Chapter 11: Limits of Propagation (Costas Array)

Helmut Simonis

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

ECLiPSe ELearning Overview

Licence

Outline

What we want to introduce

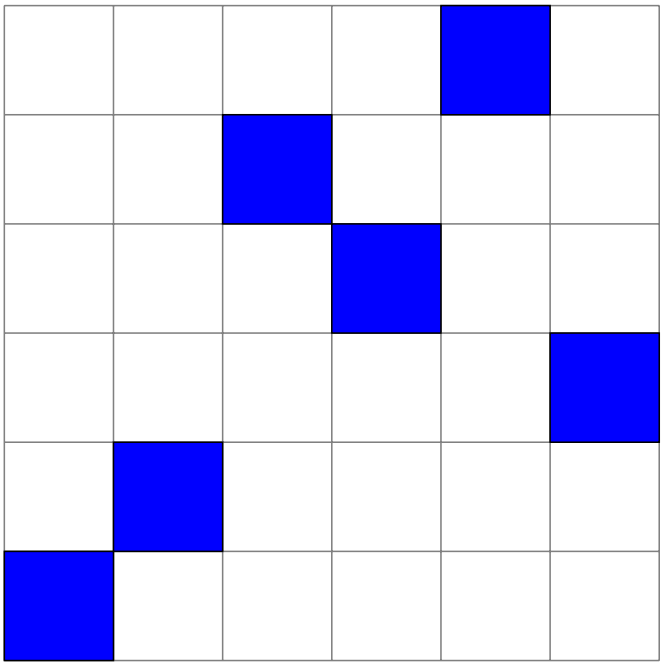
- Improving propagation does not always pay
- For some problems, simple backtracking is best
- CP may not always be the best method
- CP should always be fastest way to model problem
- Consider time to target
  - Time required to *run* program
  - Time required to *write* program
- Problem: Costas Array (Antenna design, sonar systems)

Problem Definition

Costas Array (Wikipedia)

A Costas array (named after John P. Costas) can be regarded geometrically as a set of N points lying on the squares of a NxN checkerboard, such that each row or column contains only one point, and that all of the N(N - 1)/2 vectors between each pair of dots are distinct.

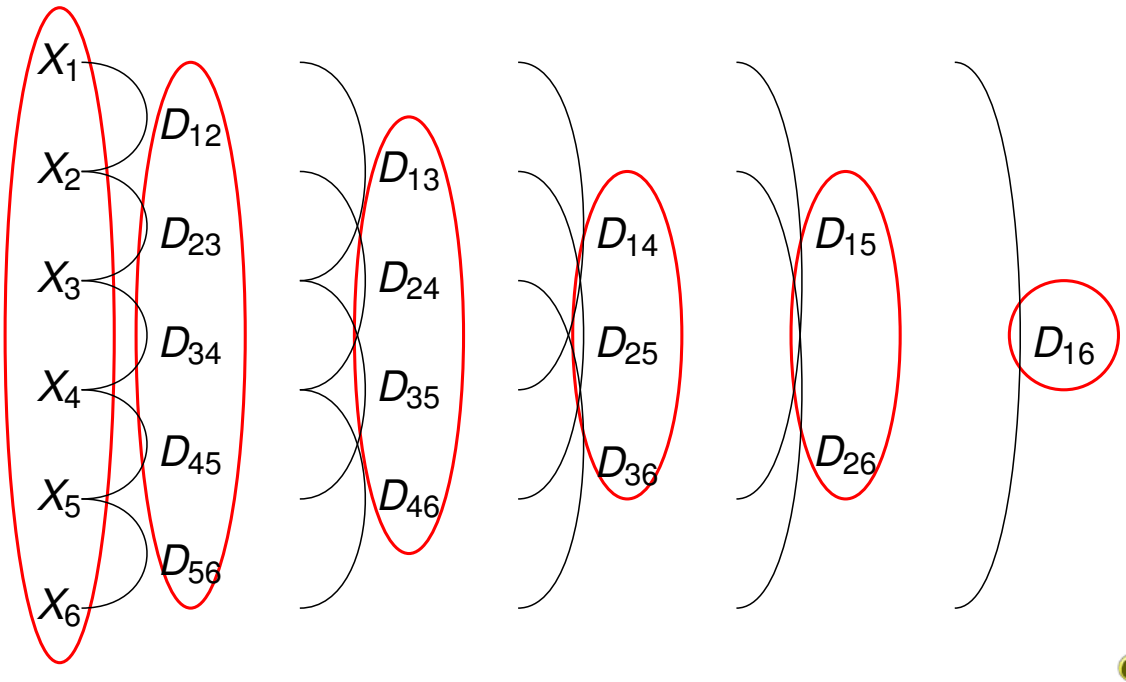
Example (Size 6)



