On the use of run time distributions to evaluate and compare stochastic local search algorithms

Celso C. Ribeiro[1], Isabel Rosseti[2], and Reinaldo Vallejos[3]

[1] Department of Computer Science, Universidade Federal Fluminense, Rua Passo da Pátria 156, Niterói, RJ 24210-240, Brazil.
[2] Department of Science and Technology, Universidade Federal Fluminense, Polo Universitário de Rio das Ostras, Rio das Ostras, RJ 28890-000, Brazil.
[3] Universidad Técnica Federico Santa María, Telematics Group, Department of Electronic Engineering, Av. España 1680, Valparaíso, Chile
{celso,rosseti}@ic.uff.br, reinaldo.vallejos@usm.cl

Abstract. Run time distributions or time-to-target plots are very useful tools to characterize the running times of stochastic algorithms for combinatorial optimization. We further explore run time distributions and describe a new tool to compare two algorithms based on stochastic local search. For the case where the running times of both algorithms fit exponential distributions, we derive a closed form index that gives the probability that one of them finds a solution at least as good as a given target value in a smaller computation time than the other. This result is extended to the case of general run time distributions and a numerical iterative procedure is described for the computation of the above probability value. Numerical examples illustrate the application of this tool in the comparison of different algorithms for three different problems.

1 Motivation

Run time distributions or time-to-target plots display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. Time-to-target plots were first used by Feo et al. [8]. Run time distributions have been advocated also by Hoos and Stützle [12, 11] as a way to characterize the running times of stochastic algorithms for combinatorial optimization.

It has been observed that in many implementations of local search heuristics for combinatorial optimization problems, such as simulated annealing, genetic algorithms, iterated local search, tabu search, and GRASP [2, 6, 7, 10, 14, 19, 27, 28, 29], the random variable *time to target value* fits an exponential (or a shifted exponential) distribution. Hoos and Stützle [13, 14] conjecture that this is true for all local search methods for combinatorial optimization.

Aiex et al. [3] describe a perl program to create time-to-target plots for measured times that are assumed to fit a shifted exponential distribution, following [2]. Such plots are very useful in the comparison of different algorithms for solving a given problem and have been widely used as a tool for algorithm design and comparison.

In this work, we further explore run time distributions to evaluate stochastic local search algorithms. We describe a new tool to compare any pair of different stochastic local search algorithms and we use it in the investigation of different applications. Under the assumption that the running times of the two algorithms follow exponential (or shifted exponential) distributions, we develop in Section 2 a closed form index that gives the probability that one of the algorithms finds a target solution value in a smaller computation time than the other. In Section 3, this result is extended to the case of general run time distributions and a numerical iterative procedure is described for the computation of such probability. Applications illustrating the comparison of different algorithms for the same problem appear in Section 4. Concluding remarks are made in the last section.

2 Comparing exponential-time algorithms

We assume the existence of two stochastic local search algorithms A_1 and A_2 for some combinatorial optimization problem. Furthermore, we assume that their running times fit exponential (or shifted exponential) distributions. We denote by X_1 (resp. X_2) the continuous random variable representing the time needed by algorithm A_1 (resp. A_2) to find a solution as good as a given target value:

$$X_1 \mapsto \begin{cases} 0, & \tau < T_1 \\ \lambda_1 e^{-\lambda_1(\tau - T_1)}, & \tau \geq T_1 \end{cases}$$

and

$$X_2 \mapsto \begin{cases} 0, & \tau < T_2 \\ \lambda_2 e^{-\lambda_2(\tau - T_2)}, & \tau \geq T_2 \end{cases}$$

where T_1, λ_1, T_2, and λ_2 are parameters. The cumulative probability distribution and the probability density function of X_1 are depicted in Figure 1.

Since both algorithms stop when they find a solution at least as good as the target, we may say that algorithm A_1 performs better than A_2 if the former stops before the latter. Therefore, we must evaluate the probability that X_1 takes a value smaller than or equal to X_2, i.e. we compute $Pr(X_1 \leq X_2)$. Conditioning on the value of X_2 and applying the total probability theorem, we obtain:

$$Pr(X_1 \leq X_2) = \int_{-\infty}^{\infty} Pr(X_1 \leq X_2 | X_2 = \tau) f_{X_2}(\tau) d\tau =$$

$$= \int_{T_2}^{\infty} Pr(X_1 \leq X_2 | X_2 = \tau) \lambda_2 e^{-\lambda_2(\tau - T_2)} d\tau = \int_{T_2}^{\infty} Pr(X_1 \leq \tau) \lambda_2 e^{-\lambda_2(\tau - T_2)} d\tau.$$

Let $\nu = \tau - T_2$. Then, $d\nu = d\tau$ and

$$Pr(X_1 \leq X_2) = \int_{0}^{\infty} Pr[X_1 \leq (\nu + T_2)] \lambda_2 e^{-\lambda_2 \nu} d\nu. \tag{1}$$

To solve the above integral, one first has to compute

$$Pr[X_1 \leq (\nu + T_2)] = \int_{-\infty}^{\nu + T_2} f_{X_1}(\tau) d\tau.$$

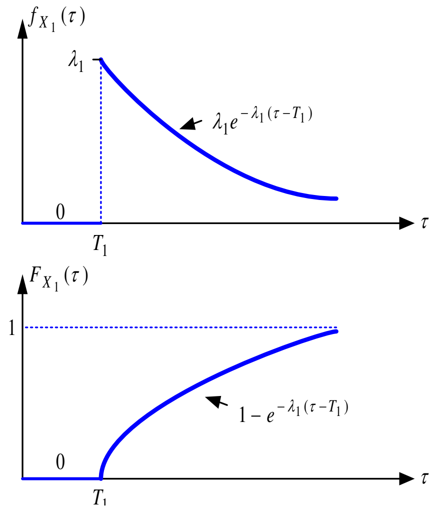


Fig. 1. Probability density function and cumulative probability distribution of X_1.

