Extending time-to-target plots to multiple instances and targets: mttt-plots

Alberto Reyes, Celso C. Ribeiro

Institute of Computing, Universidade Federal Fluminense,
Niterói, RJ 24210-346, Brazil.
albertord84@gmail.com, celso@ic.uff.br

Abstract

Time-to-target plots (ttt-plots) are a useful tool to characterize, evaluate, and compare the behavior of randomized heuristics for a given problem instance of some combinatorial optimization problem. Multiple time-to-target plots (mttt-plots) are their natural extension to sets of multiple instances. We show how to build an mttt-plots from the individual ttt-plots for each instance in the set. Finally, we give an illustrative example of the construction of the mttt-plots from the individual ttt-plots for a case study involving GRASP algorithms for the 2-path network design problem.

1 Introduction

Runtime distributions or time-to-target plots (or, simply, ttt-plots) display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value for a given problem instance within a given running time, shown on the abscissa axis. They provide a very useful tool to characterize the running times of stochastic algorithms for combinatorial optimization problems and to compare different algorithms or strategies for solving a given problem. Time-to-target plots were first used by Feo et al. [4] and have been widely used as a tool for algorithm design and comparison. They have been also advocated by Hoos and Stützle [5, 6] as a way to characterize the execution times of stochastic algorithms for combinatorial optimization.

Let $\mathcal{P}$ be an optimization problem and $\mathcal{H}$ a randomized heuristic for this problem. Furthermore, let $\mathcal{I}$ be a specific instance of $\mathcal{P}$ and let $look4$ be a target value for this instance.

Aiex et al. [2] developed a perl program to create time-to-target plots for measured times that are assumed to fit a shifted exponential distribution, following closely the work of Aiex et al. [1]. To build the ttt-plot, the heuristic $\mathcal{H}$ is run N times on the fixed instance $\mathcal{I}$ and the algorithm is made to stop as soon as a solution whose objective function is at least as good as the given target value $look4$ is found. For each of the N runs, the random number generator used in the implementation of the heuristic is initialized with a distinct seed. Therefore, the runs are assumed to be independent. The solution time of each run is recorded and saved.

After concluding the N independent runs, the solution times are sorted in increasing order. We associate with the ith sorted solution time t_i a probability $p_i = (i - 1/2)/N$ (see [12], page 114) and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, N$.

If $X \geq 0$ denotes the continuous random variable representing the time taken by heuristic $\mathcal{H}$ to find a solution as good as the target value $look4$ for instance $\mathcal{I}$, then $\hat{F}_X(t_i) = p_i$ is an estimator of $P(X \leq x) = F_X(x)$ for every $x = t_i, i = 1, \ldots, N$. Figure 1 illustrates the plot of this estimated cumulative probability distribution for some problem $\mathcal{P}$, a GRASP heuristic $\mathcal{H}$, an instance $\mathcal{I}$, and a target $look4$. We can see that the probability that the heuristic finds a solution at least as good as the target value in at most 416 seconds is about 50%, in at most 1064 seconds is about 80%, and in at most 1569 seconds is about 90%.

Ribeiro et al. [16] developed a closed form result to compare two exponential algorithms and an iterative procedure to compare two algorithms following generic runtime distributions. This work was extended by Ribeiro et al. [17] and was also applied in the comparison of parallel heuristics. Ribeiro and Rosseti [15] developed a code to compare runtime distributions of randomized algorithms. However, a possible limitation of the use of time-to-target plots to compare different algorithms for the same problem is that they convey information that is valid strictly for a single pair of problem instance and target value at

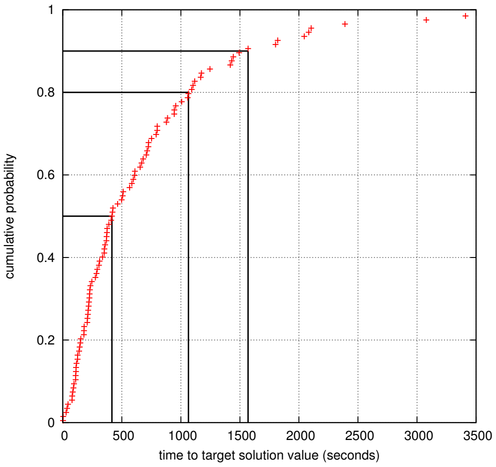


Figure 1: Cumulative probability distribution plot of measured data.

