DM826 – Spring 2011

Modeling and Solving Constrained Optimization Problems

Lecture 8
Search

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

[*Based on slides by Stefano Gualandi, Politecnico di Milano*]

Outline

1. Search

2. Random Restart

Search

- Complete
  - backtracking
  - dynamic programming
- incomplete
  - local search

Backtracking: Terminology

- backtracktracking: depth first search of a search tree
- branching strategy: method to extend a node in the tree
- node visited if generated by the algorithm
- constraint propagation prunes subtrees
- deadend: if the node does not lead to a solution
- thrashing repeated exploration of failing subtree differening only in assignments to variables irrelevant to the failure of the subtree.

Simple Backtracking

- at level $j \longleftarrow$ instantiation $I = \{x_1 = a_1, \ldots, x_j = a_j\}$
- branches: different choices for an unassigned variable: $I \cup \{x = a\}$
- branching constraints $\mathcal{P} = \{b_1, \ldots, b_j\}$, $b_i, 1 \leq i \leq j$
- $\mathcal{P} \cup \{b^1_{j+1}\}, \ldots, \mathcal{P} \cup \{b^k_{j+1}\}$ extension of a node by mutually exclusive branching constraints

Branching strategies

Assume a variable order and a value order (e.g., lexicographic):

A. Generic branching with unary constraints:

1. Enumeration, d-way

$$x = 1 \quad | \quad x = 2 \quad | \ldots$$

2. Binary choice points, 2-way

$$x = 1 \quad | \quad x \neq 1$$

3. Domain splitting

$$x \leq 3 \quad | \quad x\ 3$$

⇝ d-way can be simulated by 2-way with no loss of efficiency. The contrary does not old.

⇝ 2-way seems theoretically more powerful than d-way

Branching strategies

B. Problem specific:

- Disjunctive scheduling
- Zykov's branching rule for graph coloring

Constraint propagation

- constaint propagation performed at each node: mechanism to avoid thrashing
- typically best to enfore domain based but with some exceptions (e.g., forward checking is best in SAT)
- nogood constraints added after deadend is encountered:
  - set of assignements and branching constraints that is not consistnet with a solution
  - backtracking has laready ruled out the subtree but inserting nogood constraints the hope is they contribute to propagate
  - e.g., $I = \{x_1 = 2, x_2 = 5, x_3 = 3, x_5 = 4\}$ and $x = 6$ deadend post $\neg\{x_1 = 2, x_2 = 5, x_3 = 3, x_5 = 4\}$

Backjumping

- standard backtracking: chronological backtracking
- non-chronological backtracking: retracts the closest branching constraint that bears responsability.
  backjumping or intelligent backtracking:
  $\mathcal{P} = \{b_1, \ldots, b_j\}$
  $J(\mathcal{P}) \subseteq \mathcal{P}$ jumpback nogood for $\mathcal{P}$
  largest i $1 \leq i \leq j : b_i \in J(\mathcal{P})$
  jumpback and retracts b_i and all those posted after b_i
  and delete nogoods recorded after b_i

Restoration Service

What do we have at the nodes of the search tree?
A computational space:

1. Partial assignments of values to variables
2. Unassigned variables
3. Suspended propagators

How to restore when backtracking?

- Trailing Changes to nodes are recorded such that they can be undone later
- Copying A copy of a node is created before the node is changed
- Recomputation If needed, a node is recomputed from scratch

Variable-Value ordering

Possible goals

- Minimize the underlying search space
- Minimize expected depth of any branch
- Minimize expected number of branches
- Minimize size of search space explored by backtracking algorithm
  (intractable to find "best" variable)

Variable ordering

dynamic vs static

- it is optimal if it visits the fewest number of nodes in the search tree
- finding optimal ordering is hard

dynamic heuristics:

- based on domain size $textrem(x|\mathcal{P})$ remaining after propagation
- dom + deg (# constraints that involve a variable still unassigned)
- $\frac{\text{dom}}{\text{wdeg}}$ weight incremented when a constraint is responsible for a deadend
- min regret
- structure guided var ordering:
  instantiate first variables that decompose the constraint graph
  graph separators: subset of vertices or edges that when removed separates the graph into disjoint subcomponents

Value ordering

- estimate number of solutions:
  counting solutions to a problem with tree structrure can be done in polytime
  reduce the graph to a tree by dropping constraints
- if optimization constraints: reduced cost to rank values

Variants to best search

- Limite Discrepancy search

  Discrepancy: when the search does not follow the value ordering heuristic and does not take the left most branch out of a node.

  explopre tree by iteratively increasing number of discrepancies, preferring discrepancies near the root
  (thus easier to recover from early mistakes)

  Ex: ith iteration: visit all leaf nodes up to i discrepancies
  $i = 0, 1, \ldots, k$ (if $k \geq n$ depth trhen alg is complete)

- Interleaved depth first search
  each subtree rooted at a branch is searched for a given time-slice using depth-first.
  If no solution found, search suspended, next branch active.
  Upon suspending in the last the first again becomes active.
  Similar idea in credit based.

Outline

Algorithm Survival Analysis

Run time distributions

- $T \in [0, \infty]$
- $F(t) = \Pr\{T \leq t\} \quad F : [0, \infty] \mapsto [0, 1]$
- $f(t) = \frac{dF(t)}{dt}$ pdf
- $S(t) = \Pr\{T > t\} = 1 - F(t)$

Characterization of runtime

Parametric models used in the analysis of run-times to
exploit the properties of the model (eg, the character of tails and completion rate)

Procedure:

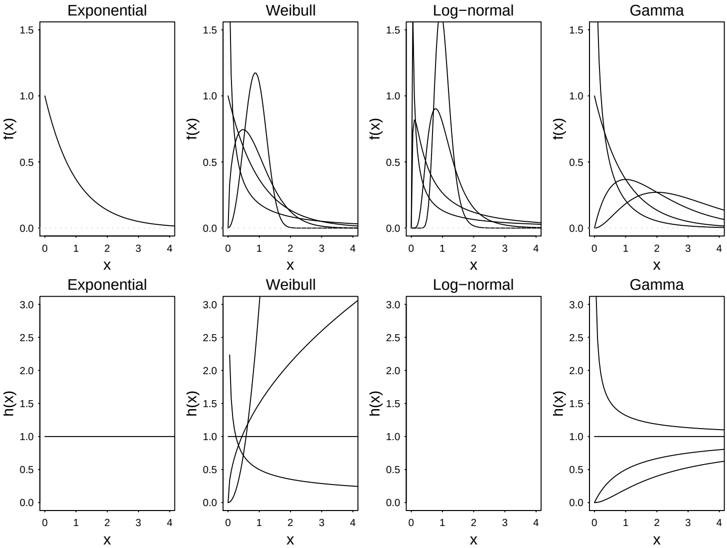
- choose a model
- apply fitting method
  maximum likelihood estimation method:

$$\max_{\theta \in \Theta} \log \prod_{i=1}^{n} p(X_i, \theta)$$

- test the model

Parametric models

The distributions used are [Frost et al., 1997; Gomes et al., 2000]:



Characterization of Run-time

Motivations for these distributions:

- qualitative information on the completion rate (= hazard function)
- empirical good fitting

To check whether a parametric family of models is reasonable the idea is to make plots that should be linear. Departures from linearity of the data can be easily appreciated by eye.

