The "Clones" of the Incursive Genetic Algorithms

Arnaud VINCENT
GRPI
Place du 8 mai 1945
93206 St Denis cedex
Tel : 01 39 98 34 00 E-mail arnovinc@hol.fr

ABSTRACT

This paper deals with a singular comportment of an incursive algorithm. This algorithm has been primary developed to optimise a production process. It is closed to classical genetic algorithms because it uses these major stages : determination of n random solutions, evaluation of the solutions (in our case with the simulation of production process), selection of the best solution and reproduction according to the biologic rules of crossing over.

Nevertheless the differences between this algorithm and a classical genetic algorithm are : the use of incursion in the selection's stage and the singularity of the reproduction stage. The algorithm enables indeed to optimise the system configuration by optimising the laws of reproduction. The "crossing over" becomes a specific case among huge other configurations.

Despite we expect that the algorithm will be able to adapt his reproduction laws to the specificity of the system studied, the comportment of the optimisation, due to the use of incursion is unexpected and unfortunately unable to solve our production matter. Nevertheless this comportment is very interesting to study the consequences of the use of incursion with a genetic algorithm.

Keywords : genetic, incursion, clone, optimisation, algorithm

1 The algorithm's basis

1.1 The basic genetic algorithm

The genetic algorithm enables to optimise a system without an overall view of it. In our production matter we use a genetic algorithm with a simulation of our process. The genetic algorithm proposes a configuration which is evaluated by the simulation.

The principle of a genetic algorithm is to consider some individuals which are defined by a particular configuration of the system and the result provided by the simulator.

A "population" is defined as a group of individuals.

The genetic algorithm is divided in four stages

```
- Determination of n random individuals
Loop
- Evaluation of the solution provided by each individual
- Selection of the best one
- Reproduction
End of loop.
```

1.2 Stage of reproduction

The classical approach is to use the biologic principle of "crossing-over" which mixes two individuals "parents" to create a third one.

Example

Individual 1
Configuration proposed : $\underline{\mathbf{7\ 8\ 6\ 5}}$ 2 1 4 8 8
Performance : X
Individual 2
Configuration proposed : 2 2 2 7 $\underline{\mathbf{7\ 6\ 6\ 6\ 6}}$
Performance : Y
New individual
Configuration proposed : $\underline{\mathbf{7\ 8\ 6\ 5\ 7\ 6\ 6\ 6\ 6}}$
Performance : Unknown (will be evaluated in the next stage)

The use of crossing-over to optimise a system provides most generally good results. Nevertheless our approach was to reconsider this law of reproduction. where the crossing over would be only a particular case among huge other solutions.

1.3 A different way of reproduction

To search the best reproductions laws, we try to optimise them with the genetic algorithm itself, so we choose a binary code using four different letters A, B, C, D

A and B mean "0"
C and D mean "1"

Then, a configuration proposed by an individual of our algorithm will be described like this : AABBCDBBAADCDCDCBAAAA etc... that means 00001100001111...

The stage of reproduction will mix two individuals to create a third one. But the law of reproduction will be quite different :

Individual 1
Configuration proposed : AABBBCDDCBAAAAAA
Performance : X

Individual 2
Configuration proposed : BBBBCCDCCDDDACCD
Performance : Y

New individual
Configuration proposed : A+B A+B B+B B+B B+C .... A+D
Performance : Unknown (will be evaluated in the next stage)

Now the problem is to determine the value of the law "+". Our idea was to add to each individual a "sentence" where all the potential results of the law "+" are considered.

A+A = ?
A+B = ?
......
D+D =?

We add a sentence to the definition of each individual which correspond to particular laws of reproduction.

For example, this sentence : AAAABBBBCCCCBBBB describes the result of A+A =A, A+B=A, A+C=A, A+D=A, B+A =B......D+C= B, D+D = B

So, an individual is described with three different informations : the configuration proposed, which will be tested on the system, the sentence which describes laws of reproduction, and the performance :

Individual X

Configuration proposed	: AAABBBCDCDCDBBAACCDCDDDC
Reproduction laws	: AAAABBBBCCDDAACC
Performance : x	

In the stage of reproduction, the first individual considered imposes his reproduction laws :

Individual 1
Configuration proposed : ABCD
Reproduction laws :BBBBCCCCDDDDAAAA
Performance : X

Individual 2
Configuration proposed : DDDA
Reproduction laws : AAABBBBCDDDDCCCC
Performance : Y

New individual

Configuration proposed :	A+D = B, B+D = C, C+D = D, D+A = A
	So BCDA
Reproduction laws :	B+A, B+A, B+A, B+B,................,A+C
	So CCCC.........B

Performance : Unknown (will be evaluated in the next stage)

A more detail's example is proposed on the fig.1

In spite of his interest, this algorithm is unable to optimise our system because an individual with a good performance is not inevitably an individual with the best laws of reproduction. To know if an individual has got the best laws of reproduction, we have to study the comportment of his "descendants". So we have to consider the future states of the algorithm to select the best individuals.