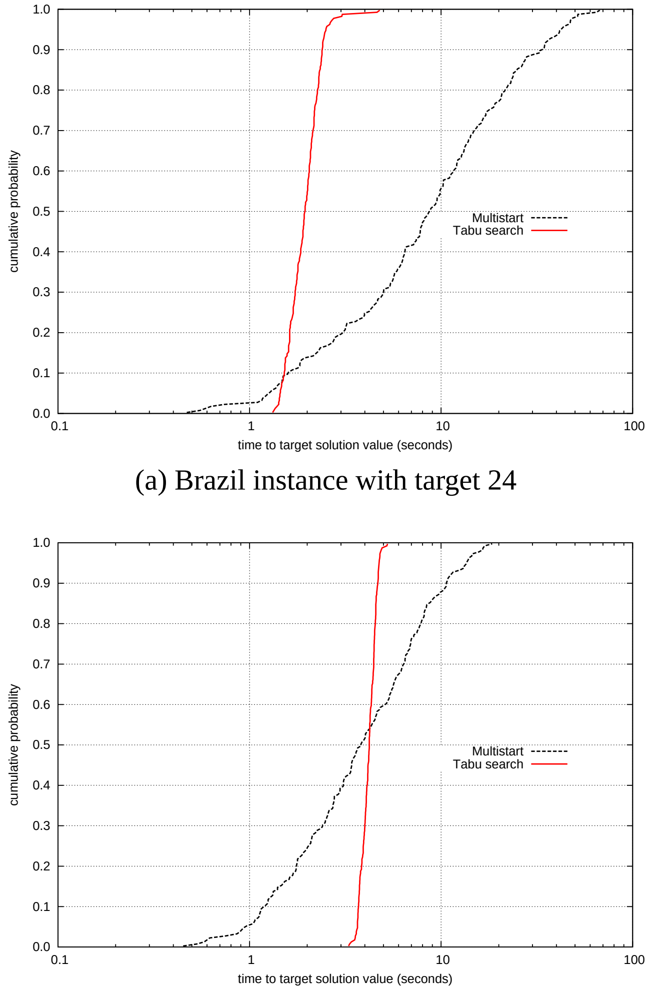
a time. To consider the implications of this assumption, let us consider one numerical example involving an instance of the routing and wavelength assignment problem [18].

In the context of this problem, a point-to-point connection between two endnodes of an optical network is called a lightpath. Two lightpaths may use the same wavelength, provided they do not share any common link. The routing and wavelength assignment problem is that of routing a set of lightpaths and assigning a wavelength to each of them, minimizing the number of wavelengths needed. Suppose that two randomized approaches are available for approximately solving this problem and one wants to compare their performance from the standpoint of their time-to-target plots. The approaches considered are a multistart greedy heuristic [8] and a tabu search decomposition scheme [9]. In addition, suppose that two networks are used for benchmarking. The first has 27 nodes representing the capitals of the 27 states of Brazil, with 70 links connecting them. There are 702 lightpaths to be routed. The second is formed by 31 nodes and 51 links, with 930 lightpaths to be routed [7], and represents an optical network in Finland. The target was set at 24 (the best known solution value) for instance Brazil and at 50 for instance Finland (the best known solution value is 47).

The time-to-target plots of the decomposition and multistart strategies are superimposed in Figure 2. The direct comparison of the two approaches shows that decomposition approach clearly outperformed the multistart strategy for instance Brazil. However, the situation changes for instance Finland. Although both algorithms have similar performances, multistart performs slightly better: Ribeiro et al. [16] showed that the probability that it will find a solution as good as the target in less time than tabu search is 0.536787 for instance Finland.