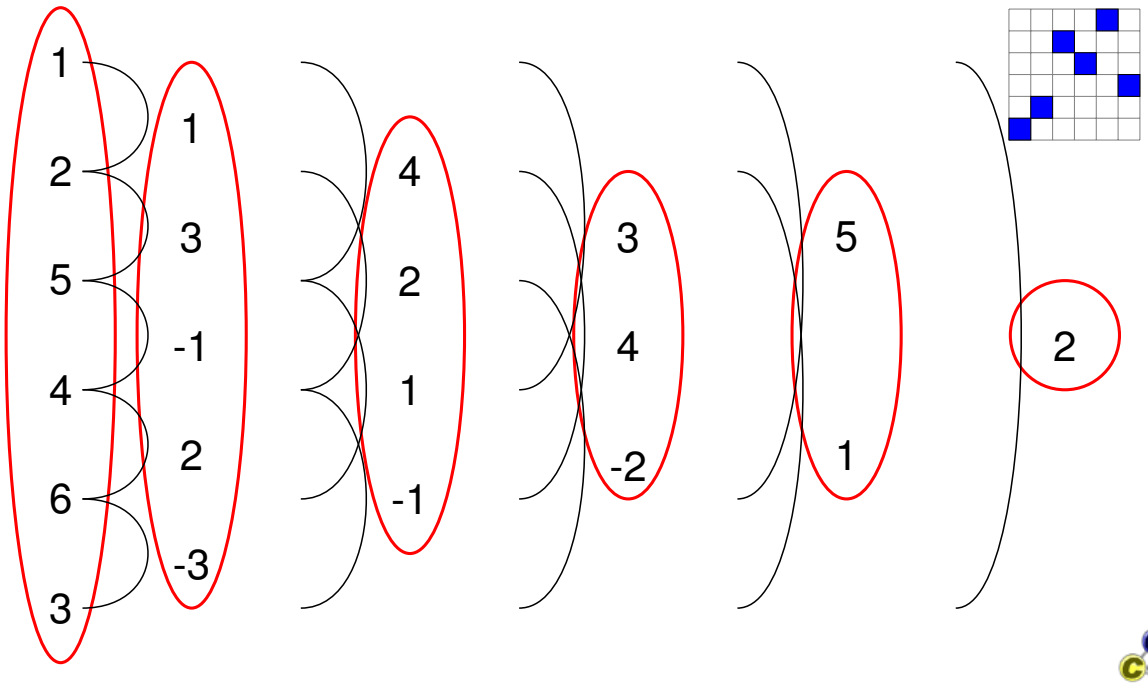
Model

- A variable for each column, ranging from 1 to N
- A list of N variables for the columns
- A difference variable between each ordered pair of variables
- `alldifferent` constraint between variables
- `alldifferent` constraints for all differences

Model



Example



Declarations

```
:-module(costas).

:-export(top/0).

:-lib(ic).
```

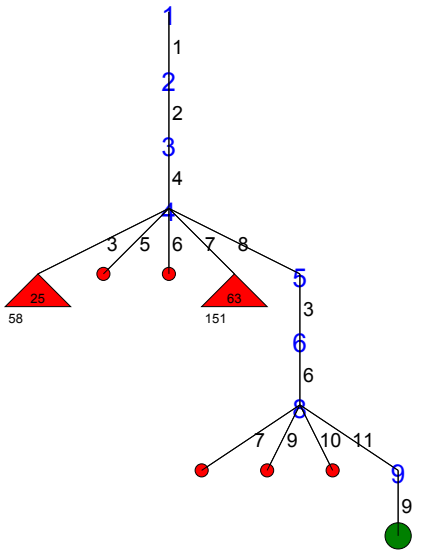
Main Program

```
top:-
        (for(N,3,20) do
            costas(N,_)
        ).

costas(N,L):-
        length(L,N),
        L :: 1..N,
        alldifferent(L),
        L = [_|L1],
        diffs(L,L1),
        search(L,0,first_fail,indomain,
               complete,[]).
```

Differences

```
diffs(_,[]).
diffs(L,[H|T]):-
        diff_pairs(L,[H|T],Diffs),
        alldifferent(Diffs),
        diffs(L,T).

diff_pairs(_,[],[]).
diff_pairs([X|X1],[Y|Y1],[D|D1]):-
        X #= Y+D,
        diff_pairs(X1,Y1,D1).
```

