GRASP with Path-Relinking for the Maximum Contact Map Overlap Problem

R.M.A. Silva[1], M.G.C. Resende[2], P. Festa[3], F.L. Valentim[4] and F.N. Junior[1]

[1] Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, Brazil
Emails: rmas@cin.ufpe.br, fnj@cin.ufpe.br
[2] AT&T Labs Research, Florham Park, NJ, USA – Email: mgcr@research.att.com
[3] Department of Mathematics and Applications "R. Caccioppoli"
University of Napoli FEDERICO II, Italy – Email: paola.festa@unina.it
[4] Biology Department, Federal University of Lavras, MG, Brazil
Email: felipe.flv@gmail.com

Abstract. This paper proposes a hybrid Greedy Randomized Adaptive Search Procedure with path-relinking for the maximum contact map overlap problem, an NP-hard combinatorial optimization problem that arises in computational biology. Preliminary experimental results illustrate the effectiveness and efficiency of the algorithm.

Key words: Maximum contact map overlap, GRASP, Path-relinking

1 Introduction

Knowledge about the function of a given protein can be attained by verifying any similarities between that protein and other proteins whose functions are already known. One promising way of accomplishing this task is to evaluate the alignment of their contact maps. A *contact map* consists of either a graph or a two-dimensional matrix (binary or real). In the graph representation, the contact map is a graph with a sequence of nodes corresponding to the sequence of residues and an edge for each pair of non-consecutive residues whose distance is below a given threshold. Given two contact maps $G_A = (V_A, E_A)$ and $G_B = (V_B, E_B)$ such that $|V_A| = n$ and $|V_B| = m$, the Maximum Contact Map Overlap Problem (MAX-CMO) [9] is an NP-hard problem consisting in finding two subsets $S_A \subseteq V_A$ and $S_B \subseteq V_B$ with $|S_A| = |S_B|$ and an order preserving bijection f between S_A and S_B such that the cardinality of the *overlap set* $\mathcal{L}(S_A, S_B, f) = \{(u, v) \in E_A \ : \ u, v \in S_A, \ (f(u), f(v)) \in E_B\}$ is maximized. A solution (S_A, S_B, f) of the contact map overlap problem can be represented as an assignment vector p of size n such that $p_u = v$ if $(u, v) \in \mathcal{L}(S_A, S_B, f)$; or `nil`, otherwise. The MAX-CMO was introduced in 1992 [9]. Since then, several heuristic and exact algorithms have been proposed ([2, 10]). In a recent paper appeared in 2011 [1], Andronov et al. proposed a Branch & Bound approach that is based on a novel and more performing Lagrangian relaxation, but that can be used only to solve small sized instances of the problem.

2 GRASP with Path-relinking for the MAX-CMO

A GRASP heuristic [3–5, 7] is a multi-start procedure where at each iteration a greedy randomized solution is constructed to be used as a starting solution for local search. The best local optimum found over all GRASP iterations is output as the solution. In GRASP with path-relinking [11, 6, 8], an elite set of diverse good-quality solutions is maintained and updated. At each GRASP iteration, the current local optimal solution is combined with a randomly selected solution from the elite set using the path-relinking operator. The combined solution is a candidate for inclusion in the elite set and is added to the elite set if it meets quality and diversity criteria.

Algorithm 1 shows pseudo-code for the GRASP with path-relinking heuristic for the MAX-CMO (GRASP-PR). The algorithm takes as input two contact maps C^A and C^B of proteins A and B, with n and m residues ($m > n$), respectively. It outputs an array p^* of length n, with $p^*_i = \texttt{nil}$, if node $i \in C^A$ representing residue $i \in A$ is not aligned, and $p^*_i = j$, if node $i \in C^A$ is aligned with node $j \in C^B$. After initializing the elite set P as empty in line 1, the GRASP with path-relinking iterations are computed in lines 2 to 19 until a stopping criterion is satisfied. This criterion could be, for example, a maximum number of iterations, a target solution quality, or a maximum number of iterations without improvement. During each iteration, a greedy randomized solution p is generated in line 3 and tentatively improved in line 4 with an approximate local search. If the elite set P is empty, solution p is added to it in line 15. If P is not empty, then while it is not full, solution p is added to it in line 16 if it is sufficiently different from the solutions already in the elite set. To define the term "sufficiently different" more precisely, let $\Delta(p, q)$ denote the number of assignments in p that are different from those in q. For a given level of difference δ, we say p is sufficiently different from all elite solutions in P if $\Delta(p, q) > \delta$ for all $q \in P$, which we indicate with the notation $p \not\approx P$. If the elite set P is full, then path-relinking is applied in line 7 between p and some elite solution q randomly chosen from P in line 6, resulting in solution r. In line 8, r is updated by an approximate local minimum in its neighborhood. If r is the best solution found so far, then it replaces t, the solution most similar to it, computed in line 10. Otherwise, if r is better than the worst solution in P and $r \not\approx P$, then it replaces t, the solution most similar to it, computed in line 12.

