A Qualitative Analysis of Search Behavior: A Visual Approach

Ian Howell,[1,2] Robert J.Woodward,[1] Berthe Y.Choueiry,[1] and Hongfeng Yu[2]
[1]Constraint Systems Laboratory • University of Nebraska-Lincoln • USA
[2]Visualization Laboratory • University of Nebraska-Lincoln • USA

Motivation

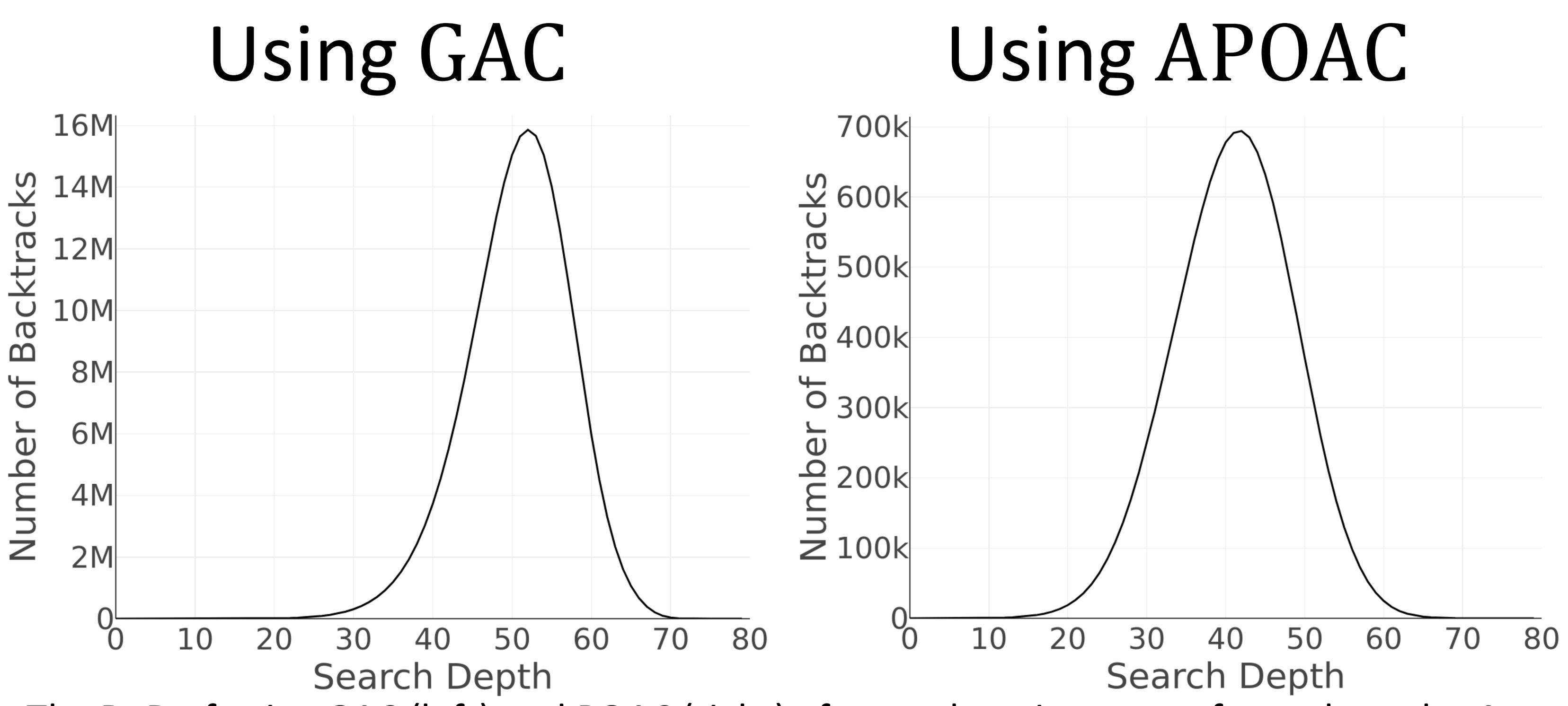
Thrashing is the main malady of Backtrack (BT) search through CSP instances.

Contributions

1. Analyze locations of thrashing and propagation strength using Backtracks per Depth (BpD) and Calls per Depth (CpD).
2. Efficiently analyze the evolution of location of where search struggles and performs.

