In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., & Smith, R. E. (eds.). GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 612-619, Orlando, Florida, USA, 13-17 July, 1999. Morgan Kaufmann.

Transposition versus Crossover: An Empirical Study

Anabela Borges Simões
Centre for Informatics and Systems
of the University of Coimbra
Polo II - Pinhal de Marrocos
3030 Coimbra - Portugal

Ernesto Costa
Centre for Informatics and Systems
of the University of Coimbra
Polo II - Pinhal de Marrocos
3030 Coimbra - Portugal

Abstract

Genetic algorithms are adaptive systems biologically motivated which have been used to solve different problems. Since Holland's proposals back in 1975, two main genetic operators, crossover and mutation, have been explored with success. Nonetheless, nature presents many other mechanisms of genetic recombination, based on phenomena like gene insertion, duplication or movement. The aim of this paper is to study one of these mechanisms: transposition. Transposition is a context-sensitive operator that promotes gene movement intra or inter chromosomes. This work presents an empirical study of the genetic algorithm performance, being the traditional crossover operator replaced by transposition. Such empirical study, based on an extensive set of test functions, shows that, under certain circumstances, transposition allows the GA to achieve higher quality solutions.

1 INTRODUCTION

Genetic Algorithms (GA) are a search paradigm that applies ideas from evolutionary biology (crossover, mutation, natural selection) in order to deal with intractable search spaces (Holland 1992). The power and success of GA is mostly due to the diversity of the individuals of a population that evolve according to the principle of "the survival of the fittest". In the standard GA, the population diversity is obtained and maintained using the genetic operators of crossover and mutation, which allow the GA to find more promising solutions and avoid premature convergence to a local maximum (Goldberg 1989).

In order to find the most efficient ways of using GA, many researchers have carried out extensive studies to understand specific aspects such as the role of types of selection, representation issues and how to apply different types of genetic operators.

The use of the genetic operators has been the object of study of many researchers. Some important work related with crossover and mutation can be found in (Davis 1989; De Jong et al. 1992; Schaffer et al. 1991; Spears et al. 1991; Spears 1992; Spears 1993; Syswerda 1989).

In addition to the traditional genetic operators, many authors have presented new genetic operators dependent of the problem domain, for instance, (Davidor 1989; D'Haeseleer 1993; Mathias et al. 1992; Parsons et al. 1995).

Nevertheless, no new biologically inspired genetic operators have been widely adopted since the advent of GAs. Rather, the inversion operator included in John Holland's original work (Holland 1992) has been largely abandoned. Mitchell et al. (1994) point out the importance of studying new genetic operators. In addition, the authors emphasize the last discoveries of molecular biology as a good source of inspiration for new mechanisms of genetic material. Mitchell et al. (1994) and Mitchell (1996) state that it would be interesting to analyze if any of these mechanisms, incorporated in a GA, could lead to any significant advantages. Banzhaf et al. (1998) share the same opinion: the authors highlight the significance of implementing evolutionary approaches using mechanisms such as conjugation, transduction or transposition.

Following these ideas, some authors have proposed other biologically inspired genetic operators, besides crossover and mutation. Furuhashi et al. (1994) introduced an application using a bacterial mechanism called transduction. Transduction is a process involving bacteriophages which carry a copy of a gene from a host cell and insert it in the chromosome of an infected cell. By transduction it is possible to spread the characteristics of a single bacterium to the rest of population. Furuhashi et al. (1994) presented a new approach for finding fuzzy rules for an obstacle avoidance problem involving a

mobile robot. The authors showed that using transduction to locally improve the chromosomes, the GA would be more efficient finding the solution. Transduction was also used by Yoshikawa et al. (1997) and Nawa et al. (1997, 1998, 1999).

Later, Harvey (1996) and Smith (1996a, 1996b) suggested alternative genetic operators inspired in a bacterial form of recombination called conjugation. This process involves the unidirectional transfer of genetic material by direct cellular contact between a donor bacterial cell and a recipient cell.