Assuming that $T_2 \geq T_1$, without loss of generality, we have that:

$$Pr[X_1 \leq (\nu + T_2)] = \int_{T_1}^{\nu+T_2} \lambda_1 e^{-\lambda_1(\tau - T_1)} d\tau.$$

Now, let $w = \tau - T_1$. Then, $dw = d\tau$ and

$$Pr[X_1 \leq (\nu + T_2)] = \int_{0}^{\nu+T_2-T_1} \lambda_1 e^{-\lambda_1 w} dw = 1 - e^{-\lambda_1(\nu+T_2-T_1)}. \tag{2}$$

Replacing (2) in equation (1), we obtain

$$Pr(X_1 \leq X_2) = \int_{0}^{\infty} [1 - e^{-\lambda_1(\nu+T_2-T_1)}]\lambda_2 e^{-\lambda_2 \nu} d\nu =$$

$$= 1 - e^{-\lambda_1(T_2-T_1)} \int_{0}^{\infty} e^{-\nu(\lambda_1+\lambda_2)} d\nu = 1 + e^{-\lambda_1(T_2-T_1)} \lambda_2 \frac{e^{-\nu(\lambda_1+\lambda_2)}}{\lambda_1+\lambda_2} \Bigg|_{\nu=0}^{\nu=\infty} .$$

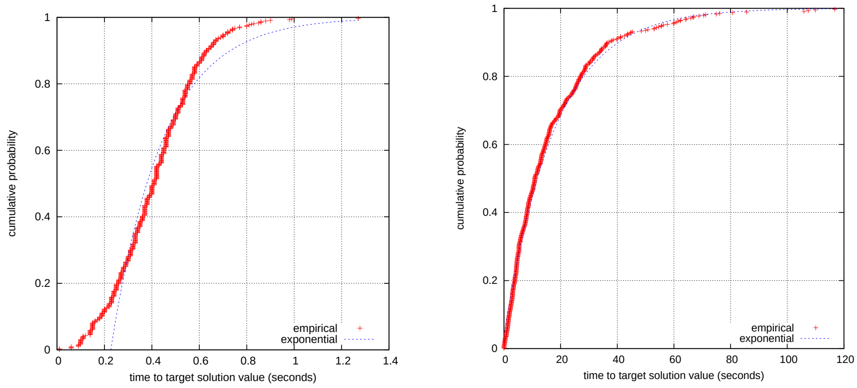
Finally,

$$Pr(X_1 \leq X_2) = 1 - e^{-\lambda_1(T_2-T_1)} \frac{\lambda_2}{\lambda_1+\lambda_2}. \tag{3}$$

This result can be better interpreted by rewriting expression (3) as:

$$Pr(X_1 \leq X_2) = (1 - e^{-\lambda_1(T_2-T_1)}) + e^{-\lambda_1(T_2-T_1)} \frac{\lambda_1}{\lambda_1+\lambda_2}. \tag{4}$$

The first term of the right-hand side of equation (4) is the probability that $0 \leq X_1 \leq T_2$, in which case X_1 is clearly less than or equal to X_2. The second



(a) GRASP with bidirectional path-relinking

(b) Pure GRASP

Fig. 2. Run time distributions on an instance of the 2-path network design problem with 80 nodes and 800 origin-destination pairs, with target value set at 588.

term of (4) is the probability that X_1 be greater than T_2 and less than or equal to X_2, given that $X_1 \geq T_2$, which completes the interpretation.

To illustrate the above result, we consider two algorithms described in [25] for solving the 2-path network design problem. Algorithm A_1 is an implementation of GRASP with bidirectional path-relinking, while algorithm A_2 is a pure GRASP heuristic. Figure 2 depicts the run time distributions obtained after 500 runs with different seeds on an instance with 80 nodes and 800 origin-destination pairs, with the target value set at 588. The plots have been obtained with the perl tool provided in [3], which also computed the parameters of the two distributions: $\lambda_1 = 0.218988$, $T_1 = 0.01$, $\lambda_2 = 17.829236$, and $T_2 = 0.01$. Applying expression (3), we get $Pr(X_1 \leq X_2) = 0.943516$. This probability is consistent with Figure 3, in which the run time distribution of GRASP with bidirectional path-relinking is much to the left of that of pure GRASP for the same instance.

Aiex et al. [2] have shown experimentally that the time taken by a GRASP heuristic to find a solution at least as good as a given target value fits an exponential distribution. If the setup times are not negligible, it fits a two-parameter shifted exponential distribution. The experiments involved 2,400 runs of five problems: maximum stable set [8], quadratic assignment [16], graph planarization [22], maximum weighted satisfiability [21], and maximum covering [20].