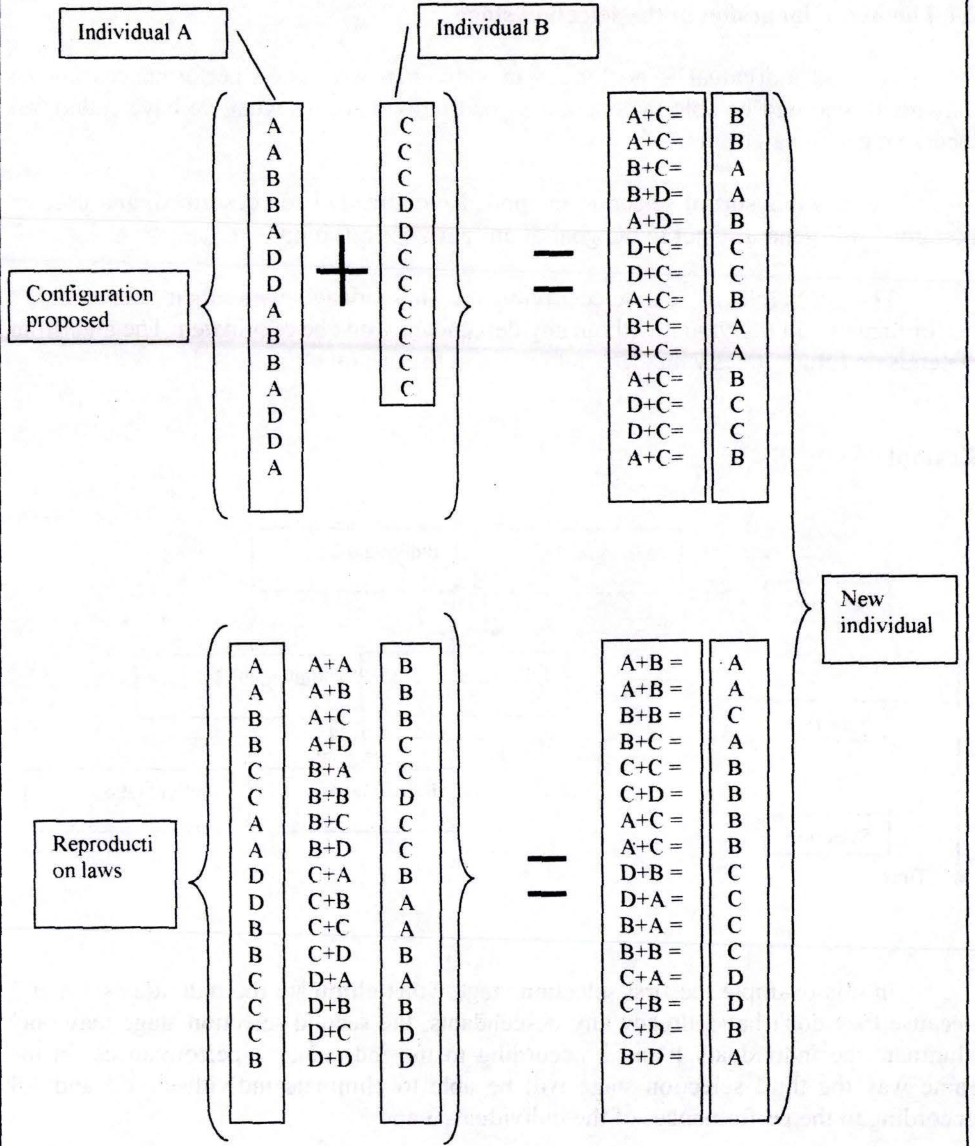

Individual A
Individual B
Configuration proposed
A A B B A D D A B B A D D A
C C C D D C C C C C C
+
=
A+C= A+C= B+C= B+D= A+D= D+C= D+C= A+C= B+C= B+C= A+C= D+C= D+C= A+C=
B B A A B C C B A A B C C B
New individual
Reproducti on laws
A A B B C C A A D D B B C C C B
A+A A+B A+C A+D B+A B+B B+C B+D C+A C+B C+C C+D D+A D+B D+C D+D
B B B C C D C C B A A B A B D D
=
A+B = A+B = B+B = B+C = C+C = C+D = A+C = A+C = D+B = D+A = B+A = B+B = C+A = C+B = C+D = B+D =
A A C A B B B B C C C C D D B A