This example shows that different choices for the pair of problem instance and target value may lead to different conclusions regarding the behavior of two algorithms for the same problem. This observation reinforces the well-known fact that the comparison of different algorithms should be based on a representative set of benchmark test instances, and not only on a single one or a on a few problem instances. Therefore, the main limitation of time-to-target plots is that they can consider only one problem instance at a time and that conclusions regarding algorithm efficiency are limited to the problem instance and to the target value that generated the plot.

In this work, we define multiple time-to-target plots (or, simply, mttt-plots) in the next section, as the natural extension of ttt-plots to sets of multiple instances. In Section 3, we show how to build an mttt-plot from the individual ttt-plots for each instance in the set. We present one case study with a numerical example in Section 4, to illustrate the applicability and the usefulness of the newly proposed tool. Concluding remarks are drawn in the last section.



(a) Brazil instance with target 24

(b) Finland instance with target 50

Figure 2: Superimposed runtime distributions of multistart and tabu search.

2 Multiple ttt-plots – mttt-plots

We first introduce some basic definitions and the experimental framework that will lead to the construction of multiple time-to-target plots (mttt-plots).

As before, we are given an optimization problem $\mathcal{P}$ and a randomized heuristic $\mathcal{H}$ for this problem. However, instead of one single instance and one single target value, we now have n instances $\mathcal{I}_j$ and their corresponding targets $look4_j$, for $j = 1, \ldots, n$.

Let each $X_j \geq 0$ be a continuous random variable representing the time taken by heuristic $\mathcal{H}$ to find a solution as good as the target value $look4_j$ for instance $\mathcal{I}_j$, for $j = 1, \ldots, n$. In addition, let $F_{X_j}(x) = P(X_j \leq x)$ and $f_{X_j}(x)$ be, respectively, the cumulative distribution function and the probability density function of X_j.

Heuristic $\mathcal{A}$ is run N times for each instance $\mathcal{I}_j$, for $j = 1, \ldots, n$. Each run is interrupted when a solution at least as good as the target value $look4_j$ is found. Let $t_1^j \leq t_2^j \leq \ldots \leq t_N^j$ be the ordered running times for instance $\mathcal{I}_j$ and target $look4_j$, with $T^j = \{t_1^j, \ldots, t_N^j\}$. Then, $\hat{F}_{X_j}(x) = |\{t_i^j \leq x : i = 1, \ldots, N\}|/N$ is an estimator of $F_{X_j}(x)$ for every $x \in T^j$.

In order to define an extension of the time-to-target plot of a single instance, we consider the cumulative distribution function $F_{X_1+\ldots+X_n}(x) = P(X_1+\ldots+X_n \leq x)$ of the random variable $X_1+\ldots+X_n$, i.e., the probability that all targets of their corresponding instances be reached in total time less than or equal to x.

The mttt-plot of the instances $\mathcal{I}_j$ and their corresponding targets $look4_j$, for $j = 1, \ldots, n$, is an estimator $\hat{F}_{X_1+\ldots+X_n}(x)$ of the cumulative distribution function $F_{X_1+\ldots+X_n}(x)$.

To illustrate the computation of the estimator $\hat{F}_{X_1+\ldots+X_n}(x)$, we first consider the case $n = 2$. By the total probability theorem,

$$F_{X_1+X_2}(x) = P(X_1 + X_2 \leq x) = \int_{-\infty}^{+\infty} P(X_1 + X_2 \leq x | X_2 = v) \cdot f_{X_2}(v) dv.$$

Since both random variables X_1 and X_2 are non-negative,

$$P(X_1 + X_2 \leq x) = \int_{0}^{x} P(X_1 + X_2 \leq x | X_2 = v) \cdot f_{X_2}(v) dv$$

$$P(X_1 + X_2 \leq x) = \int_{0}^{x} P(X_1 \leq x - v) \cdot f_{X_2}(v) dv = \int_{0}^{x} F_{X_1}(x - v) \cdot f_{X_2}(v) dv.$$

For any arbitrary small real number $\varepsilon > 0$, the above expression can be approximated by

$$P(X_1 + X_2 \leq x) = \sum_{k=0}^{k=\frac{x}{\varepsilon}-1} \int_{k\varepsilon}^{(k+1)\varepsilon} F_{X_1}(x - v) \cdot f_{X_2}(v) dv. \tag{1}$$