1. Search Efficiency

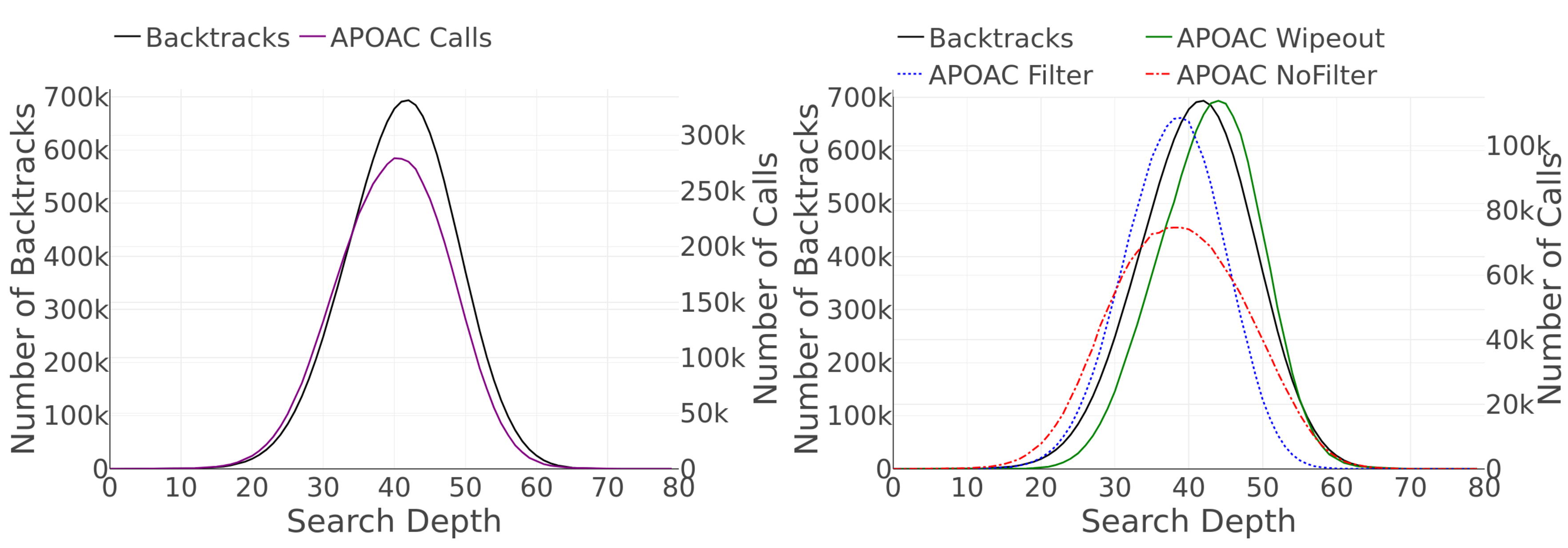
Tracking the Backtracks per Depth shows where different consistencies find inconsistency in the instance. Sonner detecting inconsistency represents less of the search space needing to be explored.



The BpD of using GAC (left) and POAC (right) after each assignment of search on the 4-insertions 3-3 instance.

	GAC	APOAC
Peak Depth	52	42
Peak Backtracks	15,863,603	693,829
CPU Time (sec)	>8,099.9	2,981.9

Consistency can filter some values, wipeout the domain of a variable, or do no filtering, wasting the time of the solver. We track this through the Calls per Depth.

