Journal of Theoretical and Applied Information Technology
30th April 2017. Vol.95. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NEW CROSSOVER OPERATOR FOR GENETIC ALGORITHM TO RESOLVE THE FIXED CHARGE TRANSPORTATION PROBLEM

[1]AHMED LAHJOUJI EL IDRISSI, [2]CHAKIR TAJANI, [3]MOHAMMED SABBANE

[1,3]Laboratory Modeling, Analysis, and Control of Systems (MACS), Faculty of Science Meknes, Morocco

[2]Department of Mathematics, Polydisciplinary faculty of Larache, Morocco

[1]idrissila@gmail.com, [2]chakir_tajani@hotmail.fr, [3]m.sabbane@umi.ac.ma

ABSTRACT

Genetic algorithms (GAs) are a class of global optimization methods. It has been used to solve combinatorial problems. Among the difficulties in GAs, the parameter setting and the choice of the crossover operator adapted to the problem. In this paper, we studied the influence of these operators on the performance of the GAs by making a comparative study with different adapted operators to the Fixed Charge Transportation Problem (FCTP) and described the genetic algorithm to find an optimal solution. In addition, we proposed a new crossover operator for solving the FCTP. The experimental results show that the choice of adequate crossover is important to solve each combinatorial problem by genetic algorithm. Moreover, the GA with our developed crossover operator is more efficient.

Keywords: *Combinatorial Problem, Fixed Charge Transportation Problem, Genetic Algorithm, Crossover Operator, Priority Based Encoding.*

1. INTRODUCTION

The fixed charge transportation problem (FCTP) is a problem of optimization and operations research; it is a generalization of the classical transportation problem [1]. It contains two costs, variable costs proportional to the amount shipped and fixed cost regardless of the quantity transported.

Many problems in the real world can be modeled as a FCTP such as the problem of allocation of launch vehicles for space missions [2], the allocation of teaching duties for teachers to minimize in the average number of distinct subjects assigned to each teacher [3] and goods distribution problem which is treated with Adlakha [4].

FCTP is classified as NP-complete problem which is difficult to resolve by exact methods [5], because of the fixed costs which are reflected by discontinuities in the objective function [6] as well as the computation time required to find a solution which is likely to increase exponentially with the size of the problem. However, the approximate methods and meta-heuristics can be used [7].

The GAs techniques are very powerful and widely applicable in the field of optimization [8]. It offers a population of chromosomes, each of which is a feasible coded solution. The values in standard are assigned to each chromosome, and the population is changing by a set of operators (crossover, mutation) until a stop criterion is reached [9]. A selection mechanism will be applied after the evaluation to choose the best chromosomes of the population with a high probability of choice [10].

To apply the GA approach, it is important to choose an adequate representation of chromosomes for the problem. Several representation methods are used to solve the FCTP. Z. Michalewicz and al. [11] proposed a matrix representation to solve the problem of distribution of goods where the chromosome was represented with a matrix $m \times n$ with $m + n - 1$ positive elements. The Prüfer number representation introduced by M. Gen and al. [12] can be used to solve different network problems; M. Hajiaghaei-Keshteli and al. [13] proposed a method to choose the chromosomes not need a repair procedure for feasibility with all the produced chromosomes are feasible. M. Gen and al. [14] have developed a new process of encoding and decoding for two-stages transportation problem called "Priority based Representation method" (pb-GA) which has several advantages and has been used successfully in several problems [15].

In this work, we consider the resolution of the FCTP by genetic algorithm using Priority based representation method "pb-GA" where we analyze and compare the performance of GA with four different suitable crossover operators to obtain optimal solution and to show the influence of these operators in the resolution of this kind of problem and to conclude the most adapted crossover operator for FCTP. After providing the comparative analysis, we propose a new crossover operator in order to improve the performance of the GA applied to FCTP and obtain best solution.

The second section is devoted to the presentation of the mathematical formulation of the FCTP and an introduction to the problem with examples and graphs to explain the transport flows. The Section 3 presents a brief discussion of adaptive genetic algorithm to solve the FCTP and all parameters that can influence the performance of the GAs (encoding, selection, crossover and mutation ...). Particularly, the crossover operator which is the most important operation in the genetic algorithm process. Hence; we present four types of suitable crossover used with the pb-GA. Moreover, we propose a new crossover operator that we called Inversion Position-based Crossover "IPX". In section 4, we present numerical results where a comparative study of the four presented crossover operators (OPEX, OX, PX and PEX) with our operator has developed based on different standard existing instances in the literature to conclude the most performing operator for FCTP. The results show that the new proposed operator improves the performance of the GA.

2. PROBLEM DESCRIPTION

The Fixed Charge Transportation Problem (FCTP) is a particular case of the transportation problem, in which a variable cost that is proportional to the quantity shipped, with a fixed cost independent of the quantity transported. The goal is to find the combination of flow that minimizes the total variable and fixed costs while satisfying the demands of each origin and destination (Figure 1).

We have a group of warehouses D_j (destination) $j=1, 2,...,n$ that is served by a group of production sites S_i (sources) $i=1,2,...,m$ while each producer has a given production capacity and each destination S_i a request to meet D_j. The problem is to determine the amount of product to be sent from each place of production for each warehouse to minimize the total cost, fixed and variable to serve all destinations. To obtain the formulation of this problem, it is necessary to add the fixed cost in the modeling of the linear transport problem. In this case, the transport costs have a variable part $c_{ij}\, x_{ij}$ and a fixed part $f_{ij}\, y_{ij}$.

