.) \( \rightarrow \) B
3.) \( \rightarrow \) C
4.) \( \rightarrow \) D
5.) \( \rightarrow \) E

Notes:
Helpful notes will go here.

# of Nodes Expanded
1
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[\begin{array}{c} \text{red} \end{array} \quad \text{yellow} \quad \text{blue} \]

Notes:

2.) Conflict-Directed Backjumping

# of Nodes Expanded

2
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \(\text{A}\)

2.) \(\text{B}\)

3.) \(\text{C}\)

4.) \(\text{D}\)

5.) \(\text{E}\)

Notes:

# of Nodes Expanded: 3
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) $\rightarrow$ A
2.) $\rightarrow$ B
3.) $\rightarrow$ C
4.) $\rightarrow$ D
5.) $\rightarrow$ E

Notes:

1.) Conflict-Directed Backjumping

# of Nodes Expanded

4
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td>A,B</td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \( \overset{\rightarrow}{\text{A}} \)
2.) \( \overset{\rightarrow}{\text{B}} \)
3.) \( \overset{\rightarrow}{\text{C}} \)
4.) \( \overset{\rightarrow}{\text{D}} \)
5.) \( \overset{\rightarrow}{\text{E}} \)

Notes:

# of Nodes Expanded

5
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
<tr>
<td>E</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) red yellow A

2.) yellow B

3.) blue C

4.) blue D

5.) yellow E

Notes:

2.) Conflict-Directed Backjumping

# of Nodes Expanded

6
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
</tbody>
</table>

A,B,D

Instantiation Order:

1.)  

2.)

3.)

4.)

5.)

Notes:

2.) Conflict-Directed Backjumping

# of Nodes Expanded

7
2.) **Conflict-Directed Backjumping**

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td>red</td>
<td>A,B</td>
<td>A,B,D</td>
<td>A,B,D</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[\text{A}\]

2.) \[\text{B}\]

3.) \[\text{C}\]

4.) \[\text{D}\]

5.) \[\text{E}\]

Notes:

- 2.) Conflict-Directed Backjumping

- # of Nodes Expanded: 8
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td>A,B,D</td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td>red</td>
<td>A,B,D</td>
<td>A,B,D</td>
<td>A,B,D</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

2.) Conflict-Directed Backjumping

# of Nodes Expanded

8
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td>red</td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  

2.)  

3.)  

4.)  

5.)  

# of Nodes Expanded 8

Notes:
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>red</td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) ![Region A Color Red](image)
2.) ![Region B Color Yellow](image)
3.) ![Region C Color Blue](image)
4.) ![Region D Color Red](image)
5.) ![Region E](image)

Notes:

- 2.) Conflict-Directed Backjumping
- # of Nodes Expanded: 8
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>D</td>
<td>red</td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:
1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

# of Nodes Expanded: 9
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[ \text{red, yellow, blue} \]
2.) \[ \text{A} \]
3.) \[ \text{B} \]
4.) \[ \text{C} \]
5.) \[ \text{D} \]
6.) \[ \text{E} \]

Notes:
Conflict-Directed Backjumping knows to skip the circled nodes

\# of Nodes Expanded
9
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>A,B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)   A

2.)   B

3.)   C

4.)   D

5.)   E

Notes:

8 nodes expanded so far vs.

about 16 for chronological backtracking

# of Nodes Expanded

9
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
Will explore this tree even though no chance of success

Cost = 9 nodes better than the 16 nodes during chronological backtracking.

Instantiation Order:
1.)  

2.)  

3.)  

4.)  

5.)  

Same problem!!!

