# Using Genetic Algorithms in Integer Programming for Decision Support

## **Dr. Youcef Souar**

## **Omar Mouffok**

Taher Moulay University – Saida, Algeria Email:Syoucef12@yahoo.fr

#### Doi:10.5901/ajis.2014.v3n6p11

#### Abstract

Genetic Algorithms is a new developed quantitative method used in management decision support; it's an Artificial intelligence technique that simulates scientific explanations in genetics and natural evolution for getting an optimal solution population. Thus this study aims to recognize Genetic Algorithms including: definition, properties, basic elements and steps. Then we try to employ it in practical study. A local Mill shapes a suitable case to use Genetic Algorithms in Integer Programming as one of its application forms. After using Evolver software, and making a comparison with Branch and Bound method, we have got results around this case. At last we have concluded the efficiency of Genetic Algorithms in Integer Programming, and its superiority concerning characteristics, it also deserves to be used in more complicated problems and when usual methods fail.

Keywords: genetic algorithms ;decision support; quantitative methods; integer programming, optimization; artificial intelligence.

### 1. Introduction

Decision making process has a great importance in management, where it is the basic goal. Due to developments known in management sciences for facing internal and external factors of the institution, it becomes necessary to use more efficient methods to make the optimal decision that fits the institution policies and strategies.

Decision making depends on availability of needed information that allows having more clear and exact choices to decision maker. Data can be collected by different qualitative and quantitative methods. Quantitative methods including statistics and Operational Research provide data in form of: numbers, proportions, probabilities...could be framed to have information contributes in decision support. There is a great integration between management and these methods proven its effectiveness across many researches.

Thanks to the amazing growing of networks and artificial intelligence, Operational Research knows recent changes and developments, so it has been discovered new methods. Among these methods, there is the Genetic Algorithms that has a wide scope in this domain.

Integer programming as one of Operational Research domains, it depends usually on Branch and Bound method, but in more complicated cases it requires a more developed method such as Genetic Algorithms. Thus, to what extent is the efficiency of using genetic algorithms in Integer Programming for decision support?

#### 2. Literature Review

This section looks at the various definitions of integer programming ,genetic algorithms and Genetic Algorithms Operations and Steps.

### 2.1 Integer Programming

In some optimization problems, the variables depend to physical quantities which can't be parted, therefore it must be integer values. Using Linear Programming usually gives fractional values, so this kind of problem requires Integer Programming.

The model of Integer Programming contains objective function and constraints just like Linear Programming, but the only difference is the variables of the final solution, that must be integer values. Besides it's more difficult so it depends on complicated calculations and algorithms [Render et al., 2012].

The usual method is Branch and Bound, it consists of two steps:

- Finding the optimal solution of the original model by using Simplex method, if its variables are fractional values we pass to the second step.
- Adding new constraints to the original model to have two new models to be solved separately, that's the branching. Then we find the optimal solution again, we keep branching until we get integer values variables.

### 2.2 Genetic Algorithms

Methods that simulate nature like: Genetic Algorithms, Neural Networks, Ant Colony, Logic Flow, Simulated Annealing... produce new search points and represent some natural phenomena operations. These methods depend on artificial intelligent researches using statistical tools.

John Holland was the first who created Genetic Algorithms and founded its theoretical basis in 1975 [Holland, 1975]. De Jong's works concerning function optimization show the ability of using Genetic Algorithms, they were considered as the first efforts to determine the application parameters [De Jong, 1980]. David Goldberg has been successful in solving a complicated problem of pipeline operations using Genetic Algorithms [Goldberg, 1981], then he has introduced Genetic Algorithm as an effective tool and has made it more famous through his perfect book [Goldberg, 1989].

Due to the simplicity of genetic algorithms and its application ability to various types of problems and domains, the adoption of Genetic Algorithms is constantly increasing. It has been used to solve difficult problems in various technical domains like: computing, programming, artificial intelligent, biology, engineering, planning, industries... as well as social and economical sciences where it is applied in: economy, finance, management, decision making...., where it's used as a tool that provides data needed to support a decision attached to quantitative model. It is proven to be used in Operational Research, data mining, econometrics, forecasting and time series [Alander, 2012].

