Chapter 11: Limits of Propagation (Costas Array)

Helmut Simonis

Cork Constraint Computation Centre
Computer Science Department
University College Cork
Ireland

ECLiPSe ELearning Overview

Licence

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.
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)
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.
- 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
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).
### Other Problem Sizes

<table>
<thead>
<tr>
<th>Size</th>
<th>Backtrack</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>118</td>
<td>0.08</td>
</tr>
<tr>
<td>12</td>
<td>50</td>
<td>0.05</td>
</tr>
<tr>
<td>13</td>
<td>335</td>
<td>0.36</td>
</tr>
<tr>
<td>14</td>
<td>5008</td>
<td>6.23</td>
</tr>
<tr>
<td>15</td>
<td>47332</td>
<td>68.92</td>
</tr>
<tr>
<td>16</td>
<td>157773</td>
<td>271.22</td>
</tr>
<tr>
<td>17</td>
<td>1641685</td>
<td>3278.19</td>
</tr>
<tr>
<td>18</td>
<td>115745</td>
<td>283.97</td>
</tr>
</tbody>
</table>
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
The model is doing this

It could be doing that!
**Changed Differences**

```prolog
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**

```prolog
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]).
```

*Helmut Simonis*  
*Limits of Propagation*  
23
Changed Differences

```prolog
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)
Comparison (Search Tree, size 16)

Initial Model

![Initial Model Diagram]

Improved Model

![Improved Model Diagram]

Other Problem Sizes

<table>
<thead>
<tr>
<th>Size</th>
<th>Basic Model</th>
<th>Improved Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Backtrack</td>
<td>Time</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>118</td>
<td>0.08</td>
</tr>
<tr>
<td>12</td>
<td>50</td>
<td>0.05</td>
</tr>
<tr>
<td>13</td>
<td>335</td>
<td>0.36</td>
</tr>
<tr>
<td>14</td>
<td>5008</td>
<td>6.23</td>
</tr>
<tr>
<td>15</td>
<td>47332</td>
<td>68.92</td>
</tr>
<tr>
<td>16</td>
<td>157773</td>
<td>271.22</td>
</tr>
<tr>
<td>17</td>
<td>1641685</td>
<td>3278.19</td>
</tr>
<tr>
<td>18</td>
<td>115745</td>
<td>283.97</td>
</tr>
</tbody>
</table>
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

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

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

<table>
<thead>
<tr>
<th>Size</th>
<th>Basic Model, Middle</th>
<th>Improved Model, Middle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Backtrack</td>
<td>Time</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>97</td>
<td>0.09</td>
</tr>
<tr>
<td>13</td>
<td>644</td>
<td>0.74</td>
</tr>
<tr>
<td>14</td>
<td>746</td>
<td>1.03</td>
</tr>
<tr>
<td>15</td>
<td>10041</td>
<td>16.03</td>
</tr>
<tr>
<td>16</td>
<td>17005</td>
<td>31.12</td>
</tr>
<tr>
<td>17</td>
<td>73080</td>
<td>152.72</td>
</tr>
<tr>
<td>18</td>
<td>28837</td>
<td>60.97</td>
</tr>
<tr>
<td>19</td>
<td>1187618</td>
<td>3174.72</td>
</tr>
</tbody>
</table>

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
  - $D_{ijv} = \sum_{v1=v2+v} X_{iv1} X_{jv2}$

More Information

- [http://www.costasarrays.org/](http://www.costasarrays.org/)