Assignment 1 - Extra non-submission tasks **Kim Hammar**
ID2204 **Mallu**
Constraint Programming Due Date: 7 April 2017

Search Heuristics

Can you find a better heuristic than first-fail for the following crypto arithmetic puzzle?

$$DONALD + GERALD = ROBERT$$

Comparison between first-fail min-value (ffminv), first-fail max-value (ffmaxv), first-fail split-max(ffsmax), first-fail split-min (ffsmin).

Search Heuristic	**nodes**	**failures**
ffminv	178	87
ffmaxv	50	23
ffsmax	50	23
ffsminv	180	87

First-fail max-value and First-fail split-max are the best heuristics I've found. The main difference is that when picking max-value instead of min the failed sub-trees is a lot smaller. I.e trying the larger values first give more constraints and more propagation possible.

Is Propagation Compositional?

Assume that A, B, C, and X are finite domain variables and that you are given the following statement:

$$A + B + C + B = X$$

It looks intuitive that you can transform this into

`IntVar U`
$A + B = U$
$U + B + C = X$

Implement the two programs and analyze what is different.

The two solution sets are identical which is shown by the program. Does this mean propagation is compositional?

Is there any other aspect in which they compute differently? In which aspect?

Propagation is not compositional, propagations communicate through the variables values and assignments.

Yes they result in different search trees despite the fact that they are using the same search-heuristics (branching strategies):

First we have script $S1$ which models $A + B + C + B = X$ where $D_A = D_B = D_C = D_X = \{1, 2, 3, 4, 5\}$

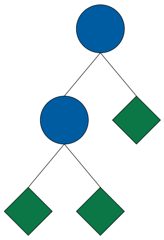

Figure 1: First-fail Minimum-value for Script S1

Second we have script $S2$ which models $A + B = U; U + B + C = X$ where $D_A = D_B = D_C = D_X = D_U = \{1, 2, 3, 4, 5\}$

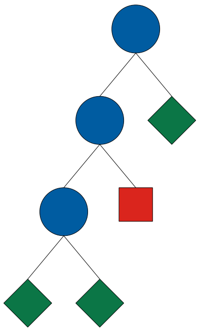

Figure 2: First-fail Minimum-value for Script S2

What is happening?

In S1 we only post one constraint that will create propagators for that single constraint while in S2 there are two constraints posted. In the search tree of S2 the propagator for $A + B = U$ is executed first which results in interleaving of constraint propagation and variable assignment until the constraint is satisfied, in particular the assignments $A = 1, B = 2, U = 3$ is tried which leads to a (red) failed node since that means that there are no satisfying assignments for C and X that gives $U + B + C = X$. This is because $U + B = 5$ and the maximum value of X is 5 and the minimum value of C is 1.

In S1 already before any assignment is made constraint propagation can infer that B must be 1. This is the main difference. In S2 the constraint propagation were not able to resolve this initially since just looking at the constraint $U + B + C = X$ B could be other things than 1. But if the constraint propagation were able to consider both constraint simultaneously it would find out that B could not be something else than 1, looking at each constraint in isolation B does not have to be only 1.

Search Heuristics for N-Queens

First-Fail Minimum-Value

The standard first-fail heuristic is the first-fail minimum-value heuristic, the search tree for it is shown in the figure below.