For every $v \in [k\varepsilon, (k+1)\varepsilon]$,

$$P(X_1 \leq x - (k+1)\varepsilon) \leq P(X_1 \leq x - v) \leq P(X_1 \leq x - k\varepsilon),$$

i.e.,

$$F_{X_1}(x - (k+1)\varepsilon) \leq F_{X_1}(x - v) \leq F_{X_1}(x - k\varepsilon). \tag{2}$$

Let

$$L(\varepsilon, x) = \sum_{k=0}^{k=\frac{x}{\epsilon}-1} \int_{k\varepsilon}^{(k+1)\varepsilon} F_{X_1}(x - (k+1)\varepsilon) \cdot f_{X_2}(v) dv$$

and

$$R(\varepsilon, x) = \sum_{k=0}^{k=\frac{x}{\varepsilon}-1} \int_{k\varepsilon}^{(k+1)\varepsilon} F_{X_1}(x - k\varepsilon) \cdot f_{X_2}(v) dv.$$

Then, from equation (1) and inequalities (2),

$$L(\varepsilon, x) \leq P(X_1 + X_2 \leq x) = \sum_{k=0}^{k=\frac{x}{\varepsilon}-1} \int_{k\varepsilon}^{(k+1)\varepsilon} F_{X_1}(x - v) \cdot f_{X_2}(v) dv \leq R(\varepsilon, x),$$

$$L(\varepsilon, x) \leq P(X_1 + X_2 \leq x) \leq R(\varepsilon, x),$$

and, consequently, an estimator of the cumulative distribution function $F(X_1 + X_2)$ is given by

$$\hat{F}_{X_1+X_2}(x) = \frac{L(\varepsilon, x) + R(\varepsilon, x)}{2}, \tag{3}$$

with the approximation error $\Delta(\varepsilon, x) = R(\varepsilon, x) - L(\varepsilon, x)$ being

$$\Delta(\varepsilon, x) = \sum_{k=0}^{k=\frac{x}{\varepsilon}-1} [F_{X_1}(x - k\varepsilon) - F_{X_1}(x - (k+1)\varepsilon)] \int_{k\varepsilon}^{(k+1)\varepsilon} f_{X_2}(v) dv. \tag{4}$$

Now, let $\delta = \max_{u \geq 0}\{f_{X_1}(u)\}$. Then, $F_{X_1}(x - k\varepsilon) - F_{X_1}(x - (k+1)\varepsilon) \leq \delta\varepsilon$ and

$$\Delta(\varepsilon, x) \leq \sum_{k=0}^{k=\frac{x}{\varepsilon}-1} \delta\varepsilon \int_{k\varepsilon}^{(k+1)\varepsilon} f_{X_2}(v) dv = \delta\varepsilon \sum_{k=0}^{k=\frac{x}{\varepsilon}-1} \int_{k\varepsilon}^{(k+1)\varepsilon} f_{X_2}(v) dv = \delta\varepsilon F_{X_2}(x) \leq \delta\varepsilon.$$

The approximation error can be made as small as desired by choosing a sufficiently small value for ε.

However, in practice, the probability distributions are unknown. Neither the cumulative distribution functions $F_{X_1}(x)$ and $F_{X_2}(x)$ nor the probability density functions $f_{X_1}(x)$ and $f_{X_2}(x)$ are known. Instead of them, all information available is a large number N of observations of the random variables X_1 and X_2, and the estimators $\hat{F}_{X_1}(x)$ of $F_{X_1}(x)$ and $\hat{F}_{X_2}(x)$ of $F_{X_2}(x)$. Since the value of $\delta = \max_{u \geq 0}\{f_{X_1}(u)\}$ is also unknown beforehand, the appropriate value of ε cannot be estimated. However, for any ε sufficiently small, $f_{X_2}(v)$ may be approximated by $\hat{f}_{X_2}(v) = (\hat{F}_{X_2}((k+1)\epsilon) - \hat{F}_{X_2}(k\varepsilon))/\varepsilon$ for every $v \in [k\varepsilon, (k+1)\varepsilon]$ in the above computations. Using this approximation, $\hat{F}_{X_1+X_2}(x)$ can be calculated following equation (3). If the approximation error $\Delta(\varepsilon) = R(\varepsilon) - L(\varepsilon)$ given by equation (4) is sufficiently small, then the procedure stops. Otherwise, the value of ε is halved and the procedure is repeated until convergence, following a similar scheme to that proposed by [17].