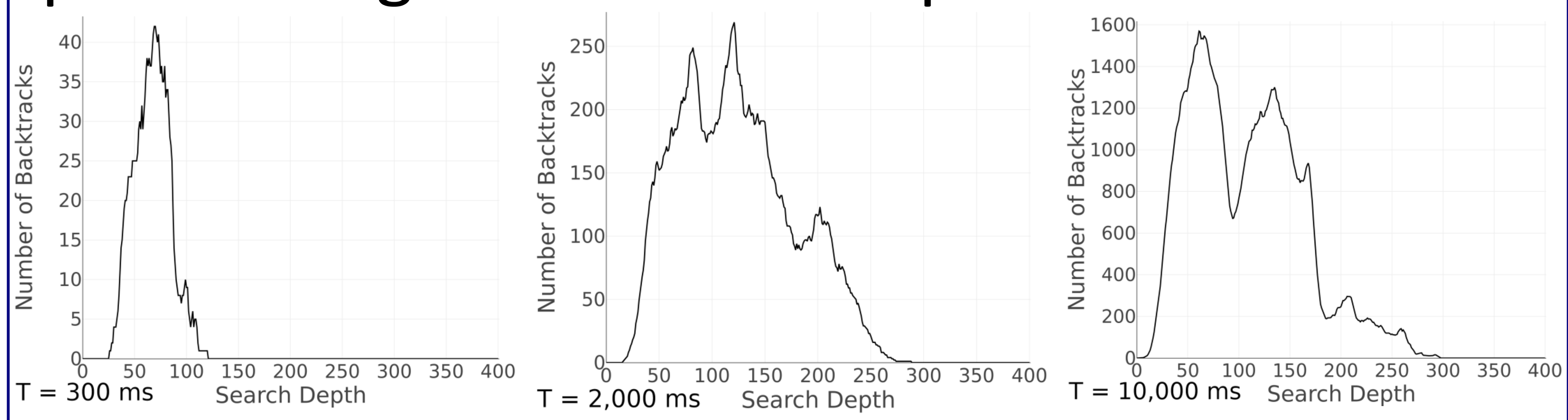


The CpD of using POAC after each assignment of search on the instance 4-insertions 3-3, cumulative (left) and split by filtering amount (right).

Here, APOAC struggles at depth 38, the peak of both no filtering occurring and partial filtering.

2. Feature Evolution

To efficiently analyze the change in peaks throughout search, we use regimes, partitionings of time with equivalent behavior.

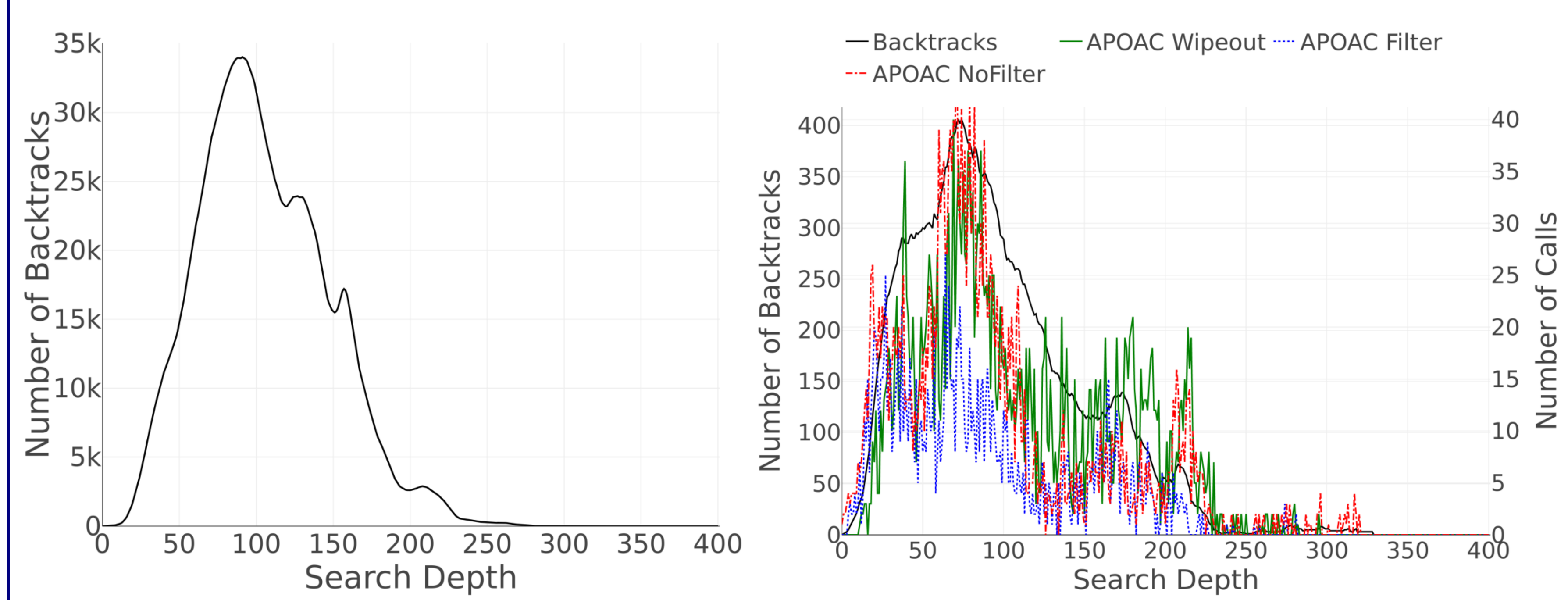


Three SHAPE regime representatives of the BpD of GAC on pseudo-aim-200-1-6-4 instance.

