UNIVERSITY OF LIÈGE

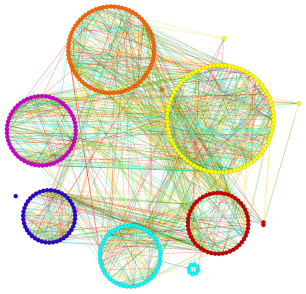
MASTER IN ENGINEERING

May 10, 2021

MATH0462
Discrete Optimization

Project :

Module analysis in cancer diagnosis



Jean-David Mukolonga - **s170679**
Maxime Lione - **s175606**

Question 1, 2 and 3

Heuristic :

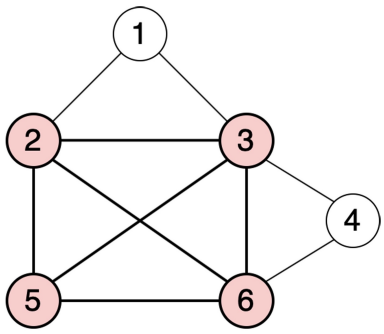
We've chosen to implement a **simulated annealing** metaheuristic in order to find the largest module. Firstly, an objective function had to be set up. To do so, let's define $G(V,E)$, the graph drawn by the set V of the n genes identified in the file that represent the vertices, and the set E of all the existing coexpressions between the genes that represent the edges. $M_G(m_{i,j})$ is the adjacency matrix of $G(V,E)$, with $i,j = 1,2,...,n$. The value s is the size of a module, and V' represents a permutation of V. According to the definition of a module in question 1, the graph $G(V,E)$ has a module of size s if it exists a permutation V' of V such that :

$$\sum_{i'=1}^{s-1} \sum_{j'=i'+1}^{s} (1 - m_{i',j'}) = 0 \tag{1}$$

The goal is thus to minimize the left part of (1), for the biggest s. The objective function can be written as :

$$min\ F(G,V') = \sum_{i'=1}^{s-1} \sum_{j'=i'+1}^{s} (1 - m_{i',j'}) \tag{2}$$

To illustrate the sum (1), here is a simple example. Figures 1 (a) and (b) respectively illustrates a graph and it's adjacency matrix, and Figure 1 (c) illustrates the permutation V' and its elements on which the sum is computed.



(a)

i \ j	1	2	3	4	5	6
1	1	1	1	0	0	0
2	1	1	1	0	1	1
3	1	1	1	1	1	1
4	0	0	1	1	0	1
5	0	1	1	0	1	1
6	0	1	1	1	1	1

(b)

i \ i' \ j'	2	3	5	6	1	4
2 \ 1	1	1	1	1	1	0
3 \ 2	1	1	1	1	1	1
5 \ 3	1	1	1	1	0	0
6 \ 4	1	1	1	1	0	1
1 \ 5	1	1	0	0	1	0
4 \ 6	0	1	0	1	0	1

(c)

Figure 1: (a) Example of a graph (b) Adjacency Matrix of $G(V,E)$ (c) Adjacency Matrix of $G(V',E')$

With this in mind, once all the coexpessions have been collected from the file, the maximum possible size of a clique s_{max} is computed. Then, the parameters T_i (initial temperature), T_f (final temperature) and c (cooling coefficent) are set. Then, the following process is repeated for s in $[2, s_{max}]$:

- While the current temperature t is higher than T_f, the initial state is defined by a random permutation of all the genes from the file. A neighbor V'' to the initial permutation V' is defined by a swap of two different genes. Thus, the increment of objective function is computed as : $\Delta F = F(G,V'') - F(G,V')$

- If $\Delta F <= 0$, the new permutation is accepted. $\Delta F > 0$, the new permutation is accepted with the probability $P = e^{\frac{-\Delta F}{t}}$.

Regarding questions 2 and 3, a module is respectively defined as :

- a set of genes in which all pairwise co-expressions are present, except a predefined number of them.
- a set of genes in which all pairwise co-expressions are present and the number of allowed missing co-expressions is bounded by an increasing function of the size of the module.

Consequently, the heuristic process to find the largest module is exactly the same as described above, except that the equality (1) respectively becomes the following inequalities :

- $\sum_{i'=1}^{s-1} \sum_{j'=i'+1}^{s} (1 - m_{i',j'}) <= k$
- $\sum_{i'=1}^{s-1} \sum_{j'=i'+1}^{s} (1 - m_{i',j'}) <= \frac{k}{100} * \frac{s!(s-1)!}{2}$

In that way, for the definition of modules of question 2, we indeed allow the modules to have k missing co-expressions. For the definition of modules of question 3, since $\frac{s!(s-1)!}{2}$ is the amount of edges in a complete graph of s nodes, we allow the modules of size s to have $k\%$ of missing co-expressions.

