International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 10 (2019) pp. 065-072
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Received: 20 Dec, 2018; Accepted: 18 March, 2019; Publish: 17 April, 2019

Evolutionary Genes Algorithm To Path Planning Problems

Paulo Salgado[1] **and Paulo Afonso**[2]

[1] CITAB/Universidade de Trás-os-Montes e Alto Douro
5000-801 Vila Real, Portugal
psal@utad.pt

[2] Instituto de Telecomunicações /ESTGA, Universidade de Aveiro,
3754-127 Águeda, Portugal
pafnaa@ua.pt

***Abstract*: Genes are fundamental pieces for reproductive processes and one force field creator of the evolutionary mechanisms of the species, whose laws and mechanism are not well known. In this paper a new evolutionary optimization strategy that combines the standard genetics algorithms (GA) with selfish perspective of evolution of genes is presented. Natural selection theory is explained by a mechanism, which is centred in individuals that are the elements of a population, characterized by their chromosomes. The primary variables are the genes (characters or words), which are non-autonomous entities, grouped in a Chromosome structure (phrases of live). However, genes make their influence felt far beyond the chromosome structure (entity of the individual). Based on this paradigm, we propose the evolutionary genes algorithm (EGA) that enriches the GA with a new line field generating of evolutions. Genes-centred evolution (GCE) improve the search engine of chromosome-centred evolution (CGE) of the GA. Its impact is apparent on the increased algorithm speed, but mainly on the improvement of genetics solutions, which may be useful to solve complex problems. This approach was used to path-planning problems, in a continuous search space, to show its effectiveness in complex and interdependent sub-paths and evolution processes. GCE improved local sub-paths search as sub-processes that catalyze the CCE engine to find an optimal trajectory solution, task that the standard genetic algorithm have no ability to solve.**

***Keywords*:** Genes algorithms, genetic algorithms, path planning**.**

I. Introduction

The evolutionary algorithm (EA) is a family of stochastic problem solvers based on principles of natural evolution. During the last four successive decades, the EA and the artificial life have been proposed to solve complex engineering problems through computational simulations [1][2], particularly in optimization tasks. Genetic algorithms (GA) have become one of the most famous EA. The base of GA paradigm is that the solution(s) for a given problem can be seen as a survival task where possible solutions compete with each other for survival and the right to reproduce. This competition is the driving force behind the progress that desirable leads to an optimal solution. It is based on Darwin's natural selection theory and were extended today to the automatic evolution of computer programs. GA are now used to solve complex and multidimensional problems where other traditional methods fail or are difficult to be use [3][4].

For the theory of natural selection [5], reproduction is the leading mechanism of evolution [6], where individuals are its main agents. These base elements are selected to participate on the reproductive process, where each individual favored by its set of characteristics revealed in the performance tests. The characteristics that favor their fitness function (FF) are then transferred to the next generation [5] making these traits more common at each generation. Hereditary traits are transferred via DNA [8]. Here, the main processes involved are natural selection and reproduction, implemented by a set of genetics operators. DNA sequences can change through mutations, producing new alleles and affecting the characteristics of the underlying gene, altering the phenotype.

Natural selection is based into following three necessary and sufficient conditions [6]:

- Diversity: Individuals within a population have different phenotypes (characteristics/traits).
- Inheritance, a combination of parent's traits is transferred to offspring.
- Competition and reproduction: Individuals possessing traits well suited for the struggle for local resources will contribute more to the next generation offspring.

Frequently, due to a high number of offspring and limited local resources there are a high rate mortality. An overview of the basic process of natural selection can be found in [5] [7].

A gene-centred view of evolution is another point of view on the evolution process, initially proposed by Richard Dawkins in his book "The Selfish Gene" (1976) . In this theory, genes are the essential units of selection [9][10], having the design to replicate themselves in order to secure and perpetuate their own existence. Individual organisms contribute weakly to the evolution process; their main role is the manufacture and hosting of genes. Genes highlight their merits or failures across individuals, defining their characteristics and behaviors [11].