Cost = 9 nodes  

# of Nodes Expanded  

18
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td>A</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>red</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  \( \rightarrow \) A
2.)  \( \rightarrow \) B
3.)  \( \rightarrow \) C
4.)  \( \rightarrow \) D
5.)  \( \rightarrow \) E

Notes:

Now we’re on the right track

\[ \text{# of Nodes Expanded} \quad 19 \]
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>red</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) Red yellow A
2.) Yellow B
3.) Blue C
4.) Blue D
5.) Red yellow E

Notes:

# of Nodes Expanded

20
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>red</td>
<td>A</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) red yellow A
2.) yellow red B
3.) blue red C
4.) red blue D
5.) red yellow E

Notes:

# of Nodes Expanded

21
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>red</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>yellow</td>
<td></td>
<td>A,B,D</td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  A
2.)  B
3.)  C
4.)  D
5.)  E

Notes:

# of Nodes Expanded 22
2.) Conflict-Directed Backjumping

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>blue</td>
<td>A</td>
<td></td>
<td>A,B,D</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[ \text{A} \]
2.) \[ \text{B} \]
3.) \[ \text{C} \]
4.) \[ \text{D} \]
5.) \[ \text{E} \]

Notes:

Solution Found!!

# of Nodes Expanded

23
3.) Dynamic Backtracking
3.) Dynamic Backtracking

Dynamic Backtracking()
1.) Set \( P = \{ \text{null} \} \) and \( \text{Set } E = \{ \text{null} \} \)
2.) If \( P = \) solution, return Success. If \( V_i = 0 \) return Failure.
   Else if \( P = \) Consistent,
   set next as \( (V_i) \) and assign next color \( (c) \), and goto step 2.
   Else if \( P = \) Inconsistent,
   remove \( (c) \) from domain of \( (V_i) \) and continue
3.) While domain of \( (V_i) \) is not empty
   choose the next domain color \( (c) \) and goto step 2.
4.) If domain of \( (V_i) \) is empty
   add \( (V_i, c) \) to \( E_i \)
   Remove \( (V_i) \) from \( P \)
   set \( V_i = \) most recent variable in \( E_i \) any variable not in \( P \)
   remove any \( (E_j) \) involving \( (V_i) \), return to step 3.

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Allows a dynamic Instantiation order!
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[ \rightarrow \quad A \]
2.) \[ \rightarrow \quad B \]
3.) \[ \rightarrow \quad C \]
4.) \[ \rightarrow \quad D \]
5.) \[ \rightarrow \quad E \]

Notes:
Helpful notes will go here.

# of Nodes Expanded
1
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

1.) 3.) Dynamic Backtracking

Instantiation Order:

1.) [Red, Yellow, Blue] A
2.) B
3.) C
4.) D
5.) E

# of Nodes Expanded

2
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \(\text{red} \rightarrow \text{yellow} \rightarrow \text{blue} \rightarrow \text{A} \)
2.) \(\text{red} \rightarrow \text{yellow} \rightarrow \text{blue} \rightarrow \text{B} \)
3.) \(\text{red} \rightarrow \text{yellow} \rightarrow \text{blue} \rightarrow \text{C} \)
4.) \(\text{red} \rightarrow \text{yellow} \rightarrow \text{blue} \rightarrow \text{D} \)
5.) \(\text{red} \rightarrow \text{yellow} \rightarrow \text{blue} \rightarrow \text{E} \)

Notes:

- Dynamic Backtracking

# of Nodes Expanded: 3
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

1.) Dynamic Backtracking

# of Nodes Expanded

4
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  A
2.)  B
3.)  C
4.)  D
5.)  E

Notes:

- Dynamic Backtracking

# of Nodes Expanded: 5
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \(\color{red}A\)
2.) \(\color{yellow}B\)
3.) \(\color{blue}C\)
4.) \(\color{blue}D\)
5.) \(\color{yellow}E\)

Notes:

3.) Dynamic Backtracking

# of Nodes Expanded

6
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
<tr>
<td>E</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) [A]

2.) [B]

3.) [C]

4.) [D]

5.) [E]

Notes:

3.) Dynamic Backtracking

# of Nodes Expanded

7
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>red</td>
<td>A,B,D</td>
<td>A,B,D</td>
<td>A,B,D</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[\text{red} \rightarrow \text{yellow} \rightarrow \text{blue} \rightarrow \text{red} \rightarrow \text{red}\]

Notes:

1.) Dynamic Backtracking

# of Nodes Expanded: 8
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td>A,B</td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td>red</td>
<td>A,B</td>
<td>A,B</td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) A
2.) B
3.) C
4.) D
5.) E

Notes:

1.) Dynamic Backtracking

# of Nodes Expanded

8
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>blue</td>
<td></td>
<td>A,B</td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) Red yellow A, B
2.) Blue A, B
3.) Blue A, B
4.) Blue A, B
5.) Red A, B

Notes:

3.) Dynamic Backtracking

# of Nodes Expanded

8
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[\text{A}\]

2.) \[\text{B}\]

3.) \[\text{C}\]

4.) \[\text{D}\]

5.) \[\text{E}\]

Notes:

Dynamic Backtracking

# of Nodes Expanded: 8
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>D</td>
<td>red</td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td>A,B</td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) Red A
2.) Yellow B
3.) Blue C
4.) Red D
5.) Yellow E

Notes:

# of Nodes Expanded 9
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow-blue</td>
<td>A,B</td>
<td>A,B</td>
<td>A,B</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) Red A
2.) Yellow B
3.) Blue C
4.) Red D
5.) Yellow E

Notes:
Dynamic Backtracking doesn’t have to erase C.

# of Nodes Expanded 9
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>yellow</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td>A,B</td>
<td></td>
<td>A,B</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>A,B</td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \[ \text{地区 A} \]  
2.) \[ \text{地区 B} \]  
3.) \[ \text{地区 C} \]  
4.) \[ \text{地区 D} \]  
5.) \[ \text{地区 E} \]

Notes:

Dynamic Backtracking doesn’t have to erase C.

# of Nodes Expanded

9
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  
2.)  
3.)  
4.)  
5.)  

Notes:

A is recorded as the reason B can't be yellow

# of Nodes Expanded 9
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  

2.)  

3.)  

4.)  

5.)  

Notes:

This is the same problem as before but only costs 6 nodes, since it is lower in the tree.

Cost = 7

Same problem!!! but less costly!

# of Nodes Expanded  16
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  🟥🔵  A
2.)  🟧🔵  B
3.)  🟧🟦  C
4.)  🟦🟦  D
5.)  🟦🔵  E

Notes:

Now we’re on the right track.

# of Nodes Expanded

17
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  A
2.)  B
3.)  C
4.)  D
5.)  E

Notes:

Now we’re on the right track.

# of Nodes Expanded

18
3.) Dynamic Backtracking

Notes:
Now we’re on the right track.

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) \( \rightarrow \) A
2.) \( \rightarrow \) B
3.) \( \rightarrow \) C
4.) \( \rightarrow \) D
5.) \( \rightarrow \) E

# of Nodes Expanded
19
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td>A</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.)  A
2.)  B
3.)  C
4.)  D
5.)  E

Notes:

Now we’re on the right track.

# of Nodes Expended

19
3.) Dynamic Backtracking

Elimination Explanations:

<table>
<thead>
<tr>
<th>Region</th>
<th>Color</th>
<th>Red</th>
<th>yellow</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>red</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>yellow</td>
<td></td>
<td></td>
<td>A,B,D</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Instantiation Order:

1.) → A
2.) → B
3.) → C
4.) → D
5.) → E

Notes:

Solution Found!!

Solution Found!

# of Nodes Expanded

20
Dynamic Backtracking Summary

• **Positive Features**
  – Allows a Variable Instantiation Order
  – Allows partial assignments to remain assigned
    (if they are not part of a conflict)

• **Negative Features**
  – Requires a conflict-detection sub-routine
  – Requires more memory than simple backtrack search
  – Completeness proof is not easy to understand or visualize
    (the proof that it doesn’t skip over any of the search space)

<table>
<thead>
<tr>
<th></th>
<th>Map Example Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chronological BT</td>
<td>37</td>
</tr>
<tr>
<td>Conflict-Directed BT</td>
<td>23</td>
</tr>
<tr>
<td>Dynamic BT</td>
<td>20</td>
</tr>
</tbody>
</table>
Any Questions?