Effective Stochastic Local Search Algorithms for the Genomic Median Problem

Renaud Lenne[1,3], Christine Solnon[2], Thomas Stützle[3],
Eric Tannier[4] and Mauro Birattari[3]
[1]Université Lyon 1, Lyon, France
[2]LIRIS, Université Lyon 1, Lyon, France
[3]IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
[4]INRIA Rhône-Alpes, LBBE, Université Lyon 1, Lyon, France

Abstract

The Genomic Median Problem is an optimization problem inspired by a biological issue: it aims at finding the genome organization of the common ancestor to multiple living species. It is formulated as the search for a genome that minimizes some distance measure among given genomes. Several attempts have been made at solving the problem. These range from simple heuristic methods to a stochastic local search (SLS) algorithm that is inspired by a well-known local search algorithm for the satisfiability problem in propositionnal logic, called `WalkSAT`. The objective of this study is to implement improved algorithmic techniques, particularly ones based on tabu search, in the quest for better quality solutions for large instances of the problem. We have engineered a new high-performing SLS algorithm, extensively tested the developed algorithm and found a new best solution for a real-world case.

1 Introduction

The objective of the Genomic Median Problem (GMP) is to find the probable organization of the genome of the common ancestor of two species, given a third more distant one as a comparison. The study and the solutions to this problem can lead to discover properties of common ancestors of existing species and help making better classifications.

The GMP is an optimization problem that can be formulated as follows. Given three genomes and a distance function that measures in some way the number of rearrangements needed to move from one genome to another one, find a fourth genome that minimizes the sum of the distances between this new one and the three given ones.

There have been various attempts at solving the problem algorithmically ranging from rather simple heuristics [8, 2] to more complex local search

algorithms [3, 6]. These existing algorithms produce solutions that are often of good quality but that are not necessarily optimal and for larger instances there may be significant gaps to optimal solutions. In addition, compared to the currently available local search techniques, the approaches are rather simple and therefore one can conjecture that there is room for improving their performance.

Motivated by these observations, we developed a new high performing stochastic local search (SLS) algorithm for the GMP. The goal of this new SLS algorithm is to improve upon the performance of the current state-of-the-art algorithms in terms of the run-time required to reach specific bounds on the solution quality and ideally to find also better quality solutions, thus, providing new state-of-the-art solutions that may be of biological relevance.

2 Model and methods

A *genome* is an unordered set of chromosomes; a *chromosome* is an ordered list of markers, where each *marker* is modelled by a different signed integer. The three genomes of a GMP instance share the same set of n markers. Hence, each genome is modelled by a signed permutation of the same n integers grouped by chromosomes. The number of chromosomes is not fixed a priori and it may vary from one genome to another.

Various methods for solving the *GMP* have been proposed. These either work on a simplified problem that considers only one chromosome and that involves finding the median of a signed permutation, like `GRAPPA` [8] or `AmGRP` [2]; or use rather simplistic search methods like `MGR-MEDIAN` [3], which uses a greedy constructive algorithm. The best performance results so far have been reported for `MedRByLS` [6], a local search algorithm that is inspired by `WalkSAT`, a well-known local search algorithm for the satisfiability problem in propositionnal logic.

We base our algorithm on the same data structures and neighborhood as used in `MedRByLS`:

- The distance between two genomes is approximated by the BreakPoint Graph distance described in [5, 7, 1], and later extended to multi-chromosomal genomes in [6]. This distance is defined with respect to the number of cycles and paths in an edge bicolored graph obtained from the two genomes by linking adjacent markers. This distance is computed in linear time with respect to the number of markers.

- A *move* consists in the change of two edges with the same colour by inverting one of their nodes. From the biological point of view, such a move corresponds to a transformation in one genome, while the distance between two genomes is the minimum number of such transformations needed to transform one genome into the other.

We have first re-implemented `MedRByLS`, thus allowing a direct comparison of our new algorithms to the original `MedRByLS` using a same implementation of the data structures. For this comparison, we verified that our re-implementation matches the performance of the original version.

As a next step, we enhanced the local search by a simple tabu search scheme. For the search diversification of the resulting tabu search algorithm, we integrated it into the iterated local search framework by adding appropriate perturbations and acceptance criteria. This resulted in an algorithm that we called `MedITaS` (for Median solver by Iterated Tabu Search). More into details, it consists of the following main algorithmic components.

1. A simple Tabu Search (TS) algorithm that forbids the reversal of the last t moves (where t is the tabu tenure), that is, the last changed nodes. We considered a *first-improvement* strategy for the choice of the move to perform because the neighbourhood is very large so that a best-improvement strategy (that implies a full scan of the neighborhood) would be too time-consuming.

2. An Iterated Local Search (ILS) algorithm that perturbates the solution when the search is stuck in plateau-moves or in a basin of attraction (that is, when too many already visited solutions are recalculated). The perturbation uses a rearrangement of k edges and then TS is rerun starting from the perturbed solution. Finally, an acceptance criterion decides whether either the solution before the perturbation or the one after is kept for the next iteration of ILS. The implemented acceptance criterion accepts a new solution if it is better than the previous one; otherwise, the previous solution has a user-defined probability of being kept (in our test, we used the default value of 0.2).