3. Case Study: PREPEAK$^+$

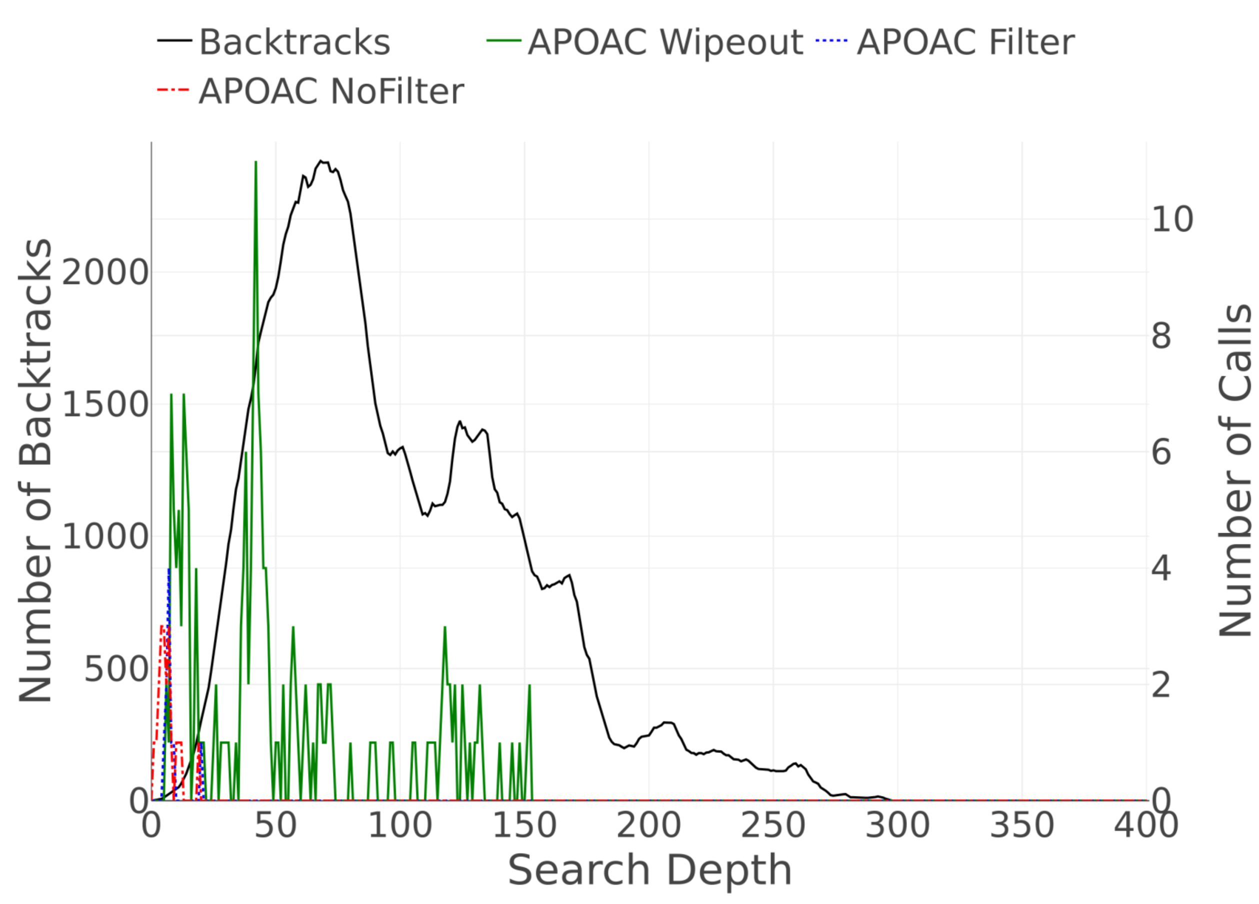
GAC, APOAC, and PREPEAK$^+$ on the pseudo-aim-200-1-6-4 from the Lecoutre benchmarks.