For the general case $n > 2$, $\hat{F}_{X_1+\ldots+X_2}(x)$ could be recursively computed by defining $Y_1 = X_1$, $Y_i = X_i + Y_{i-1}$, for $i = 2, \ldots, n$, and $\hat{F}_{X_1+\ldots+X_2}(x) = \hat{F}_{Y_n}(x)$.

However, the numerical procedure described in this section is not feasible in terms of the processing times needed to generate the mttt-plots, due to time taken by integration calculations. To overcome this limitation, we describe in the next section the construction of the mttt-plots by simulation.

3 Construction by simulation

We assume that we want to create the mttt-plot associated with an experiment resulting in N running times taken by the randomized heuristic $\mathcal{H}$ to reach each target $look4_j$ of instance $\mathcal{I}_j$ of problem $\mathcal{P}$, for $j = 1, \ldots, n$, i.e., each individual ttt-plot has N points.

The mttt-plot will be defined by a set of M points $(s_k, \hat{F}_{X_1+\ldots+X_n}(s_k))$, for $k = 1, \ldots, M$, where each s_k is a sample of $X_1 + \ldots + X_n$ and $\hat{F}_{X_1+\ldots+X_n}$ is an estimator of $F_{X_1+\ldots+X_n}$. In order to generate these M points of the mttt-plot by simulation, we simply sample M occurrences of the sum of independent variables $X_1 + \ldots + X_n$ using the algorithm whose pseudo-code appears in Figure 3.

```
begin mttt-plot($M$);
1   for $k = 1, \ldots, M$ do
2       $s_k \leftarrow 0$;
3       for $j = 1, \ldots, n$ do
4           Randomly sample $time$ from the random variable $X_j$;
5           $s_k \leftarrow s_k + time$;
6       end-for;
7   end-for;
8   Sort the sampled times $(s_k, k = 1, \ldots, M)$ in non-decreasing order;
9   Associate a probability $p_k = (k - 1/2)/M$ with the $k$th sorted time $s_k$, for $k = 1, \ldots, M$;
10  Call ttt-plot to plot the points $z_k = (s_k, p_k)$, for $k = 1, \ldots, M$;
end mttt-plot.
```

Figure 3: Computing the mttt-plot with M points by simulation.

The loop in lines 1 to 7 generates the M points, where $M \gg N$ is a parameter defined by the user. Line 2 sets to 0 the kth running time, for $k = 1, \ldots, M$. The loop in lines 3 to 6 samples one running time from each random variable X_j, for $j = 1, \ldots, n$, and adds up them all, saving the sampled value in s_k. Line 8 sorts the sampled times in non-decreasing order. Line 9 computes the probability associated with each sampled running time, as explained in Section 1. Finally, line 10 makes an external call to the ttt-plot code available in Aiex et al. [2] to draw the mttt-plot.

Furthermore, for each sampled point s_k, $\hat{F}_{X_1+\ldots+X_n}(s_k) = \sum_{\ell=1}^{\ell=k} p_\ell, \quad k = 1, \ldots, n.$