Harvey (1996) suggested a type of conjugation based on tournament selection. Parents are first selected on a random basis, and then the winner of the tournament becomes the donor and the loser the recipient of the genetic material.

Smith (1996a) used tournament-based conjugation as a method of genetic recombination in complex satisfiability problems. He constructed a simple model using a GA that operates directly on the phenotype (the satisfiability expressions) and applies the mutation operator. Later, Smith (1996b) proposed a simple conjugation operator involving two individuals randomly chosen. Both authors achieved positive results using those alternative genetic operators.

Odutayo (1996) has empirically studied the conjugation and crossover operators, using the five De Jong's test bed functions.

Simões et al. (1999a) introduced a new genetic operator, inspired in biology. The proposed mechanism is known as transposition and consists in the presence of genetic mobile units called transposons or jumping genes. These are capable to relocate themselves, or transpose, onto the chromosome and subsequently jump into new zones of the same or other chromosome. This preliminary work, employing a single test function, revealed that, replacing crossover by a simple form of transposition, the GA achieved better results, even with smaller populations.

This paper extends the comparative analysis of crossover and transposition to a wider test suit. Moreover, a new form of transposition, other than the simple one introduced by Simões et al. (1999a) - a tournament-based transposition -, is tested and compared with traditional crossover.

The rest of the paper is divided in five sections. Section 2. introduces the classical way to use the traditional GA. The following section describes how transposition works in nature and the inserted simplifications to implement it computationally. Section 4. describes our case study. Next, is presented an exhaustive comparison of the results obtained with two forms of transposition (simple and a tournament-based), 1-point, 2-point and uniform crossover. The final section highlights the main conclusions of the present study.

2 THE CLASSICAL GENETIC ALGORITHM

A GA starts with a randomly initialized population of candidate solutions and implements probabilistic and parallel exploration in the search space using the domain-independent genetic operators of selection, crossover and mutation. A GA associates each individual candidate in the population with a fitness which measures the quality of a solution. Selection chooses individuals probabilistically, according to their fitness. The higher the fitness, the more likely it is for an individual to be selected. Crossover and mutation produce new individuals: the first operator exchanges genetic information between two selected parents; mutation randomly changes one gene value to the generated offspring.

The GA searches through an iterative process: the process of one generation involving selection, crossover and mutation is called one cycle of iteration and is repeated until convergence is reached or the number of generations achieves the established limit.

The typical GA is described in Figure 1.

1. Randomly initialize population
2. Do
   2.1. Evaluate population
   2.2. Select parents
   2.3. Crossover
   2.4. Mutation
   2.5. Substitute old population

Until (DONE)

Figure 1: The Classical Genetic Algorithm

3 TRANSPOSITION

In nature, the genetic diversity of the individuals is preserved by several mechanisms that involve operations like gene insertion, duplication or movement (Russell 1998). In each one of these categories there are several processes that produce changes in the genome of the species enabling the genetic diversity. For instance, there are mechanisms involving gene insertion, like transformation, transduction, conjugation and retroinsertion; or involving either gene duplication or gene movement, like break and fusion, unequal recombination and transposition.

This paper extends Simões et al. (1999a) preliminary work, using one of these biological mechanisms, the transposition.

3.1 BIOLOGICAL TRANSPOSITION

Transposition is characterized by the presence of mobile genetic units inside the genome, moving themselves to new locations or duplicating and inserting themselves elsewhere. These mobile units are called transposons (Gould et al. 1996).

Transposons (also known as jumping genes) can be formed by one or several genes or just a control unit. The movement can take place in the same or in a different chromosome.

Transposition was first discovered by Barbara McClintock in the 50's (when the DNA structure was not yet completely understood). She proved that certain phenomena present in living beings exposed to UV radiation could not be the result of the normal recombination and mutation processes. She found that in corn certain genetic elements occasionally move producing kernels with unusual colors that could not have resulted from crossover or mutation. Transposons were for a long time considered as some sort of abnormality, but in 1983 when she was awarded the Nobel Prize, many such transposons had been discovered and their possible role in evolution was beginning to be recognized. For instance, the genetic alterations caused by transposons are responsible for the growth of cancers in human or the resistance to antibiotics in bacteria (Gould et al. 1996; Russell, 1998).