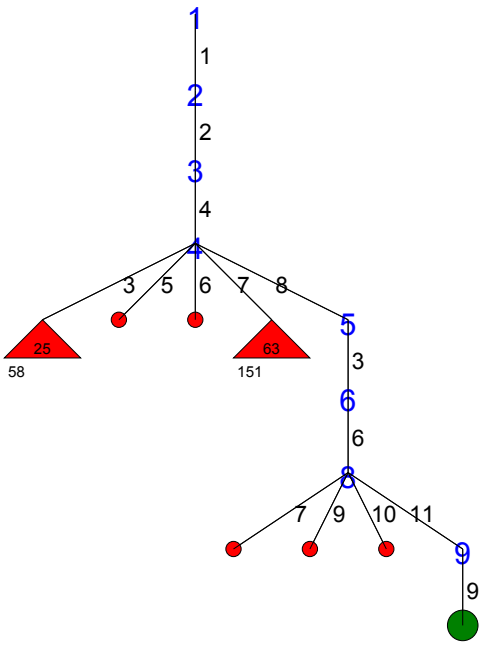
Basic Model



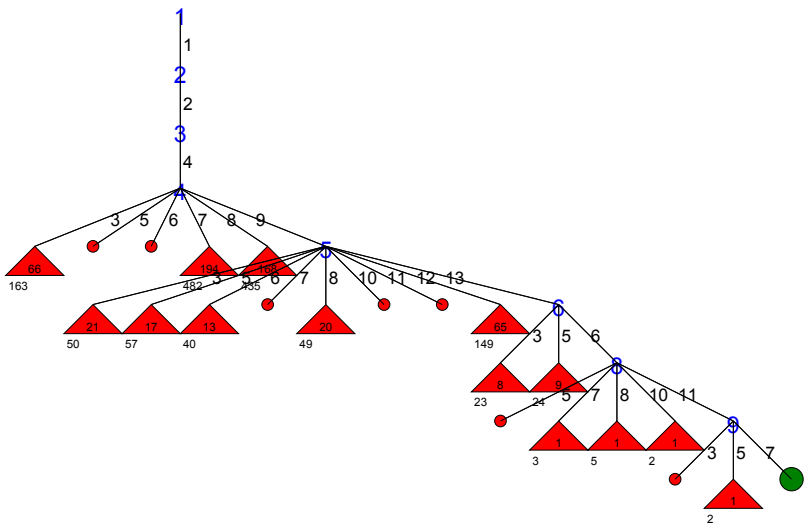
Other Problem Sizes

	Basic Model	
Size	Backtrack	Time
10	4	0.00
11	118	0.08
12	50	0.05
13	335	0.36
14	5008	6.23
15	47332	68.92
16	157773	271.22
17	1641685	3278.19
18	115745	283.97

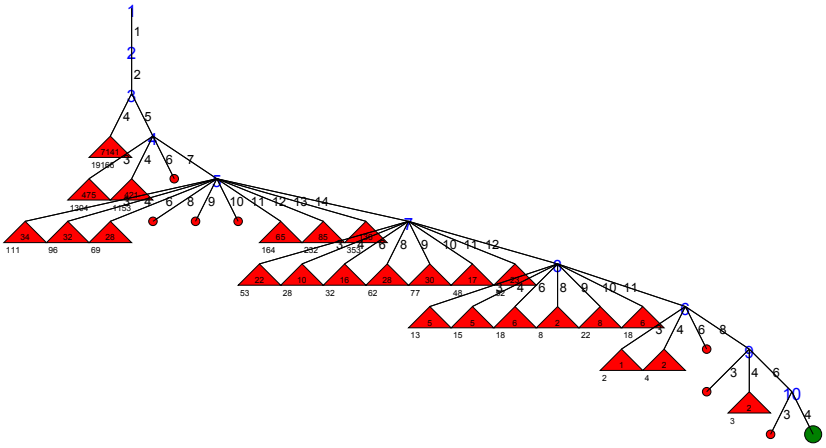
Search tree (Size 12)



