GRASP WITH PATH-RELINKING FOR THE QUADRATIC ASSIGNMENT PROBLEM

CARLOS A.S. OLIVEIRA, PANOS M. PARDALOS, AND M.G.C. RESENDE

Abstract. This paper describes a GRASP with path-relinking heuristic for the quadratic assignment problem. GRASP is a multi-start procedure, where different points in the search space are probed with local search for high-quality solutions. Each iteration of GRASP consists of the construction of a randomized greedy solution, followed by local search, starting from the constructed solution. Path-relinking is an approach to integrate intensification and diversification in search. It consists in exploring trajectories that connect high-quality solutions. The trajectory is generated by introducing in the initial solution, attributes of the guiding solution. Experimental results illustrate the effectiveness of GRASP with path-relinking over pure GRASP on the quadratic assignment problem.

1. Introduction

The quadratic assignment problem (QAP) was first proposed by Koopmans and Beckman [10] in the context of the plant location problem. Given n facilities, represented by the set $F = \{f_1, \ldots, f_n\}$, and n locations represented by the set $L = \{l_1, \ldots, l_n\}$, one must determine to which location each facility must be assigned. Let $A^{n\times n} = (a_{i,j})$ be a matrix where $a_{i,j} \in \mathbb{R}^+$ represents the flow between facilities f_i and f_j. Let $B^{n\times n} = (b_{i,j})$ be a matrix where entry $b_{i,j} \in \mathbb{R}^+$ represents the distance between locations l_i and l_j. Let $p : \{1 \ldots n\} \to \{1 \ldots n\}$ be an assignment and define the cost of this assignment to be

$$c(p) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} b_{p(i),p(j)}.$$

In the QAP, we want to find a permutation vector $p \in \Pi_n$ that minimizes the assignment cost, i.e. $\min c(p)$, subject to $p \in \Pi_n$, where Π_n is the set of all permutations of $\{1, \ldots, n\}$. The QAP is well known to be strongly NP-hard [18].

GRASP, or greedy randomized adaptive search procedures [5, 6, 8, 17], have been previously applied to the QAP [12, 14, 15]. For a survey on heuristics and metaheuristics applied to the QAP, see Voß[19]. In this paper, we present a new GRASP for the QAP, which makes use of path-relinking as an intensification mechanism. In Section 2, we briefly review GRASP and path-relinking, and give a description of how both are combined to find approximate solutions to the QAP. Experimental results with benchmark instances are presented in Section 3. Finally, in Section 4 we draw some concluding remarks.

Date: March 8, 2004.
AT&T Labs Research Technical Report TD-5WVRUW.

2. GRASP and path-relinking

GRASP is a multi-start procedure, where different points in the search space are probed with local search for high-quality solutions. Each iteration of GRASP consists of the construction of a randomized greedy solution, followed by local search, starting from the constructed solution. A high-level description of GRASP for QAP, i.e. solving $\min c(p)$ for $p \in \Pi_n$, is given in Algorithm 1.

Algorithm 1 GRASP for minimization

$c^* \leftarrow \infty$
while stopping criterion not satisfied **do**
  $p \leftarrow$ `GreedyRandomized`()
  $p \leftarrow$ `LocalSearch`(p)
  if $c(p) < c^*$ **then**
    $p^* \leftarrow p$
    $c^* \leftarrow c(p)$
  end if
end while
return p^*

The greedy randomized construction and the local search used in the new algorithm are similar to the ones described in [12]. The construction phase consists of two stages.

In stage 1, two initial assignments are made: facility F_i is assigned to location L_k and facility F_j is assigned to location L_l. To make the assignment, elements of the distance matrix are sorted in increasing order:

$$b_{i(1),j(1)} \leq b_{i(2),j(2)} \leq \cdots \leq b_{i(n),j(n)},$$

while the elements of the flow matrix are sorted in increasing order:

$$a_{k(1),l(1)} \geq a_{k(2),l(2)} \geq \cdots \geq a_{k(n),l(n)}.$$

The product elements

$$a_{k(1),l(1)} \cdot b_{i(1),j(1)}, a_{k(2),l(2)} \cdot b_{i(2),j(2)}, ..., a_{k(n),l(n)} \cdot b_{i(n),j(n)}$$

are sorted and the term $a_{k(q),l(q)} \cdot b_{i(q),j(q)}$ is selected at random from among the smallest elements. This product corresponds to the initial assignments: facility $F_{k(q)}$ is assigned to location $L_{i(q)}$ and facility $F_{l(q)}$ is assigned to location $L_{j(q)}$.