Example: for an exponential distribution:

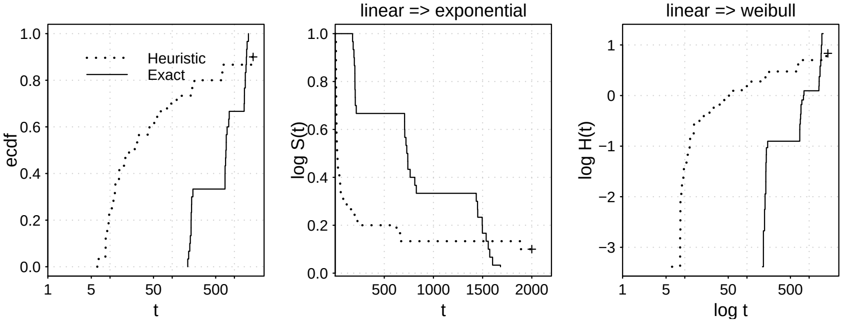
$$\log S(t) = -\lambda t \qquad S(t) = 1 - F(t) \text{ is the survivor function}$$

$\rightsquigarrow$ the plot of $\log S(t)$ against t should be linear.

Similarly, for the Weibull the cumulative hazard function is linear on a log-log plot

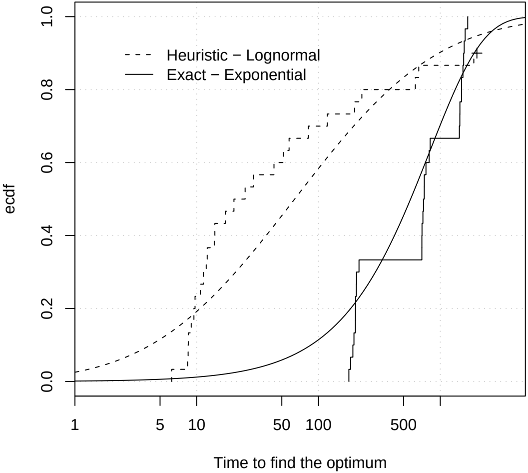
Characterization of Run-time
Example

Graphical inspection for the two censored distributions from the previous example on 2-edge-connectivity.



Characterization of Run-time
Example

Graphical inspection for the two censored distributions from the previous example on 2-edge-connectivity.



Extreme Value Statistics

- Extreme value statistics focuses on characteristics related to the tails of a distribution function
  1. extreme quantiles (*e.g.*, minima)
  2. indices describing tail decay

- 'Classical' statistical theory: analysis of means.
  Central limit theorem: $X_1, \ldots, X_n$ i.i.d. with F_X

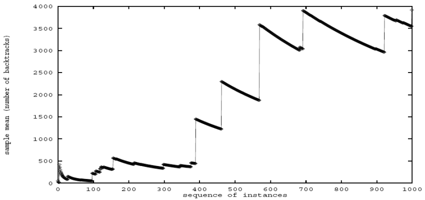
$$\sqrt{n}\frac{\bar{X} - \mu}{\sqrt{Var(X)}} \xrightarrow{D} N(0,1), \qquad \text{as } n \to \infty$$

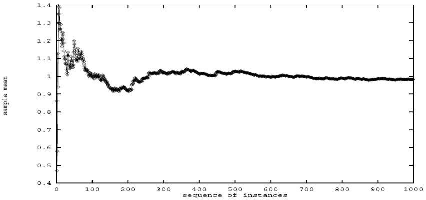
Heavy tailed distributions: mean and/or variance may not be finite!

Characterization of Run-time
Heavy Tails

Gomes et al. [2000] analyze the mean computational cost to find a solution on a single instance

On the left, the observed behavior calculated over an increasing number of runs.
On the right, the case of data drawn from normal or gamma distributions





- The use of the median instead of the mean is recommended
- The existence of the moments (*e.g.*, mean, variance) is determined by the tails behavior: a case like the left one arises in presence of long tails

Extreme Value Statistics

Extreme values theory

- $X_1, X_2, \ldots, X_n$ i.i.d. F_X
  Ascending order statistics $X_n^{(1)} \leq \ldots \leq X_n^{(n)}$
- For the minimum $X_n^{(1)}$ it is $F_{X_n^{(1)}} = 1 - [1 - F_X^{(1)}]^n$ but not very useful in practice as F_X unknown
- Theorem of [Fisher and Tippett, 1928]:
  "almost always" the normalized extreme tends in distribution to a generalized extreme distribution (GEV) as $n \rightarrow \infty$.