In order for a transposable element to transpose as a discrete entity it is necessary for its ends to be recognized. So, transposons within a chromosome are flanked by identical or inverse repeated sequences, some of which are actually part of the transposon. See Figure bellow.

INVERSE FLANKING SEQUENCES

NNNNNATTGA (Transposon) AGTTANNNNNN

IDENTICAL FLANKING SEQUENCES

NNNNNATTGA (Transposon) ATTGANNNNNN

Figure 2: Inverse and Equal Flanking Sequences

When the transposon moves to another zone of the genome one of the flanking sequences goes with it.

The insertion point can be chosen at random, but there are transposons that show a regional preference when inserting into the same gene. Other method can be a correspondence in the new position with the flanking sequence.

The point into which the transposon is inserted requires no homology with the point where the transposon was excised. This is in marked contrast to classical recombination, where relatively long sequences of DNA must share homology to permit a recombination event to occur (same cut point(s)). Consequently, transposition is sometimes referred to as illegitimate recombination.

3.2 COMPUTATIONAL TRANSPOSITION

The first form of computational transposition proposed by Simões et al. (1999a) was directly inspired in biology. After the selection of two parents for mating, the transposon is formed in one of them. The insertion point is found in the second parent. According to this point, the same amount of genetic material is exchanged between the two chromosomes. The transposon is recognized by the presence of equal or inverse flanking sequences with a fixed length. The insertion point is searched in the second chromosome and is chosen when a sequence of bits equal or inverse to the flanking sequence is found. The insertion point will be the first gene after that sequence. After that, the movement of the transposon occurs. Since it was used fixed size chromosomes, the same amount of genetic material is exchanged between the two selected parents. The detailed functioning of transposition is described in Simões et al. (1999a). In this paper, this mechanism will be referred as *simple transposition*.

The first observations of the results immediately showed that, in spite of the good results using simple transposition, the population average became very unstable. In order to minimize this effect a new form of transposition was implemented: *tournament-based transposition*.

The two selected parents become competitors in a tournament. The transposon will be searched in the winner chromosome and the insertion point will be found in the loser parent. Only this individual will be altered by inserting the transposon, which replaces the same number of bits after the insertion point. Figure 3 shows these two mechanisms:

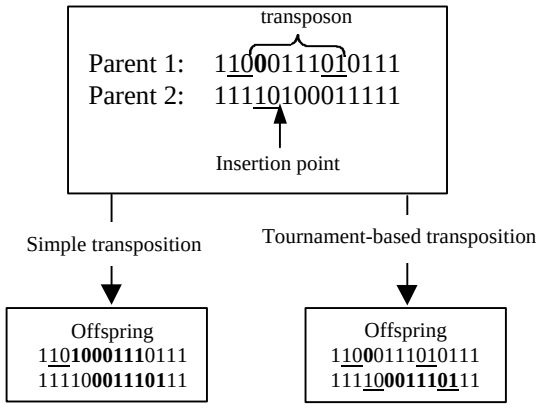

Figure 3: Simple and Tournament-based Transposition

4 THE ENVIRONMENT

The performance of transposition was studied using a test suit containing eighteen test functions (see Appendix). These functions were selected in order to cover a large set of characteristics, such as continuity/discontinuity, unimodal/multimodal, high/low/scalable dimensionality, stochastic/deterministic, quadratic/non-quadratic and convex/non-convex.

The test suit was divided in two categories: one including functions for maximization and the other for minimization. For the first category the five De Jong's test functions (De Jong 1975) were selected, since they are a well recognized test suit for measuring the GA performance. To complement the test suit was we selected other important functions proposed and used by several authors to evaluate the GA performance (Fogel 1995; Foster 1995; Michalewicz 1994; Whitley et al. 1995). For the maximization problem we used a total of eleven test functions (F1 - F11 in Appendix). In the second category we grouped seven test functions suggested by Koon et al. (1995) (F12 - F18 in Appendix).

