A METHODOLOGY FOR THE ANALYSIS OF PARALLEL GRASP STRATEGIES

RENATA M. AIEX AND MAURICIO G.C. RESENDE

ABSTRACT. In this paper, we describe a methodology for the analysis of greedy randomized adaptive search procedures (GRASP). GRASP is a metaheuristic for combinatorial optimization. It usually consists of a construction procedure based on a greedy randomized algorithm and a local search. Hybrid approaches of GRASP with path-relinking developed for the 3-index assignment problem (AP3) and for the job-shop scheduling problem (JSP) are analyzed with the methodology proposed. Independent and cooperative parallelization strategies are described and implemented for the AP3 and for the JSP. The computational results for independent parallel strategies are compared to the results obtained using the methodology. The experiments confirm the fact that the methodology proposed is a useful tool to predict qualitatively the behavior of GRASP when implemented using multiple independent processes.

1. INTRODUCTION

GRASP [6] is an iterative metaheuristic, where each GRASP iteration usually consists of two phases: construction and local search. In the construction phase, a feasible solution is built, one element at a time, using a greedy randomized algorithm. Since the solutions generated by a GRASP construction phase are not guaranteed to be locally optimal, local search is applied to attempt to improve each constructed solution. The best solution over all GRASP iterations is returned as the result.

Hybrid strategies of GRASP with path-relinking have been leading to significant improvements in solution quality when compared to the solutions obtained by the pure method. The use of path-relinking within a GRASP procedure, as an intensification strategy applied to each locally optimal solution, was first proposed by Laguna and Martí [5]. The path-relinking approach consists in exploring trajectories that connect a *initial solution* to a *guiding solution*. This is done by introducing attributes of the guiding solution in the initial solution. At each step, all moves that incorporate attributes of the guiding solution are analyzed and the best move is chosen. In general, one of these solutions is obtained by GRASP local search and the other one is chosen among the solutions kept in an elite set.

In this paper, a methodology for the analysis of GRASP is proposed. The hybrid GRASPs developed for the 3-index assignment problem (AP3) in [2] and for the job-shop scheduling problem (JSP) in [1] are analyzed according to this methodology. The methodology for GRASP analysis uses the sequential version of the algorithm to predict qualitatively how solution quality will vary in a parallel independent approach. The GRASPs for the AP3 and for the JSP are parallelized and the computational results are compared to the results obtained using the methodology proposed.

The remainder of this paper is organized as follows. Section 2 shows how GRASP and path-relinking are combined in the sequential and parallel approaches for the AP3

Date: April 7, 2003.
AT&T Labs Research Technical Report.

and the JSP. The methodology for the analysis of GRASP is described in Section 3. The computational results are reported and analyzed in Section 4. Concluding remarks are made in Section 5.

2. GRASP WITH PATH-RELINKING

We describe how we combined path-relinking and GRASP to form a hybrid GRASP with path-relinking. Let $|P|$ and *maxpool* be the current and the maximum size of the elite set, respectively. The first *maxpool* GRASP iterations contribute one solution to the elite set per GRASP iteration. Path-relinking is not carried out until the pool of elite solutions is full, i.e., $|P| = maxpool$.

Once the pool of elite solutions is full, solution S produced by the local search phase of GRASP is tested to verify its quality. If S passes the quality test used, bidirectional path-relinking is done between S and all elements of a subset $P' \subseteq P$. The degree of restrictiveness of the quality test is proportional to the computational time necessary to do path-relinking. For the AP3, the cost of a solution in the neighborhood of S can be computed from the cost of S in $O(1)$ time [2] and therefore, path-relinking is applied to all solutions obtained by the local search. For the JSP, the cost of each solution visited by path-relinking is computed in $O(|\mathcal{J}| \cdot |\mathcal{M}|)$, where $|\mathcal{J}|$ is the number of jobs and $|\mathcal{M}|$ is the number of machines in the instance [1]. Thus, for the JSP, path-relinking is applied only when GRASP solutions satisfy a given quality criterion. After each path-relinking phase, the best solution obtained is tested for inclusion in the elite pool. A solution S_{gmin} produced by path-relinking will be inserted in the pool if S_{gmin} is better than the best solution in P or if it is not worse than the worst solution in P and it differs significantly from all elite solutions.