Genetic Algorithms is a search and solving technique and heuristic method depends on genetics and natural selection principals, it can be applied to any kind of optimization problem. It allows under specific constrains to have the optimal solution among all the possible solutions - that shape the search space - by achieving the highest fitness value [Coley, 1999].

Genetic Algorithms is characterized by: the randomness as a principal of all its operations, it gives a population of solutions; it could be used in any kind of problems in several domains and its performance elasticity. As for limits, it doesn't guarantee always having the best solution in some cases; so at least, it probably finds a good acceptable solution. Its speed and its accuracy are influenced by the problem nature. Anyway Genetic Algorithms could be worthily used when other methods fail [Sivanandam & Deepa 2008].

#### 2.2.1 Basic Elements

Genetic Algorithms is based on the following elements: population, individual, encoding and fitness; these elements depend to an environment is the search space. Population consists of individuals, each individual is represented by a chromosome in form of an encoded string, and each chromosome is evaluated according the fitness function.

### 2.2.2 Population

The population is a group of individuals live in the studied environment subject to many tests and Genetic Algorithms operations. The most important concepts are: initial population generation and the population size (n).

#### 2.2.3 Individuals

An individual is one of the possible solutions; it is represented by a chromosome containing raw genetic information (Genotype) that the Genetic Algorithms deals with. Phenotype is the real value expressed by the chromosome. The chromosome is divided by genes; each gene represents a solution variable [Sivanandam & Deepa 2008].

### 2.2.4 Encoding

Encoding is the process of transforming the real values of variables or genes into string of codes adapted to Genetic

E-ISSN 2281-4612	Academic Journal of Interdisciplinary Studies	Vol 3 No 6
ISSN 2281-3993	MCSER Publishing, Rome-Italy	November 2014

Algorithms work; it can be performed by using: bits, numbers, letters.... The binary encoding is the most common way, where each chromosome is represented by a binary string (0 and 1), so each string represents a possible solution. For example:

Chromosome 1: 0010110110 Chromosome 2: 1000110011

The string length differs according to variables value, the number of values increases with string length, it equals to  $2^{L}$ . Assuming the variable X, its extreme values are  $X_{min}$  and  $X_{max}$  [Adeli & Sarma 2006]:

$$\begin{split} &X_{min} \leq X \leq X_{max} \Leftrightarrow 00 \dots 0 \leq X \leq 11 \dots 1 \\ &\text{The decoding is performed by the following formula:} \\ &X = X_{min} + \frac{A}{2^L} (X_{max} - X_{min}) \\ &A = \sum_{l=1}^{L} a_l 2^{L-l} \\ &A : \text{The string value.} \\ &a_l : \text{The bit value in the } l \text{ location.} \end{split}$$

## 2.2.5 Fitness

The individual's fitness in Genetic Algorithms is the value of its goal function; it is the basic criterion of Genetic Algorithms operations and the evaluation of solution. It can take any function could be evaluated according to problem nature [Rothlauf, 2006].

## 2.3 Genetic Algorithms Operations and Steps

Genetic Algorithms handles a population of possible solutions. First we form an initial population generation of individual and evaluate them according to the fitness function, then we reproduce new individual through a repeated loop of operations: selection, reproduction (crossover), mutation and replacement. This loop continues until attaining termination criterions. There is always a possibility to have a better solution in the next generation [Sivanandam & Deepa 2008].

The following figure represents Genetic Algorithms steps and operations with a simulation of a natural example. We suppose a population of five mice of different: colors (dark and bright), and size (big and small). Cats have difficulties to find small and dark mice, so the fitness is the dark color and the small size [Palisade Corporation, 2010]. To match between Genetic Algorithms and natural features we suppose that: each mouse represents a solution, the fittest mouse to this environment represents the optimal solution, natural factors represent problem constraints, and genes are the model variables.