Since the GA was used as a function optimizer, we chose roulette wheel with elitism as the selection method, in order to keep track of the best solution found (De Jong 1993).

The GA was first implemented with crossover (1-point. 2-poit and uniform) and then with transposition (simple and tournament-based). The population size varied between 50, 100 and 200 individuals, either for transposition and crossover. The elite size was 20% of the complete population.

The mutation and crossover/transposition rate used was 0.01 and 0.7, respectively. Ten runs of each experiment involving 1-point, 2-point and uniform crossover were executed.

Transposition was tested with flanking sequences from 1 to 8, 10, 15 or 20 - depending on the chromosome length. All the tests were run over 500, 1000 or 2000 generations - depending on the test function.

Off-line measure (De Jong 1975) was used to compare GA efficiency when applied crossover or transposition. This measure is defined by:

$$X^*_e(g) = \frac{1}{T} * \sum_{t=1}^{T} f_e^*(t)$$

Were $f_e^* = best\ \{f_e(1), f_e(2), ..., f_e(n)\}$ and T is the number of runs. In other words, off-line measure is the average of the best individuals in each generation. Due to the total of ten trials, the average of the tens runs was evaluated.

5 THE RESULTS

First, we will analyze the transposition results individually, explaining how the flanking sequences length can influence the performance of the GA. Empirical results show how we can choose the appropriate size for the flanking sequences, depending on the chromosome length.

Following, we will present the results obtained with simple and tournament-based transposition, 1-point crossover, 2-point crossover and uniform crossover.

Since we cannot show the achieved results with the 18 test functions, a representative function was selected: F14- N-dimensional test function, with N = 4. A global overview of all the results will be given at the end of the section. Simões (1999) presents the complete study, showing all the results achieved by the GA, using transposition or crossover, in the total test suit.

5.1 TRANSPOSITION PERFORMANCE

Transposition performance depends essentially on two factors: the population size and the flanking sequences length. Simões et al. (1999b) offer an extensive study about transposition performance. The authors demonstrate the importance of the choice of the flanking sequences length and its influence on the GA performance. In this paper this point will be focused only briefly.

The N-dimensional test function (with N = 4) used chromosome length of 74 bits. We analyzed the results obtained with the mechanism of transposition using flanking sequences length from 1 to 20 and, in each case, using populations with 50, 100 and 200 individuals.

Observing the average of the results obtained in the 10 simulations we conclude two main results:

1. With larger populations the results are better.
2. With larger flanking sequences the performance of the transposition decreases.

The first conclusion seems obvious. A possible justification to the second could be based on the observed fact that the length of the transposon directly depends on the flanking sequence length. In most cases the transposition mechanism doesn't occur because a matching flanking sequence is never found. In practice, with larger sequences the rate of transposition declines. With larger sequences, the amount of genetic material exchanged is larger, therefore it is harder to find the second flanking sequence. Hence, the percentage of no occurring transposition will be higher, what could lead to

a loss of the population diversity and, subsequently, to the worst results achieved.

Observing all the results obtained with the 18 test functions we can conclude that an appropriate choice for the flanking sequences size can be made using one of the following heuristics:

Simple transposition:

Seq. Length = 10% * chromosome length ± 1,
if chromosome length ∈ [1, 40]
Seq. Length = 5% * chromosome length ± 1,
if chromosome length ∈ [41, 80]
Seq. Length = 0.3% * chromosome length ± 1,
if chromosome length ∈ [81, max]

Tournament-based transposition:

Seq. Length = 18% * chromosome length ± 1,
if chromosome length ∈ [1, 80]
Seq. Length = 5% * chromosome length ± 1,
if chromosome length ∈ [81, max]

5.2 TRANSPOSITION VERSUS 1-POINT CROSSOVER

Choosing the flanking sequences size from one of the given heuristics, the GA using transposition (with populations size of 50, 100, 200 individuals) achieved results that outperformed the results obtained with 1-point crossover with the same population size.