The GRASP with path-relinking iterations continues for at most *maxitr* iterations, but can be terminated when a solution having a cost at least as good as *look*4 is found. The value of *look*4 is given as input to GRASP.

Our parallel GRASP implementations use message passing for communication between processors. The first scheme, named *independent* approach, revisits a basic parallelization strategy for GRASP with path-relinking [2]. This scheme limits communication between processors only for problem input, detection of process termination, and determination of best overall solution. When ρ processors are used, a single process reads the problem data and passes it to the remaining $\rho - 1$ processes. Each process executes a copy of the GRASP program. Processes send a message to all others when they either stop upon finding a solution at least as good as the target value *look*4 or complete the maximum number of allotted iterations. The second scheme is called *cooperative* approach. This scheme enables processes to share elite set information in addition to the communication allowed in the independent approach.

3. METHODOLOGY FOR THE ANALYSIS OF GRASP

The methodology for analysis of GRASP consists in studying the probability distribution of the random variable *time to target value* of the algorithm. The parameters estimated for the distribution of time to target value of the sequential GRASP are used to predict the quality of the solutions when the algorithm is implemented in parallel.

Aiex, Resende, and Ribeiro studied the empirical probability distributions of the random variable time to target value in [3]. They showed that, given a target solution value, the time it takes GRASP to find a solution with cost at least as good as the target fits a two-parameter exponential distribution. A standard technique for graphical analysis [4] was used to compute the empirical and theoretical distributions and to estimate the parameters of the distributions. The same technique was used to study time to target value for the GRASPs proposed for the AP3 and the JSP in [2] and [1], respectively. It is shown that these variants of GRASP also have time to target value distributions that fit a two-parameter exponential distribution.

The distributions are computed with 200 runs of the GRASP variant. For each of the 200 runs of a variant, the random number generator is initialized with a distinct seed and therefore the runs are independent. Following the technique used in [3], we draw the theoretical quantile-quantile plot for the data to estimate the parameters of the two-parameter exponential distribution. The cumulative distribution function for the two-parameter exponential distribution is given by $F(t) = 1 - e^{-(t-\mu)/\lambda}$, where λ is the mean of the distribution data (and indicates the spread of the data) and μ is the shift of the distribution with respect to the ordinate axis.

The following can be stated for a two parameter (shifted) exponential distribution [3]. Let $P_\rho(t)$ be the probability of not having found a given (target) solution in t time units with ρ independent processes. If $P_1(t) = e^{-(t-\mu)/\lambda}$ with $\lambda \in \mathbb{R}^+$ and $\mu \in \mathbb{R}$, i.e. P_1 corresponds to a two parameter exponential distribution, then $P_\rho(t) = e^{-\rho(t-\mu)/\lambda}$. This follows from the definition of the two-parameter exponential distribution. It implies that the probability of finding a solution of a given value in time ρt with a sequential process is equal to $1 - e^{-(\rho t-\mu)/\lambda}$, while the probability of finding a solution at least as good as that given value in time t with ρ independent parallel processes is $1 - e^{-\rho(t-\mu)/\lambda}$. Note that if $\mu = 0$, then both probabilities are equal and correspond to the non-shifted exponential distribution. Furthermore, if $\rho|\mu| \ll \lambda$, then the two probabilities are approximately equal and it is possible to approximately achieve linear speed-up in solution time to target value using multiple independent processes.

In the context of GRASP, the parameter μ is an estimate of the minimum amount of time necessary for the algorithm to find the target value for the pair of instance/target value considered. The parameter λ is an estimate of the spread of the measured times for the tested pair of instance/target value.