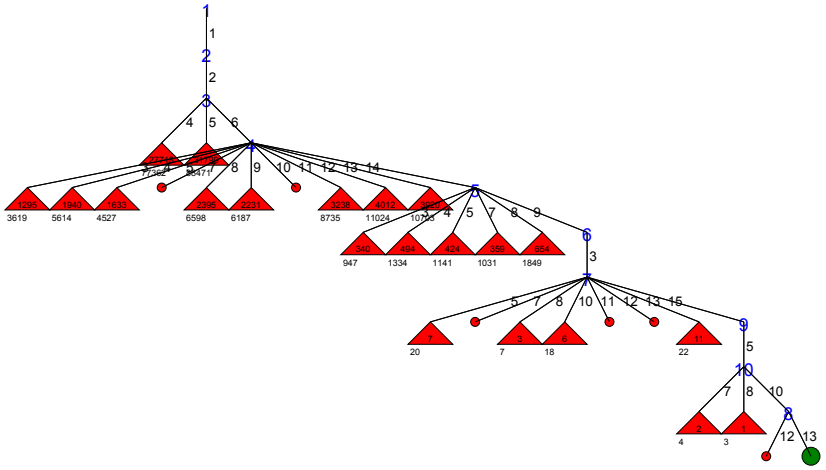
Search tree (Size 13)



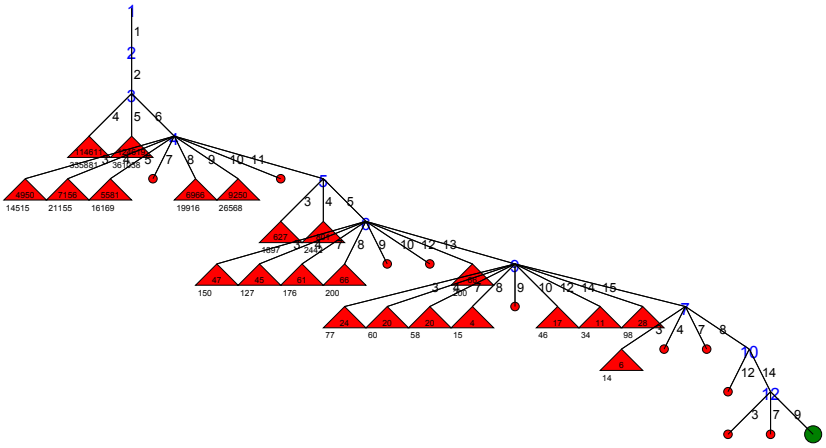
Search tree (Size 14)



Search tree (Size 15)



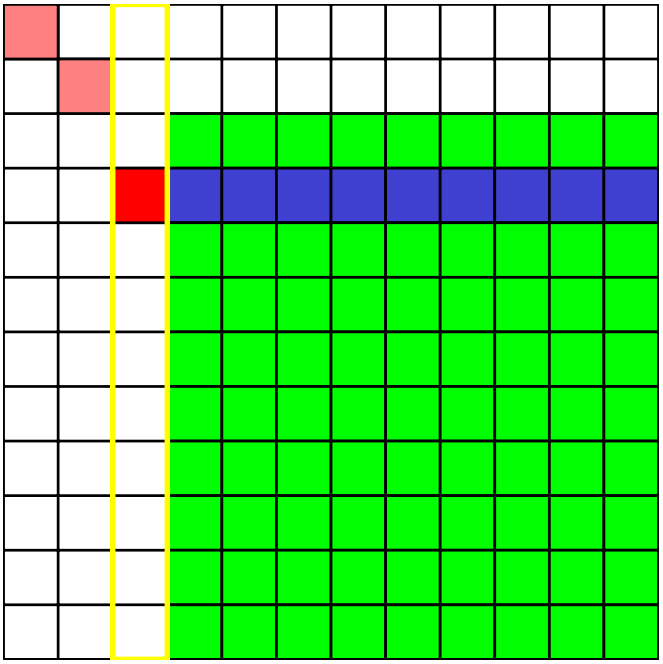
Search tree (Size 16)



Observation

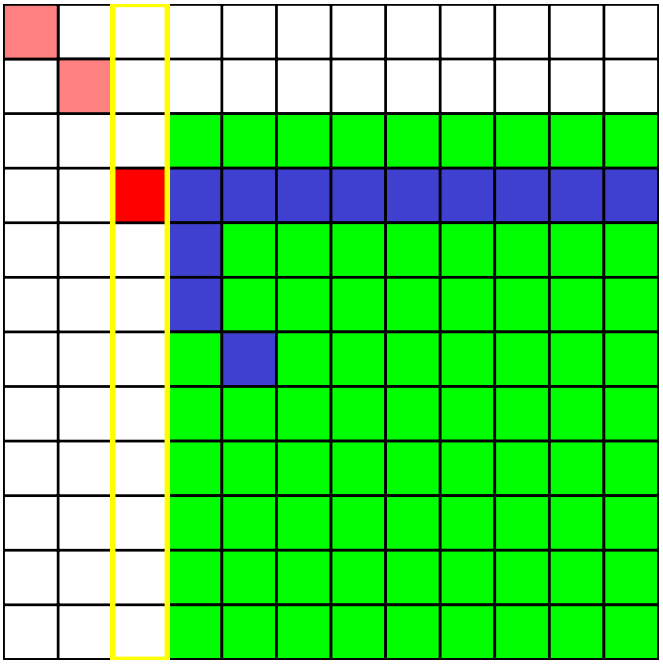
- Problem becomes harder with increasing size
- Failures occur from level 3 down
- Deep backtracking required to undo wrong choices
- Value selection not working, have to explore all choices
- Increase not uniform

Missing Propagation



The model is doing this

Missing Propagation



It could be doing that!