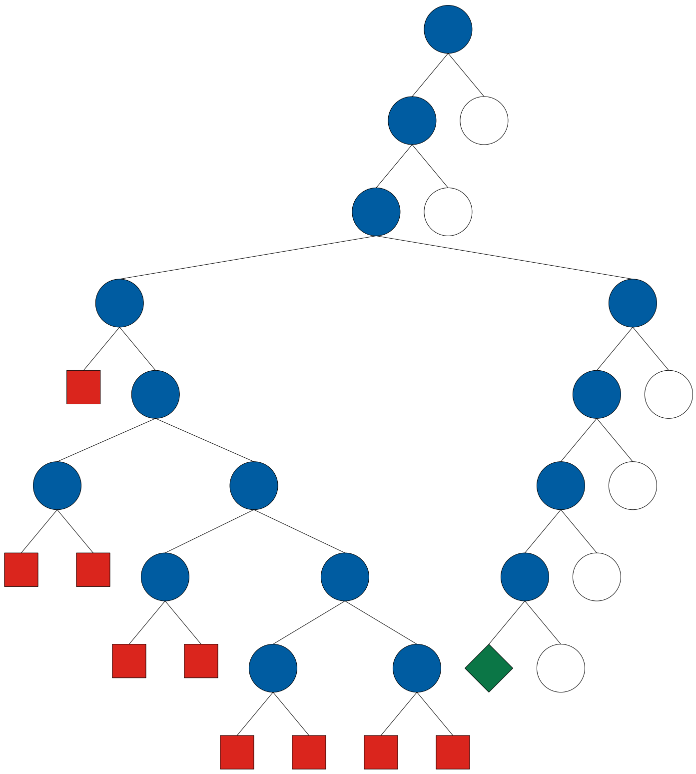

Figure 3: First-fail Minimum-value for queen problem instance of size 10

It visits 25 nodes and 9 failures before finding the first solution.

First-Fail Median-domain-value

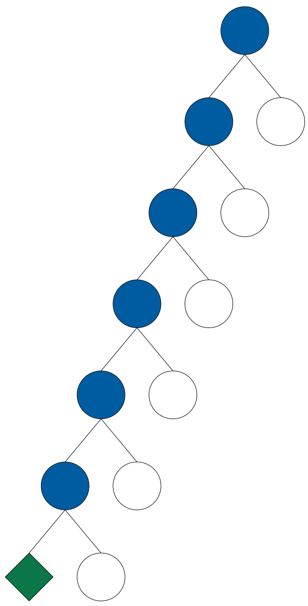

Figure 4: First-fail Median-value for queen problem instance of size 10

The search visits 7 nodes and 0 failures before finding the first solution.

Knight-move heuristic (min-min median-domain-value)

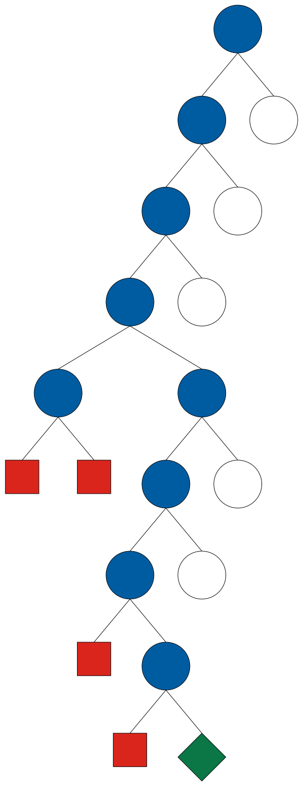

Figure 5: Knight-move Median-value for queen problem instance of size 10

This search visits 14 nodes and 4 failures before finding the first solution.

Knight-move heuristic (min-min val-min)