3 Experimental results

All experiments with GRASP-PR were run on a Dell PE1950 computer with dual quad core processors and 16 Gb of memory, running Red Hat Linux nesh version 5.1.19.6 (CentOS release 5.2, kernel $2.6.18 - 53.1.21.el5$). GRASP-PR was implemented in Java and compiled into bytecode with javac version 1.6.0_05.

```
algorithm GRASP-PR (C^A, C^B)
1     P ← ∅;
2     while (stopping criterion not satisfied) →
3         p ← GreedyRandomized(·);
4         p ← ApproximateLocalSearch(p);
5         if (P is full) then
6             Randomly select a solution q ∈ P;
7             r ← PathRelinking(p, q);
8             r ← ApproximateLocalSearch(r);
9             if (c(r) > max{c(s) : s ∈ P}) then
10                t ← argmin{Δ(r, s) : s ∈ P}; P ← P ∪ {r} \ {t};
11            else if (c(r) > min{c(s) : s ∈ P} and r ≉ P) then
12                t ← argmin{Δ(r, s) : s ∈ P : c(s) < c(r)}; P ← P ∪ {r} \ {t};
13            endif
14        else
15            if (P = ∅) then P ← {p};
16            else if (p ≉ P) then P ← P ∪ {p};
17            endif
18        endif
19    endwhile
20    return(p* = argmax{c(s) : s ∈ P});
end GRASP-PR
```

Fig. 1. Pseudo-code of the GRASP-PR heuristic for the MAX-CMO.

The random-number generator is an implementation of the Mersenne Twister algorithm from the COLT[5] library.

Three pairs of proteins were randomly selected from the dataset used by Caprara and Lancia [2] as summarized in Table 1. Each heuristic was run 200

Table 1. Test instances: *Prot.* is the PDB code for the protein; *Res.* is the number of residues; *Contacts* is the number of contacts in the contact map at 7Å; *Target* is the optimal value used as stopping criteria for the algorithms.

ID	*Prot.*1	*Res.*	*Contacts*	*Prot.*2	*Res.*	*Contacts*	*Target*
1	1gzi	58	110	9msi	59	112	106
2	1fh3	54	86	1ptx	54	93	57
3	3chy	128	378	4tmy	118	366	323

times on each pair of proteins in Table 1, using as target solution the values given in column Target. For each of the 200 runs, the random number generator was initialized with a distinct seed and, therefore, the runs are assumed to be independent. For each instance/target pair, the running times were sorted in increasing order. We associated with the $i-th$ sorted running time t_i a probability $p_i = (i - 1/2)/n$ and plot the points $z_i = [t_i, p_i], i = 1, \ldots, n$. Then, Time-to-target (TTT) plots display the probability that an algorithm will find a solution at least as good as a given target value within a given running time. Figure 2

[5] COLT is a open source library for high performance scientific and technical computing in Java.

shows the time-to-target plots for the algorithms. GRASP-PR has achieved the target values on all instances, always having the best performance in comparison with a Variable Neighborhood Search [10] (VNS) and a Lagrangian Relaxation based algorithm [2] (LAGR).

Looking at this preliminary experiments, GRASP-PR seems to be a well-suited approach for the MAX-CMO.

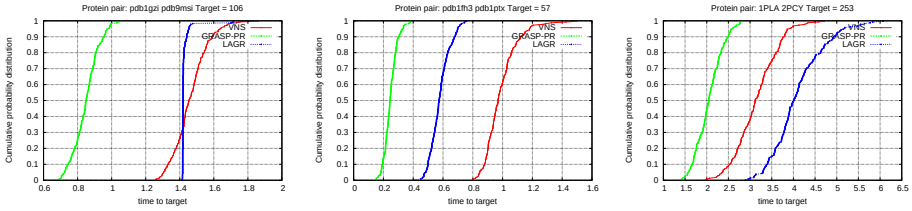

Fig. 2. Time to target distributions comparing GRASP-PR, VNS, and LAGR.

References

1. R. Andronov, N. Malod-Dognin, and N. Yanev. Maximum contact map overlap revisited. *Journal of Computational Biology*, 18(1):27–41, 2011.
2. A. Caprara and G. Lancia. Structural alignment of large-size proteins via lagrangian relaxation. In *In Proceedings of the Sixth Annual International Conference on Computational Biology*, pages 100–108. ACM press, 2002.
3. T. A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, March 1995.
4. P. Festa and M.G.C. Resende. An annotated bibliography of GRASP – Part I: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009.
5. P. Festa and M.G.C. Resende. An annotated bibliography of GRASP – Part II: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.
6. P. Festa and M.G.C. Resende. Hybrid GRASP heuristics. *Studies in Computational Intelligence*, 203:75–100, 2009.
7. P. Festa and M.G.C. Resende. GRASP: Basic components and enhancements. *Telecommunication Systems*, 46(3):253–271, 2011.
8. P. Festa and M.G.C. Resende. Hybridizations of GRASP with path-relinking. *Studies in Computational Intelligence*, 434:135–155, 2013.
9. A. Godzik, A. Kolinski, and J. Skolnick. Topology fingerprint approach to the inverse protein folding problem. *J Mol Biol*, 227(1):227–238, September 1992.
10. David A Pelta, Juan R Gonzalez, and Marcos Moreno Vega. A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, 9(1):1–16, 2008.
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, 2005.