Changed Differences

```
diffs(_,[]).
diffs(L,[H|T]):-
        diff_pairs(L,[H|T],Diffs,Triples),
        impose_triples(Triples,[]),
        alldifferent(Diffs),
        diffs(L,T).

diff_pairs(_,[],[],[]).
diff_pairs([X|X1],[Y|Y1],[D|D1],[t(X,Y,D)|T]):-
        X #= Y+D,
        diff_pairs(X1,Y1,D1,T).
```

Changed Differences

```
impose_triples([],_).
impose_triples([t(X,Y,D)|R],Others):-
        suspend(impose_triple(D,R),4,D->inst),
        suspend(impose_triple(D,Others),4,D->inst),
        impose_triples(R,[t(X,Y,D)|Others]).
```

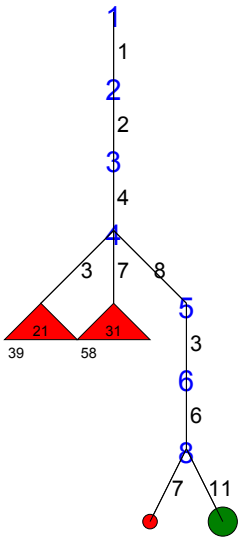
Changed Differences

```
impose_triple(_D,[]).
impose_triple(D,[t(X,Y,_)|R]):-
        (var(X) ->
            suspend(impose_one_triple12(D,X,Y),
                    4,X->inst)
        ;
            impose_one_triple12(D,X,Y)
        ),
%       ...
        impose_triple(D,R).

impose_one_triple12(D,X,Y):-
        V is X-D,
        Y #\= V.
```

Further Model Improvements

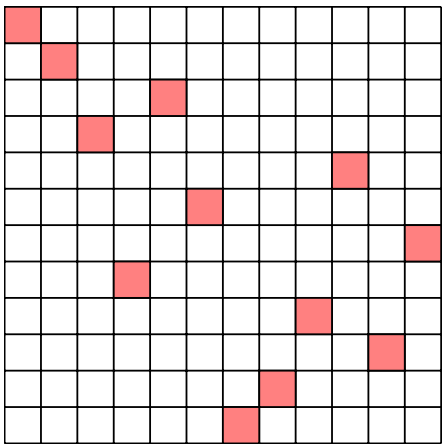
- DC consistent `alldifferent` between variables
- (DC consistent `alldifferent` between differences)
- DC difference constraint

Improved Model

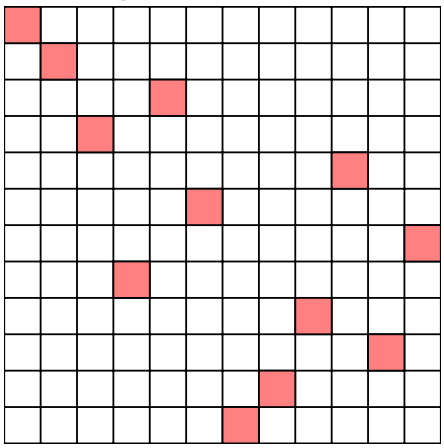


Comparison (Solutions)

Initial Model

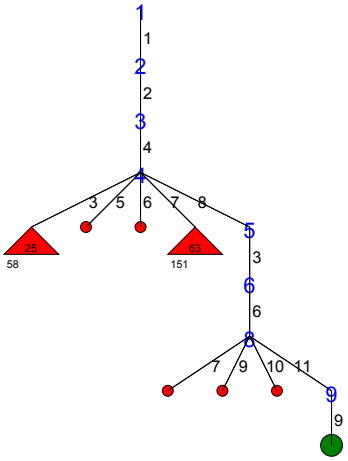


Improved Model

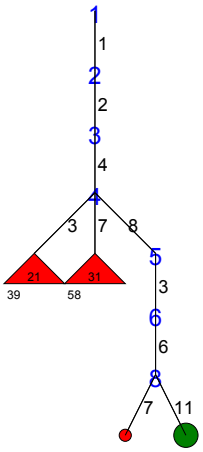


Comparison (Search Trees)