However, if path-relinking is applied as an intensification step at the end of each iteration [4, 24, 25], then the iterations are no longer independent and the memoryless characteristic of GRASP is destroyed. Consequently, the time-to-target random variable may not fit an exponential distribution.

This claim is illustrated by two implementations of GRASP with path-relinking. The first is an application to the 2-path network design problem [25].

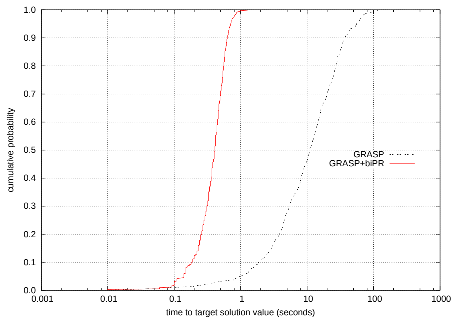


Fig. 3. Superimposed run time distributions of GRASP with bidirectional path-relinking and pure GRASP.

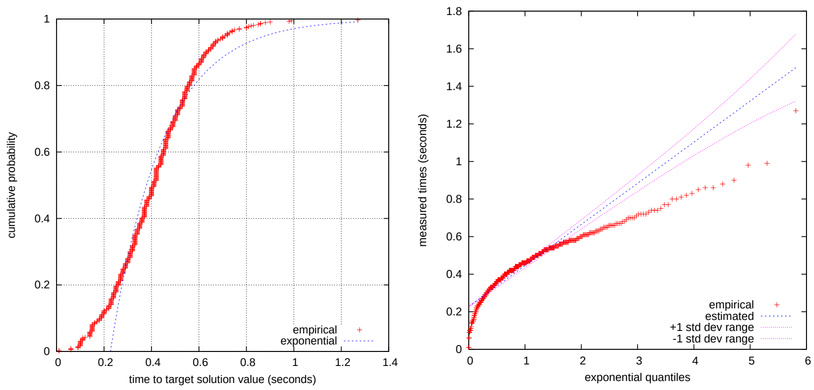


Fig. 4. Run time distribution and quantile-quantile plot for GRASP with bidirectional path-relinking on an instance of the 2-path network design problem with 80 nodes and 800 origin-destination pairs, with target set to 588.

The run time distribution and the quantile-quantile plot for an instance with 80 nodes and 800 origin-destination pairs are depicted in Figure 4. The second is an application to the three-index assignment problem [1]. Run time distributions and quantile-quantile plots for Balas and Saltzman problems 22.1 (target set to 8) and 24.1 (target set to 7) are shown in Figures 5 and 6, respectively. We observe that points steadily deviate by more than one standard deviation from the estimate for the upper quantiles in the quantile-quantile plots (i.e., many points associated with large computation times fall outside the plus or minus one standard deviation bounds). Therefore, we may say that these run time distributions are not exponential.

If the running times do not fit exponential distributions, then the result established by expression (3) does not hold. Therefore, this approach is extended to general run time distributions in the next section.

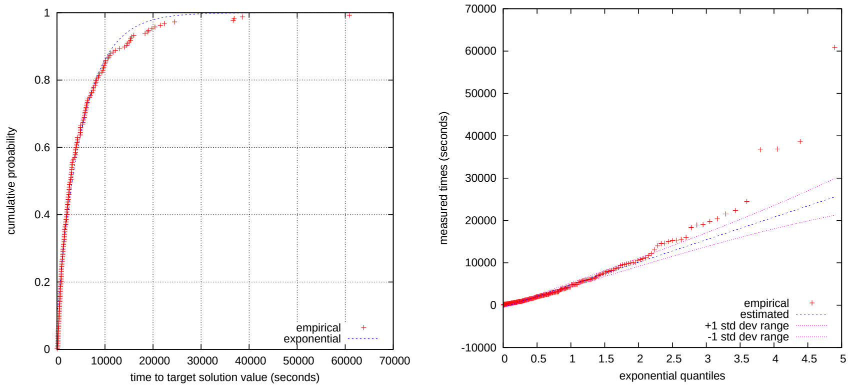


Fig. 5. Run time distribution and quantile-quantile plot for GRASP with bidirectional path-relinking on Balas and Saltzman problem 22.1, with target set to 8.

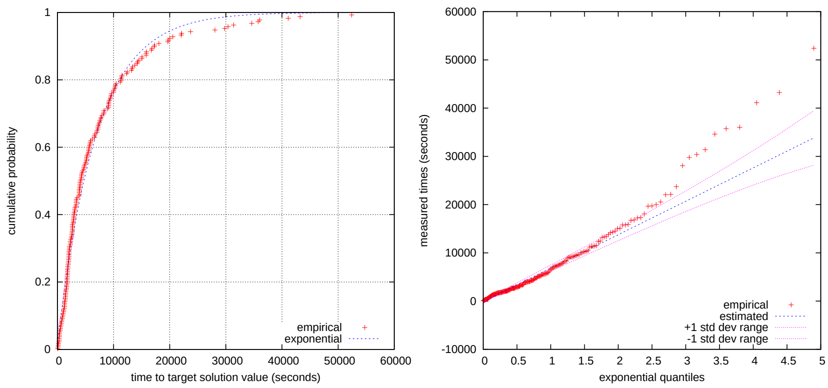


Fig. 6. Run time distribution and quantile-quantile plot for GRASP with bidirectional path-relinking on Balas and Saltzman problem 24.1, with target value to 7.