The performance of a gene is revealed through multiple individuals, generally taking into account the qualities of the best individuals of the latter set. Gene fitness is inevitably associated with these values, recorded in a gene performance table (collective memory of gene qualities), but also by the performance of individuals who have such a gene in their chromosome. On the other hand, each individual is subject during his or her life to a multiplicity of performance evaluation tests, where the individual and combined actions of a subset of their genes (activated genes) are revealed. These values measure the individual's different performance perspectives. Both measures (gene and individual performance) are taken into account in the natural selection of individuals.

Individuals are not the only owners of the chromosomes. These are selected for the reproduction group according to their suitability and the cumulative sum of the performance of their genes. They do this by promoting their best genes. On the other hand, genes do not have reproductive autonomy and do not work alone. The evolutionary mechanism uses these two competing processes. In one, the adaptation mechanism promotes the phenotypic effects of genes in order to maximize their representation in future generations. On the other hand, genes whose phenotypic effects benefit individuals successfully promote their own propagation.

Genes are the elementary elements that belong to the population of chromosomes and to the population of individuals who share the genetic code of the species inheritance. The genetic code of the species is the combination of genes and their chromosome structure.

This idea is transferred to the proposed evolutionary gene algorithms (EGA). It adopts the principle of the two mechanisms described above as main force fields of the evolutionary processes [12][13]. These use individuals as the main agents for natural evolution by measuring their phenotype characteristics. Genes with advantageous phenotypic effects benefit the individuals and increase their survival probability at the same time that alleles remain in gene population, promoting their own propagation and so maximizing their representation in future generations. In this perspective fitness function of individuals and of genes are correlated functions, two visible parts of a global fitness function [14]. As in GA algorithm, there is a population of individual, each one with their own chromosome (association of genes) and a fitness function. In parallel structure, all the genes of the population are placed on (virtual) population of genes whose fitness value was inferred from the performance values of individuals of the population that have that gene. Both populations were evaluated and subject to a set of genetic operations.

EGA has been tested in trajectory planning problems within a continuous and non-convex space of solutions. The main task is to find a sequence of continuously linked straight segments that connect a point of departure and a point of arrival, avoiding obstacles scattered in the navigation region with reasonable margin of safety precautions [15]. An evolutionary algorithm based on selfish and altruistic strategies has been used to solve this kind of path-planning problem, with good results [16]. Here, three algorithms were used to solve hard path-planning problems. All of them use the same number of individuals from one population or divided among a set of *nPop* parallel populations (with migration process) and the same genetic operators functions. They are: GA, the standard genetics algorithms; MGA, multi-populations GA (with migration) [17]; EGA, the GA with genes evolution strategy, and MEGA, MGA (with migration) with genes evolution strategy.

As will be seen, conventional GA algorithms often reveal difficulty in providing viable solutions to path planning problems, particularly when the research space is continuous and when there are multiple local minima. Contrary, the proposed EGA and MEGA algorithms, that combining an evolutionary strategy of individuals and genes, offers the best solution to this type of problem, overcoming the weaknesses of GA.

This paper is organized as follows. The path-planning problem is described in Section 2. Next section presents the main structure of the evolutionary gene algorithms (EGA). The main experimental results are presented in section 4. These results are compared with the results of the other algorithms to verify the improvements of the evolutionary gene strategy. Finally, the last section presents the main conclusions of this study and of the proposed algorithm.

II. Path-planning problem

In this work, the path-planning problem is formulated as the task of determining a safe and continuous path connecting two points in a workspace, *W*: the starting and the ending points. It should avoid collision with known obstacles, while, if possible, maximizing the safety clearance distance to them with a minimum distance route.