An interesting conclusion is that, using transposition, even with a small population of 50 individuals, the results were in almost cases much better than 1-point crossover using 50, 100 or 200 individuals. With populations of 100 or 200 individuals, transposition performance is less sensitive to the variation of the flanking sequences length. In these cases the GA using transposition obtained always better results than 1-point crossover. For instance, in Figure 4 we show the results obtained with a GA using simple and tournament-based transposition (with flanking sequences length of 4 and 3 bits, respectively) with a population of 50 candidate solutions. In the same Figure we can see the results achieved by the GA using 1-point crossover (using 50, 100 and 200 individuals). The first obvious observation is that, even with a smaller population, transposition performance is better than 1-point crossover with larger populations.

5.3 TRANSPOSITION VERSUS 2-POINT CROSSOVER

Comparing the results with 2-point crossover, we observe the same characteristics that we saw with 1-point crossover, i.e., better results with transposition in all situations, even when using smaller populations. To illustrate these results, in Figure 5, the best values obtained with simple and tournament-based transposition using 50 individuals are visible.

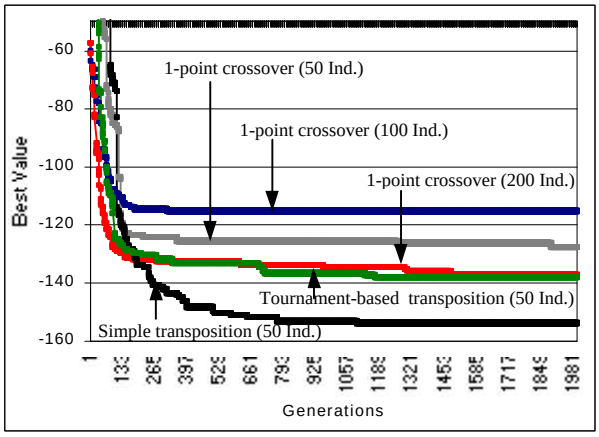

Figure 4: Transposition, 50 individuals. Comparing Results with 1-point Crossover, 50, 100 and 200 Individuals.

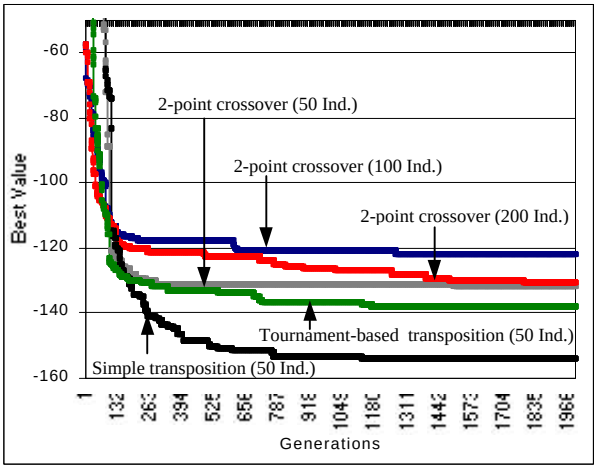

Figure 5: Transposition, 50 individuals. Comparing Results with 2-point Crossover, 50, 100 and 200 Individuals.

5.4 TRANSPOSITION VERSUS UNIFORM CROSSOVER

The GA using uniform crossover performed worse than 1-point and 2-point crossover. So, transposition mechanisms performed again much better in this case. Even using a population with only 50 individuals, the GA achieved higher quality results than uniform crossover with 100 or 200 individuals.

We show these results in Figure 6.

