

## Parallel genetic algorithm for the graph partition problem

**Abstract.** The performances of genetic algorithms relies heavily on various parameters, particularly the encoding scheme used to represent solutions. This paper introduces a novel parallel genetic algorithm that employs two sub-populations each with a specific encoding scheme. Knowing that each encoding scheme possesses its own features that may help it to better explore the search landscape or even prevent it to get through promising search areas. This approach aims to mitigate the limitations associated with these encodings and enhance the performance of the genetic algorithm for addressing the graph partitioning problem (GPP) in order to improve enhance the efficacy of the genetic algorithm. The encoding schemes employed in this study include an integer vertex based encoding and a binary edge based encoding. The core concept is the migration of best solutions among the sub-population following a specific strategy.

**Key words :** parallel genetic algorithm, encoding schemes, solutions migration.

### Introduction

Since their appearance in 1970s by J.Holland [ref], GAs still achieve impressive improvement. This stochastic bio-inspired methods are known by their simplicity and effectiveness to solve hard problems. Their principle is rather simple. Indeed, the process consists of evolving iteratively a population of candidates' solutions throughout a specific process till the fulfillment of a certain condition. Genetic operators such as natural selection, crossover and mutation are applied during the whole process.

As a blind method, GAs are strongly rely on their encoding scheme used to represent solutions to better explore the search landscape related to the treated problem [ref chaouche]. In their paper [ref\_chaouche] Ali et Menouar had presented a set of encoding schemes used by GAs to solve the graph partitioning problem. However, each of the presented encoding scheme has its limitations and drawbacks.

The main idea behind this work is to use a parallel GA, in which we make a set of sub-population each with a specific encoding scheme evolve together in order to get advantage of each of these encoding schemes. A migration strategy is applied to get better solutions transferred from a sub-population to another with the aim of getting the search process of the later redirected to new promising areas.

The next of this paper is structured as follow :

Section 1 : graph partitioning problem

Section 2 : parallel genetic algorithm

Section 3 : Encodings schemes, (DVTC, BE)

Section 4 : Results

## Conclusion

### Graph partitioning problem

The graph partitioning is widely used in many areas such as community detection in social networks [ref], as biological networks [ref], road networks [ref], image processing [ref], production management [ref], and so on.

A formal definition of GPP is given as follow :

given an undirected weighted graph  $G = (V, E)$  where  $V = \{v_1, \dots, v_n\}$  is the set of the vertices and  $E = \{e_1, \dots, e_m\} \subset (V \times V)$  the set of the edges to each we associate a *non-negative* weights denoted  $\omega(e)$ . The *graph partitioning problem* (GPP) asks for a partition  $P = \{C_1, \dots, C_k\}$  of  $V$  in  $k$  blocks of nodes namely clusters that satisfy the following constraints:

$$\forall i, j \in \{1, \dots, k\}, i \neq j: C_i \cap C_j = \emptyset \quad (1)$$

$$\bigcup_{i=1}^k C_i = \emptyset \quad (2)$$

$$\forall i \in \{1, 2, \dots, k\}: C_i \neq \emptyset \quad (3)$$

In addition, feasible solution must satisfy the following constraints in order to be accepted:

- **Coexistence constraints:** let  $C$  the set of vertex pair that must be in the same cluster. Coexistence constraints is formulated as follow :

$$\forall (v_k, v_l) \in C, k, l \in \{1, 2, \dots, n\}, \exists C_i \in P, i \in \{1, 2, \dots, |P|\}: v_k, v_l \in C_i \quad (4)$$

- **Non coexistence constraints:** let  $T$  the set of vertex pair that must be in the same cluster. Non Coexistence constraints is formulated as follow :

$$\forall (v_k, v_l) \in T, k, l \in \{1, 2, \dots, n\}, \exists C_i, C_j \in P, i, j \in \{1, 2, \dots, |P|\}, i \neq j: \\ v_k \in C_i \text{ and } v_l \in C_j \quad (5)$$