In stage 2, the remaining $n-2$ assignments of facilities to locations are made, one facility/location pair at a time. Let $\Omega = \{(i_1, k_1), (i_2, k_2), \ldots, (i_q, k_q)\}$ denote the first q assignments made. Then, the cost assigning facility F_j to location L_l is $c_{j,l} = \sum_{i,k \in \Omega} a_{i,j} b_{k,l}$. To make the $q+1$-th assignment, select at random an assignment from among the feasible assignments with smallest costs and add the assignment to Ω.

Once a solution is constructed, local search is applied to it to try to improve its cost. For each pair of assignments $(F_i \to L_k; F_j \to L_l)$ in the current solution, check if the swap $(F_i \to L_l; F_j \to L_k)$ improves the cost of the assignment. If so, make the swap, and repeat. A solution is locally optimal, when no swap improves the cost of the solution.

Path-relinking [9] is an approach to integrate intensification and diversification in search. It consists in exploring trajectories that connect high-quality solutions. The trajectory is generated by introducing in the initial solution, attributes of the guiding solution. It was first used in connection with GRASP by Laguna and Martí [11]. A recent survey of GRASP with path-relinking is given in Resende and Ribeiro [16]. The objective of path-relinking is to integrate features of good solutions, found during the iterations of GRASP, into new solutions generated in subsequent iterations. In pure GRASP (i.e. GRASP without path-relinking), all iterations are independent and therefore most good solutions are simply "forgotten." Path-relinking tries to change this, by retaining previous solutions and using them as "guides" to speed up convergence to a good-quality solution.

Path-relinking uses an *elite set* P, in which good solutions found by the GRASP are saved to be later combined with other solutions produced by the GRASP. The maximum size of the elite set is an input parameter. During path-relinking, one of the solutions $q \in P$ is selected to be combined with the current GRASP solution p. The elements of q are incrementally incorporated into p. This relinking process can result in an improved solution, since it explores distinct neighborhoods of high-quality solutions.

Algorithm 2 shows the steps of GRASP with path-relinking. Initially, the elite set P is empty, and solutions are added if they are different from the solutions already in the set. Once the elite set is full, path-relinking is done after each GRASP construction and local search.

Algorithm 2 GRASP with path-relinking

$P \leftarrow \emptyset$
while stopping criterion not satisfied **do**
    $p \leftarrow$ `GreedyRandomized`$()$
    $p \leftarrow$ `LocalSearch`(p)
    if P is full **then**
        Select elite solution $q \in P$ at random
        $r \leftarrow$ `PathRelinking`(p, q)
        if $c(r) \le \max\{c(q) \mid q \in P\}$ and $r \notin P$ **then**
            Let $P' = \{q \in P \mid c(q) \ge c(r)\}$
            Let $q' \in P'$ be the most similar solution to r
            $P \leftarrow P \cup \{r\}$
            $P \leftarrow P \setminus \{q'\}$
        end if
    else
        if $p \notin P$ **then**
            $P \leftarrow P \cup \{p\}$
        end if
    end if
end while
return $p^* = \min\{c(p) \mid p \in P\}$

A solution $q \in P$ is selected, at random, to be combined, through path-relinking, with the GRASP solution p. Since we want to favor long paths, which have a better change of producing good solutions, we would like to choose an elite solution q with a high degree of differentiation with respect to p. Each element $q \in P$, let $d(q)$ denote the number of facilities in q and p that have different assignments, and let $D = \sum_{q \in P} d(q)$. A solution q is selected from the elite set with probability $d(q)/D$. The selected solution q is called the *guiding* solution. The output of path-relinking, r, is at least as good as solutions p and q, that were combined by path-relinking.

If the combined solution r is not already in the elite set and its cost is not greater than cost of the highest-cost elite set solution, then it is inserted into the elite set. Among the elite set solutions having cost not smaller than $c(r)$, the one most similar to r is deleted from the set. This scheme keeps the size of the elite set constant and attempts to maintain the set diversified.