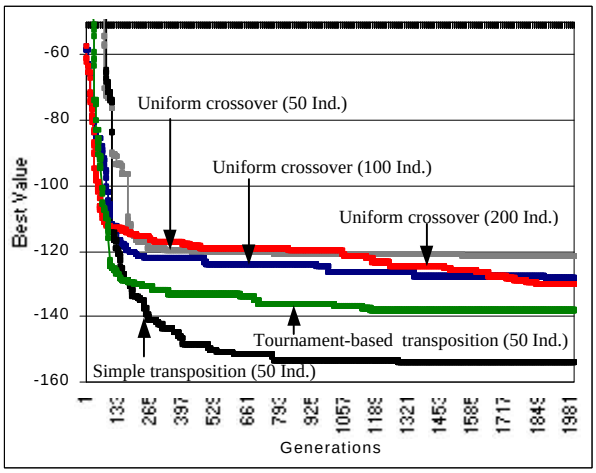

Figure 6: Transposition, 50 individuals. Comparing Results with Uniform Crossover, 50, 100 and 200 Individuals.

5.5 GENERAL OVERVIEW OF THE RESULTS

In the remaining test functions, the GA applying the transposition mechanisms (with the appropriate flanking sequence size), when compared with crossover, achieved always better results. The major advantage of transposition mechanisms is the ability to reach higher results with smaller populations than crossover operators.

In some cases (F1, F2, F3, F5, F15, F18), the maximum/minimum was always obtained either by crossover or transposition. However, the GA using the transposition mechanism had faster convergence.

The results obtained with simple and tournament-based transposition were similar in most cases. However, for some test functions, tournament-based transposition performed better than simple transposition (F4, F6, F8, F10) and for other cases the contrary was observed (F9, F12, F14 (n=4)). We couldn't find valid justifications for these results.

6 CONCLUSIONS

In this paper we extended a preliminary work using a new biologically inspired genetic operator, alternative to the traditional crossover. This genetic operator is called transposition. In addition to simple transposition, we proposed a tournament-based transposition.

The GA was implemented with two variations: one using the crossover operator; the other applying the transposition mechanisms. For both cases we compared the GA efficiency with a test suit containing eighteen test functions.

The process employed to evaluate the GA performance was off-line measure. Some parameters, such as the population size and the flanking sequences length were changed.

We concluded that transposition performance is related with the flanking sequences size: larger sequences imply worst results due to a loss of diversity. Empirical results allowed us to find some heuristics for the choice of the appropriate sequence size. Comparing the results with crossover we realized that, choosing the suitable size for the flanking sequences, transposition is always better than crossover. Besides that, even with smaller populations the GA using one of the transposition mechanisms can obtain much better results than crossover with larger populations.

Acknowledgements

This work was partially financed by the Portuguese Ministry of Science and Technology under Program Praxis XXI.

Appendix

F1: De Jong's Test Function F1 (Sphere Model)

$$F_1(x) = \sum_{i=1}^{3} x^2 * i$$

F2: De Jong's Test Function F2 (Rosenbrock's Saddle)

$$F_2(x_1, x_2) = 100 * \left(x_1^2 - x_2^2\right)^2 + \left(1 - x_1\right)^2$$

F3: De Jong's Test Function F3 (Step Function)

$$F_3(x) = \sum_{i=1}^{5} integer(x_i)$$

F4: De Jong's Test Function F4 (Gaussian Quartic)

$$F_4(x) = \sum_{i=1}^{30} i x_i^4 + Gauss(0,1)$$

F5: De Jong's Test Function F5 (Foxholes Function)

$$1/F_5(x) = \frac{1}{K} + \sum_{i=1}^{2} \frac{1}{f_j(x)}$$

$$f_j(x) = c_j + \sum_{i=1}^{2} \left(x_i - a_{ij}\right)^6$$

F6: Michalewicz's Function

$$F_6(x1,x2) = 21.5 + x1.\sin(4Px1) + x2.\sin(20Px2)$$

F7: Bohachevsky's Function I

$$F_7(x_1,x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3Px_1) - 0.4\cos(4Px_2) + 0.7$$

F8: Bohachevsky's Function II

$$F_8(x_1,x_2) = x_1^2 + 2x_2^2 - \\ -0.3[\cos(3Px_1)0.4\cos(4Px_2)] + 0.3$$

F9: Griewangk's Function

$$F_9(x) = 1 + \sum_{i=1}^{n}\left[\frac{x_i^2}{4000}\right] - \prod_{i=1}^{n}\left[\cos\left(\frac{x_i}{\sqrt{i}}\right)\right]$$