3. A reactive version of TS (that reactively adapts the tabu list length) and a reactive version of ILS (that reactively tunes the perturbation strength) have been implemented. This was done since initial experiments showed that both ILS and TS were very sensitive to parameter settings and that the optimal parameter settings were very different from one instance to one other.

3 Results

In order to evaluate our algorithms, we ran multiple comparisons. All runs were made on a same Dual-Core AMD Opteron2216 HE (2 processors at 2.4GHz) with 4GB of RAM; only one core is used for each execution since our algorithm is implemented as a fully sequential one.

The first set of data contains 22 randomly generated instances of different difficulties (with respect to the definition of the phase transition by [6]) but with the same size (500 markers). The set has 11 levels of hardness and

2 instances per level. On this set we run our `MedITaS` algorithm and our implementation of the basic local search algorithm `MedRByLS` from [6] for 20 independent trials on each instance and 40 seconds per trial. The comparison of the best solution qualities reached by both algorithms on each instances is given in Figure 1. From this figure, we can see that `MedITaS` always gives solution qualities that are at least as good as `MedRByLS` and that the gap between the two algorithms tends to increase as the instances become harder.

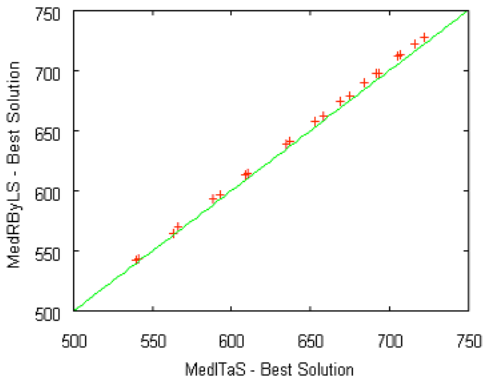

Figure 1: Solution Value Comparison

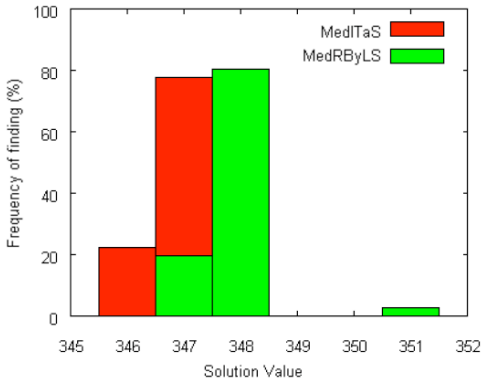

Figure 2: Frequency Comparison

In another experiment we used a real-world instance: the human-mouse-rat comparison, which was also used in [6]. This instance is made of 424 markers and the best median found so far had a value of 346. We ran each algorithm 35 times for a computation time limit of 60 seconds. From these runs, we generated Figure 2, which represents the histogram of the frequency of finding certain solution qualities with the two main algorithms (`MedITaS` and `MedRByLS`). It is clear from this graph that `MedITaS` finds solutions that are at least as good as those found by `MedRByLS` and always of a very good quality (of 347 or better); `MedRByLS` sometimes fails to find good solutions: on some runs it returned a solution of value 351). We should also notice that `MedRByLS` has a quite low probability (less than 20%) of finding a solution of 347 or better. Also, our algorithm found a new best solution for this instance with an evaluation function value of 345.

4 Discussion

Our implementation of the Iterated Tabu Search gave very promising results. First, we have seen that `MedITaS` always gives at least as good or better results, in the same computation time, than the former best algorithm (`MedRByLS`). We also found a new best solution for the human-mouse-rat common ancestor.

The developped algorithmic techniques perform significantly better than previously developped ones from a solution quality point of view. But from a biological point of view, the distance used here (as the one used in all preceding attempts at solving the problem) does not seem to reflect the biological reality of the evolution process (as it is explained in [4]). Thus, a research on a more appropriate distance measure has to be envisaged.

Also, we noted in our experiments that there were a lot of medians with exactly the same value. It could be a good idea to do some comparison between them trying to extract some valuable information on the most probable characteristics of the real ancestor.

Acknowledgements. The authors would like to thank to Yannet Interian for her kind help in any questions regarding the implementation of her algorithm. Thomas Stützle acknowledges support from the Belgian FNRS of which he is a Research Associate.

References

[1] V. Bafna and P. Pevzner. Genome rearrangements and sorting by reversal. *SIAM Journal on Computing*, 25:272–289, 1996.

[2] M. Bernt, D. Merkle, and M. Middendorf. Using median sets for inferring phylogenetic trees. *Bioinformatics - Oxford Univ Press*, Volume 23, Number 2:e129–e135, 2007.

[3] G. Bourque and P. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Res.*, 12(1):26–36, 2002.

[4] N. Eriksen. Reversal and transposition medians. *Theoretical Computer Science*, 374(1-3), 2007.

[5] S. Hannenhalli and P. A. Pevzner. Polynomial algorithm for genomic distance problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, 1995.

[6] Y. Interian and R. Durrett. Computing genomic midpoints, 2007. Submitted.

[7] J. D. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1/2):180–210, 1995.

[8] B. Moret, S. Wyman, D. Bader, T. Warnow, and M. Yan. A new implementation and detailed study of breakpoint analysis, 2001. Proc. 6th Pacific Symp. on Biocomputing (PSB 2001), Hawaii, World Scientific Pub.