Algorithm 3 Path-relinking

Require: p, the current GRASP solution; q, the guiding solution

$c^* \leftarrow \infty$
for $i \leftarrow 1, \ldots, n$ **do**
    if $p(i) \neq q(i)$ **then**
        Let j be such that $p(j) = q(i)$
        $\delta \leftarrow$ `evalij`(p, i, j)
        $\tau \leftarrow p(i)$
        $p(i) \leftarrow p(j)$
        $p(j) \leftarrow \tau$
        if $\delta > 0$ **then**
            $r \leftarrow$ `LocalSearch`(p)
            if $c(r) < c^*$ **then**
                $r^* \leftarrow r$
            end if
        end if
    end if
end for
return r^*

We next give details on our implementation of path-relinking for the QAP, shown in Algorithm 3. Let p be the mapping implied by the current solution and q the mapping implied by the guiding solution. For each location $i = 1, \ldots, n$, path-relinking attempts to exchange facility $p(i)$ assigned to location i in the current solution with facility $q(i)$ assigned to i in the guiding solution. To maintain the mapping p feasible, it exchanges $p(i)$ with $p(k)$, where $p(k) = q(i)$.

The change in objective function caused by this swap is found using the function `evalij`, which is limited to the part of the objective function affected by these elements. If the change is positive, then the algorithm applies local search to the resulting solution. This is done only for positive changes in the objective value function to reduce the total computational time spent in local search. The algorithm also checks if the generated solution is better than the best known solution and, if so, saves it.

The path-relinking procedure described above can be further generalized, by observing that path-relinking can also be done in the reverse direction, from the solution in the elite set to the current solution. This modification of the path-relinking procedure is called *reverse path-relinking*. In our implementation, a reverse

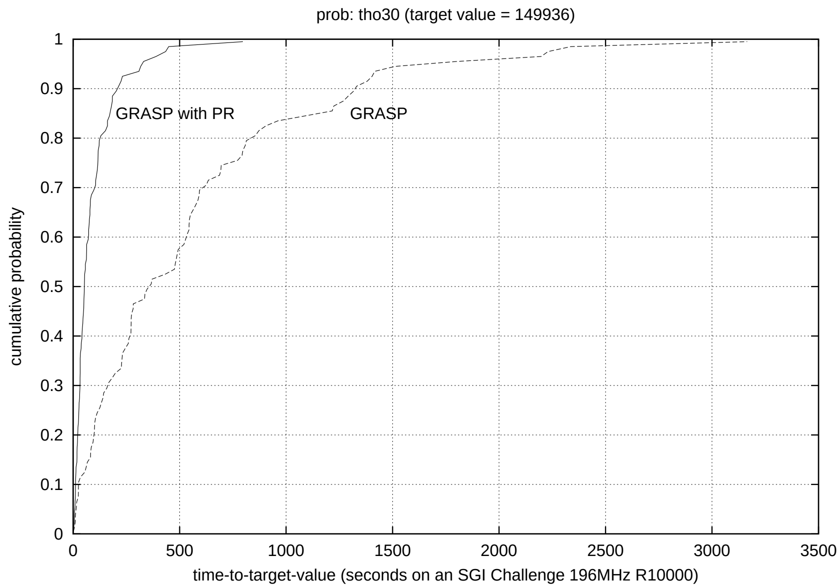

FIGURE 1. Probability distribution of time-to-target-value on instance `tho30` from QAPLIB for GRASP and GRASP with path-relinking.

path-relinking is also applied at each iteration. As a last step, we use a post-optimization procedure where path-relinking is applied among all solutions of the elite set. This procedure, which can be viewed as an extended local search, is repeated while an improvement in the best solution is possible.

One of the computational burdens associated with path-relinking is the local search done on all new solutions found during path-relinking. To ameliorate this, we modified the local search phase proposed in GRASP [12] by using a non-exhaustive improvement phase. In the local search in [12], each pair of assignments was exchanged until the best one was found. In our implementation, only one of the assignments is verified and exchanged with the one that brings the best improvement. This reduces the complexity of local search by a factor of n, leading to a $O(n^2)$ procedure. This scheme is used after the greedy randomized construction and at each iteration during path-relinking.