In practice, the distribution of extremes is approximated by a GEV:

$$F_{X_n^{(1)}}(x) \sim \begin{cases} \exp(-1(1-\gamma\frac{x-\mu}{\sigma})^{-1/\gamma}, & 1-\gamma\frac{x-\mu}{\sigma} > 0, \gamma \neq 0 \\ \exp(-\exp(\frac{x-\mu}{\sigma})), & x \in \mathbf{R}, \gamma = 0 \end{cases}$$

Parameters estimated by simulation by repeatedly sampling k values $X_{1n}, \ldots, X_{kn}$, taking the extremes $X_{kn}^{(1)}$, and fitting the distribution.
γ determines the type of distribution: Weibull, Fréchet, Gumbel, ...

Extreme Value Statistics

Tail theory

- Work with data exceeding a high threshold.
- Conditional distribution of exceedances over threshold τ

$$1 - F_\tau(y) = P(X - \tau > y \mid X > \tau) = \frac{P(X > \tau + y)}{P(X > \tau)}$$

- If the distribution of extremes tends to GEV distribution then there exist a Pareto-type function such that for some $\gamma > 0$

$$1 - F_X(x) = x^{-\frac{1}{\gamma}} \ell_F(x), \qquad x > 0,$$

with $\ell_F(x)$ a slowly varying function at infinity.

In practice, fit a function $Cx^{-\frac{1}{\gamma}}$ to the exceedances:
$Y_j = X_i - \tau$, provided $X_i > \tau$, $j = 1, \ldots, N_\tau$.
γ determines the nature of the tail

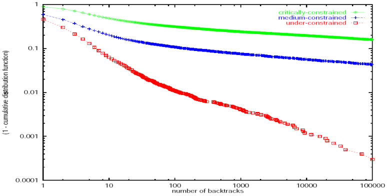
Characterization of Run-time
Heavy Tails

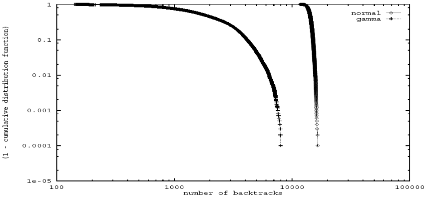
The values estimated for γ give indication on the tails:

- $\gamma > 1$: long tails hyperbolic decay (the completion rate decreases with t) and mean not finite
- $\gamma < 1$: tails exhibit exponential decay

Graphical check using a log-log plot:

- heavy tail distributions approximate linear decay,
- exponentially decreasing tail has faster-than linear decay





Long tails explain the goodness of random restart. Determining the cutoff time is however not trivial.

References

Frost D., Rish I., and Vila L. (1997). **Summarizing CSP hardness with continuous probability distributions.** In *Proceedings of AAAI/IAAI*, pp. 327–333.

Gent I.P., Petrie K.E., and Puget J.F. (2006). **Symmetry in constraint programming**. In *Handbook of Constraint Programming*, edited by F. Rossi, P. van Beek, and T. Walsh, chap. 10, pp. 329–376. Elsevier.

Gomes C., Selman B., Crato N., and Kautz H. (2000). **Heavy-tailed phenomena in satisfiability and constraint satisfaction problems**. *Journal of Automated Reasoning*, 24(1-2), pp. 67–100.

Hüsler J., Cruz P., Hall A., and Fonseca C.M. (2003). **On optimization and extreme value theory**. *Methodology and Computing in Applied Probability*, 5, pp. 183–195.

Ovacik I.M., Rajagopalan S., and Uzsoy R. (2000). **Integrating interval estimates of global optima and local search methods for combinatorial optimization problems**. *Journal of Heuristics*, 6(4), pp. 481–500.