However, in practice the probability distributions of the independent random variables X_j, for $j = 1, \ldots, n$, are not known. Therefore, instead of sampling from the probability distribution of each random variable X_j in line 4 of the previous algorithm, we sample from the observed running times $T^j = \{t_1^j, \ldots, t_N^j\}$ for each instance $\mathcal{I}_j$ and target $look4_j$, with $t_1^j \leq t_2^j \leq \ldots \leq t_N^j$. The pseudo-code of the resulting algorithm is fully described in Figure 4.

```
begin mttt-plot($M$);
1   for $k = 1, \ldots, M$ do
2       $s_k \leftarrow 0$;
3       for $j = 1, \ldots, n$ do
4           Randomly select $time$ from $\{t_i^j, i = 1, \ldots, N\}$;
5           $s_k \leftarrow s_k + time$;
6           end-for;
7       end-for;
8   Sort the sampled times $(s_k, k = 1, \ldots, M)$ in non-decreasing order;
9   Associate a probability $p_k = (k - 1/2)/M$ with the $k$th sorted time $s_k$, for $k = 1, \ldots, M$;
10  Call ttt-plot to plot the points $z_k = (s_k, p_k)$, for $k = 1, \ldots, M$;
end mttt-plot.
```

Figure 4: Computing the mttt-plot with M points by simulation (final).

We notice that the number of runs of each instance and target may vary and does not have to be necessarily equal to the same value N for all of them.

4 Case study: numerical example

In this section, we give an illustrative example of the construction of the mttt-plots from the individual ttt-plots for a case study involving GRASP algorithms for the 2-path network design problem.

Given a connected undirected graph with non-negative weights associated with its edges, together with a set of origin-destination nodes, the 2-path network design problem consists in finding a minimum weighted subset of edges containing a path formed by at most two edges between every origin-destination pair. Applications can be found in the design of communication networks, in which paths with few edges are sought to enforce high reliability and small delays. Its decision version is NP-complete [3].

We illustrate the construction of mttt-plots by comparing two heuristics for approximately solving this problem. The first is a GRASP with forward path-relinking, while the second is a GRASP with backward path-relinking [10, 11, 13, 14]. We have considered five randomly instances with 100 nodes, 4950 edges, 1000 demands, and edge weights in the interval $[1, 10]$. For each instance, two target values were considered (700 and 710), making a total of $n = 10$ pairs instance-target. Each algorithm was run $N = 200$ times for each pair instance-target, until a solution at least as good as the corresponding target was found for each instance. Figure 5 displays the individual ttt-plots for GRASP with forward path-relinking and GRASP with backward path-relinking for each of the ten pairs instance-target.

Figure 6 displays the mttt-plot resulting from the ten individual ttt-plots by the simulation algorithm described in Figure 4 using $M = 10^6$. The application of the tttplots-compare tool developed by [15] shows that GRASP with backward path-relinking performs better for this set of ten instances and ten targets, since the probability that the time taken by GRASP with backward path-relinking to find a solution at least as good as the target is less than or equal to the time taken by GRASP with forward path-relinking is 0.808776.

5 Concluding remarks

This work is a continuation of previous research reported in [1, 2] that led to the development and use of time-to-target plots (ttt-plots) as a very useful tool to characterize the running times of stochastic

