Effective probabilistic stopping rules for randomized metaheuristics: GRASP implementations

Celso C. Ribeiro
Isabel Rosseti
Reinaldo C. Souza

Universidade Federal Fluminense, Brazil

July 2012

Contents

Contents

Metaheuristics

Definition

Metaheuristics are general high-level procedures that coordinate simple heuristics and rules to find good (often optimal) approximate solutions to computationally difficult combinatorial optimization problems.

- Randomization plays an important role in algorithm design:
  - Some metaheuristics (such as GRASP and VNS) rely on randomization to sample the search space.
  - Can be used to break ties, so as that different trajectories can be followed from the same initial solution in multistart methods or to sample fractions of large neighborhoods.
  - Greedy randomized algorithms make use of randomization to build different solutions at different runs.

Need for effective stopping criteria

- Metaheuristics often suffer from the absence of effective stopping criteria, stopping after ...
  - a given maximum number of iterations, or
  - a given maximum number of consecutive iterations without improvement in the best known solution value, or
  - stabilization of the set of elite solutions found along the search.
- Algorithms ...
  - may perform an exaggerated number of iterations, or
  - may stop just before the iteration that would find an optimum.
- Dual bounds may be used in quality-based stopping rules, but they are often hard to compute or far from the optimal values.
- Bayesian rules were not followed by enough numerical results to validate their effectiveness or to illustrate their efficiency.
- This paper: effective probabilistic stopping rules for randomized metaheuristics.

Contents

GRASP template

- Minimization problem: $\min f(x) : x \in X$
- GRASP is a multi-start metaheuristic.
- Each iteration has two phases: construction and local search.

procedure $\texttt{GRASP}(\texttt{MaxIterations}, \texttt{Seed})$
1. Set $f^* \leftarrow \infty$;
2. **for** $k = 1, \ldots, \texttt{MaxIterations}$ **do**
3. $\quad x \leftarrow \texttt{GreedyRandomizedAlgorithm}(\texttt{Seed})$;
4. $\quad x \leftarrow \texttt{LocalSearch}(x)$;
5. $\quad$ **if** $f(x) < f^*$ **then**
6. $\qquad x^* \leftarrow x$;
7. $\qquad f^* \leftarrow f(x)$;
8. $\quad$ **end**;
9. $\quad f_k \leftarrow f(x)$;
10. **end**;
11. **return** x^*;

end.

2-path network design problem

Formulation

Given a connected undirected graph $G = (V, E)$ with non-negative weights associated with its edges, together with K pairs of origin-destination nodes, the 2-path network design problem consists of finding a minimum weighted subset of edges containing a path formed by at most two edges between every origin-destination pair.

- GRASP heuristic: Ribeiro & Rosseti (2002), *LNCS* 2400; Ribeiro & Rosseti (2007), *Parallel Computing* 33.

- set $\underline{\ell} = 0$ and $\overline{u}$ to the sum over all K demand pairs of the longest 2-path between each origin-destination pair.

Instance	$\lvert V \rvert$	$\lvert E \rvert$	K	$\underline{\ell}$	$\overline{u}$
2pndp50	50	1,225	500	0	6244
2pndp70	70	2,415	700	0	10353
2pndp90	90	4,005	900	246	14621
2pndp200	200	19,900	2000	0	37314

p-median problem

Formulation

Given a set F of m potential facilities, a set U of n customers, a distance function $d : U \times F \rightarrow \mathbb{R}$, and a constant $p \leq m$, the p-median problem consists of determining which p facilities to open so as to minimize the sum of the distances from each costumer to its closest open facility.

- GRASP heuristic: Resende & Werneck (2004), *Journal of Heuristics* 10.

p-median problem

- set $\underline{\ell}$ to the sum over all customers of the distance from each customer to its closest facility. Similarly, set $\overline{u}$ to the sum over all customers of the distance from each customer to its most distant facility.