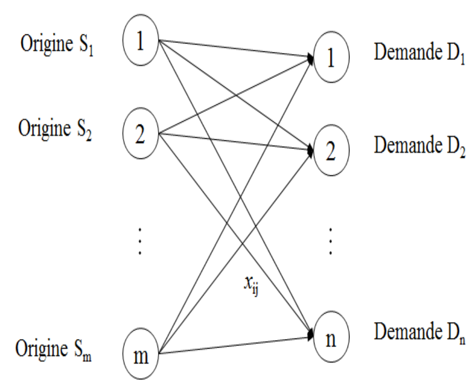


Figure 1: Example of transportation plan

The fixed charge transportation problem can be formulated as follows:

$$Min\,Z = \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij} + f_{ij}y_{ij}) \qquad (1)$$

$$y_{ij} = \begin{cases} 1, & x_{ij} > 0 \\ 0, & x_{ij} = 0 \end{cases} \qquad (2)$$

$$\sum_{j=1}^{n} x_{ij} \le S_i \qquad i = 1,2,\ldots,m \qquad (3)$$

$$\sum_{i=1}^{m} x_{ij} \ge D_j \qquad j = 1,2,\ldots,n \qquad (4)$$

$$x_{ij} \ge 0 \qquad (5)$$

$$i = 1,2,\ldots,m\,;\ j = 1,2,\ldots,n \qquad (6)$$

c_{ij} : variable cost from source i to destination j;
x_{ij} : quantity transported on the route (i,j);
f_{ij} : fixed cost associated with route (i,j);
y_{ij} : a binary variable $y_{ij}=1$ if $x_{ij}>0$ and 0 if $x_{ij}=0$;
S_i : amount of supply at source i;
D_j : amount of demand at destination j;

It is preferable to consider the balanced problem (S_i=D_j) that is to say the availability equal requests.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. GENETIC ALGORITHM FOR FCTP

Genetic algorithms (GAs) are stochastic optimization algorithms based on the mechanisms of natural selection and genetics developed by John Holland in 1975 [10]. Their fields of application are extensive; especially in the areas of economy, finance, logistics and transport ... Usually, they can be used to the NP-complete problem, for example, in the backpack problem [16] and in the traveling salesman problem [17]…

This kind of Algorithm starts with initial generation of population, the second step is to apply the evaluation process and to use the genetic operators (selection, crossover, mutation…) to reproduce new population. Moreover, the selection is used according to the adopted method to select the best solutions [18].

In GAs, one of the most important processes is the encoding of chromosomes. Therefore, it is significant to choose the adequate representation of chromosomes for this problem. In this work, we have chosen the priority based representation for FCTP [19].

3.1 Proiority based representation for FCTP

For this representation, the population is generated randomly from a series of numbers representing the transport network nodes where all the solutions are feasible even also after the crossover and mutation operations [20]. Also, in this encoding, a gene in a chromosome contains two types of information, the position of a gene to represent the nodes (source / destination) and the value of a gene, which represents the priority of the node for the construction of a transport plan. A chromosome consists of priorities of sources and destination to obtain a transport plan. Its length is equal to the total number of sources (m) and deposits (n), that is $m+n$.

From the code and the matrices costs, variable and fixed, we can construct the transportation plan using the corresponding algorithm (Figure3).

We use the formulation of M. L. Balinski [21] given by:

$$UC_{ij} = C_{ij} + \frac{f_{ij}}{min(S_i, D_j)}$$

in step 3 of the decoding algorithm (Figure 3) to calculate the cost of transport, so that, it can be solved as a problem of linear transport, instead of working by two tables, we are working by a single table.

