A Reactive Strategy for High-Level Consistency During Search

R.J. Woodward[1,2] B.Y. Choueiry[1] Christian Bessiere[2]

[1]Constraint Systems Laboratory, University of Nebraska-Lincoln
[2]CNRS, University of Montpellier

Acknowledgements

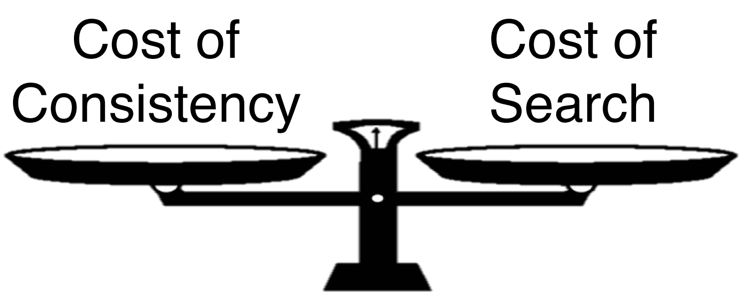
- Experiments conducted at UNL's Holland Computing Center
- NSF Grant RI-111795, RI-1619344
- NSF GRF and Chateaubriand Fellowship
- ANR Demograph (ANR-16-CE40-0028) and Contredo (ANR-16-CE33-024)

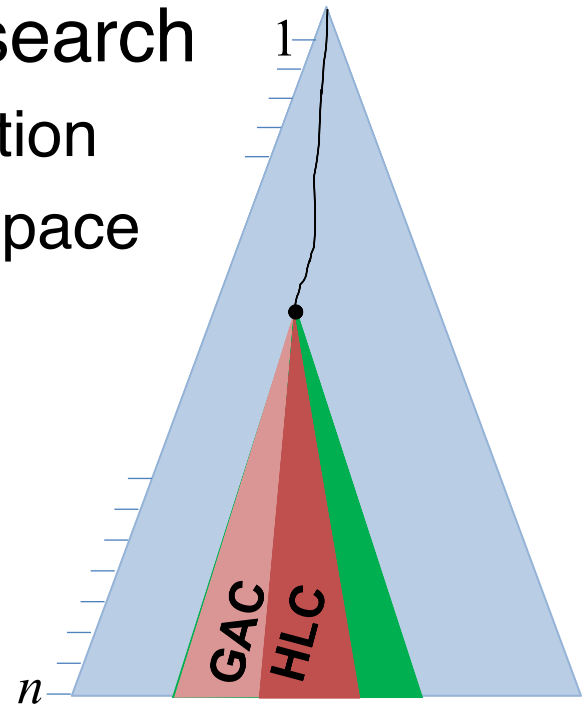
Context

- Solving a Constraint Satisfaction Problem (CSP)
  - Conditioning: Backtrack search
  - Inference: Enforcing consistency
    - Consistency **properties** (e.g., GAC)
    - Constraint propagation **algorithms**
- Consistency during search
  - Constraint Programming solvers: GAC or weaker
  - CSP research: GAC or stronger
- Our focus: Higher-level consistency (HLC)

Lesson and Problem

- Maintaining consistency during search
  - Enforced at each variable instantiation
  - Prunes subtrees, reduces search space
- Stronger consistency
  - Filters more subtrees
  - But is costlier to enforce