**Figure 1.**Genetic Algorithms Representation Simulating Natural Example **Source:** Haupt & Haupt, 2006: 28

## 2.3.1 Selection

It is the process of selecting parents from the population for reproduction according to natural selection, with hope of having a fitter individual. There are several ways of selection: using Roulette wheel divided to sections of selection probabilities according to fitness, taking the higher values individuals as parents, using a tournament between individuals distributed randomly or picking parents randomly regardless to their fitness.

## 2.3.2 Reproduction

After selection, parents reproduce new offspring, so the genes get emerged by crossover between chromosomes which allows Genotypes diversity. Thus we exchange parts of the parent's strings in one or more point to have new offspring.



Figure 2.Single point crossover Source: S.N.Sivanandam, S.N.Deepa, 2008:58

## 2.3.3 Mutation

After reproduction we move to mutation, it gives new Genotypes to cover some lost genes and prevent to fall in tight space, the simple mutation is created by changing gene codes randomly in a specific locus. We can mutate by: flipping bits (codes), interchanging two specific points, or reversing codes in a specific site.



## Figure 3.Mutation Source: S.N.Sivanandam, S.N.Deepa, 2008:68

### 2.3.4 Replacement

Replacement is the last part of Genetic Algorithms operation loop; we nominate the new offspring to enter the population. According to fitness we eliminate the weakest individuals and replace them by the best offspring ("Survival of the Fittest").

### 2.3.5 Termination

The Genetic Algorithms stops when the terminations criterions are attained, then we take the best individual to be the optimal solution, so we can stop the algorithm after: a specific time, specific number of generations, no change in the fitness or attaining a convergence value.

## 3. The Practical Study

After addressing the theoretical aspects of Genetic Algorithms, we try to apply it in a practical study, so we have chosen a local mill as a sample, it produces four products of flour:  $x_1, x_2, x_3, x_4$ .

We have collected the data surrounding the problem in order to decide product quantities to be produced in a week to maximize global profits, this data contain: production capacity, bagging capacity, trading constrains besides unit profits. The variables are the numbers of flour bags that must be integer values; we formulate these data in a mathematical E-ISSN 2281-4612 ISSN 2281-3993

model as follows:

```
Max \ Z = 108, 15x_1 + 61, 65x_2 + 91, 5x_3 + 105x_4
\begin{cases} \frac{x_1}{300} + \frac{x_2}{300} + \frac{x_3}{200} + \frac{x_4}{180} \le 84 \\ 0,5x_1 + 0,25x_2 + 0,1x_3 + 0,1x_4 \le 7700 \\ 0,1x_3 + 0,1x_4 \le 672 \\ x_1 \ge 6500 \\ x_2 \ge 3000 \\ x_3 \ge 450 \\ x_4 \ge 200 \\ x_1, x_2, x_3, x_4 \ge 0 \end{cases}
```

### 4. Findings and Discussion

Generally For results confirmation and comparison we solved the problem with two methods: Branch and Bound using Win QSB software, Genetic Algorithms using Evolver Software.

Win QSB is a software that includes quantitative tools used in decision support, it allows solving various Operational Research problems besides to statistical and simulation models. Among these tools there is Linear and Integer Programming which could be used to solve our problem.

Evolver is a Palisade Corporation software, it is one of Palisade Decision Tools software pack that could be used in decision support. Evolver can solve various problems using Genetic Algorithms. After shaping the model in Excel that includes: fitness function, variables and constraints; then determining the model in Evolver, we start the task of applying the repeated loop of Genetic Algorithms operations: selection, reproduction, mutation and replacement. The termination could be whether automatic according to chosen parameters or manual according to our estimation. The results obtained are as follows:

I	Decision Variable	Solution Value	Unit Cost or Profit c(j)	Total Contribution	Reduced Cost	Basis Status
1	X1	12 758,0000	108,1500	1 379 778,0000	-416,8500	at bound
2	X2	3 000,0000	61,6500	184 950,0000	-200,8500	at bound
3	X3	450,0000	91,5000	41 a 175,0000	-13,5000	at bound
4	×4	5260,0000	105,0000	552 <b>3</b> 00,0000	0	basic
	Objective	Function	(Max.) =	2 <b>പ്പ158പ്പ203,0000</b>		
	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price
1	C1	83,9989	<=	84,0000	0,0011	0
2	C2	7 700,0000	<=	7. 700,0000	0	1 050,0000
3	C3	571,0000	<=	672,0000	101,0000	0

Figure 4. Win QSB Rusults

Evolver Watc	ther	In other latter a li		-	and the second second	Evolver V	Vatcher						
Bogress [	Summy Log	Population   Qiversity	Stopping Options		1	Brogen	s Germany Lo	og Popylation	Qversity	Stopping Option	1		
Advertishin California					Show All Truls			*					
11	1 114	Denil ET		84	12	Trial	Elapsed Time	Result	F3	P4	P6	F7	TD-+
Best	93313	2158202,7. 12	58 3000	450	\$260		00-00-00	955300	6500	3000	450	200	1
Original	1	950100 65	00 3000	450	200	1 3	00:00:02 18	115542,70	6500	17038	450	200	
Let	2999-46	1481400 63	00 3000	450	5260		00.00.02 13	48298.50	650	3000	1304	200	
							00:00:02	1494420	6500	3000	450	5384	
10.110.00						1.1.1	00.00.02	1107663	6500	3000	7172	200	
wojuscios (	Cel oroup sensings						00:00:02	1430000	6500	3000	450	4580	
gops	hown	F3,F4,F6					00:00:02:05	192300,75	13340	3000	450	200	
Crossov.	er Rate	1		0.1000	31		00.00.02 18	02966.10.	6500	26834	450	200	-
Hutator	Rate	11		0.1000	T	1	00:00:02 10	15098,15	7101	3000	450	200	
					-		00:00:02	N/A	14923	3000	450	200	
						12	00:00:02 18	20110.05	7340	1 15635	450	200	-
						1	000002.	100.0	6300	, 2000		200	
1.1.41							- I - I				_	_	
· 国	8		-	_	OK		13				_	-	ox
roduct	Bagging C	apacity (Bag/h)	Time (h)	Time (%)	Number of	Bags	Mass (Quin	ntal) Ma	ss (%)	Unit Profit	(AD)	Global Pr	ofit (AD
X1		300	42 1/2	50,63%	12758	k I	6379	82	,84%	108,1	5	1 379 7	77,70
X2		300	10	11,90%	3000		750	9,	74%	61,65		184 95	0,00
X3	· · · · ·	200	21/4	2,68%	450	Ĩ	45	0,	58%	91,5	1. I	41 17	5,00
X4		180	29 2/9	34,79%	5260	1	526	6,	83%	105		552 30	0,00
Î	Tota	l Time (h)	84	100,00%	Total Mass (C	(uintal)	7700	100	0,00%	Total Profit	t (AD)	2 158 2	02,70

### Figure 5 . Evolver Results

The optimal solution is to produce weekly: 12,758 bags of  $x_1$ , 3,000 of  $x_2$ , 450 of  $x_3$  and 5,260 of  $x_4$ ; to attain maximum global profit of: 2,158,202.70 AD.

E-ISSN 2281-4612	Academic Journal of Interdisciplinary Studies	Vol 3 No 6
ISSN 2281-3993	MCSER Publishing, Rome-Italy	November 2014

The production capacity is exploited as follows: 82.84% for  $x_1$ , 9.74% for  $x_2$ , 0.58% for  $x_3$ , 6.83% for  $x_4$ . The bagging capacity is exploited as follows: 50.63% for  $x_1$ , 11.90% for  $x_2$ , 2.68% for  $x_3$ , 34.79% for  $x_4$ .

As confirmation of results; both methods have given the same optimum solution, we see that the profitable products are  $x_1$  and  $x_4$ , as for the others we produce just the minimum respecting the trade constraints. Genetic Algorithms gives a population of solutions, but the important in our case is only the optimal solution.