- **Cluster size constraint :** the partition's clusters size must be lower than or equal a maximum predefined number of vertices denoted  $N$ , that is :

$$\forall i \in \{1, 2, \dots, |P|\}, C_i \leq N \quad (6)$$

The aim of the objective function of the  $k$ -way graph partitioning problem is to search for a partition  $P^*$  that minimize the sum of the edge cut weight  $E_c \subset E$  defined as follows:

$$E_c = \{ (v_i, v_j) / (v_i, v_j) \in E \text{ and } \exists C_k, C_l, k \neq l : v_i \in C_k \text{ and } v_j \in C_l \} \quad (7)$$

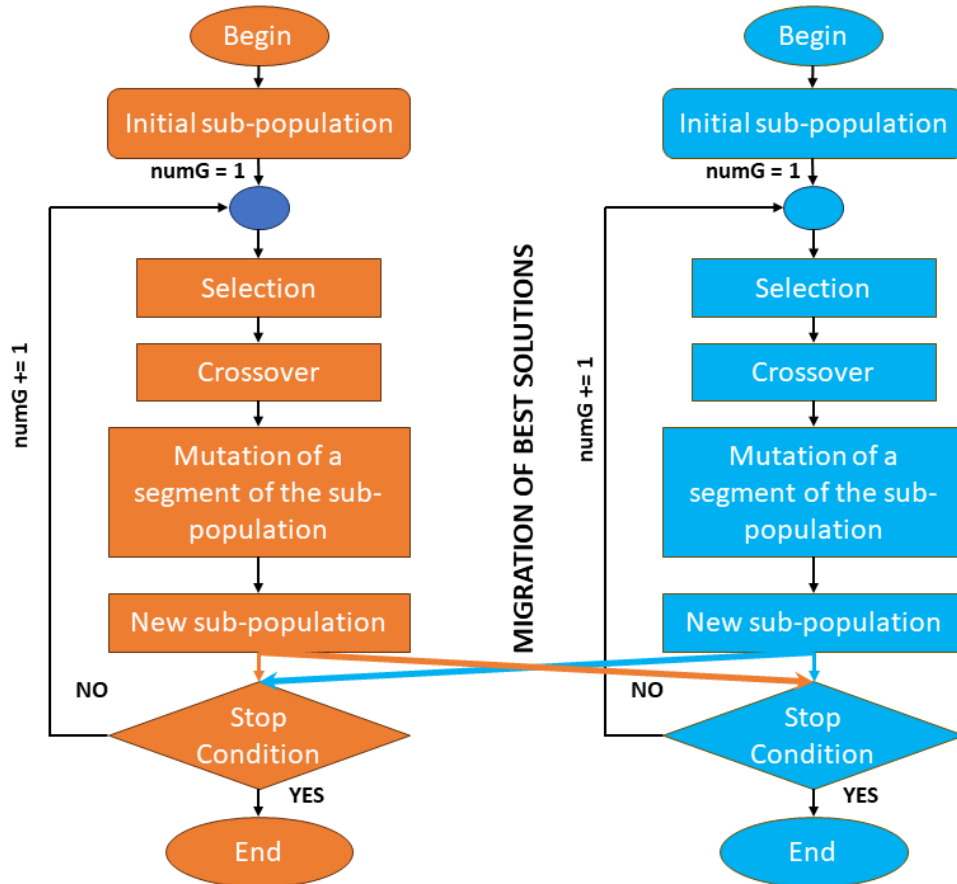
That si:  $\sum_{e \in E_c} \omega(e)$  is minimal.

## 5. Parallel genetic algorithm

Traditional genetic algorithms (GAs) suffer from the fact that they use only one encoding scheme to represent solutions. An encoding scheme possesses some characteristics such as blindness which is the incapability of the encoding to represent some kind of solutions in the search space, or redundancy where a single solution may have more than one representation at a time, to name just a few [rauthoulf]. These intrinsic features may have dramatic consequences on the performances of the GA.

To overcome these limitations, we propose a parallel genetic algorithm with two distinct sub-populations each of them uses a different encoding. A migration strategy is used to share best solutions of both sub-populations in order to redirect their respective search process to hopefully new promising areas.

The whole process is described in figure.1.



**Figure 1.** General Schema of a parallel Genetic Algorithm

As shown in Figure.1, best solutions are migrated from their own sub-population to the other one. The migration strategy adopted in this paper consists of transferring best solutions periodically after a certain number of iterations, which is an important parameter that we have settled empirically.