Without losing the generality, we consider that the trajectory is composed by a set of connected line segments, which join the initial and final node of the trajectory passing through *n* intermediate ordered nodes, i.e., the set $N=\{P_0,P_1,\cdots,P_n,P_{end}\}$. Obstacles are independents straight segments, where *O* is the set of their pairs of endpoints. Let $S=\{s_1,\cdots,s_n,s_{end}\}$ be a sequence of linked (*n*+1) segments, where $s_i=\overline{P_{i-1}P_i}$ is the segment that connects two consecutive nodes, the (*i*-1)th node with *i*th node. *N* are inside of *W* region, with position values $P_i\in W$. Their optimal values are unknown, with the exception of P_0 and P_{end} points, whose values are given by initial problem conditions. In most cases, these values cannot be obtained by analytical means or their estimation task is very hard to do. Generally, only near optimal values can be computed trough strategies or algorithms based on meta-heuristics or by optimization methods.

The solution of the presented path-planning problem, $N^*=\{P_0,P_1^*,\cdots,P_n^*,P_{end}\}$ is part of all set of solutions in the continuous space $W^n=W\times\cdots\times W$, that maximize a set of criterions and restrictions that could be (or could not) formalized through a function. The resolution of this (not convex) optimization problem is almost of type *NP*-complex.

The main objective of path-planning algorithms is to find the optimal ordered nodes N of the trajectory that maximize the fitness function:

$$Fitness(N) = Safety(N,O) + Feasibility(N,O)/Dist(N) \quad (1)$$

where $Dist(N)$ is the length of trajectory, the sum of segments length of the path N.

This function is a sum of two parts: the *Feasibility* function, which it is correlated with the number of segments of trajectories that intersect segments of the obstacles set O, and a *Safety* function, that measures features such as suavity and safety proximity of trajectory segments to obstacles segments.

The path N must be free of collisions. In this case, the *Feasibility* measure is a higher value, $F_{\max}$. If all segments of the trajectory S collide with obstacles, the *Feasibility* is a value zero. So, the *Feasibility* function measures the free intersections number between the S and O, given by:

$$Feasibility(S,O) = F_{\max}\left(1 - \frac{nc}{n+1}\right)^{\alpha} \quad (2)$$

where nc is the number of cross overlapping that happens between S and O segments and $\alpha \geq 1$ a shape factor, usually with value 1.

Straight line segments that link the initial-point and the end-point have the lowest distance value. However, in most cases that situation is not possible and $Dist(N) \geq \overline{P_0 P_{end}}$. Typically, this distance is about 2 to 10 times this value. The second term of equation (1) is a ratio between the feasibility and the distance of the trajectories, but where the sensibility to variable nc is higher and lower for the *Dist* variable. For that reason $F_{\max} > \overline{P_0 P_{end}}$.

Safety function values reflects the higher or lower proximity between the straight segments of the path S and straight segments of the set of obstacles O, given by:

$$Safety(N,O) = S_{\max}\left(1 - \frac{1}{n}\sum_{s\in S} e^{-D_s/\sigma}\right) \quad (3)$$

where Ds is the minimal distance between the segment $s \subset S$ and the set of segments O, i.e., $D_s = \min\left(\left|s - o_j\right|\right)$, $\forall o_i \in O$, where $\left|s - o_j\right|$ is the minimal distance between s and o_j segments. Its value is near zero if all trajectory segments are very close or intersect obstacle segments and with value $S_{\max}$ when $D_s \gg \sigma$, situation where all segments of S are fairly distant from the segments O (i.e., for $D_s \gg \sigma$). Generally, the *Safety* function has lower value when compared with second part of equation (1).

Path-planning problem is here taken as an optimization problem that will be solved by GA, MGA, EGA and MEGA algorithms. They will be used to find the best waypoints at right sequence to define the optimal path, by maximization the fitness equation (1). This task is hard complex because the choice of i^{th} node is dependent of the choice made for the previous one, i-1. The following parameters values were used: $F_{\max} = 200$; $S_{\max} = 20$, $\alpha = 1$, $\sigma = 5$ and n=20 for an example with 20 randomly placed obstacles.