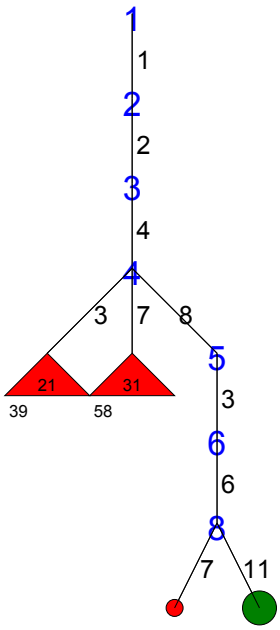
Initial Model



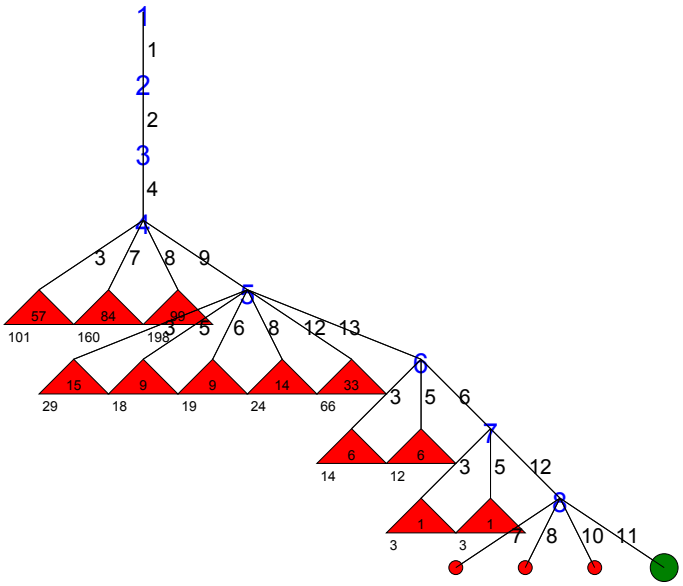
Improved Model



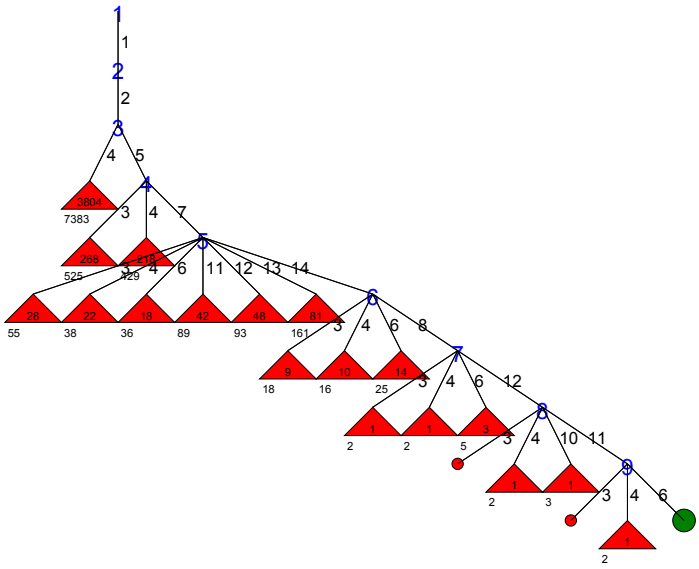
Search tree (Size 12)



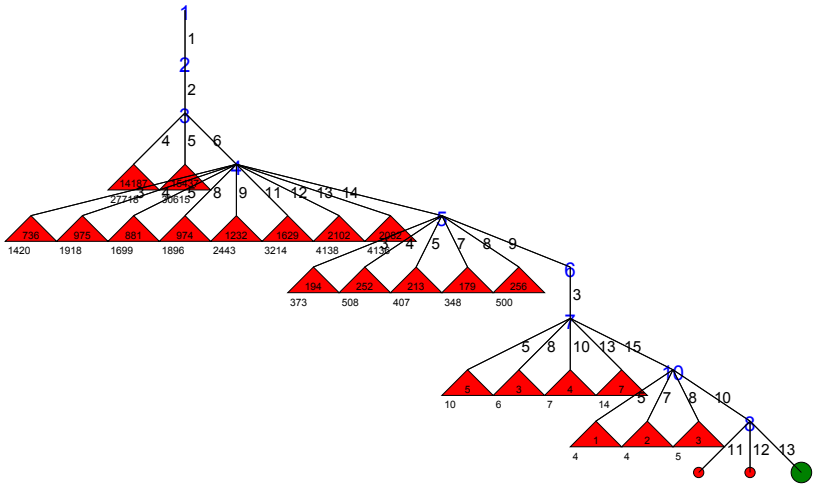
Search tree (Size 13)



