

# Insular genetic algorithm for operational management

A Vilcu<sup>1</sup>, I Herghiligi<sup>1</sup>, I Verzea<sup>1</sup>, M Pîslaru

<sup>1</sup> "Gheorghe Asachi" Technical University of Iasi, Department of Engineering and Management, Mangeron Blvd. 28, TEX1, 700050, Romania

adrian.vilcu@academic.tuiasi.ro

**Abstract.** Problems in the operational management that involve the calculation on graphs are numerous, being encountered in many economic fields: transport problems, problems of resource allocation, logistics problems. Within these problems are found the problems of partitioning graphs with partitions that meet different partitioning criteria. The paper presents a case study for the problem of partitioning graphs - organizing the graph, practical applicability of the problem, partitioning function, method of solving the problem by using three genetic algorithms working individually on islands of problem solutions, changing synchronously with each other solutions, and the application of the algorithm on a numerical example. Finally, the paper proposes conclusions and future directions of development.

## 1. Introduction

The stage of optimizing the flow of activities by balancing is an integral part of any technological process [1][2]. The importance of this stage increases in direct proportion to the increasing complexity of technological systems, the decrease of time for their design and the introduction of new restrictions [3][4]. Given the increased interest in this stage of optimization, numerous algorithms have been proposed that can be classified into two categories: exact algorithms provide optimal solutions for balancing problems in an exponentially time that exponentially increasing with problem size and class of approximate algorithms (heuristic and evolutive algorithms) which provides good quality solutions in a time independent of the size of the problem [5][6].

The objective of this paper is to balance a technological line for a textile product by developing an algorithm that takes into account the dynamic characteristic of the processing line with a new objective function that simulates a restriction of supply process line.

## 2. Material and method

### 2.1. The problem of partitioning graphs

Having in execution phases of a technological process, a graph of precedence restrictions and a technological limitation: each grouping of phases must contain at most  $L$  entries. Thus, the study problem in this paper is to determine the minimum number of groups of execution phases of a technological process that respects the supply restriction. The  $L$  integer is a grouping parameter.

With the new elements introduced, the problem is to partition a graph into (unfixed) subgraphs, each of which has at most  $L$  input elements.

Graph partitioning involves cutting arches; thus, each time an arc is cut, a primary pseudo-input (pseudo-input) and a primary pseudo-output (pseudo-output) are created. It is necessary for each

subgraph to respect the condition, where it represents the number of entries and the number of pseudo-entries.

## 2.2. Objective function

The objectives that are considered in partitioning a graph are:

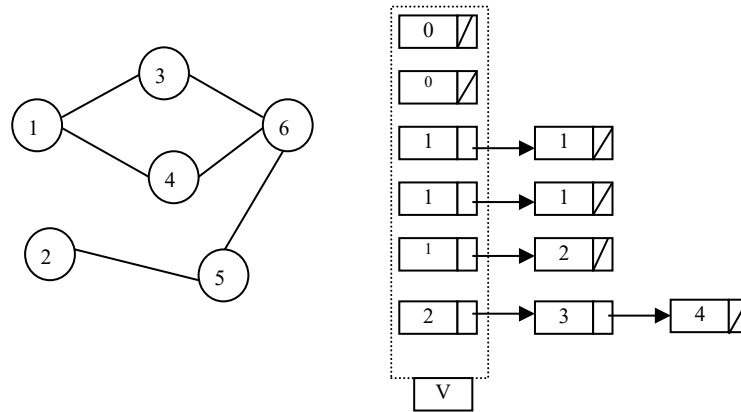
- the total number of connections between subcircuits (number of cut arcs) should be as small as possible;
- the number of subcircuits should be minimal.

These two objectives are contradictory and all the algorithms proposed in the literature try to optimize one objective to the detriment of the others. The objective function used in the present model is to minimize the number of subcircuits.

The first component of each list contains the number of predecessors of the vertices.

## 2.3. Graph representation

The graph is represented by adjacency lists (figure 1). These are massive ones that contain the ends of V chained lists, one for each node, each list containing all the predecessors of the node.