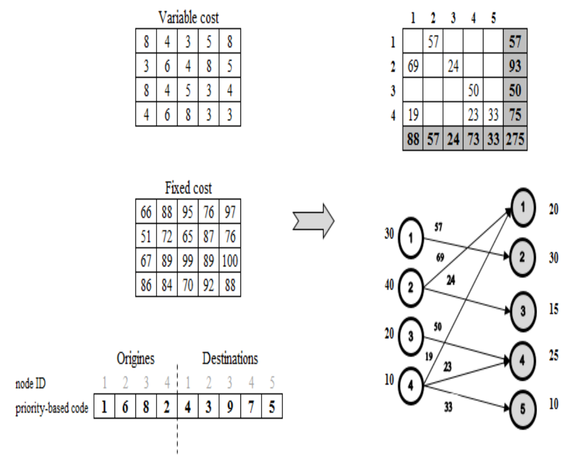


Figure 2. pb-GA for the 4x5 instance

Input :
m : number of sources, n : number of depots ;
S_i : supply of source i, i=1,2,…,m;
D_j: demand of depot j, j=1,2,…,n;
C_{ij} : variable transportation cost of one unit of product from source i to depot j ;
f_{ij} : fixed transportation cost associated with route (i,j) ;
v(i+j): chromosome;
***Step 1**: $X_{ij} \leftarrow 0$, for each i,j;*
***Step 2**: $k \leftarrow argmax\{v(t)|t=1,2,…,m+n\}$; select a node*
***Step 3**: if $k \le m$, then $i^* \leftarrow k$; select a source $j^* \leftarrow arg\ min\{UCij = C_{ij} + f_{ij}/min(S_i/D_j)\ |v(m+j) \neq 0, j=1,2,…,n\}$; select a depot with lowest cost*
else $j^* \leftarrow k-m$; select a depot; $i^* \leftarrow arg\ min\{UCij = C_{ij} + \ |v(i) \neq 0, i=1,2,…,m\}$; select a source with lowest cost
***Step 4**: $x_{i^*j^*} \leftarrow min\{S_{i^*}, D_{j^*}\}$; $S_{i^*} \leftarrow S_{i^*} - x_{i^*j}$, $D_{j^*} \leftarrow D_{j^*} - x_{i^*j}$, assign available units and update availabilities of source i^* and depot j^**
***Step 5**: if $S_{i^*}=0$ then $v(i^*) \leftarrow 0$; if $D_{j^*}=0$ the $v(m+j^*) \leftarrow 0$; remove priority of selected source or depot*
***Step 6:** if $\exists i | i \le m, v(i) \neq 0$ go to step2, else calculate the total transportation cost.*
***Ouput :** the amount of transported product from source i to depot j ;*

Figure 3. Decoding algorithm of the pb-GA for the FCTP

3.2 Initialization

The first step in the performed genetic algorithm consists to use the pb-GA to produce a initial population. We can generate a population from a number p of chromosomes, and the initialization is performed by generating a random permutation of the elements of 1 to $l = m + n$ for each chromosome. In the representation there is no need of a correction algorithm, a chromosome consists sources priorities and deposits for a transportation tree.

Journal of Theoretical and Applied Information Technology
30th April 2017. Vol.95. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

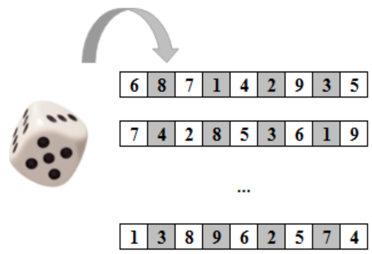


Figure 4. Initialization of the population randomly

3.3 Crossover operators

The crossover is the most important genetic operator. It works with both parents and combination of different characteristics of the two chromosomes. It allows the reproduction of a new population. There are several operators used to the problem. In this context, we have chosen four adapted crossover operator for the FCTP in order to compare their performance and conclude the best choice of crossover for FCTP. In addition, we developed a new crossover operator to improve the performance of the genetic algorithm comparing with the four cited operators.

3.3.2 Oder of priority exchange crossover (OPEX):

OPEX crossover operator consists in selecting a random crossover point. The child inherits the parent left chromosome segment and exchange the order of priority of the parents of nodes for the right of children segment [22]. The procedure illustrated in *Figure 6*. It works as follows:

Input : *two parents;*
Step 1 : *Select a cut point;*
Step 2 : *The offspring's inherit the left part of the parents;*
Step 3 : *Sort the right segment nodes based on the priority;*
Step 4 : *Exchange the priority of the nodes between the two chromosomes.*
Output : *two offspring*

Figure 5. Procedure of the OPEX crossover

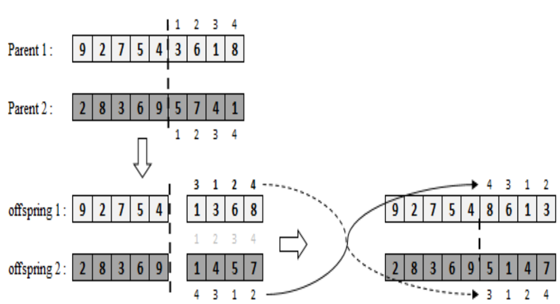


Figure 6. Example of the OPEX crossover

3.3.3 Partial-Maped Crossover (PMX)

PMX used a special procedure for repairing to solve the illegitimacy caused by two simple crossover points. Therefore, most of PMX is the two simple points, more than the crossover repair method [23]. The procedure illustrated in *Figure 8*. It works as follows:

Input*: two parents*
Step 1*: Select two positions along the string uniformly at random.*
The substrings defined by the two positions are called the mapping sections.
Step 2*: Exchange two substrings between parents to produce proto-children.*
Step 3*: Determine the mapping relationship between two mapping sections.*
Step 4: *Legalize offspring with the mapping relationship.*
Output: *two offspring*

Figure 7. Procedure of the PMX crossover

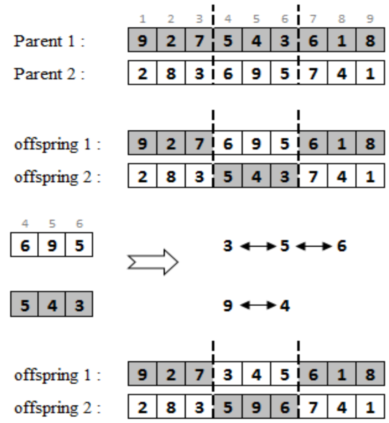


Figure 8. Example of the PMX crossover

3.3.4 Order Crossover (OX)

For this operator, two points are randomly selected from a parent, thus defining a crossover region. This region transmitted directly to the offspring; meanwhile, the remaining positions filled with the elements that do no belong to that region in the order that they appear in the second parent [24]. The procedure illustrated in *Figure 10*. It works as follows:

Input*: two parents*
Step 1*: Select a substring from one parent at random.*
Step 2*: Produce a proto-child by copying the substring into the corresponding positions of it.*
Step 3*: Delete the nodes which are already in the substring from the second parent.*
The resulted sequence of nodes contains the nodes that the proto-child needs.
Step 4*: Place the nodes into the unfixed positions of the proto-child from left to right according*
Output*: two offspring*

Figure 9. Procedure of the OX crossover

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

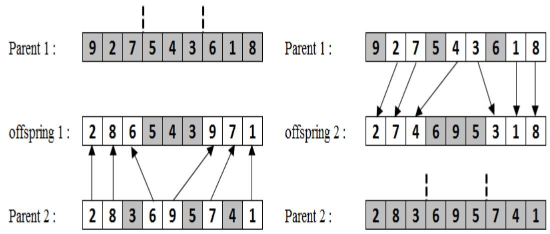


Figure 10. Example of the OX crossover

3.3.5 Position-based Crossover (PX)

PX crossover operator is essentially a kind of permutation cross with a repair procedure. It is closer to the operator OX where nodes are selected in a random manner [25]. The procedure illustrated in *Figure 12*. It works as follows:

> ***Input:*** *two parents;*
> ***Step 1:*** *Select a set of positions from one parent at random;*
> ***Step 2****: Produce a proto-child by copying the nodes on these positions into the corresponding positions of it;*
> ***Step 3:*** *Delete the nodes which are already selected from the second parent;*
> *The resulted sequence of nodes contains the nodes the proto-child needs;*
> ***Step 4:*** *Place the nodes into the unfixed positions of the proto-child from left to right according to the order of the sequence to produce one offspring;*
> ***Output:*** *two offspring.*

Figure 11. Procedure of the PX crossover

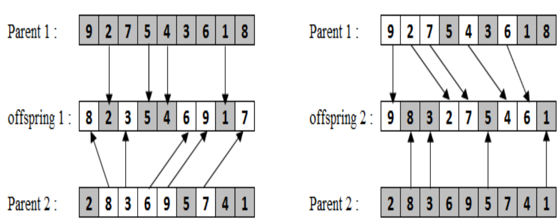


Figure 12. Example of the PX crossover

3.3.6 Proposed Inversion Position-based Crossover (IPX)

In order to improve the performance of the genetic algorithm and find a better solution than those found by the operators already used, we proposed a new crossover operator that we called "Inversion Position-based crossover" (IPX). The principle is based on the crossover operator PX, the difference consist to invert the principle of PX operator. The procedure illustrated in *Figure 14.* works as follows:

> ***Input:*** *two parents;*
> ***Step 1:*** *Select a set of positions from one parent at random;*
> ***Step 2****: Produce a proto-child by copying the nodes on these positions into the corresponding positions of it;*
> ***Step 3:*** *Delete the nodes which are already selected from the second parent;*
> ***Step 4:*** *Place the nodes into the unfixed positions of the proto-child from right to left according to the order of the sequence to produce one offspring;*
> ***Output:*** *two offspring.*

Figure 13. Procedure of the IPX crossover

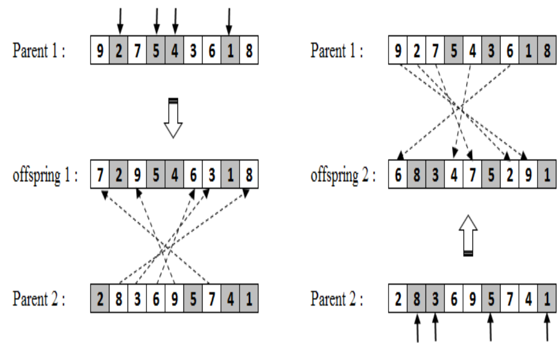


Figure 14. Example of the IPX crossover

3.1. Mutation operator

Mutation operator operates by exchanging information within a chromosome. However, instead of using this operator between two parents, we use between two segments of a parent. We have choosen a swap mutation operator which consist to permute the values of two randomly selected positions for each chromosome to reduce the risk of reproducing a chromosome with the same solution [11, 26]. Swap mutation consists to selects two elements and then randomly permutes the elements at these positions. The procedure illustrated in *Figure 16*. It works as follows:

> ***Input :*** *One paren;t*
> ***Step 1 :*** *Select two element at random;*
> ***Step 2 :*** *Swap the element on these positions;*
> ***Output :*** *One offspring.*

Figure 15. Procedure of the SWAP mutation

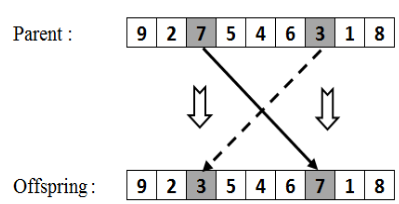


Figure 16. Example of the SWAP mutation operator

3.1. Evaluation and selection :

It is a process to evaluate a solution and compare it to the other in order to choose the best solutions. Hence; the parents selected according to their performance. The evaluation function will allow to select or to refuse an individual to retain only those individuals having the best cost as a function of the current population.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For the selection method, we used the method of selection by roulette strategy [17]. This method supplemented by the solutions resulting from operations of crossover and mutation, to ensuring that the population size remains fixed from one generation to another.