F10: Rastrigin's Function

$$F_{10}(x) = n * A + \sum_{i=1}^{n}\left[x_i^2 - A * \cos(2Px_i)\right]$$

F11: Schwefel's (Sine Root) Function

$$F_{11}(x) = V * n + \sum_{i=1}^{n}\left[-x_i * \sin\left(\sqrt{|xi|}\right)\right]$$

F12: 6-Hump CamelBack Function

$$F_{12}(x_1,x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + \\ + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2$$

F13: Shubert's Function

$$F_{13}(x_1,x_2) = \left[\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\right] * \\ * \left[\sum_{i=1}^{5} i\cos[(i+1)x_2 + i]\right]$$

F14: N-Dimensional Function (with N=1, 2, 3, 4)

$$F_{14}(x) = \frac{1}{2}\sum_{j=1}^{N}(x_j^4 - 16x_j^2 + 5x_j)$$

F15: Two-Dimensional Function (I)

$$F_{15}(x_1,x_2) = 0.5x_1^2 + 0.5[1 - \cos(2x_1)] + x_2^2$$

F16: Two-Dim. Function (II) (with n=1, 2, 3, 4)

$$F_{16}(x_1,x_2) = 10^n x_1^2 + x_2^2 - \left(x_1^2 + x_2^2\right)^2 + \\ + 10^m\left(x_1^2 + x_2^2\right)^4$$

F17: Two-Dimensional Rastrigin's Function

$$F_{17}(x_1,x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

F18: One-Dimensional Function

$$F_{18}(x) = \sum_{i=1}^{5}\sin[(i+1)x + i]$$

References

W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone (1998). *Genetic Programming - An Introduction - On the Automatic Evolution of Computer Programs and its Applications*. CA: Morgan Kaufmann.

Y. Davidor (1989). Analogous Crossover. In J. D. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, 98-103. CA: Morgan Kaufmann.

L. Davis (1989). Adapting Operator Probabilities in Genetic Algorithms. In J. D. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, 61-69. CA: Morgan Kaufmann.

K. A. De Jong (1975). *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. Dissertation, Department of Computer and Communication Science, University of Michigan.

K. A. De Jong and W. M. Spears (1992). A Formal Analysis of the Role of Multi-Point Crossover in Genetic Algorithms. *Annals of Mathematics and Artificial Intelligence* **5**(1): 1-26.

K. A. De Jong (1993). Genetic Algorithms Are NOT Function Optimizers. In L. Darrell Whitley (ed.), *Foundations of Genetic Algorithms 2*, 5-17. CA: Morgan Kaufmann.

P. D'Haeseleer (1993). Context Preserving Crossover in Genetic Programming. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, vol. 1, 256-261. IEEE Press.

D. B. Fogel (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press.

J. A. Foster (1995). *On Test Suits*. Lecture Notes on Genetic Algorithms.

T. Furuhashi, K. Nakaoka and Y. Uchikawa (1994). A New Approach to Genetic Based Machine Learning and an Efficient Finding of Fuzzy Rule. In T. Furuhashi (ed.), *Proceedings of the 1994 IEEE/Nagoya-University World Wisepersons Workshop*. Lecture Notes in Artificial Intelligence, vol. 1011, 173-189. Springer-Verlag.

D. E. Goldberg (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc.

J. L. Gould and W. T. Keeton (1996). *Biological Science*. W. W. Norton & Company.

I. Harvey (1996). The Microbial Genetic Algorithm. Submitted as a Letter to *Evolutionary Computation.* MIT Press.

J. H. Holland (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. 1st MIT Press edition, MIT Press.

G. H. Koon and A. V. Sebald (1995). Some Interesting Test Functions for Evaluating Evolutionary Programming Strategies. In J. R. McDonnell, R. G. Reynolds, D. B. Fogel (eds.), *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, 479-499. MIT Press.