**Figure 1.** Graph representation

## 2.4. Graph partitioning model (GPM)

GPM consists in partitioning the set of vertices  $X$  into  $X_1, \dots, X_j, \dots, X_k$  subsets ( $k$  not fixed) and the graph  $G$  into subgraphs  $G_1, \dots, G_j = (X_j, A_j), \dots, G_k$  so that:

- $X = \bigcup_{j=1}^k X_j$ ,  $X_j \cap X_i = \emptyset$   $\forall i \neq j, i, j = 1, \dots, k$
- $X_j \cap (P \cup M) \neq \emptyset$   $\forall j = 1, \dots, k$
- $n_j + c_j = |X_j \cap E| + |\varpi^-(X_j)| \leq L$   $\forall j = 1, \dots, k$

where:  $M$  - the set of primary inputs,  $P$  - the set of intermediate nodes,  $n_j$  - the set primary inputs of the each group,  $c_j$

iv) let  $G_j = (X_j, A_j)$  the associated subgraph is connected  $\forall j = 1, \dots, k$  or  $G_j$  is compound from  $H$  disjoint graph  $G_1, \dots, G_h = (X_h, A_h), \dots, G_H$  where  $X_h \cap (P \cup M) \neq \emptyset$  and  $G_h$  is connected  $\forall h = 1, \dots, H$

It was noted with  $\varpi^-(X_j)$  - the set of neighbors - the set of vertices that do not belong to the set and that have at least one successor in  $X_j$ :  $\varpi^-(X_j) = \{v \in X / v \notin X_j, \text{ si } \exists \text{ arc}(v, w) \text{ pentru } w \in X_j\}$

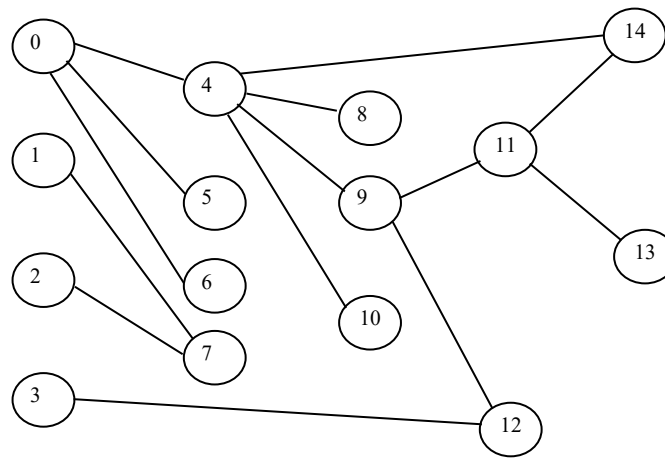
GPM is equivalent to the problem of partitioning graphs for the balancing assembly lines with supply constraints [7][8].

## 2.5. Genetic algorithm for GPM

The developed genetic algorithm respects the structure of the standard genetic algorithm through the presence of the three modules: the initial population module, the evaluation module and the recombination and modification module [9][10][11].

**2.5.1. The fitness function.** The fitness function determines the number of subgraphs, the number of cuts and the class membership of the genes of a chromosome. The following Greedy method was used: on a generated permutation, the procedure with the first gene of the chromosome is started by calculating the number of primary entries and pseudo-entries of the gene sequence in the order of position. When the restriction is violated, increase the number of subcircuits, assign the current partition vertices, and continue the procedure.

Let graph from figure 2.



**Figure 2.** Graph example for the evaluation of the fitness function

For  $L=3$  and for the generated solution,  $\text{sol}[1]=[14, 11, 4, 1, 9, 10, 3, 13, 0, 6, 5, 8, 2, 7]$

The value of the fitness function is 2 (table 1).

**Table 1.** The fitness function for the example shown in figure 2

Group	Vertices													Inputs	Cuts
1	14	11	4	1	9	10	12	3	13	0	6	5	8	3	0
2	2	7											2	1	

**2.5.2. Population management.** At the end of each stage of evolution all new chromosomes create a separate list. This list is interclassed but the initial population and the best chromosomes are retained for the next stage of evolution in the limit set for the population. This elitist selection focuses the search on the most promising regions in the solution space.

**2.5.3. Genetic operators.** The role of these operators is to introduce new individuals in the population, thus resulting in a search in the entire space of solutions.

*The mutation operator* - it prevents premature convergence, generating an additional diversity of individuals in a population. This diversity allows exploration of larger areas in the search space.