3 General run time distributions

Let X_1 and X_2 be continuous random variables, with cumulative probability distributions $F_{X_1}(\tau)$ and $F_{X_2}(\tau)$ and probability density functions $f_{X_1}(\tau)$ and

$f_{X_2}(\tau)$. Then,

$$Pr(X_1 \le X_2) = \int_{-\infty}^{\infty} Pr(X_1 \le \tau) f_{X_2}(\tau) d\tau = \int_{0}^{\infty} Pr(X_1 \le \tau) f_{X_2}(\tau) d\tau,$$

since $f_{X_1}(\tau) = f_{X_2}(\tau) = 0$ for any $\tau < 0$. For an arbitrary small real number ε, the above expression can be rewritten as

$$Pr(X_1 \le X_2) = \sum_{i=0}^{\infty} \int_{i\varepsilon}^{(i+1)\varepsilon} Pr(X_1 \le \tau) f_{X_2}(\tau) d\tau. \tag{5}$$

Since $Pr(X_1 \le i\varepsilon) \le Pr(X_1 \le \tau) \le Pr(X_1 \le (i+1)\varepsilon)$ for $i\varepsilon \le \tau \le (i+1)\varepsilon$, replacing $Pr(X_1 \le \tau)$ by $Pr(X_1 \le i\varepsilon)$ and by $Pr(X_1 \le (i+1)\varepsilon)$ in (5) leads to

$$\sum_{i=0}^{\infty} F_{X_1}(i\varepsilon) \int_{i\varepsilon}^{(i+1)\varepsilon} f_{X_2}(\tau) d\tau \le Pr(X_1 \le X_2) \le \sum_{i=0}^{\infty} F_{X_1}((i{+}1)\varepsilon) \int_{i\varepsilon}^{(i+1)\varepsilon} f_{X_2}(\tau) d\tau.$$

Let $L(\varepsilon)$ and $R(\varepsilon)$ be the value of the left and right hand sides of the above expression, respectively, with $\Delta(\varepsilon) = R(\varepsilon) - L(\varepsilon)$ being the difference between the upper and lower bounds of $Pr(X_1 \le X_2)$. Then,

$$\Delta(\varepsilon) = \sum_{i=0}^{\infty} [F_{X_1}((i+1)\varepsilon) - F_{X_1}(i\varepsilon)] \int_{i\varepsilon}^{(i+1)\varepsilon} f_{X_2}(\tau) d\tau.$$

Let $\delta = \max_{\tau \ge 0}\{f_{X_1}(\tau)\}$. Since $|F_{X_1}((i+1)\varepsilon) - F_{X_1}(i\varepsilon)| \le \delta\varepsilon$ for $i \ge 0$,

$$\Delta(\varepsilon) \le \sum_{i=0}^{\infty} \delta\varepsilon \int_{i\varepsilon}^{(i+1)\varepsilon} f_{X_2}(\tau) d\tau = \delta\varepsilon \int_{0}^{\infty} f_{X_2}(\tau) d\tau = \delta\varepsilon.$$

In order to evaluate a good approximation to $Pr(X_1 \le X_2)$, we select the appropriate value of ε such that the resulting approximation error $\Delta(\varepsilon)$ is sufficiently small. Next, we compute $L(\varepsilon)$ and $R(\varepsilon)$ to obtain the approximation

$$Pr(X_1 \le X_2) \approx \frac{L(\varepsilon) + R(\varepsilon)}{2}. \tag{6}$$

In practice, the probability distributions are unknown. Instead of them, all the information available is a large number N of observations of the random variables X_1 and X_2. Since $\delta = \max_{\tau \ge 0}\{f_{X_1}(\tau)\}$ is unknown, the value of ε cannot be estimated. Then, we proceed iteratively as follows.

Let $t_1(j)$ (resp. $t_2(j)$) be the value of the j-th smallest observation of X_1 (resp. X_2), for $j = 1, \ldots, N$. We set the bounds $a = \min\{t_1(1), t_2(1)\}$ and $b = \max\{t_1(N), t_2(N)\}$ and choose an arbitrary number h of integration intervals to compute an initial value for the integration interval $\varepsilon = (b - a)/h$. For small values of ε, the probability density function $f_{X_1}(\tau)$ in the interval $[i\varepsilon, (i+1)\varepsilon]$ can be approximated by $\hat{f}_{X_1}(\tau) = (\hat{F}_{X_1}((i+1)\varepsilon) - \hat{F}_{X_1}(i\varepsilon))/\varepsilon$, where

$$\hat{F}_{X_1}(i\varepsilon) = |\{t_1(j), j = 1, \ldots, N : t_1(j) \le i\varepsilon\}|.$$

The same approximations hold for random variable X_2.

