Computers & Operations Research 40 (2013) 3108–3120

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A GRASP metaheuristic for microarray data analysis

Roberto Cordone [a,*], Guglielmo Lulli [b]

[a] University of Milano, Department of Computer Science, Via Comelico 39, 20135 Milano, Italy
[b] University of Milano "Bicocca", Department of Informatics, Systems and Communication, viale Sarca 336, 20122 Milano, Italy

ARTICLE INFO

Available online 17 October 2012

Keywords:
Gene regulatory networks
DNA microarray data
GRASP
Path Relinking
Tabu Search

ABSTRACT

The Weighted Gene Regulatory Network (WGRN) problem consists in pruning a regulatory network obtained from DNA microarray gene expression data, in order to identify a reduced set of candidate elements which can explain the expression of all other genes. Since the problem appears to be particularly hard for general-purpose solvers, we develop a Greedy Randomized Adaptive Search Procedure (GRASP) and refine it with three alternative Path Relinking procedures. For comparison purposes, we also develop a Tabu Search algorithm with a self-adapting tabu tenure. The experimental results show that GRASP performs better than Tabu Search and that Path Relinking significantly contributes to its effectiveness.

1. Introduction

A DNA microarray is a tool for analyzing gene expression. It is a small, solid support (usually a glass microscope slide, but it can also be silicon chips or nylon membranes) onto which the sequences from thousands of different genes are attached at fixed locations. It works by exploiting the ability of a given mRNA molecule to bind specifically to the DNA template from which it originated: by measuring the amount of mRNA bound to each site on the microarray, scientists can determine the expression levels of hundreds or thousands of genes in a single experiment.

The use of DNA microarray is reshaping biomedical sciences by making available in public repositories a large amount of gene expression data. This allows to apply computationally intensive data analysis methods to unveil the functioning of the regulatory systems of which the individual gene and its interactions form a part, see [4,30]. On this subject, uncovering the gene function and operation, and their functional linkages is essential to understand how genes are implicated in the control of intracellular and intercellular processes [2], how genomic expression programs unfold during developmental processes, how the molecular machinery of cells works to respond adequately to environmental clues and to maintain homeostasis, and, consequently, how to manipulate these processes to human advantage. Hence, gaining an understanding of the emergence of complex patterns of behavior from the interactions between genes in a regulatory network poses a huge scientific challenge with potentially high industrial pay-offs.

Several reverse engineering approaches have been proposed to make sense of large, multiple time-series data sets arising in expression analysis. See [16,18,28] for relevant surveys on the subject. The purpose of these methods is to produce a high-fidelity representation of the cellular network topology as a graph, where nodes represent genes and arcs represent direct regulatory interactions (i.e., influences of gene products upon the expression of other genes), thus explaining gene expression data.

Herein, we present an optimization approach to reconstruct gene regulatory networks from DNA microarray gene expression data. More specifically, we focus on the problem of pruning a putative regulatory network to identify a small set of interesting candidate regulatory elements [7,29]. The model generates networks in which a relatively small number of regulators explain the expression of all genes, while the other elements are considered neutral, i.e., do not have any activation/inhibition influence upon other genes of the network, though they play a role in biochemical intracellular and intercellular processes. We do not assert that the result of our computation identifies the real regulatory network, but we believe that our approach quickly enables biologists to focus on interesting features extracted from raw expression array data sets.

The considered model is intrinsically difficult to solve because it admits the Set Covering problem as a very special case [8,29]. As a matter of fact, commercial solvers fail to solve even fairly small instances in reasonable computational times. To compute good quality solutions of large instances, we implemented a Greedy Randomized Adaptive Search Procedure (GRASP), which is a well-known local search metaheuristic methodology for hard combinatorial optimization problems, see [12,13,33]. To improve the

* Corresponding author.
E-mail addresses: roberto.cordone@unimi.it (R. Cordone), lulli@disco.unimib.it (G. Lulli).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.10.008

effectiveness of the algorithm, we also introduced three different intensification strategies, based on the Path Relinking (PR) framework [22]. These strategies search for better solutions along trajectories suitably designed in the solution space, but each one applies a different criterium to select the destinations of these trajectories. To assess the performance of the GRASP, we compare its results to those achieved by a Tabu Search (TS) competitor, which explores the same neighborhood. This alternative approach is also known to provide effective algorithms for hard combinatorial optimization problems [21].

The paper is organized as follows. In Section 2 we formally define the problem, through a mathematical programming formulation. In Sections 3 and 4 we present a GRASP with PR and a TS heuristic to compute good quality solutions in a reasonable amount of time. Section 5 compares the computational results of the two approaches and, finally, Section 6 draws some conclusions.

2. A formal definition of the problem

The problem of designing a gene network which provides a parsimonious explanation for the expression of a set of genes is known as Weighted Gene Regulatory Network (WGRN) problem. This problem models the gene network as a weighted directed graph $\mathcal{G}(N,A\cup I,w)$, whose set of nodes N represents the gene products, while two disjoint sets of arcs $A\subset N\times N$ and $I\subset N\times N$ represent the putative activations (A) and inhibitions (I), respectively. The weight function $w: A\cup I\to[0;1]$, derived from the activation–inhibition index [7], measures how strongly the genes activate or inhibit each other: $w_{ij}=0$ denotes a full correlation between gene products i and j, whereas $w_{ij}=1$ denotes the absence of any relation between them.