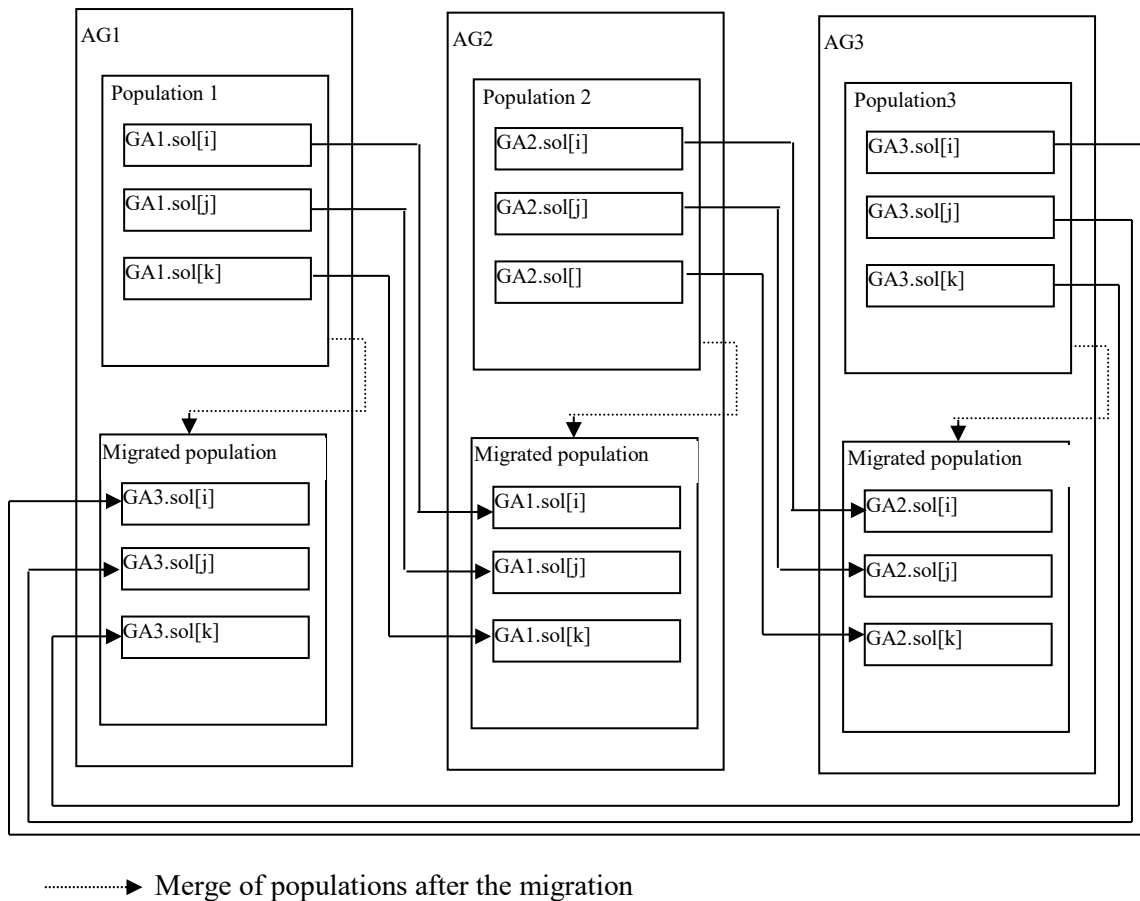
A mirroring mutation operator with a self-adaptive adjustment was used: the cutting edges are randomly generated according to the histogram of the parent chromosome - the number of elements in each class. Thus, the number of elements between the two cutting fronts is greater than the maximum number of elements in a partition. This self-adaptation of the mutation operator will avoid the generation of offspring identical to the parent chromosomes.

The crossover operator - is responsible for increasing diversity in a given population, combining what is good in the population and proliferating the most promising sequences of peaks. The offspring obtained by crossing will have the characteristics of both parents. This is a variant of the OX operator [12][13] adapted to GPM:

- 2 random positions are generated by extracting 2 chromosomes;
- 2 cutting fronts are generated for each parent chromosome so that the restriction is observed:, where it represents the maximum number of elements in a class;
- move by a circular permutation all the elements between the 2 fronts at the beginning of each descendant and fill in the remaining positions with unassigned tips from the other parent.