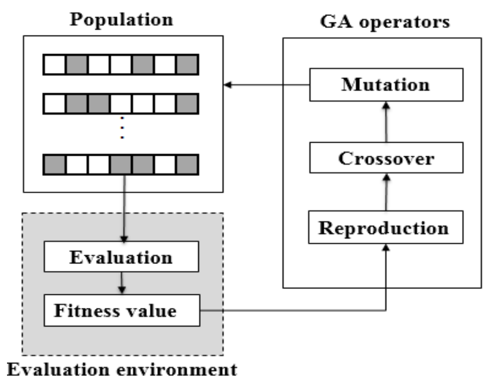


Figure 17. Principle of evaluation and selection process

3.1. Procedure of genetic algorithm :

For the proceeding of our algorithm, it takes as input the genetic algorithm parameter, and transport data to extract output as the optimal plan of transport. *Figure 19.* shows graphically the global of the proposed process problem-GA for FCTP.

Input *: data transport, GA parameter*
t ← 1;
Initialization of P(t) by priority-based representation;
Evaluation of P(t);
do (condition not finish) While
Crossover P(t) by IPX
Mutate P(t) by SWAP mutation to yield C(t);
Evaluation C(t) ;
Selecting P(t + 1) from P(t) and C(t) by roulette selection;
t ← t + 1;
Output *: Minimum total cost of transportation*

Figure 18. The proposed GA procedure for FCTP

We know that each genetic operator is applied to the population by a probability. Thus, for the crossover the probability must be chosen ($P_c > 0.5$) for the population to progress. However; a mutation probability must be small enough that it does not have a purpose to bring a great change to individuals; but it is a kind of small changes on a reduced number of individual to give opportunity to the neighboring of the current population to be introduced to the new generation.
Thus, the parameters adopted for our examples are as follows:

- ✓ Crossover rate $P_c = 0.6$;
- ✓ Mutation rate $Pm = 0.2$;
- ✓ Population size, $pop = 20$;
- ✓ Maximum generation $max\_gen = 1000$.

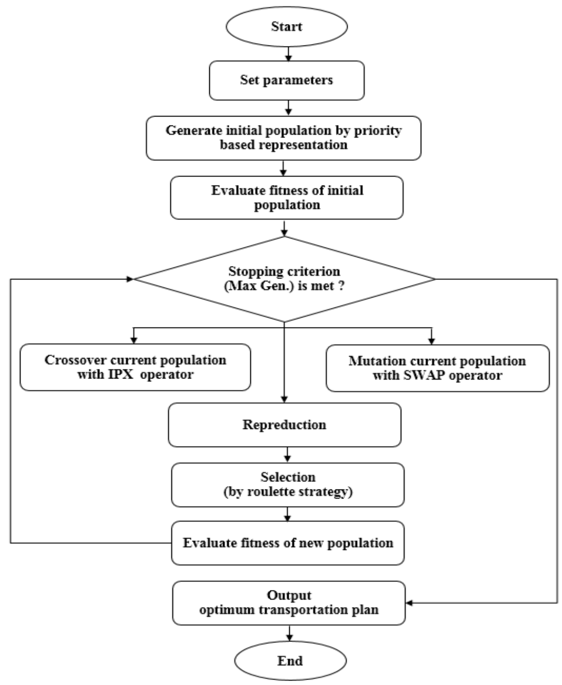


Figure 19. Process of the proposed GA for FCTP

4. RESULTS AND ANALYSIS

The fixed charge transportation problem (FCTP) is one of the most famous problems in the field of optimization and combinatorial problems. Experiments carried out to compare the performance of genetic algorithms with the crossover operators of adaptive dipped in our problem. We have proposed a new crossover operator adapted for combinatorial problems in general, and we have applied this operator on FCTP problem. The numerical results show a comparison between our operators and several operators used for FCTP. The operators of crossover are the following OPEX, PMX, OX, PX and our proposed operator IPX. In addition, we chose these operators since their use especially for FCTP.

To test, we have chosen five instances (4x5, 5x10, 10x10, 10x20) cited bellow, and 30x50 already used in several articles [13, 19] without taking into account the multiplication by 10. For the optimal solution for small instances is known for small instances.

The programming is done in JAVA (NetBeans IDE 8.0.2) on a PC machine with Intel (R) Core (TM) i5-2400 3.10 GHz in CPU and 4GB of RAM and Windows as an operating system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: The cost tables for 4x5 instance

variable cost					fixed cost				
8	4	3	5	8	60	88	95	76	97
3	6	4	8	5	51	72	65	87	76
8	4	5	3	4	67	89	99	89	100
4	6	8	3	3	86	84	70	92	88

Table 2: The optimal solution for 4x5 instance

57		57			
93	69		24		
50				50	
75	19			23	33

Table 3: The cost tables for 5x10 instance

variable cost										fixed cost									
8	4	3	5	2	1	3	5	2	6	160	488	295	376	297	360	199	292	481	162
3	3	4	8	5	3	5	1	4	5	451	172	265	487	176	260	280	300	354	201
7	4	5	3	4	2	4	3	7	3	167	250	499	189	340	216	177	495	170	414
1	2	8	1	3	1	4	6	8	2	386	184	370	292	188	206	340	205	465	273
4	5	6	3	3	4	2	1	2	1	156	244	460	382	270	180	235	355	276	190