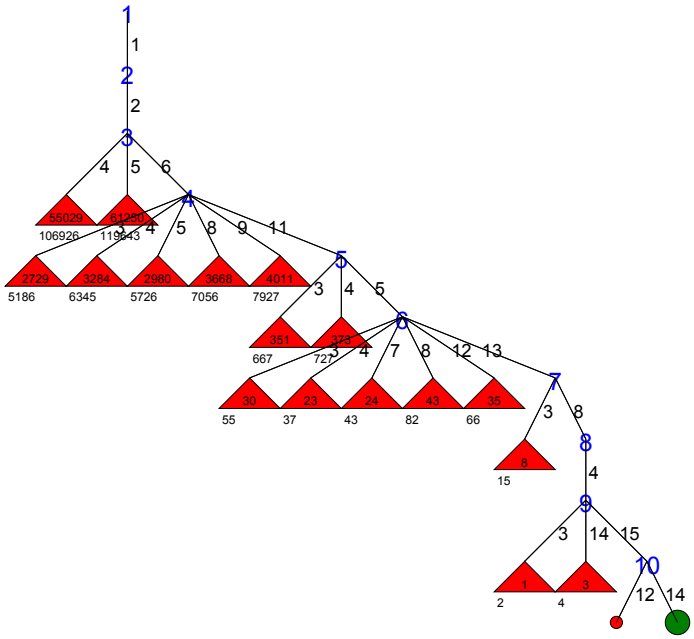
Search tree (Size 14)



Search tree (Size 15)

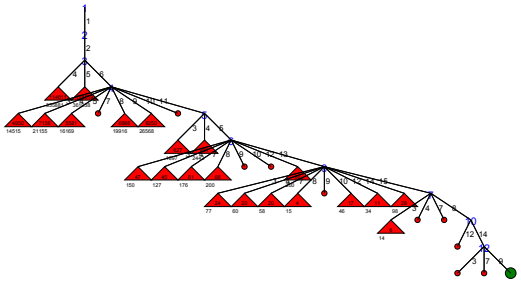


Search tree (Size 16)

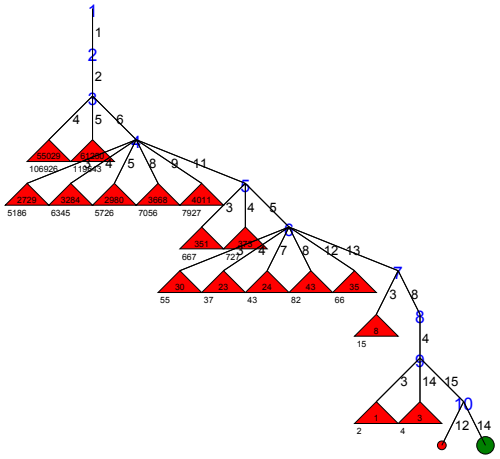


Comparison (Search Tree, size 16)

Initial Model



Improved Model



Other Problem Sizes

	Basic Model		Improved Model	
Size	Backtrack	Time	Backtrack	Time
10	4	0.00	4	0.16
11	118	0.08	77	1.44
12	50	0.05	31	0.94
13	335	0.36	216	6.22
14	5008	6.23	2875	95.94
15	47332	68.92	25820	1046.75
16	157773	271.22	84161	4099.52
17	1641685	3278.19	825590	49371.02
18	115745	283.97	55102	4530.83

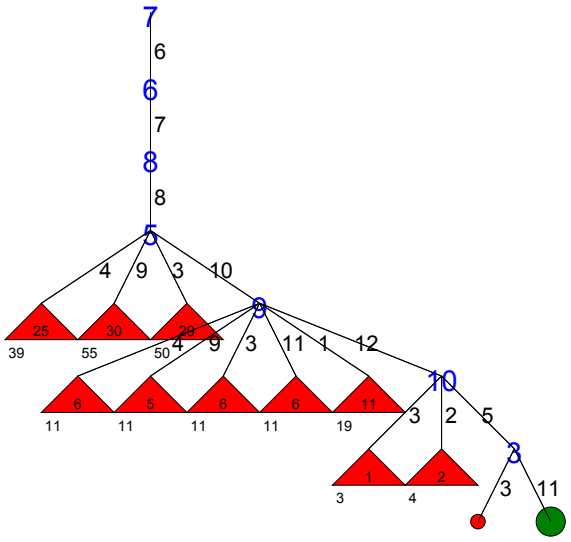
Observation

- Changes reduce backtracks by 50%
- But, run times explode
- Being clever does not always pay
- Or, perhaps, we did not make the right improvements?