## Encoding Schemes:

In genetic algorithms (GAs), the encoding scheme plays a crucial role in representing candidate solutions to the optimization problem at hand. The choice of encoding directly impacts the performances of the GA, and ultimately, the quality of solutions produced. In this section, we explore the two encoding schemes employed by the sub-populations.

### 1. vertex-to-cluster encoding (VTC)

The solutions are represented through a vector of length  $|V|$  (representing the number of vertices in the graph), wherein each allele denotes the cluster hosting the corresponding vertex, as illustrated in Figure 2. In which, both  $v_1$  and  $v_2$  are assigned to the first cluster, which explain the fact that share the same value within the chromosome.

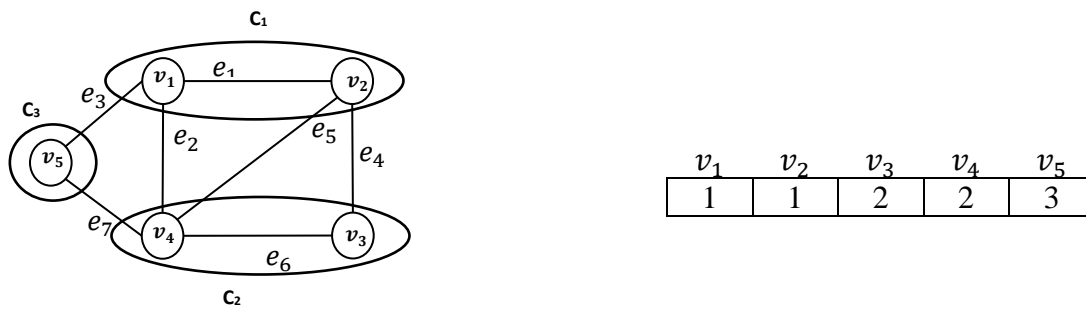


Figure 2. The VTC representative chain of a graph partition with three clusters.

### 2. Binary string encoding (BS)

This encoding is proposed by Boulif [] to solve the machine cell formation problem. It uses a chain of length  $|E|$  (number of edges in the graph) where each allele represents an edge of the graph. The values of these alleles are either 0 or 1. The zeros alleles correspond to the intra-cluster edges ( the vertices forming the extremities of this edge are in the same cluster) and the ones alleles are the inter-cluster alleles ( the vertices forming the extremities of this edge are in two deferent clusters). Figure 3 give a representative chain for the partition of figure 1.a using the binary string representation.

$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$
0	1	1	1	1	0	1

**Figure 2.** A binary string representation chain of the partition in figure 1.a

The partition is gotten by recovering the connected components formed by the intra-cluster edges. For the figure bellow, the cluster  $C_1 = \{v_1, v_2\}$  formed by the intra-cluster edge  $e_1$ , the second cluster  $C_2 = \{v_3, v_4\}$  formed by the intra-cluster edge  $e_6$ , the remaining vertex  $v_6$  form the last cluster  $C_3$ .

## 5.Results Analysis:

This section is dedicated to the description of the data used in our study, settling of the different parameters along with the metrics employed to measure the performance of the pGA, and at the end presenting and discussing the obtained results according to the chosen metrics.

### 5.1 Data description

For our experimental study we have selected a set of twenty graphs from the literature [Ali and Boulif 2018], the table below illustrates the characteristics of this dataset.

### 5.2 pGA parameters and Metrics

Table 1 summarize the set of pGA parameters and their respective values. It is worth to mention that these values are settled empirically through a set of experiences.

pGA parameters	Values
Sub-population size	100
Elitism rate	10
Reproduction rate	10
Mutation rate	0.02
Run's number	10
Iteration's number	100
Migration's number	10

Table 1. pGA parameters

In the other side, the metrics we have used to measure the performances of the pGA are:

- **Mean Best Fitness (MBF)** : the value of this metrics is calculated by dividing the sum of the fitness values of each run by the number of run settled to 10.
- **Best Fitness (BF)**: this metric represents the best fitness of over the 10 runs of this metrics is calculated by dividing the sum of the fitness values of each run by the n

**Conclusion :**

**References :**