III. Evolutionary algorithm

Evolutionary gene algorithms (EGA) use the GA structure, but they have a new additional (virtual) population whose elements are genes. These are collected from all chromosomes of the population.

In the context of the path-planning problems, a chromosome of the i^{th} individual represents a potential solution given by a sequence of nodes, $N_i = \{P_0, P_{i,1}, \cdots, P_{i,n}, P_{end}\}$ codified by a set of genes G_i and with a performance value given by the fitness function:

$$F_i = Safety(N_i,O) + Feasibility(N_i,O)/Dist(N_i) \quad (4)$$

The fitness of the i^{th} individual at k^{th} generation, $FI_i^{(k)}$, is given by a randomly weighed sum of $F_i^{(k)}$ with a sub-set of the most relevant genes, $G_i^* \subset G_i$:

$$FI_i^{(k)} = r \cdot F_i^{(k)} + (1-r) \cdot \frac{1}{ng} \sum_{g \in G_i^* \subset G_i} FG_g^{(k-1)} \quad (5)$$

where $FG_g^{(k)}$ is the fitness value of gene $g \in G_i^* \subset G_i$. ng is the number of elements of the set G_i^*. The random parameters r have values in interval [0,1] with density of probability of the uniform distribution, i.e., $r \in U([0,1])$.

The gene fitness value measures its performance in context of its historical behaviour as well as of its participation in the behaviours of news chromosomes, given by:

$$FG_g^{(k)} = \max\left(FG_g^{(k-1)}, \max_{i \supset g} F_i^{(k)}\right) \quad (6)$$

If g is a new gene, then $FG_g^{(k)} = 0$.

Moreover, if g mutates then its fitness value is computed based on a similarity factor with other genes. Let $S_{gh} \in [0,1]$ be a similarity factor between gene g and gene h.

If $S_{gh} > S_{thresh}$ and $FG_h > FG_g$, then

$$FG_g^{(k)} = FG_g^{(k)}(1 - S_{gh}) + S_{gh} FG_h^{(k)} \quad (7)$$

with S_{thresh} the threshold of similarity.

The processes of the proposed EGA are as follows:

Step 1. Create a random initial population or multi-population of n individuals.
Step 2. Evaluate the population through the fitness function (4).
Step 3. Calculate the fitness of genes with eq. (6).
Step 4. Select individuals for reproduction based on fitness values given by (5).
Step 5. Crossover parent's chromosomes to produce a child solution. Transferred genes carry their performance values.
Step 6. Mutate some chromosomes of the population. Update the fitness of individuals and genes (7).
Step 7. Renew the population with these offspring individuals.

Step 8. Repeat steps 2–6 until the specified number of generations is reached.

For the selection process are used the *Tournament Selection* and *Roulette Selection* strategies. For genetic crossover operator have been used the “Natural” and “Real” crossover. We used four mutation operators: *Uniform*, *Border*, *Perturbation* and *WeigthedGenesMutation*, the last one being used only by EGA and MEGA algorithms. With the last mutation function, genes with poor fitness have higher probability to mutate. Most of these methods and operators are well described in literature and are part of most practical implementations of evolutionary algorithms [18][19].

In the multi-population MGA and MEGA algorithms there are migration mechanisms that exchange elements between populations. Let *r* and *s* are the index of two populations that belong to a Multi-population structure that containing a number of *nPop* distinct populations. The probability of exchange of the i^{th} element of population *r* with the j^{th} element of population *s* is a random quantity directly proportional to the normalized dissimilarity measure between populations $DS(r,s)$, a value in the $[0,1]$ interval, according to the following equation:

$$\mathrm{P}(i \in r, j \in s) = p \times \mathrm{P}_m \times DS(r,s) \tag{8}$$

with $p \in U([0,1])$ a random valor and P_m the maximum probability of migration between populations.

