CapsAndRuns: An Improved Method for Approximately Optimal Algorithm Configuration

Gellért Weisz   András György   Csaba Szepesvári

DeepMind

ICML 2019

Introduction

- Problem: find good parameter settings (configurations) for general purpose solvers.
  - No structure assumed over the parameter space.

Introduction

- Problem: find good parameter settings (configurations) for general purpose solvers.
  - No structure assumed over the parameter space.

- Zillions of practical algorithms $\Leftrightarrow$ Little theory
  Want theoretical guarantees on the runtime of
  - the chosen configuration; and
  - the configuration process.

Introduction

- Problem: find good parameter settings (configurations) for general purpose solvers.
  - No structure assumed over the parameter space.
- Zillions of practical algorithms $\Leftrightarrow$ Little theory

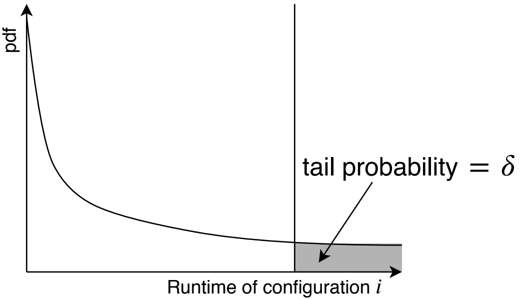
  Want theoretical guarantees on the runtime of
  - the chosen configuration; and
  - the configuration process.
- Goal: find a near-optimal configuration solving $1 - \delta$ fraction of the problems in the least expected time.
  - Since some instances (δ fraction) are hopelessly hard; don't want to solve those.

Problem formulation

Given: n configurations, distribution Γ of problem instances.

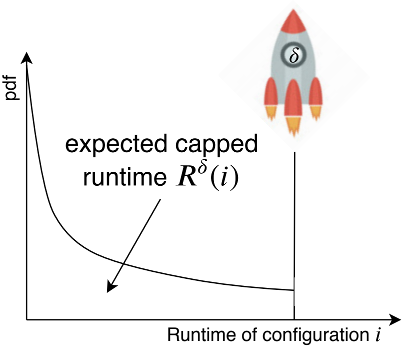
Problem formulation

Given: n configurations, distribution Γ of problem instances.



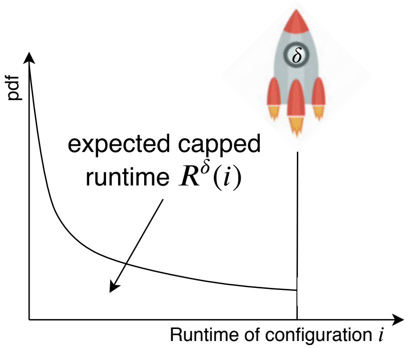
Problem formulation

Given: n configurations, distribution Γ of problem instances.



Problem formulation

Given: n configurations, distribution Γ of problem instances.

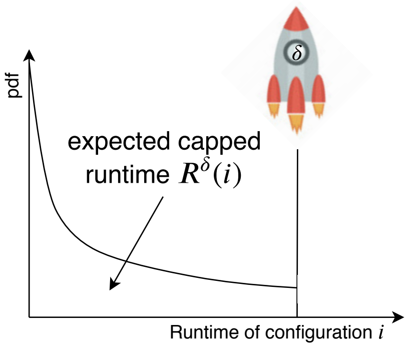


Runtime of the optimal capped configuration:

$$\mathrm{OPT}_{\delta} = \min_{i} R^{\delta}(i)$$

Problem formulation

Given: n configurations, distribution Γ of problem instances.



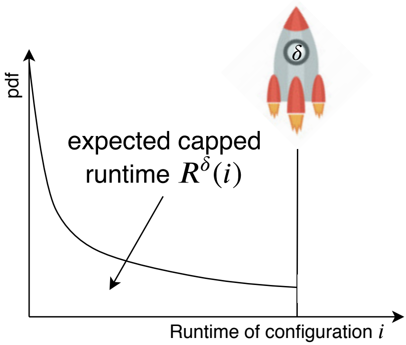
Runtime of the optimal capped configuration:

$$\mathrm{OPT}_{\delta} = \min_{i} R^{\delta}(i)$$

Configuration i is (ε, δ)-optimal if $R^{\delta}(i) \leq (1+\varepsilon)\mathrm{OPT}_{\delta/2}$.