To enhance the quality of local search outside path-relinking, after the modified local search discussed above is done, the algorithm performs a random 3-exchange step, equivalent to changing, at random, two pair of elements in the solution. The algorithm then continues with the local search, until a local optimum is found. This type of random shaking is similar to what is done in variable neighborhood search [13].

3. Computational Experiments

Before we present the results, we first describe a plot used in several of our papers to experimentally compare different randomized algorithms or different versions of the same randomized algorithm [1, 3, 7]. This plot shows empirical distributions of the random variable *time to target solution value*. To plot the empirical distribution,

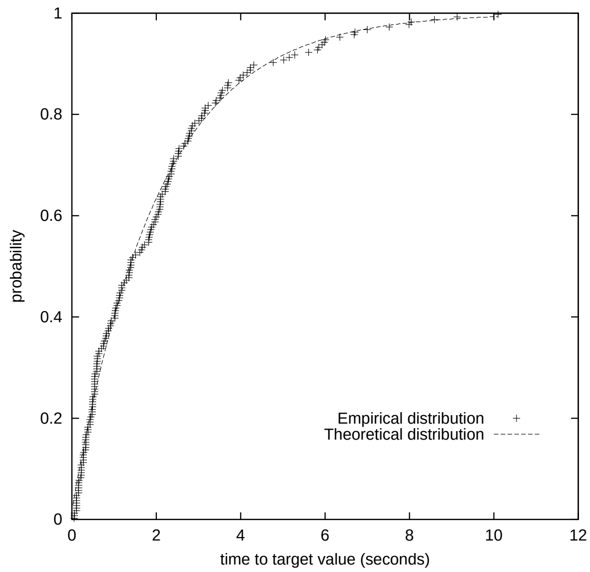

FIGURE 2. Superimposed empirical and theoretical distributions (times to target values measured in seconds on an SGI Challenge computer with 196 MHz R10000 processors).

we fix a solution target value and run each algorithm T independent times, recording the running time when a solution with cost at least as good as the target value is found. For each algorithm, we associate with the i-th sorted running time (t_i) a probability $p_i = (i - \frac{1}{2})/T$, and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, T$. Figure 1 shows one such plot comparing the pure GRASP with the GRASP with path-relinking for QAPLIB instance `tho30` with target (optimal) solution value of 149936. The figure shows clearly that GRASP with path-relinking (GRASP+PR) is much faster than pure GRASP to find a solution with cost 149936. For instance, the probability of finding such a solution in less than 100 seconds is about 55% with GRASP with path-relinking, while it is about 10% with pure GRASP. Similarly, with probability 50% GRASP with path-relinking finds such a target solution in less than 76 seconds, while for pure GRASP, with probability 50% a solution is found in less than 416 seconds.

In [3], Aiex, Resende, and Ribeiro showed experimentally that the distribution of the random variable *time to target solution value* for a GRASP is a shifted exponential. The same result holds for GRASP with path-relinking [2]. Figure 2 illustrates this result, depicting the superimposed empirical and theoretical distributions observed for one of the cases studied in [3].

In this paper, we present extensive experimental results, showing that path-relinking substantially improves the performance of GRASP. We compare an implementation of GRASP with and without path-relinking. The instances are taken from QAPLIB [4], a library of quadratic assignment test problems.

For each instance considered in our experiments, we make $T = 100$ independent runs with GRASP with and without path-relinking, recording the time taken for each algorithm to find the best known solution for each instance. (Due to the

Table 1. Summary of experiments. For each instance, the table lists for each algorithm, the number of independent runs, and the time (in seconds) for 25%, 50%, and 75% of the runs to find the target solution value.

	GRASP				GRASP with PR			
problem	runs	25%	50%	75%	runs	25%	50%	75%
esc32h	100	.5	1.4	2.5	100	.2	.5	1.0
bur26h	100	2.5	1.4	2.5	100	.7	1.4	2.8
kra30a	100	47	115	241	100	11	26	57
tho30	100	208	410	944	100	30	76	154
nug30	100	583	1334	2841	100	63	149	283
chr22a	100	723	1948	4188	100	234	449	726
lipa40a	75	12,366	23,841	39,649	100	360	526	708
ste36a	17	27,034	91,075	135,011	100	1787	4047	8503

length of the runs on a few of the instances, fewer than 100 runs were done.) The probability distributions of time-to-target-value for each algorithm are plotted for each instance considered. We consider 91 instances from QAPLIB. Since it is impractical to fit 91 plots in this paper, we show the entire collection of plots at the URL `http://www.research.att.com/~mgcr/exp/gqapspr`. In this paper, we show only a representative set of plots.