Since the Branch and Bound method is derived from Linear Programming, it deals just with linear function whether maximization or minimization, and their constraints are linear inequalities, it doesn't matter in our case but in case of different functions such as: logarithmic, exponential... it requires transformation to linear form which needs mathematics skills, in some cases it leads to confusion or falsity. While Genetic Algorithms deals with all function kinds directly, in addition to maximization and minimization the goal can be a specific value.

In Evolver the data entrance could be in any available form to Excel, while in Win QSB, the entrance of data should be directly with numbers in a specific table. In our case the entrance of the bagging constraints in Evolver was based on the fraction  $\frac{x_i}{t_i}$ , while in Win QSB it required to decimalize it with nine decimal places to prevent rounding problems, with fine decimal places to prevent rounding problems, with

five decimal places we had misleading results (Z= 2,159,576).

Output and model in general are displayed in Evolver according to our choices and preferences, so we can even use graphs. In our case percentages are displayed directly. Output and model display in Win QSB is restricted in numbers and codes in form of  $\mathbf{x}_i$ , in our case percentages are not displayed directly which requires calculation when needed.

In what concerns the taken time, it's better by using Win QSB. If we repeat again it will take the same time, whereas by using Evolver it may be more or less, it's not constant because it depends to random Genetic Algorithms operations, which could creates an uncertainty. In Integer Programming it doesn't take a very long time, in our case after experimenting several times with different computers, the taken time was between 5 and 25 minutes. It is also related to computer's speed, we advise to let the running on as long as possible and to repeat several times.

The equality of results between both methods confirms the efficiency of Genetic Algorithms in Integer Programming, and its ability to give the optimal solution. The comparison shows Genetic Algorithms superiority using Evolver concerning characteristics excepting the taken time criterion; however we can neglect that, because running on for long time according to the problem nature doesn't harm the search anyway.

### 5. Concluding Remarks

Through the practical study about production decision support, we succeed to attain the optimal solution of the problem, after analyzing the results we conclude that Genetic Algorithms could be used efficiently in Integer Programming. From the comparison we realize that Genetic Algorithms outperform the Branch and Bound method concerning characteristics except the taken time criterion, but it is preferable to use Branch and Bound in traditional and simple problems with tight search space.

Nevertheless, Integer Programming is just one of the various Genetic Algorithms scope. Our study case was very appropriate to apply Genetic Algorithms and test its efficiency, but it stills a mere simple case if compared to other management problems. We hope that it could be taken into account to treat it in other researches.

## References

Adeli, H., Sarma, K.C. (2006) Cost Optimization of Structures Fuzzy Logic, Genetic Algorithms and Parallel Computing, John Wiley & Sons, England.

Alander, J.T. (2012) An Indexed Bibliography of Genetic Algorithms in Economics, Report Series No. 94-1-ECO, Finland.

Coley, D.A. (1999) An Introduction to Genetic Algorithms for Scientists and Engineers, World Scientific, USA.

Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, USA. Goldberg, D.E. (1981) Robust Learning and Decision Algorithms for Pipeline Operations. Unpublished dissertation proposal, University

of Michigan, Ann Arbor.

Haupt, R.L., Haupt, S.E. (2006) Practical Genetic Algorithms, 2nd edition, John Wiley & Sons, USA.

Holland, J.H. (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press.

Jong, K.D. (1980) Adaptive System Design: A Genetic Approach, IEEE Transactions on Systems, Man and Cybernetics 10, p 556–574. Palisade Corporation (2010) Guide to Using Evolver the Genetic Algorithm Solver for Microsoft Excel Version 5.5, January.

Rothlauf, F. (2006) Representations for Genetic and Evolutionary Algorithms, 2nd edition, Springer, Netherlands.

Render, B., Stair, R.M. Jr, Hanna, M.E. (2012) Quantitative Analysis for Management, 11th edition, Pearson.

Sivanandam, S.N., Deepa, S.N. (2008) Introduction to Genetic Algorithms, Springer, USA.