Finally, the value of $Pr(X_1 \leq X_2)$ can be computed as in (6), using the estimates $\hat{f}_{X_1}(\tau)$ and $\hat{f}_{X_2}(\tau)$ in the computation of $L(\varepsilon)$ and $R(\varepsilon)$. If the approximation error $\Delta(\varepsilon) = R(\varepsilon) - L(\varepsilon)$ is sufficiently small, then the procedure stops. Otherwise, the value of ε is halved and the above steps are repeated.

4 Numerical applications

We apply the tool described in the previous section to compare pairs of stochastic local search algorithms running on the same instance of three different test problems: server replication for reliable multicast, routing and wavelength assignment, and 2-path network design.

4.1 DM-D5 and GRASP algorithms for server replication

Current multicast services use a delivery tree, whose root represents the sender, leaves represent the receivers, and internal nodes represent relaying servers. Transmission is performed by creating copies of the data at split points of the tree. A successful technique to provide a reliable multicast service is the server replication approach, in which data is replicated at some multicast-capable relaying servers and each of them is responsible for the retransmission of packets to receivers in its group. The problem consists of selecting the best multicast-capable relaying hosts to act as replicated servers in a multicast scenario.

DM-GRASP is a hybrid version of the GRASP metaheuristic that incorporates a data-mining process [26]. Its basic principle consists of mining for patterns found in good-quality solutions to guide the construction of new solutions. We compare two different heuristics for the server replication problem: algorithm A_1 is an implementation of the DM-D5 version [9] of DM-GRASP, in which the mining algorithm is periodically applied, while A_2 is a pure GRASP heuristic. We present illustrative results for two instances using the same network scenario, with $m = 25$ and $m = 50$ replication servers.

Each algorithm was run 200 times with different seeds. The target was set at 2,818.925 for the instance with $m = 25$ and at 2,299.07 for that with $m = 50$. Figures 7 and 8 depict run time distributions and quantile-quantile plots for DM-D5. Running times of the latter did not fit exponential distributions for any of the instances. GRASP running times were exponential for both. The run time distributions of DM-D5 and GRASP are superimposed in Figure 9. Algorithm DM-D5 outperformed GRASP, since the run-time distribution of the first is slightly to the left of that of the second for the instance with $m = 25$, and much more clearly for $m = 50$. Consistently, the computations show that $Pr(X_1 \leq X_2) = 0.614775$ and $Pr(X_1 \leq X_2) = 0.849163$ for the instances with $m = 25$ and $m = 50$, respectively.

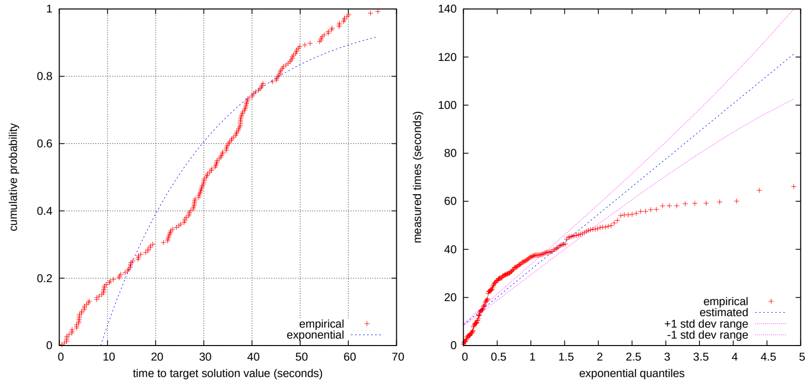


Fig. 7. Run time distribution and quantile-quantile plot for DM-D5 algorithm on the instance with $m = 25$ and target value set at 2,818.925.

4.2 Multistart and tabu search algorithms for routing and wavelength assignment

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. Noronha and Ribeiro [18] proposed a decomposition heuristic for this problem. First, a set of routes is precomputed for each lightpath. Next, one of them and a wavelength are assigned to each lightpath by a tabu search heuristic solving a partition coloring problem.

We compare this decomposition strategy with the multistart greedy heuristic of Manohar et al. [17]. Two networks are used for benchmarking. The first has 27 nodes representing the capital cities in Brazil, with 70 links connecting them. There are 702 lightpaths to be routed. Instance [15] Finland is formed by 31 nodes and 51 links, with 930 lightpaths to be routed.

Each algorithm was run 200 times with different seeds. The target was set at 24 for instance Brazil and at 50 for instance Finland. Algorithm A_1 is the multistart heuristic, while A_2 is the tabu search decomposition scheme. The multistart running times fit exponential distributions for both instances. Figures 10 and 11 display run time distributions and quantile-quantile plots for instances Brazil and Finland, respectively. The run time distributions of the decomposition and multistart strategies are superimposed in Figure 12. The direct comparison of the two approaches shows that decomposition clearly outperformed the multistart strategy for instance Brazil, since $Pr(X_1 \leq X_2) = 0.106766$ in this case. However, the situation changes for instance Finland. Although both algorithms have similar performances, multistart is slightly better with respect to the measure proposed in this work, since $Pr(X_1 \leq X_2) = 0.545619$.