Table 3 summarizes the runs in the representative set. The numbers appearing in the names of the instances indicate the dimension (n) of the problem. For each instance, the table lists for each algorithm the number of runs, and the times in seconds for 25%, 50%, and 75% of the runs to find a solution having the target value.

The distributions are depicted in Figures 1 and 3 to 9.

The table and figures illustrate the effect of path-relinking on GRASP. On all instances, path-relinking improved the performance of GRASP. The improvement went from about a factor of two speedup to over a factor of 60.

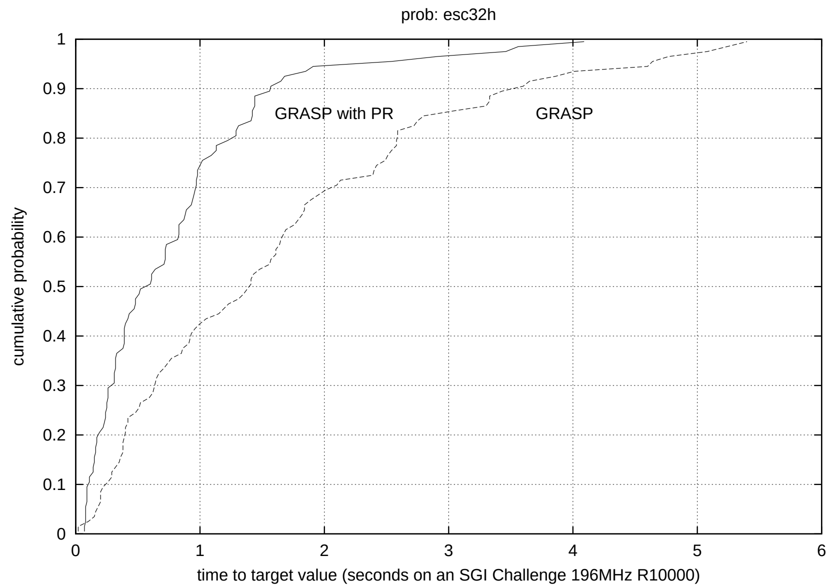

FIGURE 3. Probability distribution of time-to-target-value on instance `esc32h` from QAPLIB for GRASP and GRASP with path-relinking.

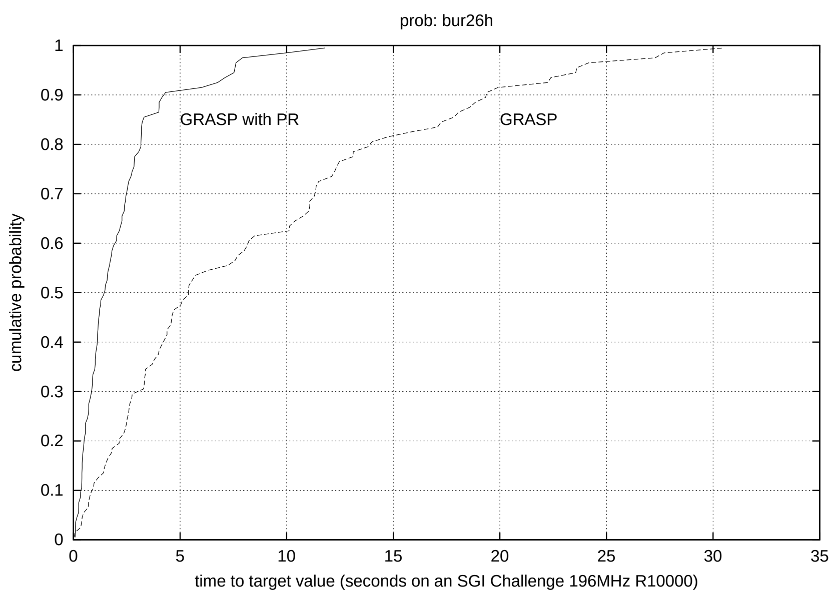

FIGURE 4. Probability distribution of time-to-target-value on instance `bur26h` from QAPLIB for GRASP and GRASP with path-relinking.