Table 4: The optimal solution for 5x10 instance

157	0	0	0	0	130	0	27	0	0	0
293	0	79	90	0	0	0	0	124	0	0
150	0	0	0	32	0	88	30	0	0	0
575	225	71	0	183	0	0	0	0	0	96
310	0	0	0	0	0	0	0	0	273	37
	225	**150**	**90**	**215**	**130**	**88**	**57**	**124**	**273**	**133**

Table 5: The cost tables for 10x10 instance

variable cost										fixed cost									
25	14	34	46	45	48	11	26	45	16	85	61	62	90	78	89	79	74	71	94
10	47	14	20	41	37	42	39	13	15	87	92	63	54	90	97	77	87	61	88
22	42	38	21	46	12	38	28	31	20	78	55	55	63	94	71	82	79	87	69
36	20	41	38	44	10	37	47	12	31	89	71	83	87	93	73	56	52	85	74
34	33	30	14	34	32	41	19	39	33	67	82	85	67	59	68	87	70	56	57
37	43	29	29	33	24	43	22	50	41	86	60	82	85	74	84	95	62	93	99
21	42	18	28	26	47	14	17	27	16	74	90	70	97	99	60	98	53	79	82
44	32	19	39	37	41	17	39	48	34	86	99	83	74	52	58	81	88	55	93
26	40	14	38	43	18	36	38	43	26	58	86	97	95	80	60	69	54	67	91
15	46	50	43	28	18	29	26	24	42	52	69	84	62	89	71	87	88	72	98

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6: The optimal solution for 10x10 instance

30	0	14	0	0	0	0	16	0	0
17	0	0	0	2	0	0	0	0	0
27	0	0	0	6	0	21	0	0	0
29	0	3	0	0	0	0	0	0	26
20	0	0	0	20	0	0	0	0	0
18	0	0	0	0	10	0	0	8	0
11	0	0	0	0	0	0	0	11	0
15	0	0	0	0	10	0	5	0	0
14	0	0	13	0	0	0	0	1	0
23	23	0	0	0	0	0	0	0	0
	23	**17**	**13**	**28**	**20**	**21**	**21**	**20**	**26**

Table 7: The variable cost for instance (10x20)

Variable cost																			
25	14	34	46	45	48	11	26	45	16	20	12	11	17	19	11	21	24	32	24
10	47	14	20	41	37	42	39	13	15	40	35	17	26	32	38	32	17	49	37
22	42	38	21	46	12	38	28	31	20	22	42	42	37	40	48	35	39	49	25
36	20	41	38	44	10	37	47	12	31	46	48	34	32	49	29	20	43	30	44
34	33	30	14	34	32	41	19	39	33	32	42	49	14	14	38	45	18	11	17
37	43	29	29	33	24	43	22	50	41	39	17	42	47	19	25	14	46	44	33
21	42	18	28	26	47	14	17	27	16	13	11	39	25	36	11	45	11	22	17
44	32	19	39	37	41	17	39	48	34	24	34	49	10	30	28	44	46	48	11
26	40	14	38	43	18	36	38	43	26	34	24	38	48	27	16	17	41	48	21
15	46	50	43	28	18	29	26	24	42	26	35	41	34	12	30	40	27	12	12

Table 8: The fixed cost for instance (10x20)

Fixed cost																			
85	61	62	90	78	89	79	74	71	94	58	97	70	58	64	100	73	52	68	79
87	92	63	54	90	97	77	87	61	88	67	74	95	87	70	64	89	100	78	65
78	55	55	63	94	71	82	79	87	69	71	89	82	100	77	53	58	77	99	96
89	71	83	87	93	73	56	52	85	74	73	62	55	59	77	92	51	52	51	97
67	82	85	67	59	68	87	70	56	57	84	89	58	86	95	93	95	73	72	59
86	60	82	85	74	84	95	62	93	99	92	96	84	83	79	55	91	96	93	63
74	90	70	97	99	60	98	53	79	82	98	63	57	66	67	85	63	79	56	81
86	99	83	74	52	58	81	88	55	93	95	85	88	95	99	78	60	65	76	52
58	86	97	95	80	60	69	54	67	91	75	60	74	93	55	65	71	55	88	53
52	69	84	62	89	71	87	88	72	98	59	87	64	58	76	93	70	56	78	68

Table 9: The optimal solution for instance (10x20)

45	0	11	0	0	0	0	21	0	0	0	0	0	13	0	0	0	0	0	0	0
27	15	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	1	0	0	6	0	21	0	2	0	15	20	0	0	0	0	0	0	0	0	0
37	0	6	0	0	0	0	0	0	16	0	0	0	0	0	0	0	15	0	0	0
30	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	1	17	0
38	0	0	0	0	20	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	17	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	13
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	7
	16	**17**	**12**	**18**	**20**	**21**	**21**	**20**	**16**	**15**	**20**	**14**	**13**	**22**	**17**	**24**	**15**	**18**	**17**	**20**

Journal of Theoretical and Applied Information Technology
30th April 2017. Vol.95. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 10: Best chromosome for the test problem.