Instance	m	n	p	$\underline{\ell}$	$\overline{u}$
pmed14	300	1800	60	2968	5898
pmed15	300	1800	100	1729	9791
pmed25	500	5000	167	1828	16477
pmed30	600	7200	200	1989	19826

Quadratic assignment problem

Formulation

Given n facilities and n locations represented by sets $F = \{f_1, \ldots, f_n\}$ and $L = \{\ell_1, \ldots, \ell_n\}$, assign each facility to a location. Let $A^{n \times n} = (a_{ij}) : a_{ij} \in \mathbb{R}^+$ represents the flow between f_i and f_j. Let $B^{n \times n} = (b_{ij}) : b_{ij} \in \mathbb{R}^+$ represents the distance between ℓ_i and ℓ_j. Assignment $p : \{1, \ldots, n\} \to \{1, \ldots, n\}$ has cost $c(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b_{p(i),p(j)}$. Find a permutation vector $p \in \Pi_n$ that minimizes the assignment cost $c(p)$, where Π_n stands for the set of all permutations of $\{1, \ldots, n\}$.

- GRASP heuristic: Oliveira, Pardalos & Resende (2004), *LNCS* 3059.

Quadratic assignment problem

- compute $\underline{\ell}$ and $\overline{u}$ by ordering the elements of A and B and multiplying them appropriately.

Instance	n	$\underline{\ell}$	$\overline{u}$
tai30a	30	1706855	8596620
tai35a	35	2216627	11803330
tai40a	40	2843274	15469120
tai50a	50	4390920	24275700

The set k-covering problem

Formulation

Given a set $I = \{1, \ldots, m\}$ of objects, let $\{P_1, \ldots, P_n\}$ be a collection of subsets of I, with a non-negative cost c_j associated with each subset P_j, for $j = 1, \ldots, n$. A subset $\hat{J} \subseteq J = \{1, \ldots, n\}$ is a *cover* of I if $\cup_{j \in \hat{J}} P_j = I$. The cost of a cover $\hat{J}$ is $\sum_{j \in \hat{J}} c_j$. The *set covering problem* consists of finding a minimum cost cover J^*. The *set multi-covering problem* is a generalization of the set covering problem, in which each object $i \in I$ must be covered by at least $\ell_i \in \mathbb{Z}_+$ elements of $\{P_1, \ldots, P_n\}$. A special case of the set multi-covering problem arises when $\ell_i = k$, for all $i \in I$. We refer to this problem as the *set k-covering problem*.

- L.S. Pessoa, M.G.C. Resende, and C.C. Ribeiro, A hybrid Lagrangean heuristic with GRASP and path relinking for set k-covering, *Computers and Operations Research*, 2012 (online).

- set $\underline{\ell}$ to the optimal value of the linear relaxation. To compute $\overline{u}$, create a list associated to each row of the constraint matrix, formed by the k largest costs of the variables that cover this row. Next, build the set of variables formed by the union of the m individual lists and set $\overline{u}$ to the sum of the costs of these variables.

Instance	Dimension	k_{min}	$\underline{\ell}$	$\overline{u}$
scp42	200x1000	2	1205	37132
scp47	200x1000	2	1115	36570
scp55	200x2000	2	550	38972
scpa2	300x3000	2	560	58550

Contents

Normal approximation

Assumption

We assume that the solution values obtained by a GRASP procedure fit a Normal distribution and experimentally validate this hypothesis for all test problems and test instances.

- $f_1, \ldots, f_N$: sample formed by all solution values obtained along N GRASP iterations:
  - Null hypothesis H_0: sample $f_1, \ldots, f_N$ fits a Normal distribution
  - Alternative hypothesis H_1: sample $f_1, \ldots, f_N$ does not fit a Normal distribution
  - Chi-square test commonly used to determine if a given set of observations fits a specified distribution.