MIP :

In order to develop a Mixed-integer formulation of the problem, we will use the same parameter definitions as above. The MIP formulation can be written as :

$$
\begin{aligned}
& max \sum_{i=1}^{n} x_i \\
& s.t. \qquad x_i + x_j \leq 1, \ \forall (i,j) \notin E \ (1) \\
& \qquad\quad x_i \in \{0,1\}, \ i = 1,...,n \ (2)
\end{aligned}
$$

The variable x_i is equal to 1 if the gene i is in the maximum module and 0 otherwise. The solver is thus told to find the module with the highest cardinality (thus containing the highest amount of genes), with the constraint (1) explaining that for any pair of genes i and j that don't share expression, there is at the most one of them that is in the module (since they can't be not adjacent and both included the module).

This MIP formulation is only valid for the definition of a module from question 1. Using the solver *Gurobi*, we didn't manage to adapt this formulation to answer to the question 2 and 3. We thus had to find another formulation of the problem as a mixed-integer programming problem :

In this new formulation, we will introduce the notion of *density d* of a subgraph (a module) such as $d \in (0,1)$. This density of a module is defined as the ratio of the number of edges with respect to the maximum possible edges. With this in mind, still using the same parameters as above, we can write the following MIP formulation :

$$
\begin{aligned}
& max \sum_{i=1}^{n} x_i \\
& s.t \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} m_{i,j} x_i x_j \geq d \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_i x_j \quad (1) \\
& x_i \in \{0,1\}, \ i = 1,...,n \qquad\qquad (2)
\end{aligned}
$$

The constraint (1) is indicating that a module will be considered as such if and only if it has at least the required predefined density. However, since this constraint is quadratic, we will use a trick that we've seen during the practicals, by introducing a new variable $w_{i,j} = x_i x_j$, which will lead to four new linear constraints. We can re-formulate the MIP problem as :

$$max \sum_{i=1}^{n} x_i$$

$$s.t. \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (d - m_{i,j}) w_{i,j} \leq 0 \quad (1)$$

$$w_{i,j} \leq x_i, w_{i,j} \leq x_j, w_{i,j} \geq x_i + x_j, i = 1,...,n,\ j = i,...,n \quad (2)$$

$$w_{i,j} \geq 0,\ x_i \in \{0,1\},\ i = 1,...,n\ j = i,...,n \quad (3)$$

Using this formulation, for the question 3, we can fix the value of d to $1 - \frac{k}{100}$ in order to accept $k\%$ of missing co-expressions. For the question 2, the constraint 1 becomes $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (1 - m_{i,j}) w_{i,j} \leq k$ where k is the predefined number of missing co-expressions.

Question 4

MIP :

In this section, we first develop a MIP formulation to find a covering of the different genes in modules, with the union of all modules covering the full list of genes and not overlapping "too much". To do so, we had do set up some assumptions. Indeed, firstly we have decided that we'll use the same definition of a module as in question 1, which means that a module is defined as a set of genes in which all pairwise co-expressions are present. Secondly, we have defined that "not overlapping too much" would mean that a gene can't appear in more than 2 modules. Finally, we assume that a module is made up of 2 genes at least.

We still use the definition of a graph $G(V,E)$, but we introduce a new set C, which contains all the index of the candidate modules, such as $C = \{1,2,...|E|\}$. Furthermore, we use 2 variables : For $i \in N$ and for $c \in C$, the variable $x_{i,c}$ is equal to 1 if the gene i is in module c and 0 otherwise. For $c \in C$, the variable z_c is equal to 1 if the module c is used and 0 otherwise. With this in mind, we can build the following MIP formulation to find a minimum module cover :

$$min \sum_{c} z_c$$

$$s.t.$$

$$x_{i,c} \leq z_c \quad (1)$$

$$x_{i,c} + x_{j,c} \leq 1,\ \forall (i,j) \notin E \quad (2)$$

$$1 \leq \sum_{c} x_{i,c} \leq 2 \quad (3)$$

$$\sum_{i} z_c x_{i,c} \geq 2 z_c \quad (4)$$

$$z_c \in \{0,1\},\ c \in \{1,...,|E|\} \quad (5)$$

$$x_{i,c} \in \{0,1\},\ i \times c \in \{1,...,n\} \times \{1,...,|E|\} \quad (6)$$

The solver is thus charged to find a covering of the different genes using as few modules as possible.

Heuristic :

Since the resolution of our MIP program is excessively slow, we've also implemented a heuristic technique to solve this problem. In our heuristic program, the assumptions that we use are that a module is defined as in question 2, which means it's defined as a set of genes in which all pairwise co-expressions are present, except a predefined number of them. We've set this number to 7. Moreover, we'll consider here that the modules don't overlap at all.

Knowing this, our heuristic algorithm is a simple extension of the heuristic used for questions 1 to 3 : We're looking for the highest module in V (in the exact same way as questions 1 to 3). Once this module is found, all its nodes are removed from the set of nodes V, leading to a shorter set of nodes V', on which the search for the highest module is repeated, and so on until all the nodes have been covered.

References

[1] https://www.researchgate.net/publication/227238953_The_Maximum_Clique_Problem

[2] https://www.researchgate.net/publication/220312579_A_simple_simulated_annealing_algorithm_for_the_maximum_clique_problem

[3] https://www.sciencedirect.com/science/article/pii/S0166218X12002843