4. Computational results

The experiments were done on an SGI Challenge computer (16 196-MHz MIPS R10000 processors and 12 194-MHz MIPS R10000 processors) with 7.6 Gb of memory. The algorithms were coded in Fortran and were compiled with the SGI MIPSpro F77 compiler using flags `-O3 -static -u`. The parallel codes used SGI's Message Passing Toolkit 1.4, which contains a fully compliant implementation of the MPI 1.2 specification.

Times measured were wall clock times. The average speedups were computed dividing the sum of the execution times of the independent parallel program executing in 1 processor by the sum of the execution times of the parallel program on 2, 4, 8, and 16 processors, for 60 runs.

For the AP3, we tested one problem of each size $n = 20, 22, 24$ and 26 generated by Balas and Saltzman [2]. We named these problems `B-S 20.1`, `B-S 22.1`, `B-S 24.1` and `B-S 26.1`. The problems `abz6`, `mt10`, `orb5`, and `la21` were tested for the JSP were obtained from four classes of standard problems [1].

The goal of the experiments in this section is to verify computationally the results of the proposition for the two parameter exponential distribution. Therefore, the stopping criterion for maximum number of iterations is disabled, and the programs are terminated only when a solution with cost at least as good as the target value is found

TABLE 1. Speedups for instances of the AP3. Algorithms are independent and cooperative GRASP. Instances are `B-S 20.1`, `B-S 22.1`, `B-S 24.1`, and `B-S 26.1`, with target values 7, 8, 7, and 8, respectively.

	estimated parameter			speedup independent (number of processors)				speedup cooperative (number of processors)			
prob.	μ	λ	$\lvert\mu\rvert/\lambda$	2	4	8	16	2	4	8	16
`B-S 20.1`	-26.46	1223.80	.021	1.67	3.34	6.22	10.82	1.56	3.47	7.37	14.36
`B-S 22.1`	-135.12	3085.32	.043	2.25	4.57	9.01	14.37	1.64	4.22	8.83	18.78
`B-S 24.1`	-16.76	4004.11	.0041	1.71	4.00	7.87	12.19	2.16	4.00	9.38	19.29
`B-S 26.1`	32.12	2255.55	.014	2.11	3.89	6.10	11.49	2.16	5.30	9.55	16.00
average:			**.020**	**1.935**	**3.95**	**7.3**	**12.21**	**1.88**	**4.24**	**8.78**	**17.10**

TABLE 2. Speedups for instances of the JSP. Algorithms are independent, and cooperative GRASPs. Instances are `abz6`, `mt10`, `orb5`, and `la21`, with target values 943, 938, 895, and 1100, respectively.

	estimated parameter			speedup independent (number of processors)				speedup cooperative (number of processors)			
prob.	μ	λ	$\lvert\mu\rvert/\lambda$	2	4	8	16	2	4	8	16
`abz6`	47.67	756.56	.063	2.00	3.36	6.44	10.51	2.40	4.21	11.43	23.58
`mt10`	305.27	524.23	.58	1.57	2.12	3.03	4.05	1.75	4.58	8.36	16.97
`orb5`	130.12	395.41	.32	1.95	2.97	3.99	5.36	2.10	4.91	8.89	15.76
`la21`	175.20	407.73	.42	1.64	2.25	3.14	3.72	2.23	4.47	7.54	11.41
average:			**.34**	**1.79**	**2.67**	**4.15**	**5.91**	**2.12**	**4.54**	**9.05**	**16.93**

4.1. **Parallel results for the AP3.** In Table 1 the speedups observed for the independent and cooperative GRASPs for the AP3 are summarized. The values of μ, λ and $|\mu|/\lambda$, estimated in [2] are also shown. An approximately linear speedup is observed for the independent GRASP for up to 8 processors. With 16 processors, we observe a reduction in the speedup, although we can still consider that the program scales well for up to 16 processors. In fact, as observed in Table 1, the values of $|\mu|/\lambda$ for the tested instances for the independent GRASP are low and their average is 0.02. Therefore, approximate linear speedups were expected for the independent GRASP. Furthermore, according to the proposition for the two-parameter exponential distribution, a reduction in the speedup as the number of processors ρ increases was expected, and was observed.