Chi-square test

- First step: estimate a histogram of the sample data.
- Second step: compare the observed frequencies with those obtained from the specified density function.
- Histogram formed by k cells: o_i and e_i are the observed and expected frequencies for the i-th cell, with $i = 1, \ldots, k$.
- Compute $D = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$
- D follows a chi-square distribution with $k - 1$ degrees of freedom under the null hypothesis.
- Since the mean and the standard deviation are unknown, they should be estimated from the sample:
  - two degrees of freedom are lost to compensate.
- Null hypothesis cannot be rejected at a level of significance α if $D < \chi^2_{[1-\alpha;k-3]}$.

Numerical experiments

- Validation for all test problems and test instances previously described:
  - results summarized for some instances for clearness of presentation.
- Level of significance set at $\alpha = 0.1$.
- Histograms with $k = 14$ cells, corresponding to the intervals $(-\infty, -3.0)$, $[-3.0, -2.5)$, $[-2.5, -2.0)$, $[-2.0, -1.5)$, $[-1.5, -1.0)$, $[-1, -0.5)$, $[-0.5, 0.0)$, $[0.0, 0.5)$, $[0.5, 1.0)$, $[1.0, 1.5)$, $[1.5, 2.0)$, $[2.0, 2.5)$, $[2.5, 3.0)$, and $[3.0, \infty)$.
- Normal fittings after $N = 50$, 100, 500, 1000, 5000, and 10000 GRASP iterations for each instance.

Validation: 2-path network design problem

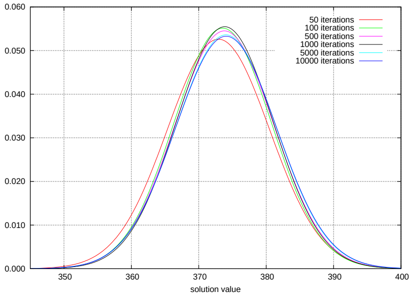
- Chi-square test after $N = 50$ iterations:

Instance	Iterations	D	$\chi^2_{[1-\alpha;k-3]}$
2pndp50	50	0.398049	17.275000
2pndp70	50	0.119183	17.275000
2pndp90	50	0.174208	17.275000
2pndp200	50	0.414327	17.275000

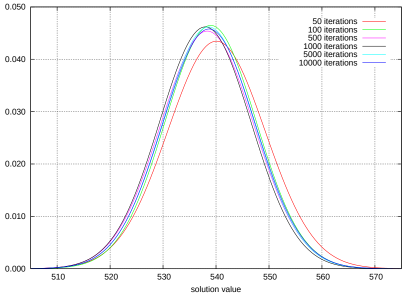
- Statistics for Normal fittings:

Instance	Iterations	Mean	Std. dev.	Skewness	Kurtosis
2pndp50	50	372.920000	7.583772	0.060352	3.065799
	100	373.550000	7.235157	-0.082404	2.897830
	500	373.802000	7.318661	-0.002923	2.942312
	1000	373.854000	7.192127	0.044952	3.007478
	5000	374.031400	7.442044	0.019068	3.065486
	10000	374.063500	7.487167	-0.010021	3.068129
2pndp70	50	540.080000	9.180065	0.411839	2.775086
	100	538.990000	8.584282	0.314778	2.821599
	500	538.334000	8.789451	0.184305	3.146800
	1000	537.967000	8.637703	0.099512	3.007691
	5000	538.576600	8.638989	0.076935	3.016206
	10000	538.675600	8.713436	0.062057	2.969389

- Fitted Normal distributions:



(a) 50-node instance



(b) 70-node instance

Validation: p-median problem

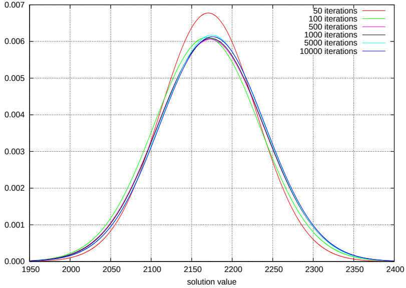
- Chi-square test after $N = 50$ iterations:

Instance	Iterations	D	$\chi^2_{[1-\alpha;k-3]}$
pmed14	50	0.374863	17.275000
pmed15	50	0.167526	17.275000
pmed25	50	0.249443	17.275000
pmed30	50	0.160131	17.275000

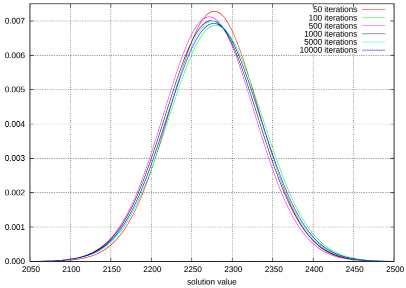
- Statistics for Normal fittings:

Instance	Iterations	Mean	Std. dev.	Skewness	Kurtosis
pmed15 $p = 100$	50	2170.500000	58.880642	-0.041262	1.949923
	100	2168.450000	65.313609	0.270892	2.693553
	500	2173.060000	65.881958	0.202400	2.828056
	1000	2173.484000	65.590272	0.129234	2.784433
	5000	2174.860000	64.639604	0.086450	2.940204
	10000	2175.651600	65.101495	0.096328	2.954639
pmed25 $p = 167$	50	2277.780000	54.782220	0.330959	3.028905
	100	2279.610000	58.034799	0.360133	3.466265
	500	2271.546000	56.029848	0.219415	3.311486
	1000	2274.182000	56.915366	0.081878	3.068963
	5000	2276.305200	56.985195	-0.041096	3.108109
	10000	2277.151600	57.583524	-0.041570	3.073374

- Fitted Normal distributions:



(a) Instance pmed15: $p = 100$



(b) Instance pmed25: $p = 167$

Validation: quadratic assignment problem

- Chi-square test after $N = 50$ iterations:

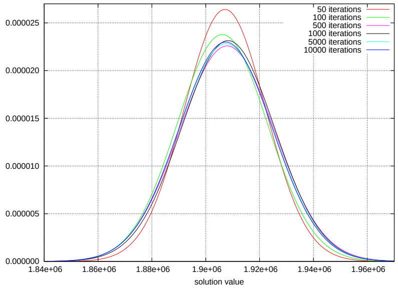
Instance	Iterations	D	$\chi^2_{[1-\alpha;k-3]}$
tai30a	50	0.127260	17.275000
tai35a	50	0.213226	17.275000
tai40a	50	0.080164	17.275000
tai50a	50	0.075752	17.275000

- Statistics for Normal fittings:

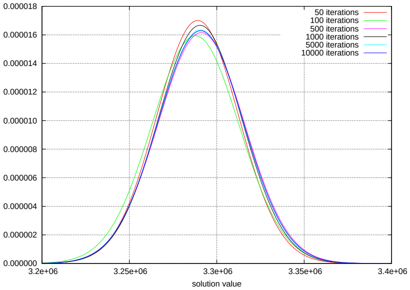
Instance	Iterations	Mean	Std. dev.	Skewness	Kurtosis
tai30a	50	1907129.960000	15106.752548	-0.068782	2.562099
	100	1906149.760000	16779.060456	0.112965	3.028193
	500	1907924.412000	17663.997163	-0.005122	3.071314
	1000	1908292.204000	17241.785219	-0.058100	2.982074
	5000	1907542.144400	17484.852454	0.077001	2.978316
	10000	1907411.800800	17354.183037	0.044985	2.982363
tai35a	50	2544227.480000	24293.234765	0.260849	2.906127
	100	2541730.980000	21782.204670	0.374843	3.131055
	500	2541151.156000	20167.926106	0.098408	2.990821
	1000	2541735.064000	20809.432271	0.079094	3.073285
	5000	2541625.512800	20952.352020	0.057649	3.069945
	10000	2541104.138000	21191.460956	0.055055	3.089498

Validation: quadratic assignment problem

- Fitted Normal distributions:



(a) Instance tai30a: $n = 30$



(b) Instance tai40a: $n = 40$

Validation: The set k-covering problem

- Chi-square test after $N = 50$ iterations:

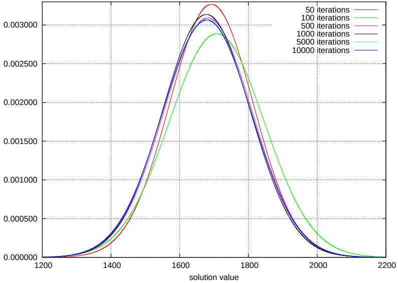
Instance	Iterations	D	$\chi^2_{[1-\alpha;k-3]}$
scp42	50	0.119939	17.275000
scp47	50	0.147765	17.275000
scp55	50	0.164476	17.275000
scpa2	50	0.092947	17.275000

- Statistics for Normal fittings:

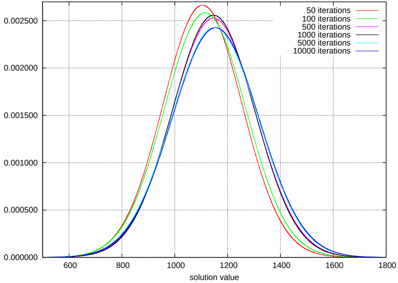
Instance	Iterations	Mean	Std. dev.	Skewness	Kurtosis
scp42	50	1692.200000	122.108968	0.346549	2.485267
	100	1707.790000	138.210513	0.747575	3.727116
	500	1682.012000	129.047681	0.453641	3.395710
	1000	1677.603000	127.209156	0.424774	3.437712
	5000	1678.960800	129.853048	0.481598	3.395114
	10000	1678.848600	130.216475	0.478711	3.328128
scp55	50	1105.160000	149.749439	0.139493	2.671918
	100	1115.010000	154.429304	0.585166	4.036000
	500	1146.800000	157.817350	0.299096	3.059246
	1000	1146.450000	155.945348	0.332401	3.045766
	5000	1151.254200	164.425966	0.384420	3.099880
	10000	1154.463700	164.456147	0.397244	3.144651

Validation: The set k-covering problem

- Fitted Normal distributions:



(a) Instance scp42: $k_{min} = 2$



(b) Instance scp55: $k_{min} = 2$

Validation: conclusions

- Skewness measures the symmetry of the original data (equal to 0 for a perfect Normal fitting).
- Kurtosis measures the shape of the fitted distribution (equal to 3 for a perfect Normal fitting).
- Mean values quickly converge to a steady-state value when the number of iterations increases.
- Mean after 50 iterations is already very close to that of the Normal fitting after 10000 iterations.
- Skewness and kurtosis values are consistently close to 0 and 3.
- Null hypothesis cannot be rejected with 90% of confidence:
  - Solution values may be approximated by a Normal distribution that can be progressively fitted and improved.
- Approximation can be used to establish and validate a probabilistic stopping rule for GRASP heuristics.

Contents

Basics

- Normal approximation of solution values can be used to give an online estimation of the number of solutions that are at least as good as the best known solution.
- Estimation will be used to implement stopping rules based on the time needed to find a solution at least as good as the incumbent.
- X is a random variable associated with the objective value of the local minimum obtained at each GRASP iteration.
- $f_1, \ldots, f_k$: solution values obtained along k iterations.
- m^k and S^k: mean and standard deviation of $f_1, \ldots, f_k$.
- We assume that X fits a Normal distribution $N(m^k, S^k)$ with average m^k and standard deviation S^k, whose probability density function and cumulative probability distribution are, respectively, $f_X^k(.)$ and $F_X^k(.)$.