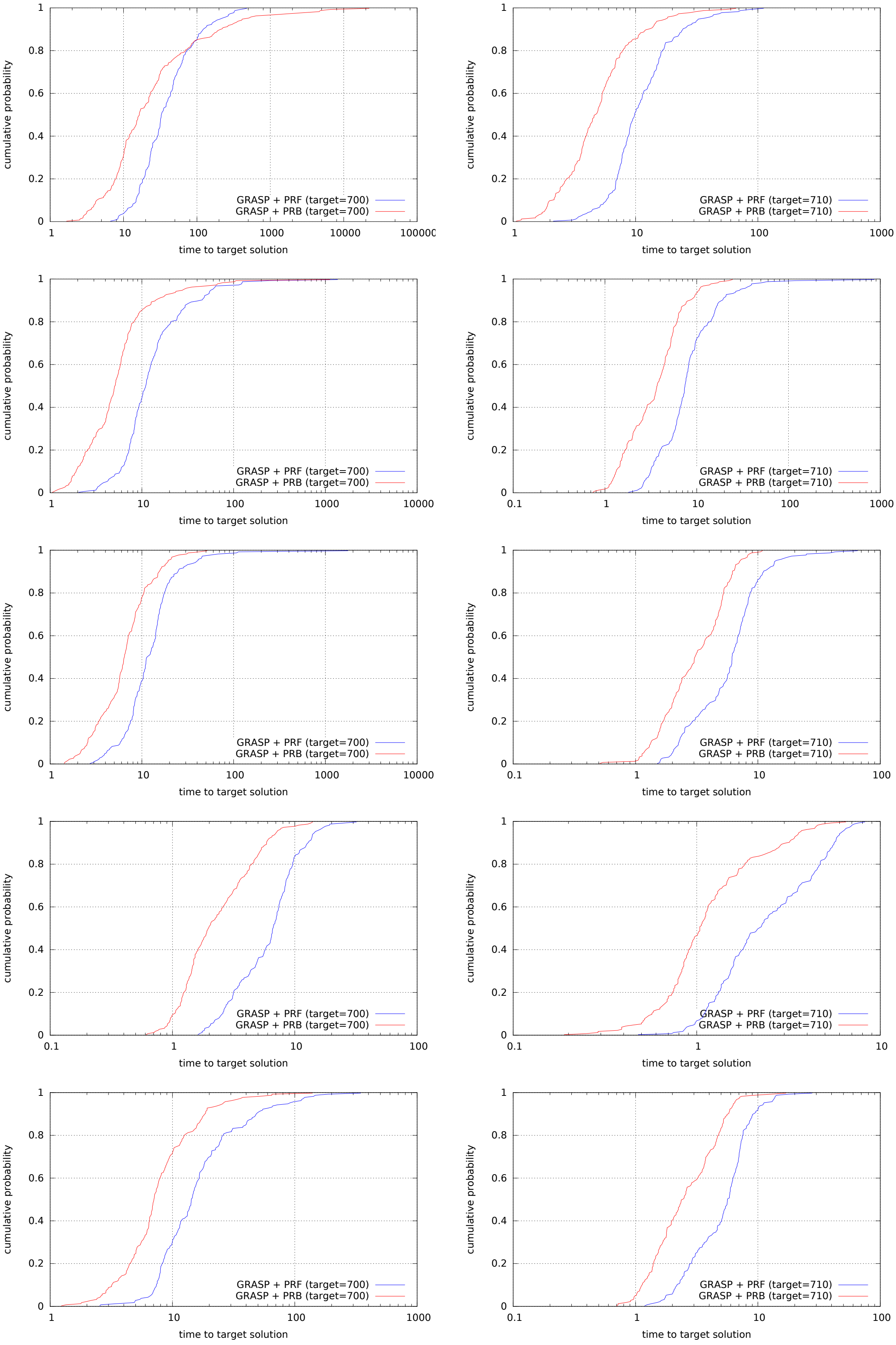


Figure 5: 2-path network design problem: individual ttt-plots for GRASP with forward path-relinking vs. GRASP with backward path-relinking for ten instance-target pairs.

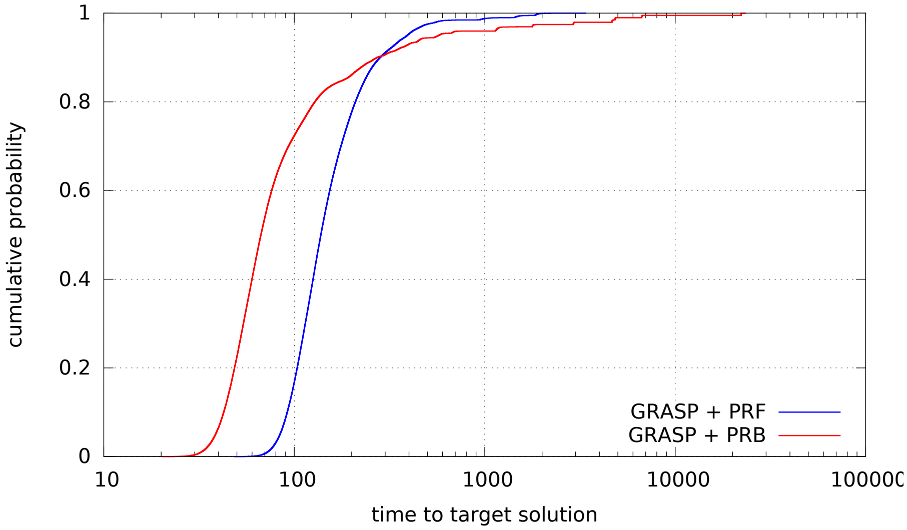


Figure 6: 2-path network design problem: combined mttt-plots for GRASP with forward path-relinking vs. GRASP with backward path-relinking for ten instance-target pairs obtained by simulation with $M = 10^6$ points each.

algorithms for combinatorial optimization problems and to compare different algorithms or strategies for solving a given problem.