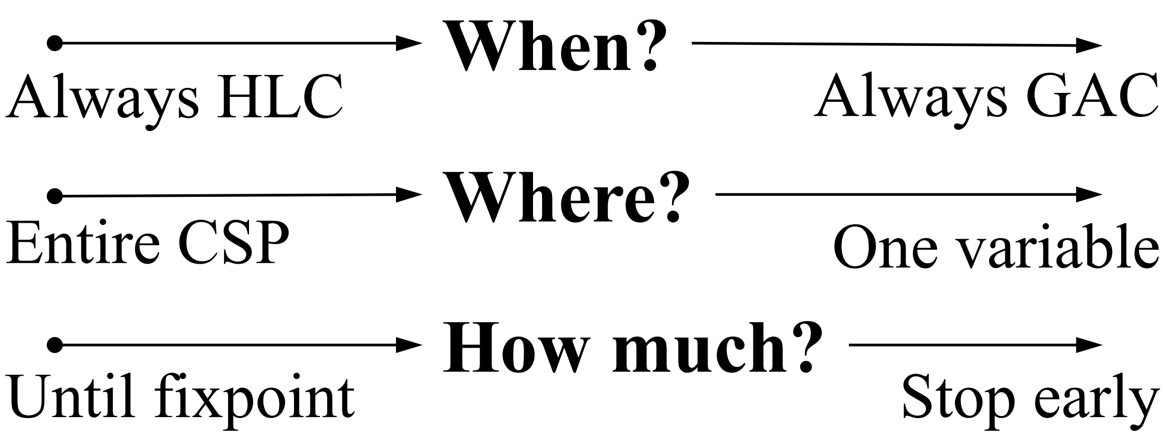


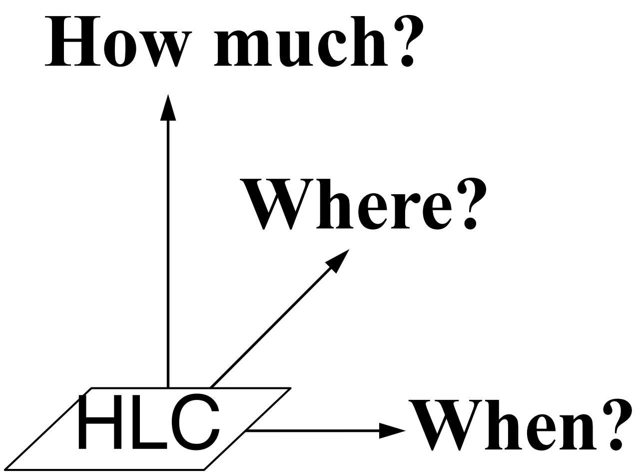


GAC vs HLC

Challenge

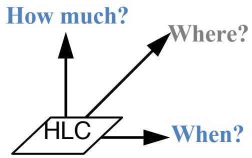
Decide **when, where,** and **how much** HLC to enforce during search