Stopping rule 1/4

- ub^k: best solution value along the k first iterations.
- Probability of finding a solution value smaller than or equal to ub^k in the next iteration can be estimated by

$$F_X^k(ub^k) = \int_{-\infty}^{ub^k} f_X^k(\tau)d\tau, \text{ with}$$

$$f_X^k(\tau) = 1/(S^k\sqrt{2\pi}) \cdot e^{\frac{-(\tau - m^k)^2}{2S^{k^2}}}$$

- If lower (L) and upper (U) bounds exist for the cost of any feasible solution, then a better approximation for the probability density function $f_X^k(\tau)$ is given by the Truncated Normal distribution whose probability density function is:

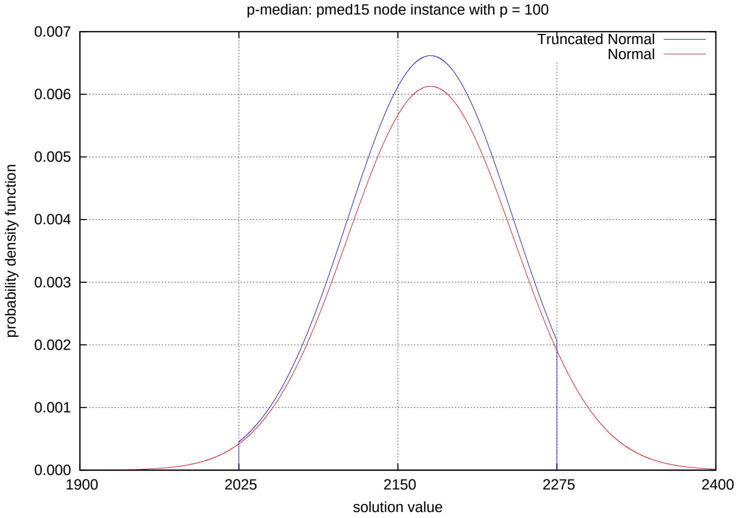
$$\hat{f}_X^k(\tau) = \begin{cases} 0, \text{ if } \tau < L \\ \frac{f_X^k(\tau)}{1-(\Delta^- + \Delta^+)}, \text{ if } L \leq \tau \leq U \\ 0, \text{ if } \tau > U \end{cases}$$

$$\Delta^- = \int_{-\infty}^{L} f_X^k(\tau) d\tau$$

$$\Delta^+ = \int_{U}^{+\infty} f_X^k(\tau) d\tau$$

Stopping rule 3/4

- Truncated Normal distribution:



Stopping rule 4/4

- Value of

$$\hat{F}_X^k(ub^k) = \int_L^{ub^k} \hat{f}_X^k(\tau)d\tau$$

is periodically recomputed after a given number of iterations is performed or whenever the value of the best known solution improves.

Computational experiment

- Experiment is performed for each test problem and instance.
- For each value of the threshold $\beta = 10^{-1}, 10^{-2}, 10^{-3}$, 10^{-4}, 10^{-5}, we run the heuristic until $\hat{F}_X^k(ub^k)$ becomes less than or equal to β:
  - $\overline{k}$ is the iteration counter where this condition is met.
  - $\overline{ub}$ is the best known solution value at this time.
- Recall that $\hat{F}_X^k(ub^k)$ is an estimation of the probability of finding in the next iteration a solution whose objective value is smaller than or equal to ub^k.
- Estimate by $\hat{N}^{\leq} = \lfloor N \cdot \hat{F}_X^{\overline{k}}(\overline{ub}) \rfloor$ the number of solutions whose value will be at least as good as $\overline{ub}$ if N additional iterations are performed:
  - $N = 1,000,000$ is empirically set in the experiments.
- Perform N additional iterations and count the number $N^{\leq}$ of solutions whose value is smaller than or equal to $\hat{F}_X^{\overline{k}}(\overline{ub})$.

Validation: 2-path network design problem

- Stopping criterion vs. estimated and counted number of solutions at least as good as the incumbent after $N = 1,000,000$ additional iterations.