We showed in this work that time-to-target plots of individual problems can be extended to sets of multiple instances. Multiple ttt-plots (mttt-plots) may be skillfully numerically computed from the individual ttt-plots of each instance by simulation. Their use allow for more solid conclusions about the comparison of different randomized heuristics for the same combinatorial optimization problem.

References

[1] R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. Probability distribution of solution time in GRASP: An experimental investigation. *Journal of Heuristics*, 8:343–373, 2002.

[2] R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. TTTPLOTS: A perl program to create time-to-target plots. *Optimization Letters*, 1:355–366, 2007.

[3] G. Dahl and B. Johannessen. The 2-path network design problem. *Networks*, 43:190–199, 2004.

[4] T.A. Feo, M.G.C. Resende, and S.H. Smith. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42:860–878, 1994.

[5] H.H. Hoos and T. Stützle. Evaluating Las Vegas algorithms – Pitfalls and remedies. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 238–245, Madison, 1998.

[6] H.H. Hoos and T. Stützle. On the empirical evaluation of Las Vegas algorithms - Position paper. Technical report, Computer Science Department, University of British Columbia, Vancouver, 1998.

[7] E. Hyytiä and J. Virtamo. Wavelength assignment and routing in WDM networks. In *Proceedings of the Fourteenth Nordic Teletraffic Seminar NTS-14*, pages 31–40, Lyngby, 1998.

[8] P. Manohar, D. Manjunath, and R.K. Shevgaonkar. Routing and wavelength assignment in optical networks from edge disjoint path algorithms. *IEEE Communications Letters*, 5:211–213, 2002.

[9] T.F. Noronha and C.C. Ribeiro. Routing and wavelength assignment by partition coloring. *European Journal of Operational Research*, 171:797–810, 2006.

[10] M.G.C. Resende and C.C. Ribeiro. A GRASP with path-relinking for private virtual circuit routing. *Networks*, 41:104–114, 2003.

[11] M.G.C. Resende and C.C. Ribeiro. GRASP with path-relinking: Recent advances and applications. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: Progress as real problem solvers*, pages 29–63. Springer, New York, 2005.

[12] M.G.C. Resende and C.C. Ribeiro. *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*. Springer, New York, 2016.

[13] C.C. Ribeiro and I. Rosseti. A parallel GRASP heuristic for the 2-path network design problem. In B. Monien and R. Feldmann, editors, *Euro-Par 2002 Parallel Processing*, volume 2400 of *Lecture Notes in Computer Science*, pages 922–926. Springer, Berlin, 2002.

[14] C.C. Ribeiro and I. Rosseti. Efficient parallel cooperative implementations of GRASP heuristics. *Parallel Computing*, 33:21–35, 2007.

[15] C.C. Ribeiro and I. Rosseti. tttplots-compare: A perl program to compare time-to-target plots or general runtime distributions of randomized algorithms. *Optimization Letters*, 9:601–614, 2015.

[16] C.C. Ribeiro, I. Rosseti, and R. Vallejos. On the use of run time distributions to evaluate and compare stochastic local search algorithms. In T. Stützle, M. Biratari, and H.H. Hoos, editors, *Engineering stochastic local search algorithms*, volume 5752 of *Lecture Notes in Computer Science*, pages 16–30. Springer, Berlin, 2009.

[17] C.C. Ribeiro, I. Rosseti, and R. Vallejos. Exploiting run time distributions to compare sequential and parallel stochastic local search algorithms. *Journal of Global Optimization*, 54:405–429, 2012.

[18] H. Zang, J.P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 1:47–60, 2000.