Our solution

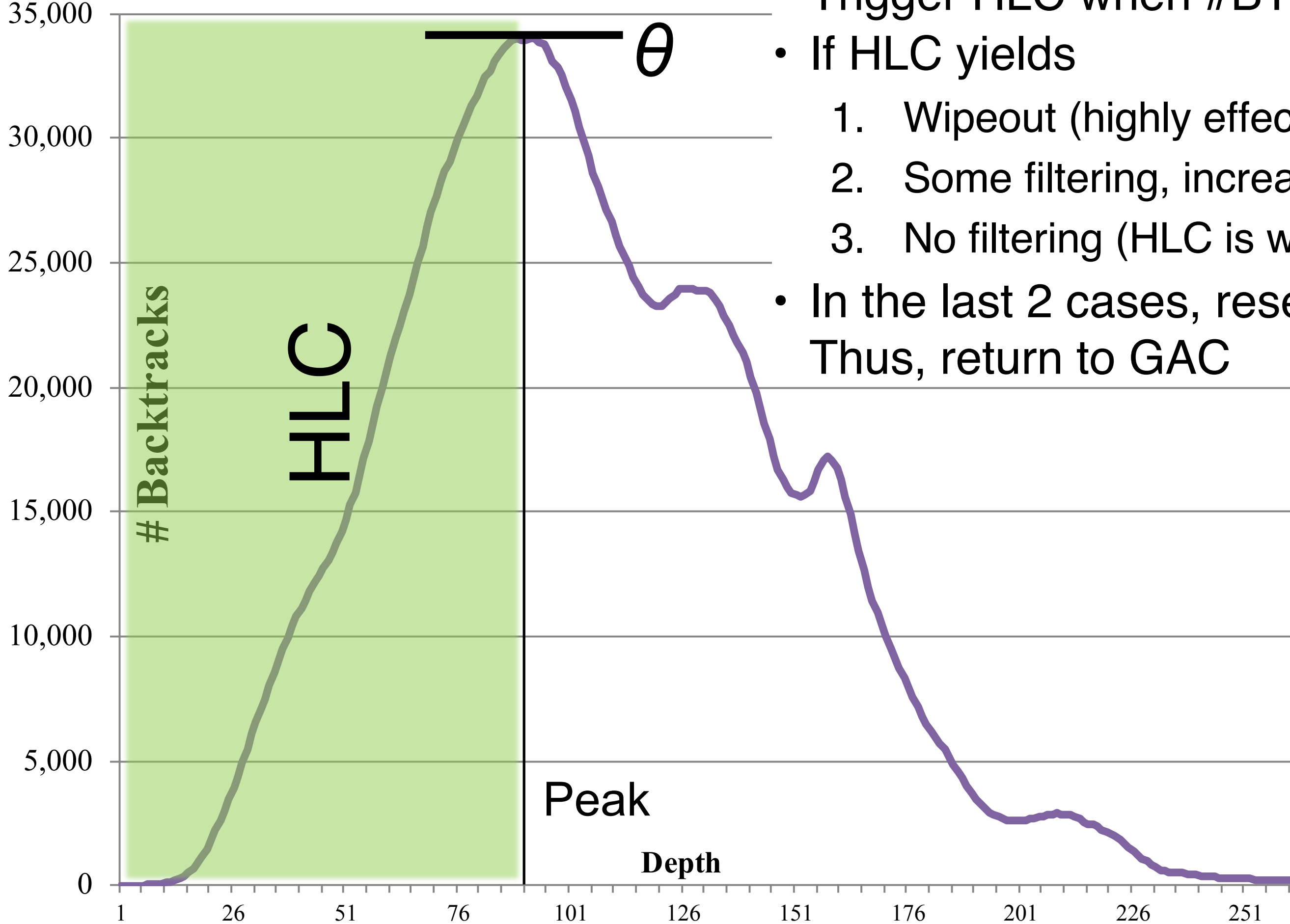
1. **When: PREPEAK**
   - Monitors search performance
   - When search starts thrashing, triggers an HLC
   - Then, conservatively reverts to GAC



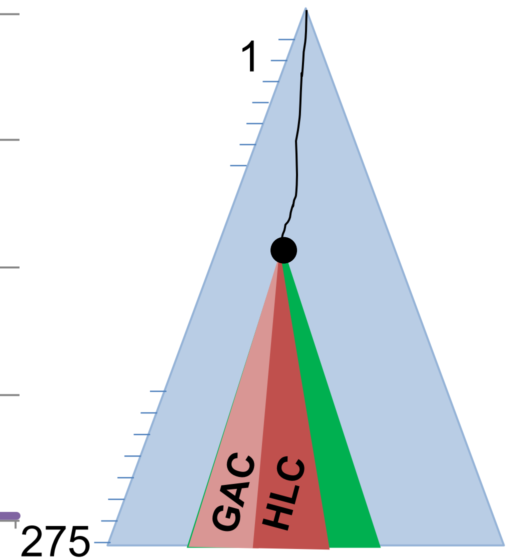
2. **How much**
   - Monitor propagation and interrupt before fixpoint

PREPEAK⁺ = PREPEAK + ‘How Much’