## 2.6. Insular genetic algorithm (IGA) for GPM

2.6.1. The structure of the insular genetic algorithm. During each stage of evolution of the IGA, each of the three GA contains mutation and crossing operations on the island of solutions given by the initial population, locally improving the value of its objective function. After the elitist selection is applied on each solution island of the three GAs, the migration operation is performed (figure 3). The policy is to make a circular infusion of solutions from one island to another in order to avoid premature achievement of a local optimum of the value of the estimation function.



**Figure 3.** A stage of evolution of IGA

This operation will result in a list of individuals (GPM solutions) copies of the individuals selected for migration.

2.6.2. *The migration operator.* Steps of the migration procedure of individuals from:

- random numbers are generated between 0 and 1, where it represents the size of the population set by the court of the problem;
- for each random number with the property where (migration probability) - parameter provided by the problem instance, one is generated and it will indicate the individual that will migrate;

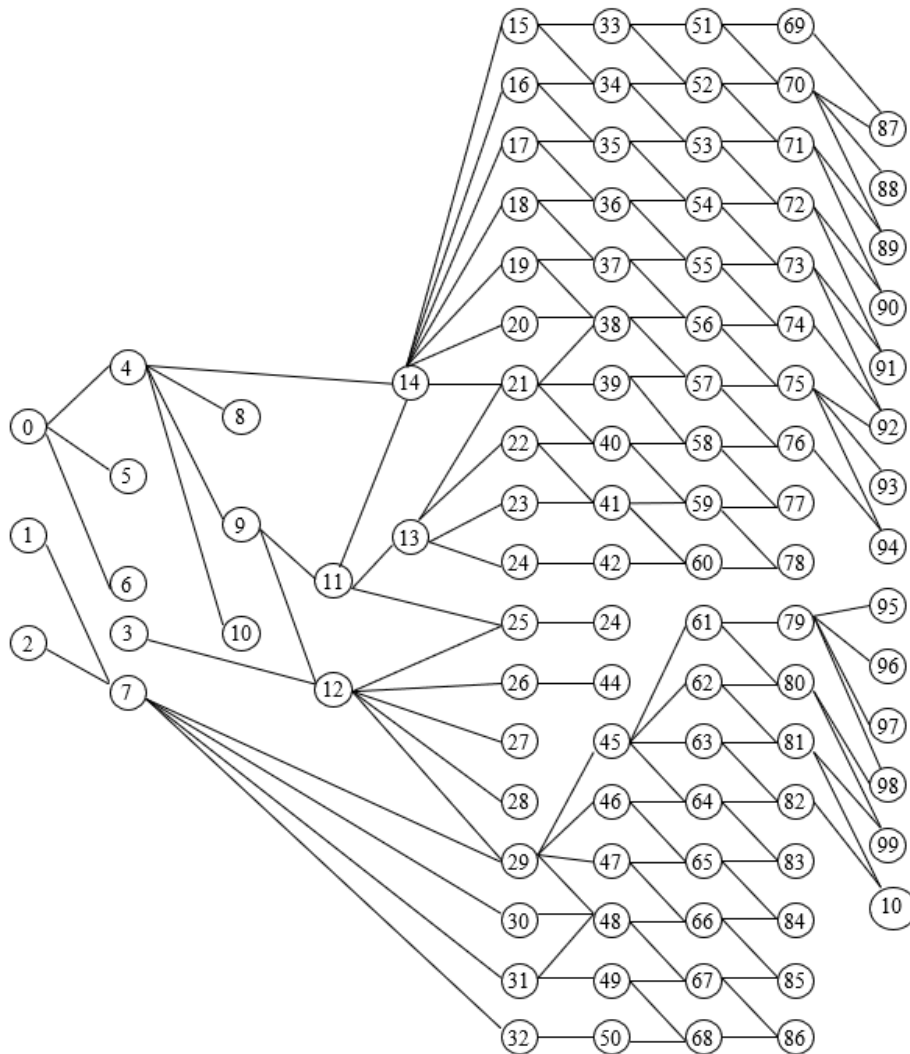
2.6.3. *Master program function - global control.* The evolution of the entire IGA is controlled by the master program function by:

- establishing the probabilities of mutation, crossing and migration;
- synchronization of processes after each stage of evolution;
- global stop criterion:

The global stop criterion. For each GA at the end of each stage of evolution, the average value of the fitness function of the individuals in the population marked with is calculated. The evolution of the process stops when the relationship becomes true for each of the GA, where there is a prescribed tolerance.