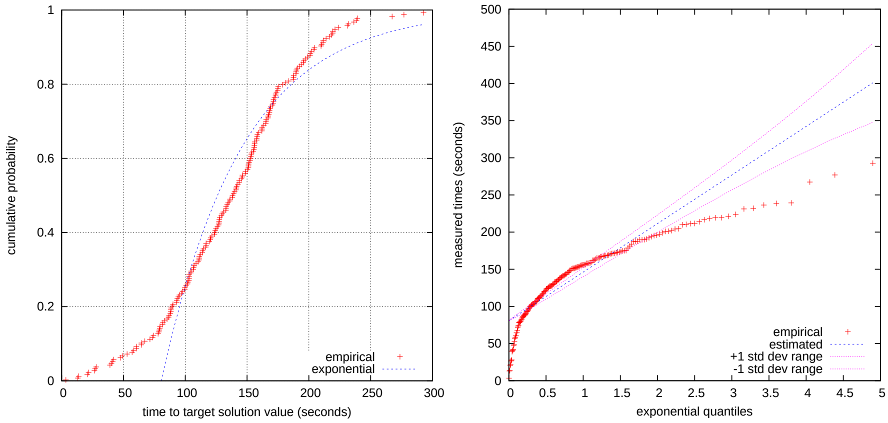
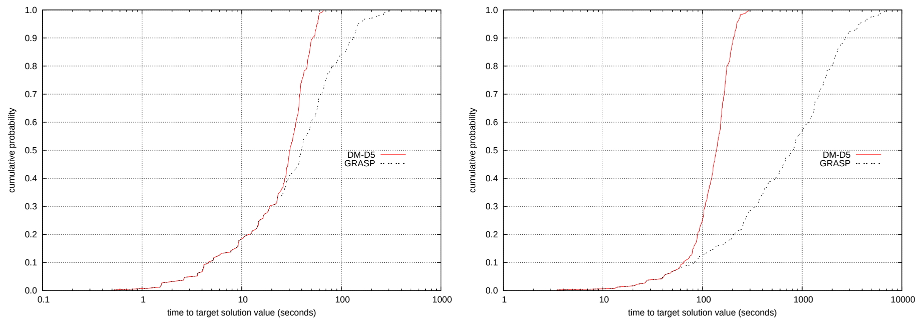


Fig. 8. Run time distribution and quantile-quantile plot for DM-D5 algorithm on the instance with $m = 50$ and target value set at 2,299.07.



(a) $m = 25$ with target 2,818.925 (b) $m = 50$ with target 2,299.07

Fig. 9. Superimposed run time distributions of DM-D5 and GRASP: (a) $Pr(X_1 \leq X_2) = 0.614775$, and (b) $Pr(X_1 \leq X_2) = 0.849163$.

4.3 GRASP algorithms for 2-path network design

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 of 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 was proved to be NP-complete by Dahl and Johannessen [5].

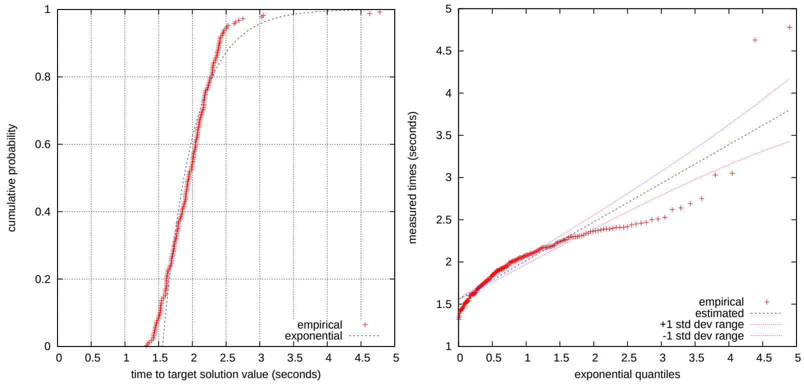


Fig. 10. Run time distribution and quantile-quantile plot for tabu search on Brazil instance with target value set at 24.

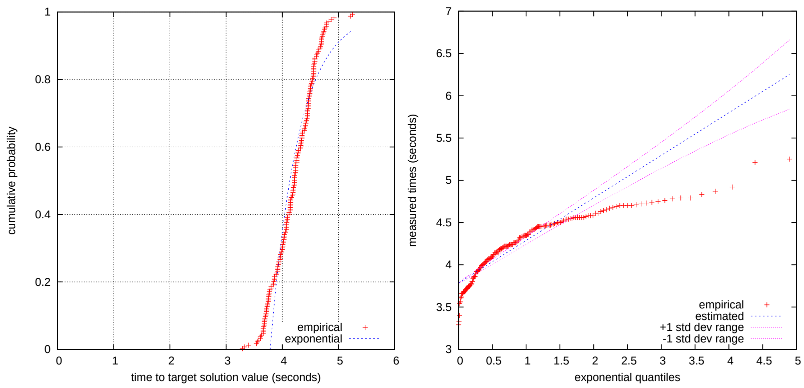
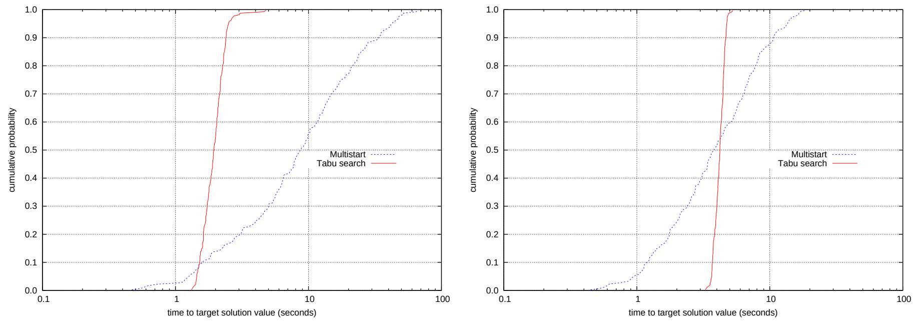


Fig. 11. Run time distribution and quantile-quantile plot for tabu search on Finland instance with target value set at 50.