Instance	Best chromosome
4x5	1 6 8 2 4 3 9 7 5
5x10	11 1 7 3 13 14 2 5 6 12 4 10 9 15 8
10x10	8 5 14 12 19 2 1 6 15 20 9 10 7 11 3 16 4 13 18 17
10x20	4 20 27 12 2 13 21 24 16 26 25 11 22 17 18 6 9 14 8 1 23 15 19 3 29 10 5 30 28 7
30x50	10 54 59 48 49 13 23 57 7 60 53 42 63 45 58 46 15 14 43 47 79 26 16 30 17 37 78 3 38 71 34 68 11 74 9 33 22 76 50 67 75 12 56 4 25 24 2 21 72 55 40 77 69 27 62 52 5 65 28 61 32 41 8 29 6 51 31 35 36 73 19 18 1 39 64 66 70 20 44 80

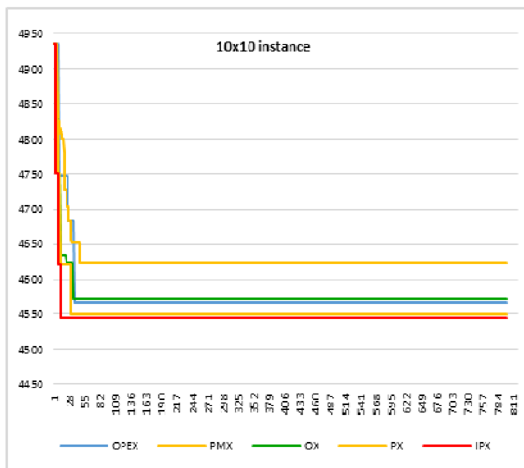
In this comparative study, for small instances, 4*5 and 5*10, the results obtained show that GA with all crossover operators used allow to obtain the best chromosome satisfying the optimal solution. However, with the proposed crossover operator IPX, you need a reduced number of iterations. For 5*10 instance, we need 69 iterations with the operator OPEX, 34 iterations with PMX operator, 27 iterations with OX, 16 iterations with PX operator to achieve the optimal solution that is Z=6195. By cons, 16 iterations are enough with the proposed IPX operator to get the same optimal solution.

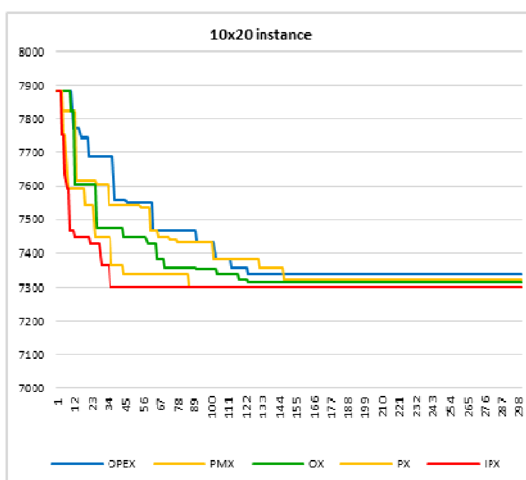
However, for problems with important size, the proposed operator IPX is very efficient compared to others, with a slight precision for the IPX operator in terms of optimal solution. Although, the latter is more advantageous to the level of number of iterations. Indeed, for 10*20 instance, the proposed operator IPX leads to the optimal solution Z=7303, this shows that our proposed operator is best compared to other operators.

On the other hand, the proposed operator IPX needs only 35 iterations to achieve optimal obtained solution. However, other operators do not achieve the optimal solution; they arrived at best and not the optimum solution after important iterations.

The performance of the IPX operator is clearer when the instances are more important. Indeed, for instance 30*50, the genetic algorithm with the proposed IPX operator converges to the optimal solution 16129 in 93 iterations which is not accessible to other operators. This shows the great performance of the proposed operator IPX, it allows achieving a better optimal solution in a reduced number of iterations over other operators.

In addition, the experimental results show that the execution time varies from one operator to another. The proposed operator IPX is the best than the other, they were always more quickly, especially for the most important instances.





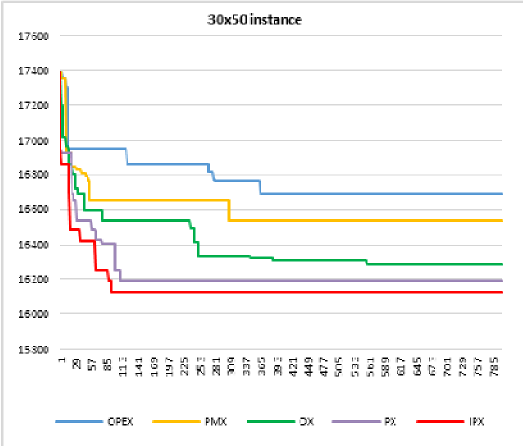


Figure 20. Performance of the GA with different crossover operators for FCTP

Find an algorithm that solves a problem, this is not always an easy thing, but what can be even more difficult and especially much more interesting, is to find an algorithm that provides this solution quickly. A factor in the performance of the algorithm is to solve the problem in a manner more quickly.

The convergence and the time of execution of the algorithm to achieve the optimal solution are very important; it varies from one operator to another. For this reason, the choice of the operator is an important factor to improve the solution.