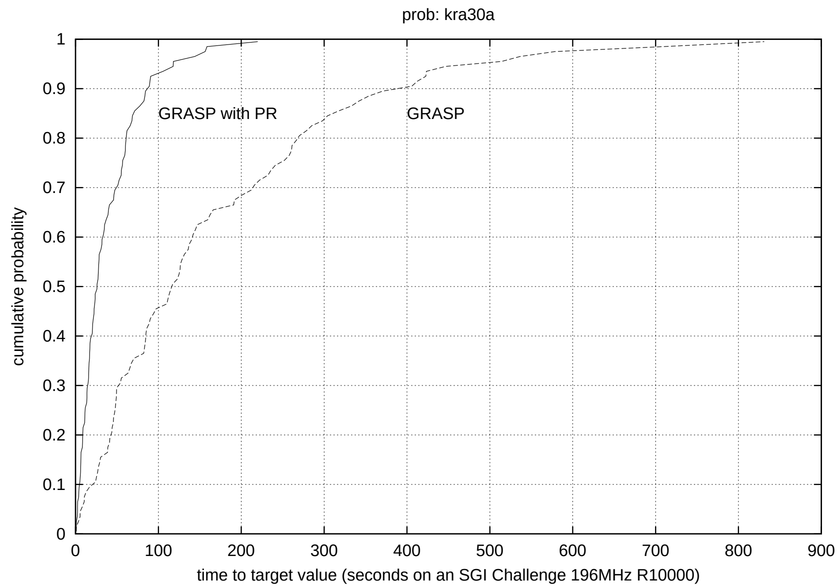

FIGURE 5. Probability distribution of time-to-target-value on instance `kra30a` from QAPLIB for GRASP and GRASP with path-relinking.

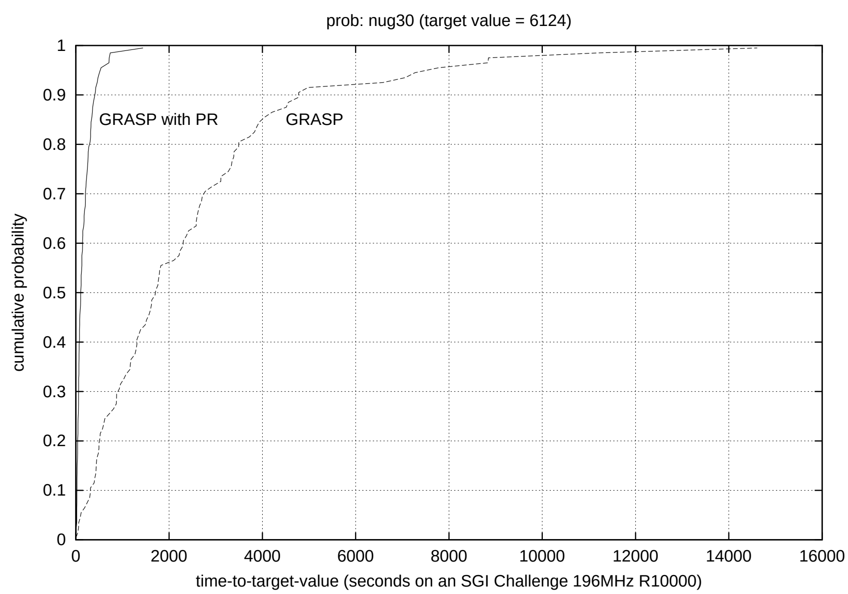

FIGURE 6. Probability distribution of time-to-target-value on instance `nug30`, from QAPLIB for GRASP and GRASP with path-relinking.