The dissimilarity measure between the *r* and *s* populations is computed by sum of square difference among the correspondent’s alleles of the best chromosome of two populations. The normalize dissimilarity measure is done by:

$$DS(r,s) = \frac{\sum\limits_g \left(G_g^r - G_g^s\right)^2}{\max\limits_{k,l} \sum\limits_g \left(G_g^k - G_g^l\right)^2} \tag{9}$$

where G_g^i, for $g = 1,\cdots,n$, are the genes of the best chromosome of the population *r* in the current generation. So, $DS(r,s)$ is a null value when the best chromosome of the r^{th} population is equal to the best chromosome of the s^{th} population. It is an unitary value when the dissimilarity of the best chromosomes of *r* and *s* populations is a maximum value, for all set of combinational dissimilarities measure of populations, i.e., with:

$$\sum_g \left(G_g^r - G_g^s\right)^2 \geq \sum_g \left(G_g^k - G_g^l\right)^2, \ \forall k,l \tag{10}$$

for any pair of combinations (k,l) of populations.

GA, EGA, MGA and MEGA are used to solve the same path-planning problem. Each gene consists of a sequence of pair-wise positive real values that represent a node of the trajectory. A chromosome is a sequence of genes, i.e. a sequence of waypoints of the path. Structures with one or multiple population (with migration facilities) are tested to solve this problem. The EGA and MEGA are the algorithms that incorporate the proposed evolutionary gene strategy. In literature there are many other method based on GA algorithms to solve the path planning problem [20][21][22][23][24].

IV. Results

Two examples problems has used to test and validate the algorithms. Conventional GA and MGA algorithms as well as EGA and MEGA algorithms has used to solve the same complex path-planning problem. These have the same structure of preceding algorithms, but incorporate evolutionary gene strategy. For both examples, test results of four algorithms are presented and discussed in this section.

For both example problems, the total number of individuals was 250, size of population of the GA or EGA algorithms. The same number of individuals was used by MGA and MEGA algorithms, by divided them among 10 populations of the multidimensional structures. MGA and MEGA incorporate migration processes with a maximum probability value P_m of 1%. Experiments has executed until the 50th generation. The statistical results here presented are for the last generation, namely the mean fitness values, the means of maximum and minimum of fitness values of populations. For both example problems, the path must link the initial point of coordinates (0,0) with the end point with coordinates (100,0) and the workspace, *W*, has the square frontier border with a side length of 100 units. All geometrical components are described in two dimensions (2D).

In first example, an object diamond crystal shape, with an upper cover, are both placed inside *W*. The diamond object is draw by the connection of five closed straight segments lines while three connected segments lines draw the hat. This workspace is used in all experiments test, namely by GA, MGA, EGA and MEGA algorithms.

The path problem have three feasibly solution with near performances values, but with an optimal trajectory line curve that connect the initial point to end point by a straight-lines that passing between the hat and the diamond object. However, find the optimal value is not a trivial task. Each chromosome has 12 elements, corresponding to 6 waypoints (nodes) *N*, all inside the workspace.

GA results are shown in figure 1, where the red line represents the best solution. It was not able to find a feasible path, i.e., by avoiding obstacles. Moreover, the population evolutions converge for a restrict zone of *W*. The fitness value of population over generations is depicted in figure 2.

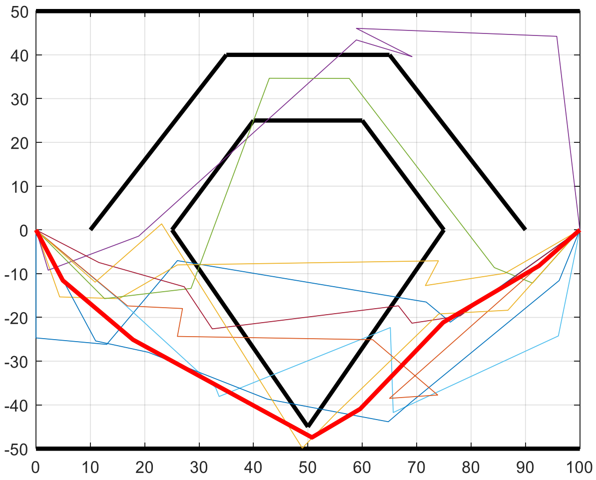

Figure 1. Trajectories results of the GA algorithms (1 population with 250 individuals)