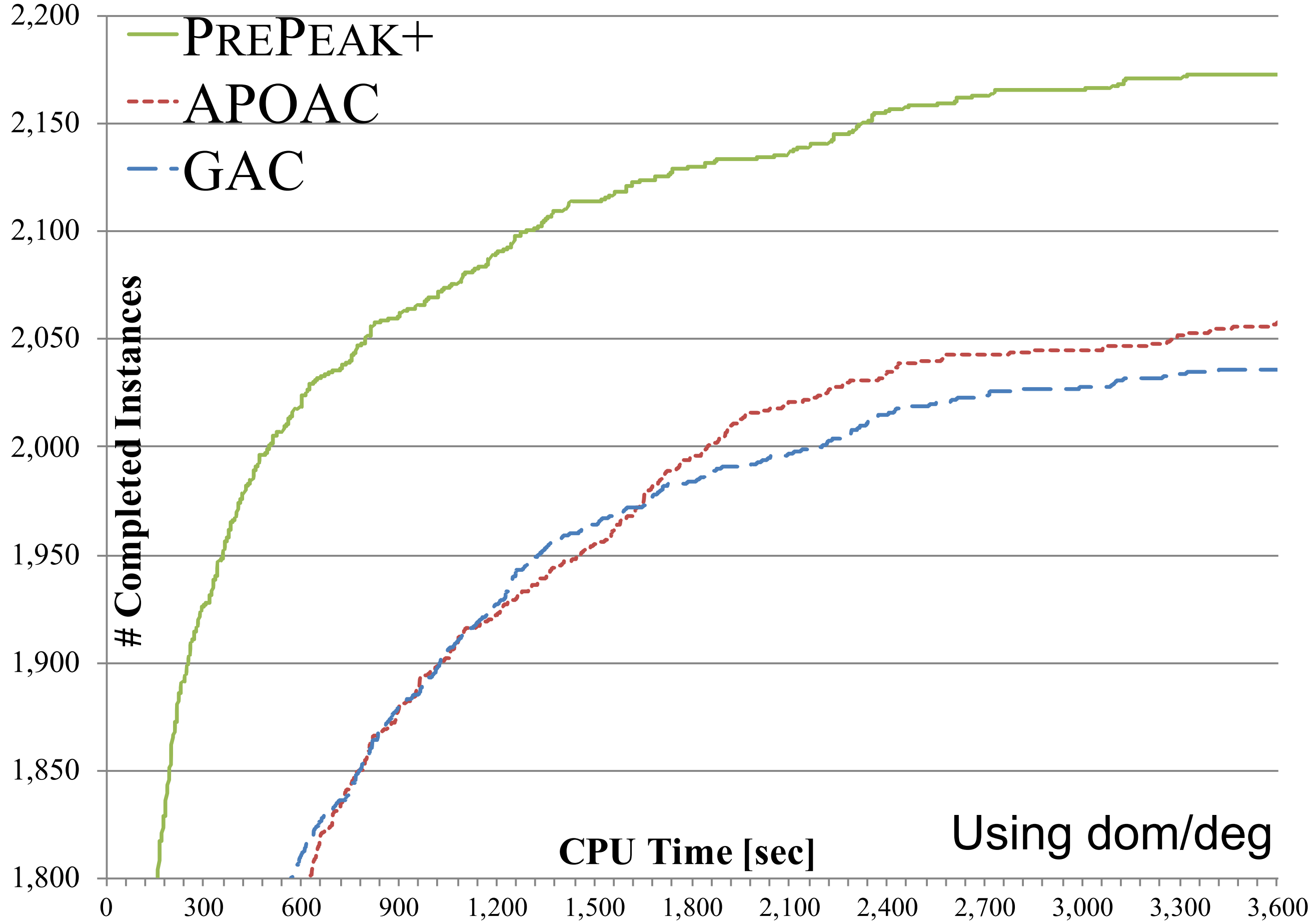
PrePeak examines #BT per depth



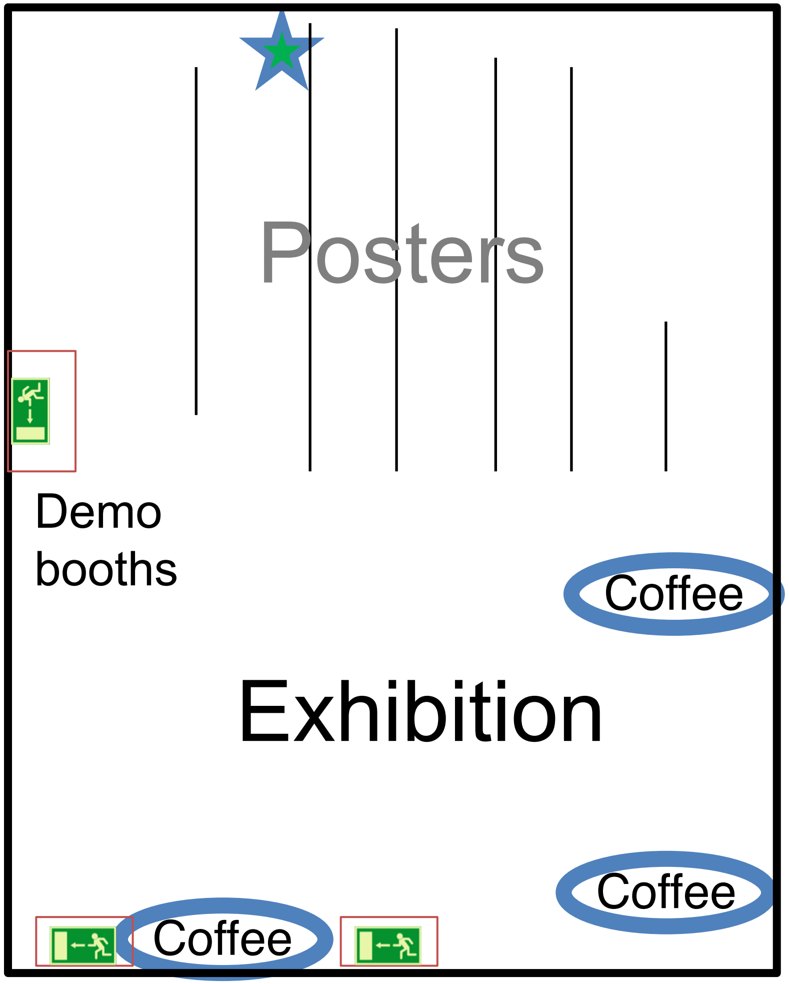
- Trigger HLC when #BT reaches a threshold θ
- If HLC yields
  1. Wipeout (highly effective), reduce θ
  2. Some filtering, increase θ a little
  3. No filtering (HLC is wasteful), increase θ a lot
- In the last 2 cases, reset #BT per depth to 0
  Thus, return to GAC