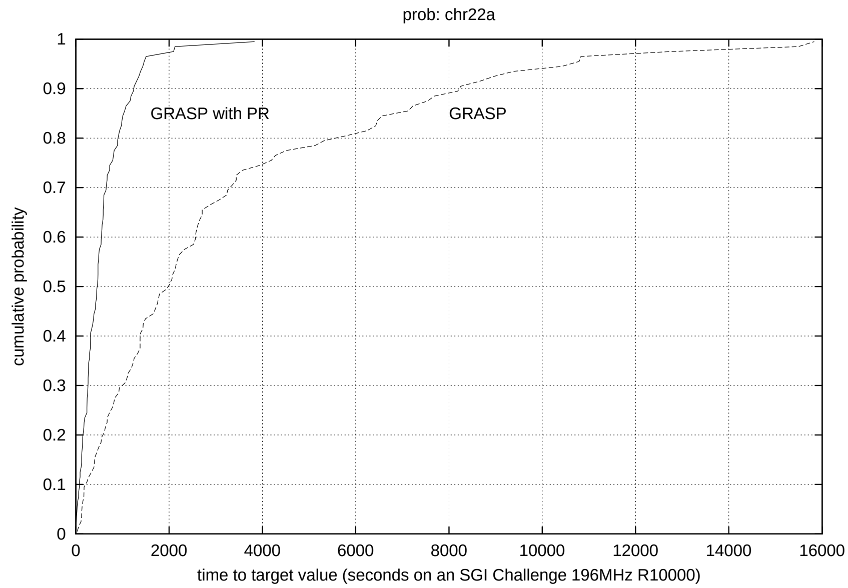

FIGURE 7. Probability distribution of time-to-target-value on instance `chr22a` from QAPLIB for GRASP and GRASP with path-relinking.

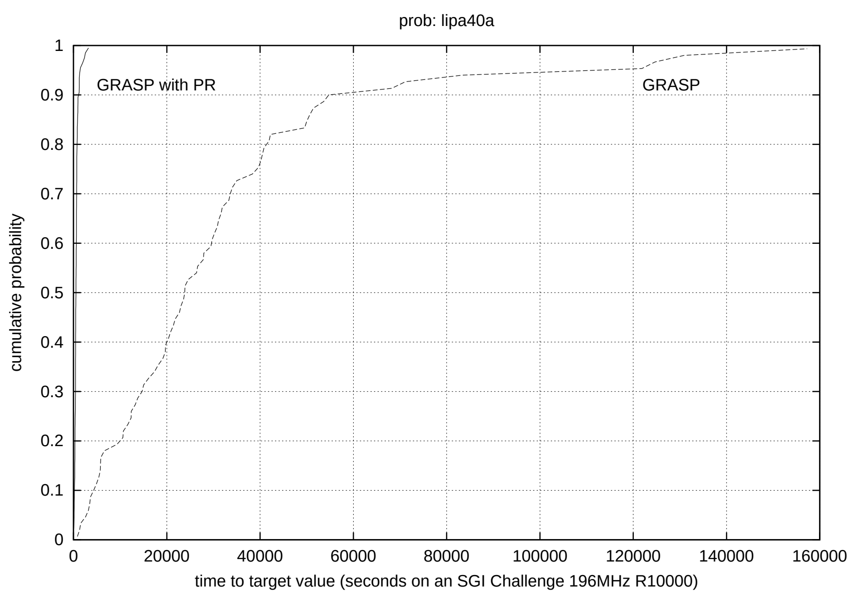

FIGURE 8. Probability distribution of time-to-target-value on instance `lipa40a` from QAPLIB for GRASP and GRASP with path-relinking.