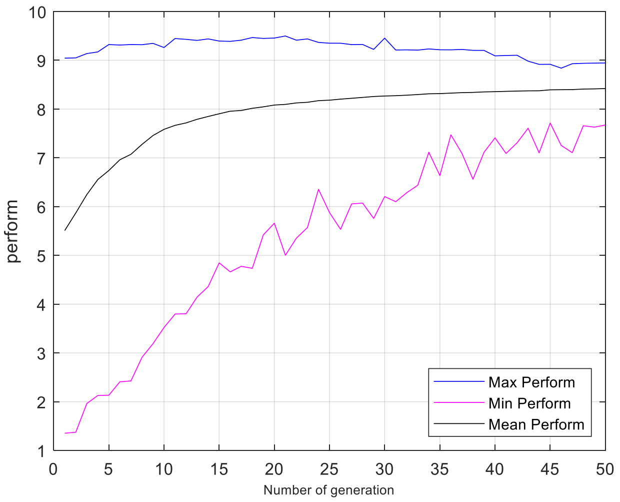

Figure 2. Fitness value of populations (minimum, maximum and mean values) over the generations.

With multi-population GA structure (MGA) the results are a little better and present a feasible solution, but with the same convergence problem of GA, as shown in figure 3. The performance of populations over generations is shown in the figure 4.

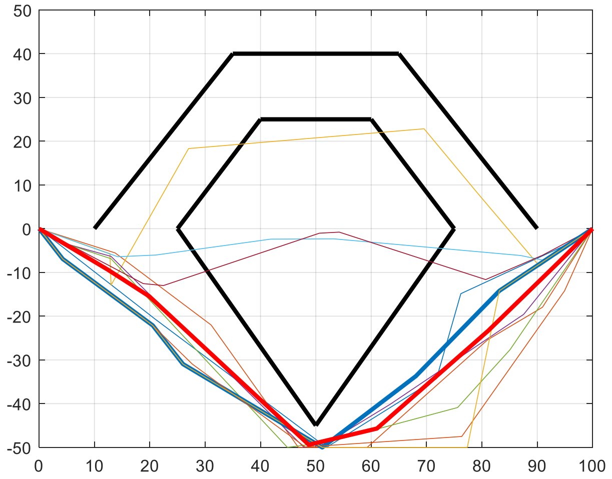

Figure 3. Trajectories results of the MGA algorithm (10 populations each with 25 individuals).

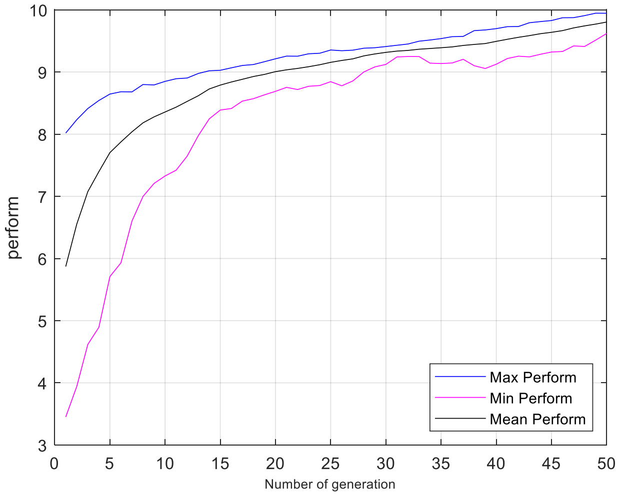

Figure 4. Performance of MGA algorithm over generations.