The proposition for the two-parameter exponential distribution does not take into account the sharing of information among the processors during program execution. Therefore, no conclusions from the distributions plotted for the sequential GRASP can be drawn for the cooperative approach. During the experiments, we observed that for $\rho > 2$ processors, the speedups of the cooperative approach were, on average, higher than the speedups for the independent approach. Therefore, for the cooperative GRASP, the computational time spent with process communication is compensated by the gain due to the sharing of information, for $\rho > 2$ processors.

Super-linear speedups are observed for the cooperative GRASP in Table 1. This is due to the sharing of information among processors that improves the quality of the solutions in each pool during program execution. In addition, in the cooperative GRASP, ρ pools are kept and shared by the processors, while in the sequential and in the independent GRASPs only one pool is used by each process. This allows for the use of a greater number of quality solutions, since the pools differ slightly.

4.2. **Parallel results for the JSP.** Table 2 summarizes the speedups for the independent and the cooperative GRASP approaches for the JSP. The values of μ, λ and $|\mu|/\lambda$, estimated in [1] are also shown. Sub-linear speedups are observed for the independent GRASP. We notice that the ratios $|\mu|/\lambda$, computed with the parameters in Table 2, are much higher than the values of $|\mu|/\lambda$, computed for the pairs of instance/target values used to study the AP3. For the AP3, the cost of a solution in the neighborhood of S can be computed more efficiently than for the JSP. Thus, GRASP with path-relinking for the JSP iterations need higher CPU times, which corresponds to higher values of μ. Path-relinking also speedups GRASP, reducing the spread of the solution time, i.e., the parameter λ. Therefore, μ values are higher and λ values are lower for GRASP with path-relinking for the JSP with respect to the parameters of a pure GRASP. For these reasons, the distributions estimated for the GRASP for the JSP cannot be approximated by a simple exponential distribution.

Table 2 shows linear and super-linear speedups for most of the tested instances for the cooperative approach. These speedups are considerably higher than the speedups observed for the independent parallel approach. Recall that for the hybrid approach of GRASP, path-relinking is not carried out at each iteration. Only good-quality solutions produced by local search are relinked with elite solutions and, eventually, will change the information in the pool. Thus, the solutions received in the messages exchanged in the cooperative GRASP are essential to guarantee the renewal and the improvement of the solutions in the pool of each process.

5. Conclusion

In this paper, the GRASP strategies proposed for the AP3 and for the JSP in [2] and [1], respectively, were parallelized and analyzed. We observed that the average speedups for the cooperative parallel approach were higher than the average speedups for the independent parallel approach in most of the cases.

The estimated parameters λ and μ of the two parameter exponential distribution for the tested pairs of instance/target values were used to explain the speedups achieved by the independent GRASP approaches. We observed that, according to [3], GRASPs with low values of $|\mu|/\lambda$ when parallelized achieve an approximately linear average speedup. One should keep in mind that the values of $|\mu|/\lambda$ cannot be used to study quantitatively the speedups achieved by GRASP, considering that these parameters are estimates calculated from measured real data. Alternatively, the analysis of the two parameter exponential distribution of a sequential GRASP can be considered as a useful tool to predict qualitatively the behavior of the corresponding GRASP when implemented using multiple independent processes.

References

[1] 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.

[2] R.M. Aiex, M.G.C. Resende, P.M. Pardalos, and G. Toraldo. GRASP with path relinking for the three-index assignment problem. Technical report, Information Sciences Research Center, AT&T Labs Research, Florham Park, NJ 07932 USA, 2000. To appear in *INFORMS Journal on Computing*.

[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] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Chapman & Hall, 1983.

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

[6] 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.

(R.M. Aiex) DEPARTMENT OF COPUTER SCIENCE, CATHOLIC UNIVERSITY OF RIO DE JANEIRO, RIO DE JANEIRO, RJ 22451, BRAZIL
E-mail address, R.M. Aiex: `rma@inf.puc-rio.br`

(M.G.C. Resende) INTERNET AND NETWORK SYSTEMS RESEARCH, 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`