	GAC	APOAC	PREPEAK$^+$
CPU time (sec)	185,045	66,816	17,836
#NV	3,978,074	47,457	284,289
$\max_{BpD}$	34,023	407	2,421
#HLC calls		7,739	228



The BpD of GAC (left) and BpD and CpD of POAC superimposed (right) on pseudo-aim-200-1-6-4

On this instance, GAC does not do enough filtering and explores far too much of the search space. APOAC does much better, but doesn't filter anything too often. PREPEAK$^+$ adaptively uses GAC and POAC, reducing the number of no-filters from POAC, achieving the best results.



The BpD and CpD of PREPEAK$^+$ superimposed on pseudo-aim-200-1-6-4

Experiments were conducted on the equipment of the Holland Computing Center at the University of Nebraska-Lincoln.
This research was supported by NSF Grant No. RI-1619344.
Howell was supported by a UCARE grant.

Paper: IJCAI 2018. July 5th, 2018

Motivation: Searching the space of a Constraint Satisfaction Problem (CSP) is an NP-Complete task. The complete method of Backtrack Search suffers heavily from thrashing, or search similar path repeatedly and failing to find solutions. We visualize this thrashing through tracking the number of times search backtracks at each depth of the search tree. In addition, we propose to track the evolution of features over time, to track where the difficult part of search lies over time.

To visualize thrashing in d-way backtrack search of Constraint Satisfaction Problems (CSP) and to determine the change in location of where search struggles and performs over time.

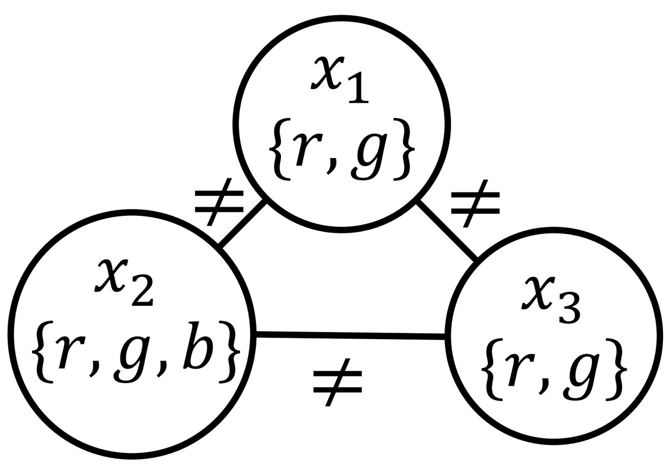
1. Background

Variables: $\{x_1, x_2, x_3\}$

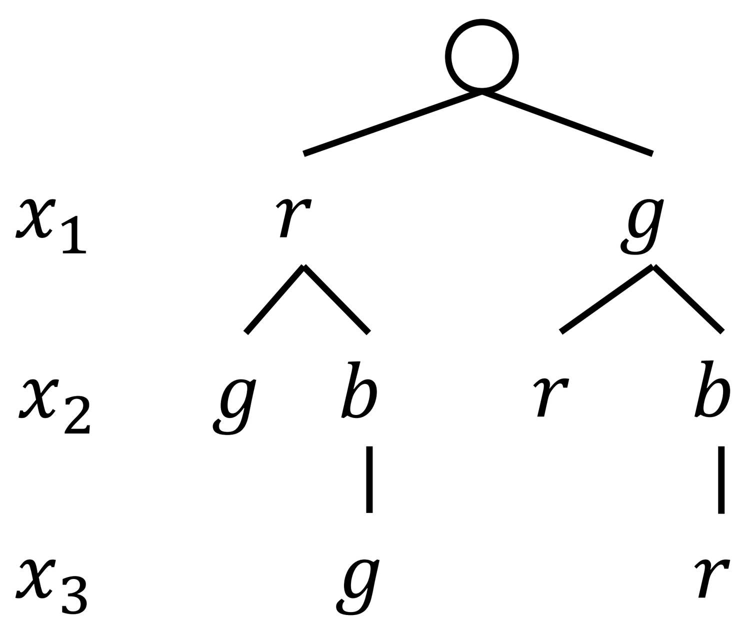
Domains: $D(x_1) = \{r, g\}$, $D(x_2) = \{r, g, b\}$, $D(x_3) = \{r, b\}$

Constraints: $\{x_1 \neq x_2, x_1 \neq x_3, x_2 \neq x_3\}$

Primal Graph:



D-way Backtrack (BT) search:



Backtracks:

$(r, g) \rightarrow (r)$

$(g, r) \rightarrow (g)$