Empirical Evaluations



Thank You



Goal state

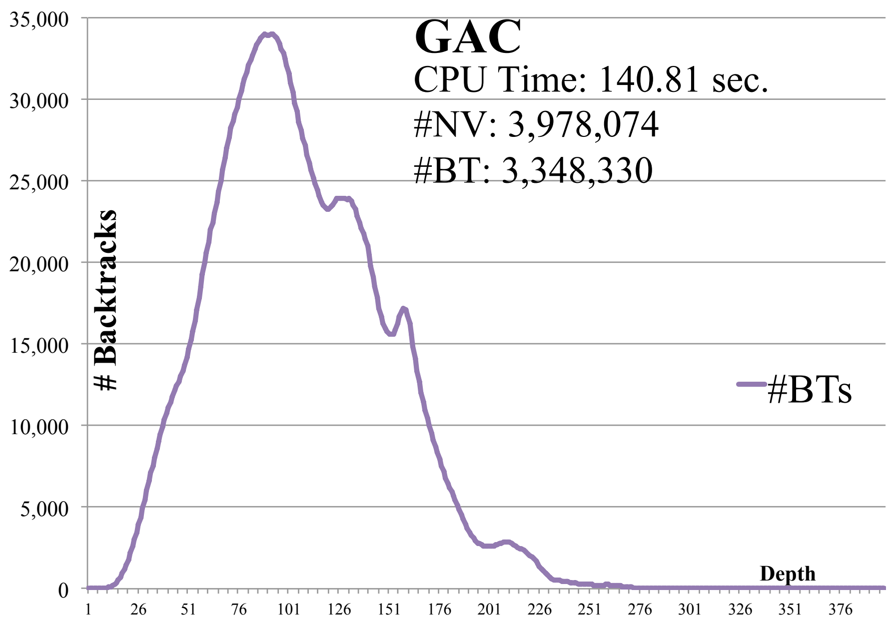
Possible start states

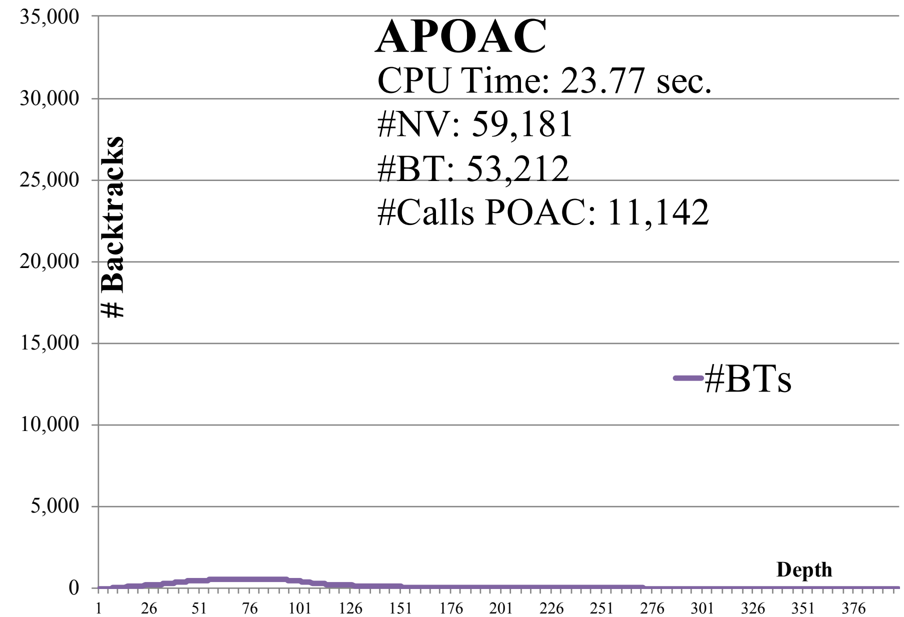
Thank You

Questions & Comments

Please stop by
the poster #51

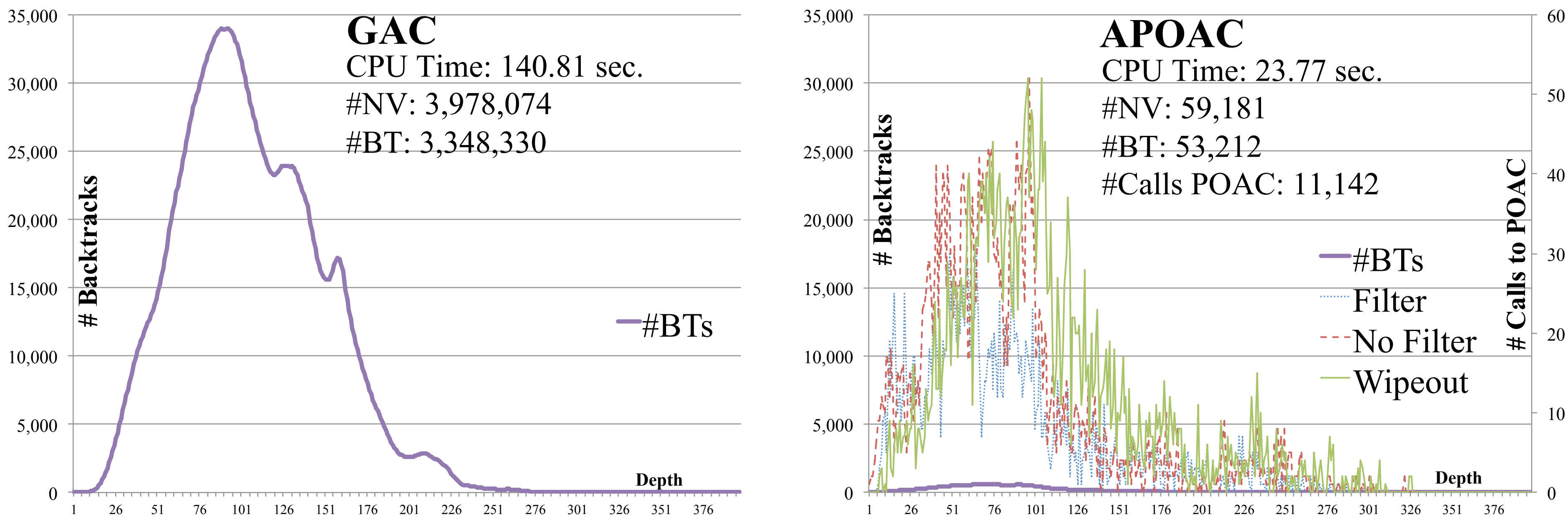
Visualization of Benefit





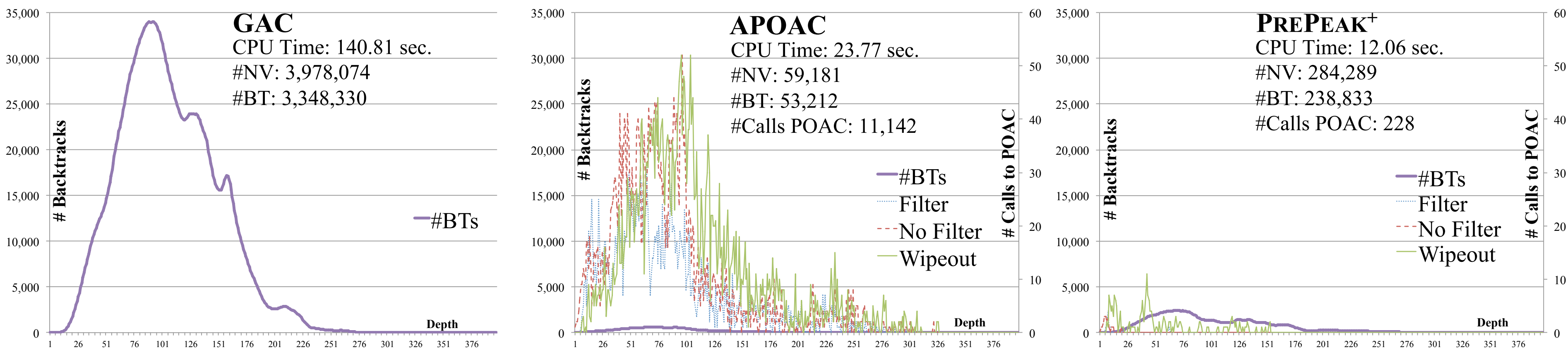
pseudo-aim-200-1-6-4, dom/wdeg

Visualization of Benefit



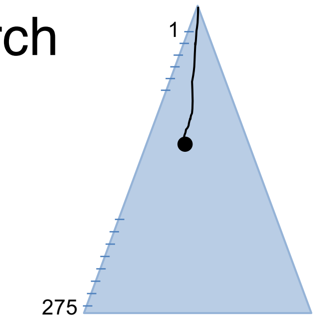
pseudo-aim-200-1-6-4, dom/wdeg

Visualization of Benefit



PREPEAK A Reactive Strategy for HLC

- Keep track of $btcount[\cdot]$, number of backtrack during search
- When $btcount[\cdot]$ reaches a given threshold θ
  - Enforce GAC then HLC as long as HLC yields domain wipeout for all values in domain of current variable
  - If backtrack, reduce threshold and keep enforcing HLC
  - If HLC finds a consistent value, reset $btcount[\cdot]$, increase threshold a little
  - If GAC finds a consistent value, reset $btcount[\cdot]$, increase threshold a lot



- Geometric laws to update threshold
  - Wipeout: $\theta^{bt}_{k+1} \leftarrow r_w \cdot \theta^{bt}_k, r_w = 1.2^{-1}$
  - Filtering: $\theta^{bt}_{k+1} \leftarrow r_f \cdot \theta^{bt}_k, r_f = 1.2^2$
  - No filtering: $\theta^{bt}_{k+1} \leftarrow r_n \cdot \theta^{bt}_k, r_n = 1.2^3$