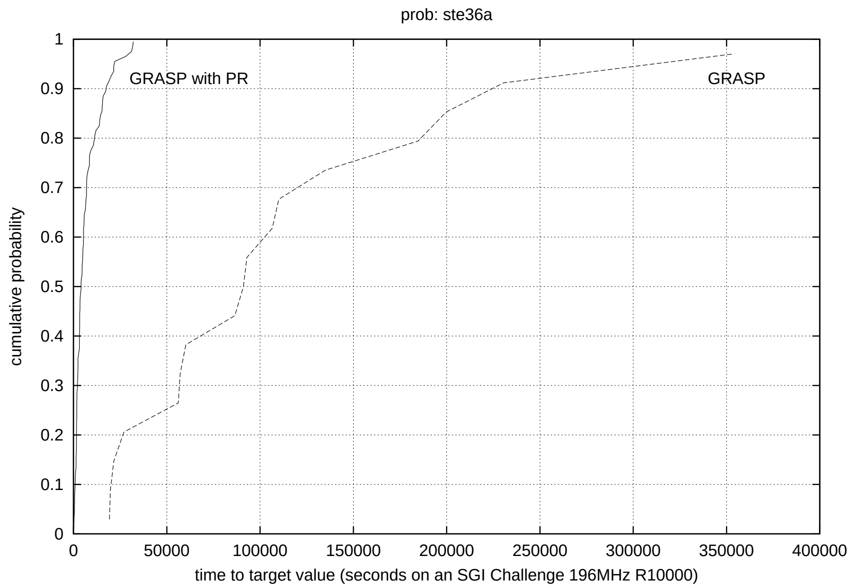

FIGURE 9. Probability distribution of time-to-target-value on instance `ste36a` from QAPLIB for GRASP and GRASP with path-relinking.

4. Concluding Remarks

In this paper, we propose a GRASP with path-relinking for the quadratic assignment problem. The algorithm was implemented in the ANSI-C language and was extensively tested. Computational results show that path-relinking speeds up convergence, sometimes by up to two orders of magnitude. The source code for both GRASP and GRASP with path-relinking, as well as the plots for the extended experiment, can be downloaded from the URL `http://www.research.att.com/~mgcr/exp/gqapspr`.

References

[1] R.M. Aiex. *Uma investigação experimental da distribuição de probabilidade de tempo de solução em heurísticas GRASP e sua aplicação na análise de implementações paralelas*. PhD thesis, Department of Computer Science, Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil, 2002.

[2] R.M. Aiex, S. Binato, and M.G.C. Resende. Parallel GRASP with path-relinking for job shop scheduling. *Parallel Computing*, 29:393–430, 2003.

[3] 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.

[4] R. Burkhard, S. Karisch, and F. Rendl. QAPLIB – A quadratic assignment problem library. *Eur. J. Oper. Res.*, 55:115–119, 1991. Online version on `http://www.opt.math.tu-graz.ac.at/qaplib`.

[5] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[6] T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

[7] P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heuristics for the max-cut problem. *Optimization Methods and Software*, 7:1033–1058, 2002.

[8] P. Festa and M. G. C. Resende. GRASP: An annotated bibliography. In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys on Metaheuristics*, pages 325–368. Kluwer Academic Publishers, 2002.

[9] F. Glover. Tabu search and adaptive memory programing – Advances, applications and challenges. In R.S. Barr, R.V. Helgason, and J.L. Kennington, editors, *Interfaces in Computer Science and Operations Research*, pages 1–75. Kluwer, 1996.

[10] T. C. Koopmans and M. J. Berkmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[11] M. Laguna and R. Martí. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11:44–52, 1999.

[12] 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.

[13] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

[14] P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. Algorithm 769: FORTRAN subroutines for approximate solution of sparse quadratic assignment problems using GRASP. *ACM Trans. Math. Software*, 23(2):196–208, 1997.

[15] M.G.C. Resende, P.M. Pardalos, and Y. Li. Algorithm 754: FORTRAN subroutines for approximate solution of dense quadratic assignment problems using GRASP. *ACM Transactions on Mathematical Software*, 22:104–118, March 1996.

[16] M.G.C. Resende and C.C. Ribeiro. GRASP and path-relinking: Recent advances and applications. Technical Report TD-5TU726, AT&T Labs Research, 2003.

[17] 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 Academic Publishers, 2003.

[18] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the Association of Computing Machinery*, 23:555–565, 1976.

[19] S. Voß. Heuristics for nonlinear assignment problems. In P.M. Pardalos and L. Pitsoulis, editors, *Nonlinear Assignment Problems*, pages 175–215. Kluwer, Boston, 2000.

(C.A.S. Oliveira) Dept. of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, Gainesville, FL 32611, USA

E-mail address, C.A.S. Oliveira: oliveira@ufl.edu

(P.M. Pardalos) Dept. of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, Gainesville, FL 32611, USA

E-mail address, P.M. Pardalos: pardalos@ufl.edu

(M.G.C. Resende) Internet and Network Systems Research Center, AT&T Labs Research, 180 Park Avenue, Room C241, Florham Park, NJ 07932 USA.

E-mail address, M.G.C. Resende: mgcr@research.att.com