### 3. Experimental results

Let be a graph with 100 vertices, 4 primary inputs and  $L=3$ . The graph is shown in the figure 4.



**Figure 4.** Graph for the studied technological process

In table 2 is presented the adjacent list for the problem graph, for each of the vertex the number of predecessors and their list are generated.

**Table 2.** Adjacent list for the graph

Vertex	No	List of predecessors	Vertex	No	List of predecessors	Vertex	No	List of predecessors
0	0		36	2	18 17	72	2	53 54
1	0		37	2	18 19	73	2	54 55
2	0		38	3	19 20 21	74	2	55 56
3	0		39	1	21	75	2	56 57
4	1	0	40	2	21 22	76	2	57 58
5	1	0	41	2	22 23	77	2	58 59
6	1	0	42	1	24	78	2	59 60
7	2	1 2	43	1	25	79	1	61
8	1	4	44	1	26	80	2	61 62
9	1	4	45	1	29	81	2	62 63
10	1	4	46	1	29	82	2	63 64
11	1	9	47	1	29	83	2	64 65
12	2	3 9	48	3	29 30 31	84	2	65 66
13	1	11	49	1	31	85	2	66 67
14	2	4 11	50	1	32	86	2	67 68
15	1	14	51	1	33	87	2	69 70
16	1	14	52	2	33 34	88	1	70
17	1	14	53	2	34 35	89	2	70 71
18	1	14	54	2	35 36	90	2	71 72
19	1	14	55	2	36 37	91	2	72 73
20	1	14	56	2	37 38	92	3	73 74 75
21	2	13 14	57	2	38 39	93	1	75
22	1	13	58	2	39 40	94	2	75 76
23	1	13	59	2	40 41	95	1	79
24	1	13	60	2	41 42	96	1	79
25	2	11 12	61	1	45	97	1	79
26	1	12	62	1	45	98	2	79 80
27	1	12	63	1	45	99	2	80 81
28	1	12	64	2	45 46	100	2	81 82
29	2	7 12	65	2	46 47			
30	1	7	66	2	47 48			
31	1	7	67	2	48 49			
32	1	7	68	2	49 50			
33	1	15	69	1	51			
34	2	15 16	70	2	51 52			
35	2	16 17	71	2	52 53			
36	2	18 17	72	2	53 54			

Population size was set at 100 individuals and tolerance at  $2 * 10^{-7}$ . For each instance of the problem (including probability values) a group of 10 runs of the GA as performed. The following characteristics were calculated for each group:

- BGP - the best graph partitioning
- ABGP - the average of the best partitions of the graph
- LABGP - the average of the best partitions of the graph in the last stage of evolution;

- ANI - the average number of iterations made, until the stop condition is verified;
- ANC - average number of cuts

We will consider the following values for the probabilities of crossing, mutation and for the migration operator:  $pM = 0.2, 0.4$ ;  $pC = 0.4, 0.6$ ,  $pMi = 0.0, 0.2$ .

The value 0.0 for means that the migration operator was not considered and GPM is solved independently of the three GA.

The results for each of the groups of 10 runs considering all combinations of probabilities were centralized in the following table 3:

**Table 3.** Results of IGA

No	pM	pC	pMi	GA1					GA2					GA3				
				BGP	ABGP	LABGP	ANI	ANC	BGP	ABGP	LABGP	ANI	ANC	BGP	ABGP	LABGP	ANI	ANC
1	0.2	0.4	0.0	5	5.7	6.2	70	12	6	6	6	70	14	5	5	6	70	12
2	0.2	0.6	0.0	5	5.5	6.1	85	11	5	5.7	5.6	85	12	4	5.5	6	85	11
3	0.4	0.4	0.0	4	5	5.5	95	9	3	3.5	3.5	95	8	5	5	5.5	95	9
4	0.4	0.6	0.0	3	4	4	107	5	3	3	3.7	107	5	2	3	3	107	4
5	0.2	0.4	0.2	3	3.4	3.4	98	5	3	3	3.3	98	5	3	3.4	3.4	98	5
6	0.2	0.6	0.2	2	3	3	95	4	2	2	3	95	4	2	3	3	95	4
7	0.4	0.4	0.2	2	2	2	97	1.5	2	2	2	97	2	2	2	2	97	1
8	0.4	0.6	0.2	2	2	2	102	1	2	2	2	102	1	2	2	2	102	1

The role of mutation and crossover operators is observed in the evolution of the number of partitions of the graph, which decreases with increasing probabilities of mutation and crossover.

