Sampling Strategies and Variable Selection in Weighted Degree Heuristics

Diarmuid Grimes and Richard J. Wallace

1 Weighted-Degree and Random Probing

Variable Ordering Heuristics

- Classical heuristics - decisions based on initial state or current state of search.
- Recent heuristics use within-problem learning, collect information relevant to decisions.
- Example: **Weighted-Degree** heuristic of (Boussemart et al.) which has been shown to be extremely efficient.
- Alternative method: **Random Probing**

- Both procedures (weighted-degree with systematic search, and random probing) store information as constraint weights: domain wipeout $\rightarrow$ increment weight of constraint whose propagation caused wipeout.
- Both procedures - weights can be used to boost maximum-degree or minimum domain/degree heuristics by replacing degree with weighted degree.
- Random probing - Information gathering phase where search is repeated with random variable selection. This is followed by complete search with boosted weighted-degree heuristic.

Both procedures follow 2 principles

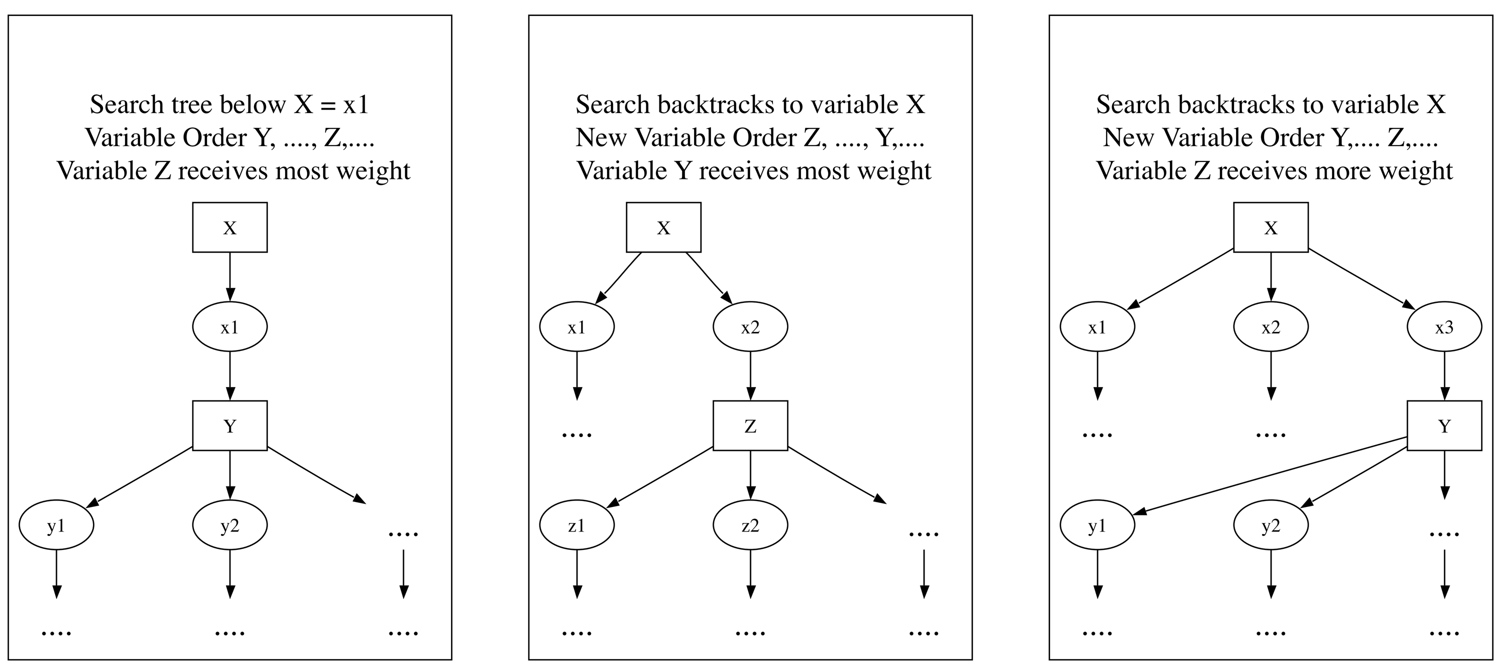
- **Fail-First Principle**: To succeed, try first where you are most likely to fail.
- **Contention Principle**: If a constraint is identified as a source of contention, then variables associated with that constraint are more likely to cause failure upon instantiation.