We compare different heuristics [25] for approximately solving this problem. The first is a pure GRASP algorithm (algorithm A_1). The others integrate different path-relinking strategies for search intensification at the end of each GRASP iteration: forward (algorithm A_2), bidirectional (algorithm A_3), and backward (algorithm A_4) [23, 24].

Each algorithm was run 500 independent times. The experiments are summarized by the results obtained on a benchmarking instance with 90 nodes and 900 origin-destination pairs, with the target value set at 673. Run time distributions and quantile-quantile plots for the different versions of GRASP with path-relinking are illustrated in Figures 13 to 15. The run time distributions of the four algorithms are superimposed in Figure 16. Algorithm A_2 (as well as A_3 and A_4) performs much better than A_1, since $Pr(X_2 \leq X_1) = 0.984470$. Algorithm A_3 outperforms A_2, as illustrated by the fact that $Pr(X_3 \leq X_2) = 0.634002$. Fi-



(a) Brazil instance with target 24 (b) Finland instance with target 50

Fig. 12. Superimposed run time distributions of multistart and tabu search: (a) $Pr(X_1 \leq X_2) = 0.106766$, and (b) $Pr(X_1 \leq X_2) = 0.545619$.

nally, we observe that algorithms A_3 and A_4 behave very similarly, although A_4 performs slightly better for this instance with respect to the measure proposed in this work, since $Pr(X_4 \leq X_3) = 0.536016$.

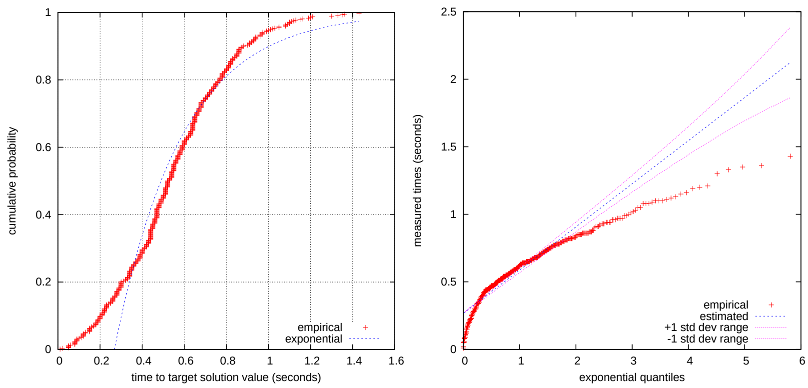


Fig. 13. Run time distribution and quantile-quantile plot for GRASP with forward path-relinking on 90-node instance with target 673.

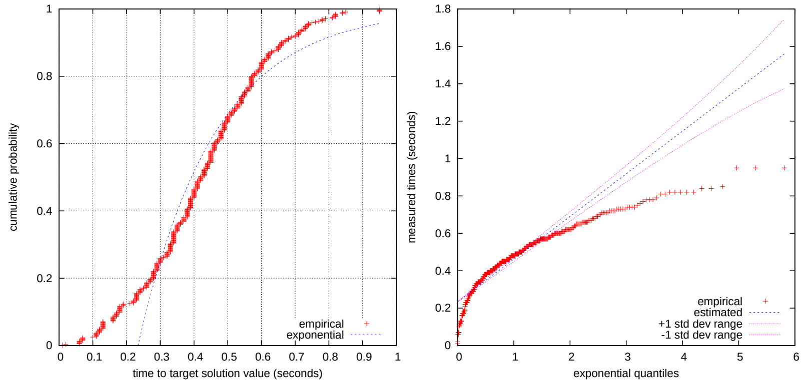


Fig. 14. Run time distribution and quantile-quantile plot for GRASP with bidirectional path-relinking on 90-node instance with target 673.

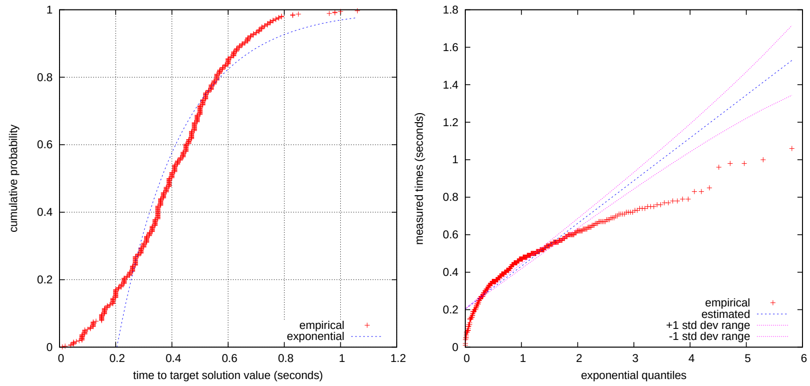


Fig. 15. Run time distribution and quantile-quantile plot for GRASP with backward path-relinking on 90-node instance with target 673.