5. CONCLUSION

In this work, we are interested to the FCTP problem that is an NP-complete and combinatorial problem. View that its resolution by the exact methods is very difficult, we opted for genetic algorithms where the crossover operator subject of our study has a great effect. After having presented four crossover operators (OX, PMX, OPEX, PX) adapted to our problem, we proposed a new operator which we have called (IPX). A comparative study was elaborated supporting the conclusion that the proposed IPX operator higher performance than the four operators presented, especially for larger instances.

REFRENCES:

[1] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities", *Journal of Mathematical Physics,* vol. 20, 1941, pp. 224–230.

[2] J. W. Stroup, "Allocation of launch vehicles to space missions: a fixed-cost transportation problem", *Operations Research*, vol. 15, No. 6, 1967, pp. 11-57.

[3] T. H. Hultberg, D. M. Cardoso, "The teacher assignment problem: A special case of fixed charge transportation problem", *European Journal of Operational Research*, vol. 101, 1997, pp. 463-473.

[4] V. Adlakha, K. Kowalski, "A heuristic method for more-for-less in distribution related problems", *International Journal of Mathematical Education in Science and Technology*, vol. 32, 2001, pp. 61-71.

[5] W. M. Hirsch, G. B. Dantzig, "The fixed charge problem", *Naval Research Logistics Quarterly,* vol. 15, 1968, pp. 413–424.

[6] K. G. Murty, "Solving the fixed charge problem by ranking the extreme points", *Operations research*, 1968, vol. 16, No. 2, pp. 268 -279.

[7] A. Sadighmanesh, K. Zare, M. Sabahi, "Distributed Generation unit and Capacitor Placement for Multi-objective Optimization", *International Journal of Electrical and Computer Engineering*, vol. 2, 2012, pp. 615-620.

[8] J. Gottlieb, L. Paulmann, "Genetic algorithm for the fixed charge transportation problem", *in: Proc. of IEEE Inter. Conf. on Evolutionary Computation*, 1998, pp. 330–335.

[9] J. R. Mohammed, "A Study on the Suitability of Genetic Algorithm for Adaptive Channel Equalization", *International Journal of Electrical and Computer Engineering,* 2012, pp. 285-292.

[10] J. Holland, "Adaptation in natural and artificial systems", University of Michigan Press, 1975.

[11] Z. Michalewicz, G.A. Vignaux, M. Hobbs, "A non-standard genetic algorithm for the nonlinear transportation problem", *ORSA Journal on Computing,* vol. 3, 1991, pp. 307–316.

[12] M. Gen, R. Cheng, "Genetic Algorithms and Engineering Optimization", second ed., John Wiley & Sons, New York, 2000.

[13] M. Hajiaghaei - Keshteli, S. Molla – Alizadeh - Zavardehi, R. Tavakkoli - Moghaddam, "Addressing a nonlinear fixed charge transportation problem using a spanning tree based genetic algorithm", *Computers & Industrial Engineering,* vol. 59, 2010, pp. 259–271.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] M. Gen, F. Altiparmak, L. Lin, "A genetic algorithm for two-stage transportation problem using priority-based encoding", *OR Spectrum,* vol. 28, 2006, pp. 337–354.

[15] L. Lin, M. Gen, X. Wang, "Integrated multi-stage logistics network design by using hybrid evolutionary algorithm", *Computers & Industrial Engineering,* vol. 56, 2009, pp. 854–873.

[16] J.E. Lee, M. Gen, K.G. Rhee, "Network model and optimization of reverse logistics by hybrid genetic algorithm", *Computers & Industrial Engineering*, vol. 56, 2009, pp. 951–964.

[17] O. Abdoun, C. Tajani, J. Abouchabaka, "Hybridizing PSM and RSM Operator for Solving NP-Complete Problems: Application to Traveling Salesman Problem", *International Journal of Computer Science Issues*, vol. 9, No. 1, 2012, pp. 374-378.

[18] A. Sadighmanesh, K. Zare, M. Sabahi, "Distributed Generation unit and Capacitor Placement for Loss, Voltage profile and ATC Optimization", *International Journal of Electrical and Computer Engineering*, vol. 2, 2012, pp. 774-780.

[19] M. I. Azim, M. F. Rahman, "Genetic Algorithm Based Reactive Power Management by SVC", *International Journal of Electrical and Computer Engineering,* 2014, pp. 200-206.

[20] K.W. Kim, M. Gen, G. Yamazaki, "Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling", *Applied Soft Computing*, vol. 2, 2003, pp. 174–188.

[21] M.L. Balinski, "Fixed cost transportation problems", *Naval Research Logistic Quarterly* vol. 8, 1961, pp. 41–54.

[22] M. Lotfia, R. Tavakkoli, "A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems", *Applied Soft Computing*, vol. 13, 2013, pp. 2711–2726.

[23] L. Davis,"Applying adaptive algorithms to domains". *Proceeding of the International Joint Conference on Artificial Intelligence*, vol. 1, 1985, pp. 162–164.

[24] G. Syswerda, "Schedule Optimization Using Genetic Algorithms", Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990.

[25] B. Chakraborty, P. Chaudhuri, " On The Use of Genetic Algorithm with Elitism in Robust and Non parametric Multivariate Analysis", Austrian journal of statistics, vol. 32, No. 1-2, 2003, pp. 13–27.

[26] V. Adlakha, K. Kowalski, R.R. Vemuganti, B. Lev, "More-for-less algorithm for fixed-charge transportation problems", *Omega*, vol. 35, pp. 116–127.