The introduction of the migration operator has the following effects:

- standardization of the solutions provided by the three genetic algorithms;
- improving the solution, Gr. no. 8 providing the optimal solution to the problem: 2 partitions and a single cutting operation.
- the number of iterations does not show a significant increase compared to the execution groups when the other probabilities remain constant.

#### 4. Conclusions and future directions of investigations

As a conclusion, the implementation of a genetic algorithm for GPM using the island model with migration generates solutions of a quality close to the optimal ones, with a shorter execution time than the exact and pseudo-exact methods.

Future directions of investigation will be in hybridizing the algorithm, implementing and executing it on a distributed system using the MPI function library and performing a statistical analysis of the results obtained after running the program on a graph with a number of vertices of thousands.

#### References

- [1] Vîlcu Adrian 2012 *A hybrid genetic algorithm for a combinational circuits partitioning problem* Buletinul Institutului Politehnic din Iași Automatic Control and Computer Science Section Volumul 3 Tome LVIII (LXII) fasc.3 p 43-51
- [2] Vîlcu Adrian 2013 *Double hybridized embryonic genetic algorithm for optimal delivery routing* Scientific Studies and Research Series Mathematics and Informatics Bacau Vol. 23 No.1 p 213

- [3] Adrian Vilcu, Ion Verzea, Marius Pislaru & Ionut-Viorel Herghiligiu 2019 *Fuzzy modeling of dependability optimization for supporting the production-quality strategies - case study in technical field* ModTech International Conference Modern Technologies in Industrial Engineering June 19-22 2019 Iasi Romania
- [4] Marius Pislaru, Ramona-Diana Leon, Vilcu Adrian 2018 *Using a fuzzy expert system for service quality improvement. the case of a car wash station* STRATEGICA International Academic Conference -Sixth Edition-Bucharest, Romania, October 11-12, 2018 Challenging the Status Quo in Management and Economics p 490-500
- [5] Vilcu Adrian, Ion Verzea, Catalin Vilcu 2016 *Heuristic algorithm for optimization of manufacturing process by balancing a production line from a textile company*, 16th Romanian Textiles and Leather Conference, CORTEP, Iasi, 27-29.10
- [6] Ion Verzea, Gabriel-Petru Luca, Rachid Chaib, Vilcu Adrian 2016 *Conventional work units breakeven point assessment in the garment Industrie*, 16th Romanian Textiles and Leather Conference–CORTEP, Iasi, 27-29.10.2016
- [7] Labontu Oana 2014 *Constructive aesthetic design of a blouse model for women and optimization of the manufacturing process through balancing* bachelor's thesis DIMA Faculty, Iasi
- [8] Verzea I and Luca GP 2009 *Production and maintenance management methods*
- [9] Aminur Rahman Khan, Adrian Vilcu, Nahid Sultana, Syed Sabbir Ahmed 2015 *Determination of Initial Basic Feasible Solution of a Transportation Problem: A TOCM-SUM Approach* Buletinul Institutului Politehnic din Iași Technical University of Iasi, Tomul LVIII (LXII) Fasc. 1 Automatic Control and Computer Science Section
- [10] Adrian Vilcu 2012 *A hybrid genetic algorithm for a combinational circuits partitioning problem*, Buletinul Institutului POLITEHNIC Technical University of Iasi Tomul LVIII (LXII) Fasc. 3 Automatic Control and Computer Science Section
- [11] M Davis-Moradkhan 1993 *Problemes de partitionnement dans la Technologie des VLSI* These de Doctorat Universite Paris VI
- [12] D Dumitrescu 2000 *Genetic algorithms and evolutionary strategies - applications in artificial intelligence and related fields* Ed. Albastră Cluj Napoca
- [13] Michalewicz Z 1994 *Genetic Algorithms + Data Structures = Evolution Program*, 2nd ed. Artificial Intelligence Springer Verlag Berlin