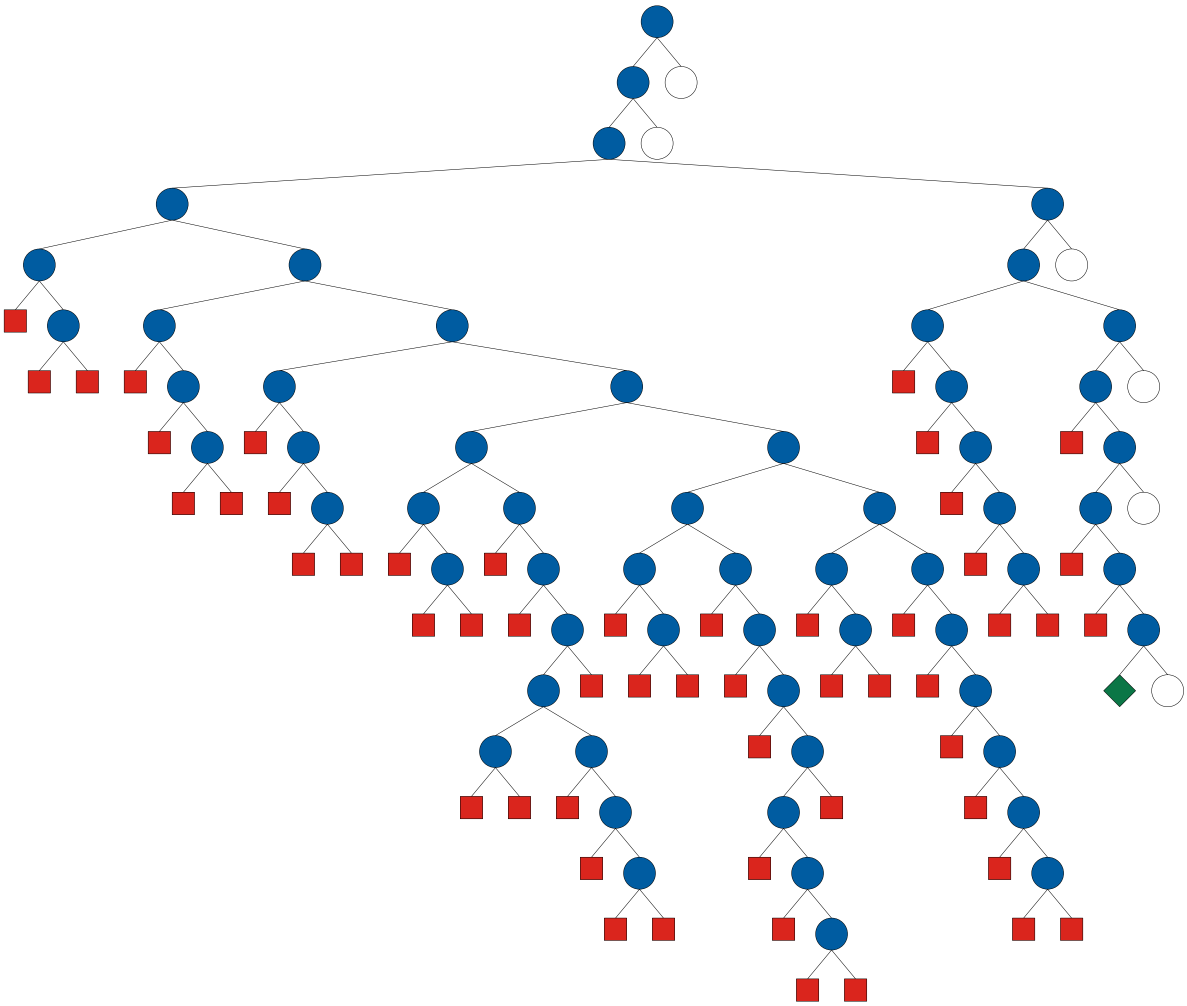

Figure 6: Knight-move minimum-value for queen problem instance of size 10

This search visits 113 nodes and 53 failures before finding the first solution.

Conclusion

Clearly for the problem instance $n = 10$, first-fail median-domain-value was the most effective search heuristic. For $n = 10$ we can order the heuristics as follows:

first-fail median-domain-value (ffmdv) $\succ$ *knight-move median-domain-value (kmmdv)* $\succ$ *first-fail minimum-value (ffmv)* $\succ$ *knight-move minimum-value (kmmv)*

However what heuristic is best depends a lot on the problem instance. For example as shown in the table below, the first-fail heuristic greatly out-performs the knight-move heuristic for larger n. Also the minimum-value first-fail outperforms the medium-value first-fail for larger n even though medium-value was more efficient for n=10.

Search Heuristic	**n**	**nodes**	**failures**
ffmdv	10	25	9
kmmdv	10	14	4
ffmv	10	25	9
kmmv	10	113	53
ffmdv	100	202	60
kmmdv	100	?	? - No solution in reasonable time
ffmv	100	138	22
kmmv	100	?	? - No solution in reasonable time
ffmdv	1000	2993	998
kmmdv	1000	?	? - No solution in reasonable time
ffmv	1000	996	2
kmmv	1000	?	? - No solution in reasonable time

Why is the heuristic called knight-move heuristic? Because the move that threatens the most other queens and is safe is the knight's move typically.