Problem formulation

Given: n configurations, distribution Γ of problem instances.



Runtime of the optimal capped configuration:

$$\mathrm{OPT}_\delta = \min_i R^\delta(i)$$

Configuration i is (ε, δ)-optimal if $R^\delta(i) \leq (1+\varepsilon)\mathrm{OPT}_{\delta/2}$.

Previous work (Kleinberg et al., 2017; Weisz et al., 2018): no capping of OPT: using OPT_0 instead of $\mathrm{OPT}_{\delta/2}$.

Guarantees

- Previous work (Kleinberg et al., 2017; Weisz et al., 2018): with high probability,
  (i) the algorithm finds an (ε, δ)-optimal configuration;
  (ii) with total work
  $$\tilde{O}\left(\mathrm{OPT}_0 \frac{n}{\varepsilon^2 \delta}\right) .$$
  - ★ Worst case lower bound: $\Omega\left(\mathrm{OPT}_0 \frac{n}{\varepsilon^2 \delta}\right)$ (Kleinberg et al., 2017).

Guarantees

- Previous work (Kleinberg et al., 2017; Weisz et al., 2018): with high probability,
  (i) the algorithm finds an (ε, δ)-optimal configuration;
  (ii) with total work

$$\tilde{O}\left(\mathrm{OPT}_0 \frac{n}{\varepsilon^2 \delta}\right) .$$

  - ★ Worst case lower bound: $\Omega\left(\mathrm{OPT}_0 \frac{n}{\varepsilon^2 \delta}\right)$ (Kleinberg et al., 2017).

- This work with high probability finds an (ε, δ)-optimal configuration:
  - ▸ Total work (simplified version):

$$\tilde{O}\left(n\mathrm{OPT}_{\delta/2}\left(\frac{1}{\delta} + \max\left\{\frac{\sigma^2}{\max\{\varepsilon^2, \Delta^2\}}, \frac{r}{\max\{\varepsilon, \Delta\}}\right\}\right)\right) ,$$

  where
  - ★ $\Delta \sim$ gap between the best two configurations
  - ★ $\sigma^2 \sim$ runtime variances,
  - ★ $r \sim$ range of runtimes.

CapsAndRuns algorithm

$$\tilde{O}\left(n\mathrm{OPT}_{\delta/2}\left(\frac{1}{\delta}+\max\left\{\frac{\sigma^2}{\max\{\varepsilon^2,\Delta^2\}},\frac{r}{\max\{\varepsilon,\Delta\}}\right\}\right)\right),$$

Phase I:

- For each configuration i find a runtime cap τ_i
  - that solves between $1-\delta$ and $1-\delta/2$ fraction of problem instances,
  - not wasting time on bad configurations.

CapsAndRuns algorithm

$$\tilde{O}\left(n\mathrm{OPT}_{\delta/2}\left(\frac{1}{\delta}+\max\left\{\frac{\sigma^2}{\max\{\varepsilon^2,\Delta^2\}},\frac{r}{\max\{\varepsilon,\Delta\}}\right\}\right)\right),$$

Phase I:

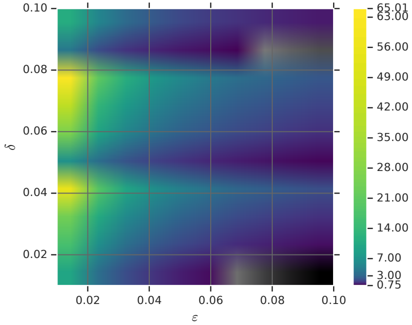
- For each configuration i find a runtime cap τ_i
  - that solves between $1-\delta$ and $1-\delta/2$ fraction of problem instances,
  - not wasting time on bad configurations.

Phase II:

- Run a Bernstein race (Mnih et al., 2008) over the configurations.
  - Evaluate configurations in parallel, giving preference to better ones, shrinking confidence regions using Bernstein's inequality.

Experiments

Configuring SAT solvers (Weisz et al., 2018):



Factor of total runtime improvement from LEAPSANDBOUNDS to CAPSANDRUNS for various values of ε and δ.

STRUCTURED PROCRASTINATION	LEAPSANDBOUNDS	CAPSANDRUNS
20643 (± 5) days	1451 (± 83) days	586 (± 7) days

Thank you!

Poster #201