Instance	Threshold β	Probability $\hat{F}_X^{\bar{k}}(\overline{ub})$	Estimation $\hat{N}^{\leq}$	Count $N^{\leq}$
50	0.1	0.079046	79046	1843
	0.01	0.009970	9970	1843
	0.001	0.000757	757	738
	0.0001	0.000001	1	0
	0.00001	0.000001	1	0
70	0.1	0.078669	78669	148028
	0.01	0.008923	8923	9537
	0.001	0.000643	643	465
	0.0001	0.000036	36	26
	0.00001	0.000005	5	4
90	0.1	0.085933	85933	2066
	0.01	0.009257	9257	2066
	0.001	0.000326	326	190
	0.0001	0.000015	15	7
	0.00001	0.000001	1	0
200	0.1	0.028989	28989	32151
	0.01	0.001821	1821	1539
	0.001	0.000566	566	503
	0.0001	0.000100	100	95
	0.00001	0.000001	1	1

Validation: p-median problem

- Stopping criterion vs. estimated and counted number of solutions at least as good as the incumbent after $N = 1,000,000$ additional iterations.

Instance	Threshold β	Probability $\hat{F}_X^{\bar{k}}(\overline{ub})$	Estimation $\hat{N}^{\leq}$	Count $N^{\leq}$
pmed14	0.1	0.060647100	60647	79535
	0.01	0.008542180	8542	7507
	0.001	0.000786594	787	215
	0.0001	6.93313e-05	69	5
	0.00001	5.55867e-06	6	0
pmed15	0.1	0.069694000	69694	117054
	0.01	0.009213730	9214	16968
	0.001	0.000626060	626	311
	0.0001	6.36170e-05	63	26
	0.00001	9.99977e-06	10	3
pmed25	0.1	0.089011300	89011	12428
	0.01	0.009309490	9309	4176
	0.001	0.000997936	998	1232
	0.0001	2.82536e-05	28	38
	0.00001	4.84266e-06	5	4
pmed30	0.1	0.089941100	89941	120598
	0.01	0.004634650	4635	1426
	0.001	0.000991829	992	1133
	0.0001	2.86224e-06	3	1
	0.00001	2.86224e-06	3	1

Validation: quadratic assignment problem

- Stopping criterion vs. estimated and counted number of solutions at least as good as the incumbent after $N = 1,000,000$ additional iterations.

Instance	Threshold β	Probability $\hat{F}_X^{\overline{k}}(\overline{ub})$	Estimation $\hat{N}^{\leq}$	Count $N^{\leq}$
tai30a	0.1	0.0451511	45151	94107
	0.01	0.0037732	3773	3759
	0.001	0.0009956	996	1031
	0.0001	—	—	—
	0.00001	—	—	—
tai35a	0.1	0.0980506	98051	39090
	0.01	0.0097537	9754	5389
	0.001	0.0009996	1000	1152
	0.0001	—	—	—
	0.00001	—	—	—
tai40a	0.1	0.0645173	64517	15748
	0.01	0.0090935	9094	15748
	0.001	0.0001213	121	111
	0.0001	—	—	—
	0.00001	—	—	—
tai50a	0.1	0.0787829	78783	281757
	0.01	0.0096325	9633	10214
	0.001	0.0004766	477	373
	0.0001	—	—	—
	0.00001	—	—	—

Validation: The set k-covering problem

- Stopping criterion vs. estimated and counted number of solutions at least as good as the incumbent after $N = 1,000,000$ additional iterations.