Figure 1 - Reproduction of two individuals according to their reproduction rules

2. The selection stage

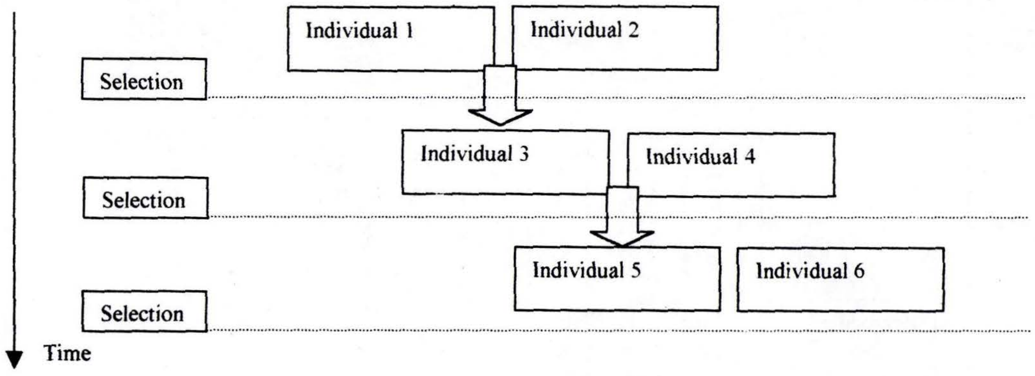
2.1 The use of incursion in the selection stage

A good individual is no longer an individual with good performances but an individual who will be able to generate a good individual. It's what we have called "an incursive genetic algorithm".

Despite this sort of selection methods have already been described, and used in literature, it is generally not in the goal of an "auto-optimisation".

The principle is to select only the individuals with their descendant's performances, an individual without any descendant won't be eliminated. The algorithm depends on future states which are not known at the instant t.

Example



In this example the first selection stage won't eliminate the individuals 1 and 2 because they don't have already any descendants, the second selection stage may only eliminate the individuals 1 and 2 according to the individual 3 performances. In the same way the third selection stage will be able to eliminate individuals 1,2 and 3,4 according to the performances of the individuals 3 and 5.

Every individual may have a lot of different descendants, in this case this individual will keep the best performance of the best descendant.

2.2 The selection criterion

After a lot empirical tests we have decided to choose a dynamic rule of selection. We have chosen to reduce the number of individuals progressively with this method :

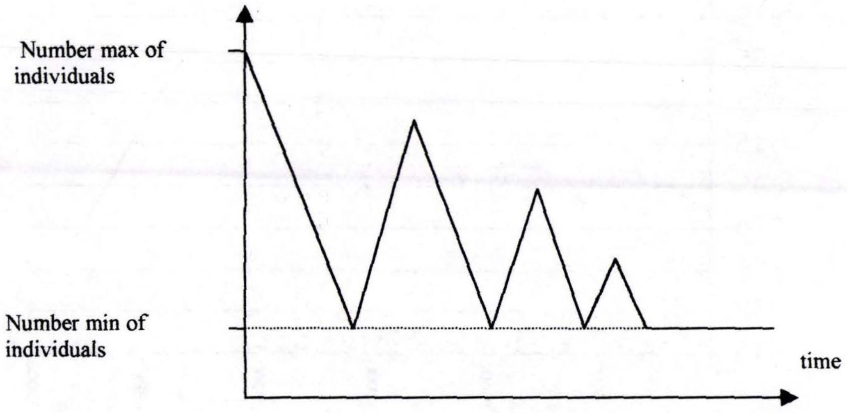


Figure 2 - Evolution of the selection criterion with time

3. Results

Without the use of incursion the comportment of the algorithm seems to be chaotic, indeed when an individual has a good performance, his law of reproduction is not systematically the best one.

With the use of incursion we assist to a singular comportment. In a very few iterations (less than 1000) some individuals are able to resist at the selection stage with very bad performances. In fact, step by step they transform their reproduction laws to be able to replicate themselves as identical.