The problem amounts to determining a subset of gene products which explain both the activation and the inhibition of all the genes in the network. This means that each node must have at least one activator and one inhibitor arc incoming from the identified subset of nodes. Each gene product should be labeled as activator, inhibitor or neutral. Neutral products exert no relevant influence. In general, activator (resp. inhibitor) products should only exert activation (resp. inhibition) influences, but few exceptions to this cornerstone are allowed, that is some activator or inhibitor gene products exert influences opposite to their label. Their presence is consistent with biological evidence, but their number is limited [31]. Therefore, the WGRN model minimizes the total weight of activation (resp. inhibition) influences exerted by a node labeled as inhibitor (resp. activator). In the following, such influences are named *incoherent*. In Fig. 1, we depict an example of a gene regulatory network. In this example, activations (*resp.* inhibitions) are depicted with a black solid (*resp.* dashed) arrow. The solution represented has two activators (full-black nodes) and one inhibitor (full-grey node). Note that, though node 4 is labeled as activator, it inhibits node 1. This incoherence increases the objective function value by an amount equal to the weight of the arc (4,1).

We formulate the problem by assigning two binary variables to each node: $z_i^{(A)}=1$ if node i is labeled as activator, 0 otherwise; $z_i^{(I)}=1$ if node i is labeled as inhibitor, 0 otherwise. For each arc (i,j) of the putative network, binary variable x_{ij} states whether the arc is used as an incoherent influence ($x_{ij}=1$) or not ($x_{ij}=0$):

$$\min \quad \phi=\sum_{(i,j)\in A\cup I} w_{ij}\cdot x_{ij}$$

$$\text{s.t.} \quad \sum_{i:(i,j)\in A}(z_i^{(A)}+x_{ij})\geq 1 \quad \forall j\in N \tag{1}$$

$$\sum_{i:(i,j)\in I}(z_i^{(I)}+x_{ij})\geq 1 \quad \forall j\in N \tag{2}$$

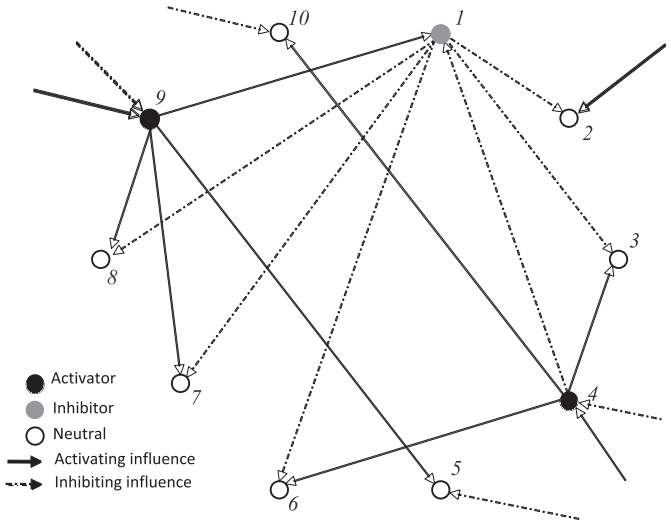

Fig. 1. Example of gene regulatory network.

$$z_i^{(A)}+z_i^{(I)}\leq 1 \quad \forall i\in N \tag{3}$$

$$\sum_{i\in N}(z_i^{(A)}+z_i^{(I)})\leq M \tag{4}$$

$$x_{ij}\leq z_i^{(I)} \quad \forall (i,j)\in A \tag{5}$$

$$x_{ij}\leq z_i^{(A)} \quad \forall (i,j)\in I \tag{6}$$

$$x_{ij}\in\{0,1\} \quad \forall (i,j)\in A\cup I \tag{7}$$

$$z_i^{(A)},z_i^{(I)}\in\{0,1\} \quad \forall i\in N \tag{8}$$

Constraints (1) and (2) force each node to have at least one activator and one inhibitor arc incoming from the subset of nodes identified. As they are set covering constraints, when an activation (resp. inhibition) constraint is satisfied, we will say that the corresponding node is covered in activation (resp. inhibition).

Constraints (3) require each node to be labeled either as activator, inhibitor or neutral (in this case, both $z_i^{(A)}$ and $z_i^{(I)}$ are set to 0). To guarantee a parsimonious explanation, the number of activator and inhibitor nodes is bounded above by Constraint (4). Notice that if an optimal solution includes less than M labeled (activator or inhibitor) nodes, it is always possible to build an equivalent solution with exactly M labeled nodes. In fact, introducing any additional labeled node in the solution would still satisfy the covering and disjunction constraints without inserting any further incoherent influence. In view of this observation, in all algorithms described in the sequel we replaced the inequality of constraints (4) by an equality. Finally, Constraints (5) and (6) impose that no neutral node exerts any influence.

3. A GRASP metaheuristic with PR

GRASP is a multi-start metaheuristic which alternatively builds starting solutions and improves them. The constructive phase uses a greedy randomized construction procedure; the improvement phase uses local search [12,13,33]. PR is a post-processing mechanism, which is periodically applied to the local optima found by the search procedure and combines them to a subset of elite solutions suitably maintained [22]. The best solution found during the whole process is returned as the final result. Hybridizations of GRASP and PR have been applied to a wide range of combinatorial optimization problems [33]. The following subsections