Instance	Threshold β	Probability $\hat{F}_X^{\overline{k}}(\overline{ub})$	Estimation $\hat{N}^{\leq}$	Count $N^{\leq}$
scp42	0.1	0.09041400	90414	111059
	0.01	0.00967873	9679	5944
	0.001	0.00045695	457	0
	0.0001	—	—	—
	0.00001	—	—	—
scp47	0.1	0.08882630	88826	125995
	0.01	0.00817110	8171	1897
	0.001	0.00035939	359	0
	0.0001	—	—	—
	0.00001	—	—	—
scp55	0.1	0.05851490	58515	36119
	0.01	0.00495659	4957	604
	0.001	0.00099997	1000	7
	0.0001	—	—	—
	0.00001	—	—	—
scpa2	0.1	0.07897290	78973	160036
	0.01	0.00995155	9952	8496
	0.001	0.00019888	199	0
	0.0001	—	—	—
	0.00001	—	—	—

Validation: conclusions

- Results show that $\hat{N}^{\leq} = \lfloor N \cdot \hat{F}_X^k(\overline{ub}) \rfloor$ is a good estimator for the number $N^{\leq}$ of solutions that might be found after N additional iterations whose value is smaller than or equal to the best solution value at the time the algorithm would stop for each threshold value β used in the stopping criterion.
- Probability $\hat{F}_X^k(ub^k)$ may be used to estimate the number of iterations that must be performed by the algorithm to find a new solution at least as good as the current best.
- Since the user is able to account for the average time taken by each GRASP iteration, the threshold defining the stopping criterion can either be fixed or determined online so as to bound the computation time when the probability of finding improving solutions becomes very small.

Revised GRASP template with stopping rule

- New template implements the termination rule that stops the iterations whenever the probability of improving the best known solution value gets smaller than or equal to β.

procedure `GRASP`(β, `Seed`)
    Set $f^* \leftarrow \infty$ and $k \leftarrow 0$;
    repeat
        $x \leftarrow$ `GreedyRandomizedAlgorithm`(`Seed`);
        $x \leftarrow$ `LocalSearch`(x);
        if $f(x) < f^*$ **then** $x^* \leftarrow x$; $f^* \leftarrow f(x)$; **end**;
        $k \leftarrow k + 1$;
        $f_k \leftarrow f(x)$;
        $ub^k \leftarrow f^*$;
        Update the average m^k and the standard deviation S^k of $f_1, \ldots, f_k$;
        Compute the estimate $\hat{F}_X^k(ub^k) = \hat{F}_X^k(f^*) = \int_{-\infty}^{f^*} \hat{f}_X^k(\tau) d\tau$;
    until $\hat{F}_X^k(f^*) < \beta$;
    return x^*;
end.

Alternative stopping rules

- Threshold β used to implement the stopping criterion may either be fixed a priori as a parameter or iteratively computed:
  - in this case, its value will be computed considering the probability of finding an improving solution (or, alternatively, the estimated number of iterations to find an improving solution) and the average computation time per iteration.
- Since the average time consumed by each GRASP iteration is known, another promising avenue of research consists in investigating stopping rules based on estimating the amount of time needed to improve the best solution found (e.g. by each percent point).

Contents

Concluding remarks

- Probabilistic stopping rules have been proposed for randomized metaheuristics.
- Solution values obtained by a metaheuristic such as GRASP have been shown experimentally to fit a Normal distribution.
- Normal approximation used to estimate the probability of finding a solution at least as good as the currently best.
- Stopping criterion is based on the probability of finding a solution at least as good as the incumbent:
  - GRASP iterations will be interrupted whenever the probability of finding a solution at least as good as the incumbent becomes smaller than or equal to a certain threshold.
  - Robustness of this strategy was illustrated and validated by a thorough computational study reporting results obtained with GRASP implementations for three combinatorial optimization problems.

Extensions and ongoing work

- Investigation of alternative termination rules:
  - Threshold β may be iteratively computed, considering the probability of finding an improving solution (or, alternatively, the estimated number of iterations to find an improving solution) and the average computation time per iteration.
  - Since the average time consumed by each GRASP iteration is known, investigate stopping rules based on estimating the amount of time needed to improve the best known solution.
- Applications to other problems will be reported elsewhere.
- Extension to other randomized metaheuristics:
  - Variable Neighborhood Search (VNS)
  - Genetic Algorithms
- Extension to memory-based metaheuristics:
  - Solution values fit Erlang distributions.
  - Application to GRASP with path-relinking heuristics.