Example :
The most simple "clone" is :
Individual X

Configuration proposed	: AAAAAAAAAAAAAAAAAAAA
Reproduction laws	: AAAAAAAAAAAAAAAA
Performance	: (bad)

This individual is only able to replicate himself as identical. When a new individual is generated he won't be eliminated in the selection stage because he doesn't have already any descendants. Then, this individual is escaping at all selection's stages.

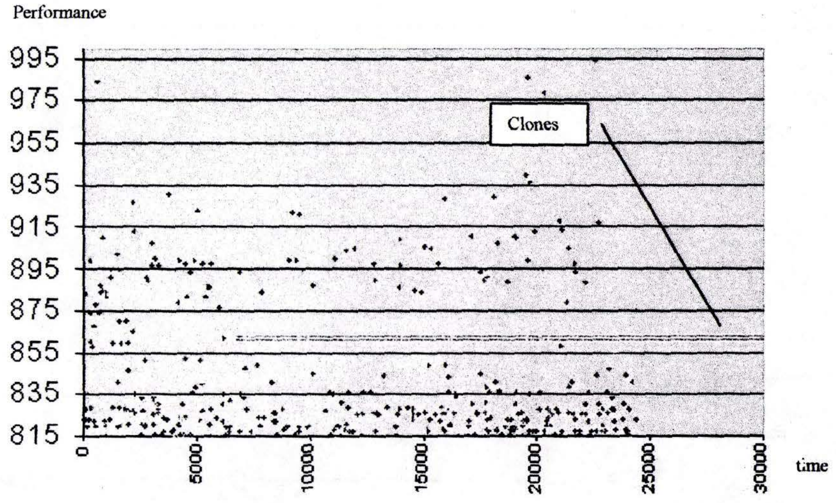


Figure 3

This figure (fig. 3) shows the performance of each individual tested to our system. The apparition of clones is sudden past 5000 and the clones eliminate all the other individuals at the end of the selection.

It is interesting to notice that the clone apparition probability is very weak in only 1000 generations. The population has resisted to the selection process only by optimising his reproduction rules.

This phenomenon has been called a "clone" apparition. Despite his interest, this "clone" apparition makes this algorithm unable to optimise our production system. So we tried to use new methods to eliminate the "clone" of our system. We "break" the incursion by testing if an individual is only able to replicate himself as identical, in this case we authorise the selection at the instant t (instead of t+1)

Despite these precautions we assist at the apparition of new "clones" which are able to replicate themselves in association with other individuals. They replicate themselves all the 2 or 3 successive generations instead of 1.

Conclusion

This algorithm throws light on the capacity of co-operation of virtual individuals and a singular adaptation capacity which could be interesting to use and study in other cases.

The debate remains to demonstrate that the apparition of "clones" is effectively due to the use of incursion in this algorithm.

If these considerations are interesting, this algorithm remains unable to solve our production problem, so we have recently made new experimentation of an association of incursion and other methods which provides good results. The use of incursion in association with other methods seems to be a good way of searching.

REFERENCES

D.M. DUBOIS
"Introduction of Aristole's Final Causation in CAST : Concept and Method of Incursion and Hyperincursion". In F. Pichler, R.Moreno Diaz, R. Albrecht (Eds.) : Computer Aided System Theory - EUROCAST'95. Lecture Notes in Computer Science, 1030, Springer- Verlag Berlin Heideberg, pp. 477-493. - 1996

D.M. DUBOIS
"Hyperincursive Stack Memory in Chaotic Automata, in Actes of Symposium ECHO "Emergence - Complexité hiérarchique - Organisation : Modèles de la boucle évolutive".
Université de Picardie Jules Vernes, pp. 77-82 - 1996

David E. GOLDBERG
Algorithmes génétiques, Exploration, optimisation et apprentissage automatique.
Editions Addison - Wesley 1994

Jean-Louis DESSALLES
L'ordinateur génétique
Edition Hermes 1996

Arnaud VINCENT, Franck FONTANILI
« Comment optimiser le fonctionnement d'un atelier avec la simulation de flux »
Revue Française de Gestion Industrielle – Volume 16 – Numero 3/1997

Arnaud VINCENT, Franck FONTANILI, Raymond PONSONNET, Thierry SORIANO
« A ring shaped mechanical assembly line optimized by a genetic algorithm »
IDMME – International Conférence on integrated Design and Manufacturing in Mechanical Engineering 1998
pages 803 à 810

M. WIDMER
« Modèles mathématiques pour une gestion efficace des ateliers flexibles »
PPUR 1991

J.M.RENDERS
« Algorithmes génétiques et réseaux de neurones »
HERMES 1997