5 Concluding remarks

Run time distributions are very useful tools to characterize the running times of stochastic algorithms for combinatorial optimization. In this work, we extended previous tools for plotting and evaluating run time distributions.

Under the assumption that running times of two stochastic local search algorithms follow exponential distributions, we derived a closed form index to compute the probability that one of them finds a target solution value in a smaller computation time than the other. A numerical iterative procedure was described for the computation of such index in the case of general run time distributions.

This new tool and the resulting probability index revealed themselves as very promising and provide a new, additional measure for comparing the performance of stochastic local search algorithms or different versions of the same algorithm.

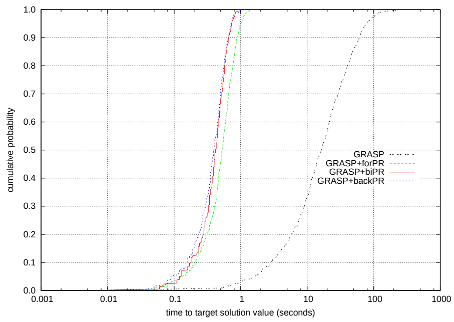


Fig. 16. Superimposed run time distributions of pure GRASP and three versions of GRASP with path-relinking.

They can also be used for setting the best parameters of a given algorithm. Numerical applications to different algorithm paradigms, problem types, and test instances illustrated the applicability of the tool.

In another context, they can also be used in the evaluation of parallel implementations of local search algorithms, providing a numerical indicator to evaluate the trade-offs between computation times and the number of processors.

References

[1] R.M. Aiex, P.M. Pardalos, M.G.C. Resende, and G. Toraldo. GRASP with path relinking for three-index assignment. *INFORMS Journal on Computing*, 17:224–247, 2005.

[2] 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.

[3] 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.

[4] S.A. Canuto, M.G.C. Resende, and C.C. Ribeiro. Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 38:50–58, 2001.

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

[6] N. Dodd. Slow annealing versus multiple fast annealing runs: An empirical investigation. *Parallel Computing*, 16:269–272, 1990.

[7] H.M.M. Ten Eikelder, M.G.A. Verhoeven, T.W.M. Vossen, and E.H.L. Aarts. A probabilistic analysis of local search. In I.H. Osman and J.P. Kelly, editors, *Metaheuristics: Theory and Applications*, pages 605–618. Kluwer, 1996.

[8] 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.

[9] E. Fonseca, R. Fuchsuber, L.F.M. Santos, A. Plastino, and S.L. Martins. Exploring the hybrid metaheuristic DM-GRASP for efficient server replication for reliable multicast. In *International Conference on Metaheuristics and Nature Inspired Computing*, Hammamet, 2008.

[10] H.H. Hoos. On the run-time behaviour of stochastic local search algorithms for SAT. In *Proc. AAAI-99*, pages 661–666. MIT Press, 1999.

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

[12] 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, 1998.

[13] H.H. Hoos and T. Stützle. Some surprising regularities in the behaviour of stochastic local search. *Lecture Notes in Computer Science*, 1520:470, 1998.

[14] H.H. Hoos and T. Stützle. Towards a characterisation of the behaviour of stochastic local search algorithms for SAT. *Artificial Intelligence*, 112:213–232, 1999.

[15] E. Hyytiä and J. Virtamo. Wavelength assignment and routing in WDM networks. In *Nordic Teletraffic Seminar 14*, pages 31–40, 1998.

[16] Y. Li, P.M. Pardalos, and M.G.C. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. In P.M. Pardalos and H. Wolkowicz, editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 237–261. American Mathematical Society, 1994.

[17] 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.

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

[19] L.J. Osborne and B.E. Gillett. A comparison of two simulated annealing algorithms applied to the directed Steiner problem on networks. *ORSA Journal on Computing*, 3:213–225, 1991.

[20] M.G.C. Resende. Computing approximate solutions of the maximum covering problem using GRASP. *Journal of Heuristics*, 4:161–171, 1998.

[21] M.G.C. Resende, L.S. Pitsoulis, and P.M. Pardalos. Fortran subroutines for computing approximate solutions of MAX-SAT problems using GRASP. *Discrete Applied Mathematics*, 100:95–113, 2000.

[22] M.G.C. Resende and C.C. Ribeiro. A GRASP for graph planarization. *Networks*, 29:173–189, 1997.

[23] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer, 2003.

[24] 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, 2005.

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

[26] L.F. Santos, S.L. Martins, and A. Plastino. Applications of the DM-GRASP heuristic: A survey. *International Transactions in Operational Research*, 15:387–416, 2008.

[27] B. Selman, H.A. Kautz, and B. Cohen. Noise strategies for improving local search. In *Proceedings of the AAAI-94*, pages 337–343. MIT Press, 1994.

[28] E.D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.

[29] M.G.A. Verhoeven and E.H.L. Aarts. Parallel local search. *Journal of Heuristics*, 1:43–66, 1995.