Goals of Present study

- To characterise the nature and quality of the information gained during sampling.
- To characterise the manner in which this information is used during subsequent search.

2 Weighted-degree with systematic search

- Heuristic has potential limitation: no information when making initial selections.
- With good variable ordering heuristics, there is normally a small failure depth range:
  - begins after K instantiations ($K > 1$, usually $K > 4,5$)
  - ends after K'th instantiation ($K' \ll n$)
- Therefore no weight information available for first K choices.
- Weighted-degree - $\triangle$wt of variable x potentially subject to negative feedback, resulting in "variable convection" effect (as illustrated in toy example below):
  - variables selected at depth $(K + e)$ receive little or no weight increase
  - variables deeper in search receive more weight
  - if search returns to depth $(K + e)$ the ordering will change
  - variables with contentious constraints rise up the ordering, upon backtracking are selected higher in search tree, receive little weight, fall back down the ordering



3 Experimental Methods

- Experiments were performed on 100 random binary problems, with 50 variables and maximum domain size of 10.
- The random probing approach involved restarting search a fixed number of times with a fixed cutoff in terms of nodes.
- variable selection was random during these "probes" of the search space, constraints were weighted as normal but not used to guide search.
- This was followed by complete search using the $dom/wdeg$ variable heuristic.
- Weights were not updated during the run to completion in order to assess the quality of information learnt solely by the random probes.
- In all experiments, results reported are the average search nodes over 100 problems when solving (nodes explored during the probing phase were not included for the random probing approach).
- Each data point for the random probes is the average of ten experiments.

4 Sampling based on various measures of contention

The normal constraint weighting method is to increment the weight of a constraint by 1 each time it causes a domain wipeout during propagation. We consider the following alternatives:

- "wipe by del": wipeout-tallies in which in each case the relevant constraint weight is increased by the size of the domain reduction leading to the wipeout
- "alldel": tallies of all deletions; i.e. whenever a domain is reduced in size during constraint propagation, the weight of the constraint involved is incremented by 1
- "alldel by #del": tallies of all deletions where constraint weights are increased by the size of the domain reduction
- "dels/nowipe": tallies of all deletions *except* those leading to a wipeout

	dom/wdeg	random probe-dom/wdeg (40R 50C)
normal weighting	1538	1265
wipe by #del	1592	1261
alldel	1523	1426
alldel by #del	1496	1461
dels/nowipe	1530	1499

As one can see from the table above, all sampling methods performed well when used with the $dom/wdeg$ heuristic without restarting. However for the random probing approach, sampling directly associated with failures provides information of a better quality.

5 Sampling based on different search procedures

We compared the quality of information learnt by three different constraint-weighting search procedures on the same problem set as above:

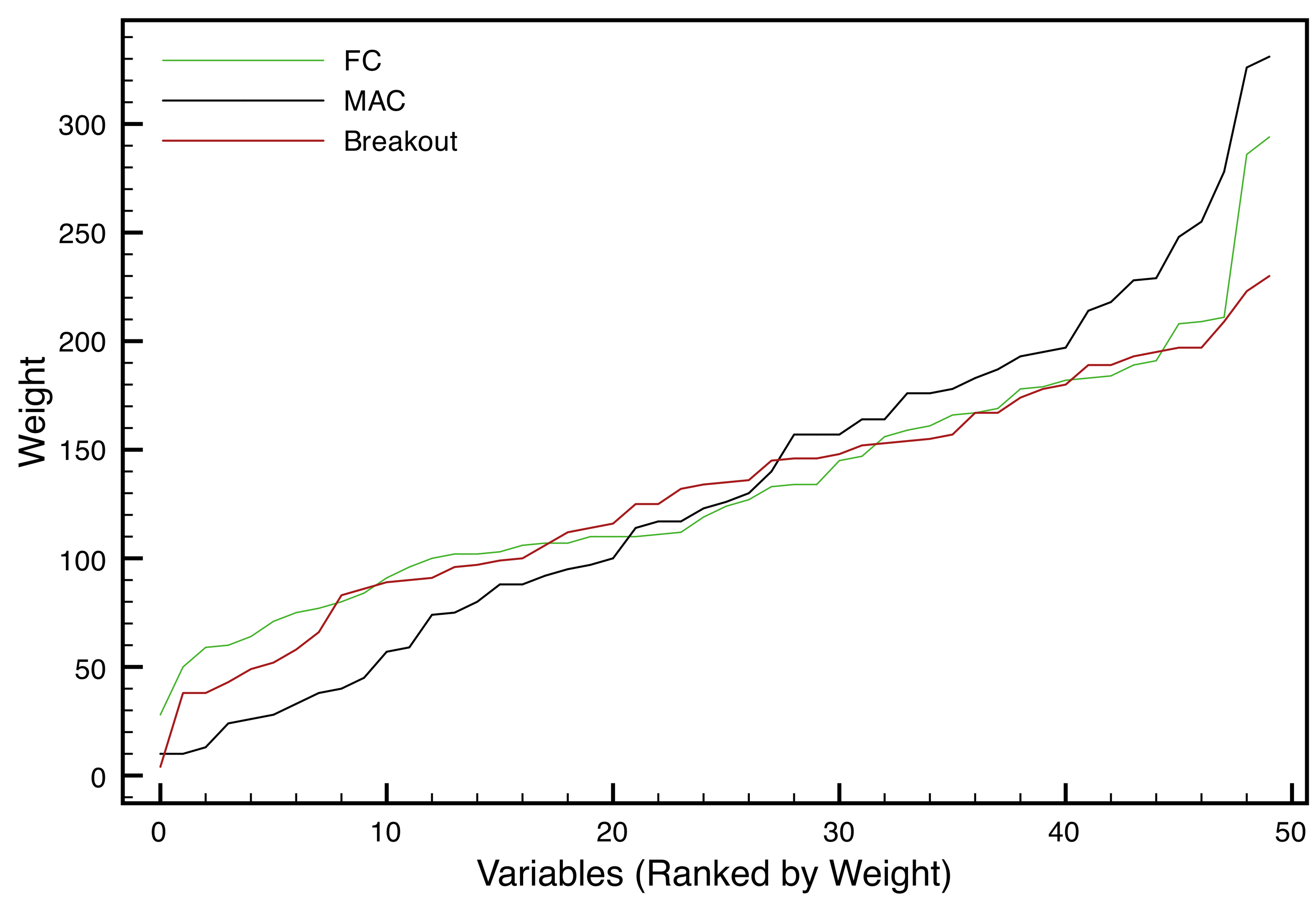
- Morris' breakout algorithm - local search algorithm where constraints in conflict are weighted. Stopped after total weight increase of 3000.
- Forward checking (FC) with random probing, 100 restarts, cutoff of 30 failures per probe.
- MAC with random probing, 100 restarts, cutoff of 30 failures per probe.

The weights learnt by the procedures were then used by a weighted-degree heuristic to solve the problem using complete search with MAC. All procedures generated the same amount of information (total weight increase of 3000). Results are average search nodes for the run to completion over the 100 problems.

	probe-$dom/wdeg$	probe-$wdeg$
Breakout Method	1863	3890
FC-probes	1492	3198
MAC-probes	1198	1595

MAC produced information of the highest quality in the probing phase. It should be noted that the only difference between the experiments was the approach used in generating the weight profiles for solving the problems.

Below is a plot of the weight profiles produced by the three methods on a sample problem. The variables are ranked according to their weight after information gathering.



Cork Constraint Computation Centre, University College Cork, Ireland (Supported by SFI Grants 00/PI.1/C075 and 05/IN/1886)

eMail: {d.grimes|r.wallace}@4c.ucc.ie