Change of Search Strategy

- Idea: Make more difficult choices first
- Reorder variables to start from middle
- Assign values starting in middle

Labeling From Middle



Other Problem Sizes

	Improved Model		Improved Model, Middle	
Size	Backtrack	Time	Backtrack	Time
10	4	0.16	1	0.01
11	77	1.44	13	0.03
12	31	0.94	72	0.26
13	216	6.22	513	1.81
14	2875	95.94	589	2.37
15	25820	1046.75	7840	34.30
16	84161	4099.52	13158	63.91
17	825590	49371.02	56390	298.16
18	55102	4530.83	19750	115.64

Observation

- Big improvement in backtracks and time
- Not for all problem sizes
- Question: Do we need improvement of model for this to work?
- Experiment: Run changes search routine on basic model

Labeling Basic Model from Middle

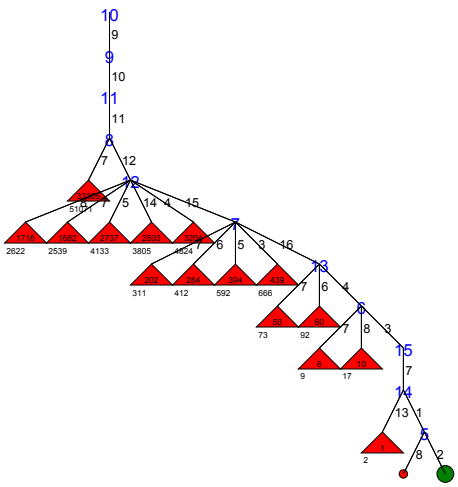
	Basic Model		Basic Model, Middle	
Size	Backtrack	Time	Backtrack	Time
10	4	0.00	1	0.00
11	118	0.08	17	0.01
12	50	0.05	97	0.09
13	335	0.36	644	0.74
14	5008	6.23	746	1.03
15	47332	68.92	10041	16.03
16	157773	271.22	17005	31.12
17	1641685	3278.19	73080	152.72
18	115745	283.97	28837	60.97

Comparison: Model Impact

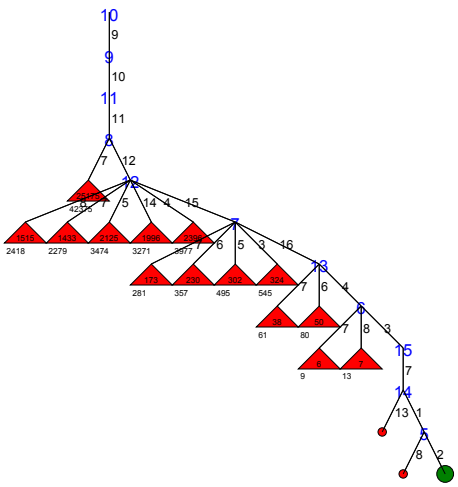
	Basic Model,Middle		Improved Model, Middle	
Size	Backtrack	Time	Backtrack	Time
10	1	0.00	1	0.01
11	17	0.01	13	0.03
12	97	0.09	72	0.26
13	644	0.74	513	1.81
14	746	1.03	589	2.37
15	10041	16.03	7840	34.30
16	17005	31.12	13158	63.91
17	73080	152.72	56390	298.16
18	28837	60.97	19750	115.64
19	1187618	3174.72	1044751	4474.56

Comparison (Search Tree, size 18)

Initial Model



Improved Model



Observation

- Search strategy does not depend on model
- Variable selection is the same!
- Basic model is about two times faster
- About 50% more backtrack steps
- Again, sometimes reasoning does not pay!
- Better search strategy pays off dramatically

A Different Model

- Model shown is not the only way to express problem

0/1 Models

- SAT (Minisat)
- Pseudo Boolean (Minisat+)
- MIP (Coin-OR)

0/1 Models: Variables

- X_{iv}: Variable i takes value v
- D_{ijv}: Difference between variables i and j is v

MIP Model: Constraints

- alldifferent between variables
  - $\sum_i X_{iv} = 1$
  - $\sum_v X_{iv} = 1$
- alldifferent between differences
  - $\sum_v D_{ijv} = 1$
  - $\sum_{i-j=c} D_{ijv} \leq 1$
- link between variables and differences
  - $D_{ijv} = \sum_{v1=v2+v} X_{iv_1} X_{jv_2}$

More Information

- `http://www.costasarrays.org/`