K. Mathias and D. Whitley (1992). Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem. In R. Männer and B. Manderick (eds.), *Proceedings of Parallel Problem Solving from Nature 2*, 219-228. North Holland-Elsevier.

Z. Michalewicz (1994). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag.

M. Mitchell and S. Forrest (1994). Genetic Algorithms and Artificial Life. *Artificial Life* **1**(3):267-289.

M. Mitchell (1996). *An Introduction to Genetic Algorithms*. MIT Press.

N. Nawa, T. Hashiyama, T. Furuhashi and Y. Uchikawa (1997). A Study on Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm with Adaptive Operator. *Proceedings of the IEEE International Conference on Evolutionary Computation*, 589-593. IEEE Press.

N. Nawa and T. Furuhashi (1998). Bacterial Evolutionary for Fuzzy System Design. *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*. IEEE Press.

N. Nawa, T. Furuhashi, T. Hashiyama and Y. Uchikawa (1999). A Study of the Discovery of Relevant Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm. *IEEE Transactions on Industrial Electronics*.

M. A. Odutayo (1996). *An Empirical Study of Crossover and Conjugation Using the De Jong's Test Bed of Optimization Functions.* M. Eng. Degree in Software Engineering Dissertation. (unpublished work), City University.

R. Parsons, S. Forrest and C. Burks (1995). Genetic Algorithms, Operators, and DNA Fragment Assembly. *Machine Learning*, 1-24.

P. J. Russell (1998). *Genetics*. 5th edition, Addison-Wesley.

J. D. Schaffer and L. J. Eshelman (1991). On Crossover as an Evolutionarily Viable Strategy. In R. K. Belew, L. B. Booker (eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, 61-68. CA: Morgan Kaufmann.

A. Simões and E. Costa (1999a). Transposition: A Biologically Inspired Mechanism to Use with Genetic Algorithms. In the *Proceedings of the Fourth International Conference on Neural Networks and Genetic Algorithms.* Springer-Verlag.

A. Simões and E. Costa (1999b). Enhancing Transposition Performance. *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press.

A. Simões (1999). *Transposition: A Biologically Inspired Mechanism and its Integration in the Classical Genetic Algorithm.* Unpublished M.Sc. Dissertation Thesis (in Portuguese), University of Coimbra.

P. Smith (1996a). Finding Hard Satisfiability Problems Using Bacterial Conjugation. Presented at *the AISB96 Workshop on Evolutionary Computing*. University of Sussex.

P. Smith (1996b). Conjugation - A Bacterially Inspired Form of Genetic Recombination. *In Late Breaking Papers of the First International Conference on Genetic Programming.* Stanford University, California.

W. M. Spears and K. A. De Jong (1991). An Analysis of Multi-Point Crossover. In G. J. E. Rawlins (ed.), *Foundations of Genetic Algorithms 1*, 301-315. CA: Morgan Kaufmann.

W. M. Spears (1992). Adapting Crossover in a Genetic Algorithm. *Technical Report AIC-92-025*. Washington, DC: Naval Research Laboratory, Navy Centre for Applied Research on Artificial Intelligence.

W. M. Spears (1993). Crossover or Mutation?. In Darrell Whitley (ed.), *Foundations of Genetic Algorithms 2*, 221-238. CA: Morgan Kaufmann.

G. Syswerda (1989). Uniform Crossover in Genetic Algorithms. In J. D. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, 2-9. CA: Morgan Kaufmann.

D. Whitley, K. Mathias, S. Rana and J. Dzubera (1995). Building Better Test Functions. In L. J. Eshelman (ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, 239-246. CA: Morgan Kaufmann.

T. Yoshikawa, T. Furuhashi and Y. Uchikawa (1997). Knowledge Acquisition of Fuzzy Control Rules for Mobile Robots Using DNA Coding Method and Pseudo-Bacterial Genetic Algorithm. *Proceedings of the First Asia-Pacific Conference on Simulated Evolution and Learning*. Lecture Notes in Artificial Intelligence, vol. 1285, 126-135. Springer-Verlag.