Results are significantly better when the algorithms incorporate the evolutionary gene strategy, as is the case of EGA and MEGA, as shown in figure 5 and 7. There, the best paths solutions found by each population are plotted, where red line is the best solution. All best solutions have fitness values around 11.3. Most present solutions circumvent the obstacles safely with a minimum length path. Moreover, the genetic diversity of the population also ensures a more global demand for the solution, with each populations providing a good and un-repeated solution. MEGA needs 14 generations to achieve the average performance of value 9 (rising time, *rt*) whereas the MGA algorithm needs 20 generations. These and other values are presented in Table 1.

Algorithm	Performance of BI	Mean performance of BI’s	*rt* (level 9)
GA	11.241	8,421	>> 50
EGA	11.391	9.263	45
MGA	11.380	9.804	20
MEGA	11.406	10.122	14

BI- best individual; *rt*- rising time;

Table 1. Main results of GA, EGA, MGA and MEGA algorithms.

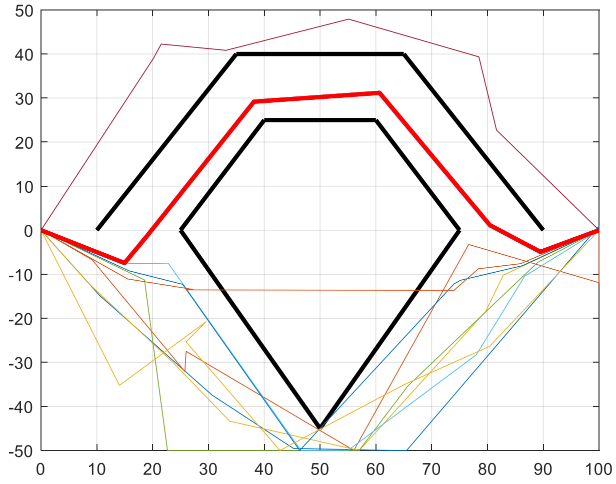

Figure 5. Trajectories results of the EGA algorithm´ (1 population with 125 individuals).

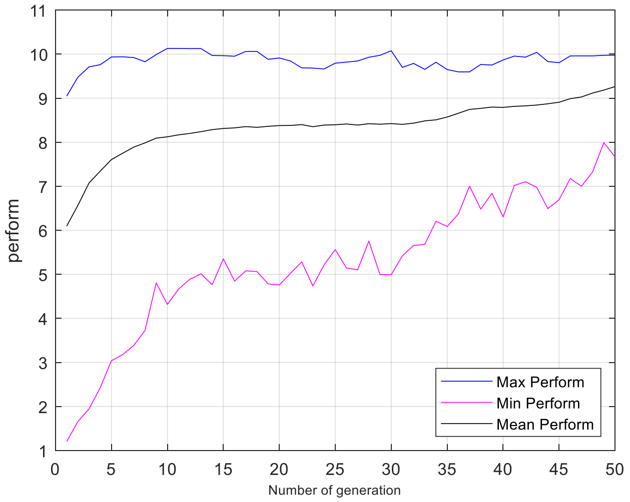

Figure 6. Performance of EGA algorithm over generations.

In the second example, the workspace, *W*, has the square border with length of 100 units. Inside there are 20 straight lines objects randomly placed. The GA, MGA, EGA and MEGA algorithms are used to find a feasible and safety path the connect the initial point (0,0) to end point (100,0). The problem has various feasible solutions, but to find the optimal value is not a trivial task. Moreover, this test example has enough pitfalls to make it difficult to execute most algorithms, including those of the evolutionary type. Each chromosome has 22 elements, corresponding to 11 waypoints (nodes) *N*, all inside the workspace.

GA and MGA results are shown in figure 7 and figure 9, respectively. They was not able to find a feasible path, i.e., by avoiding obstacles. The red line represents the best solution. With multi-population GA structure (MGA) the results are a little better and present a feasible solution, but with the same convergence problem of GA (see figures 8 and 10).

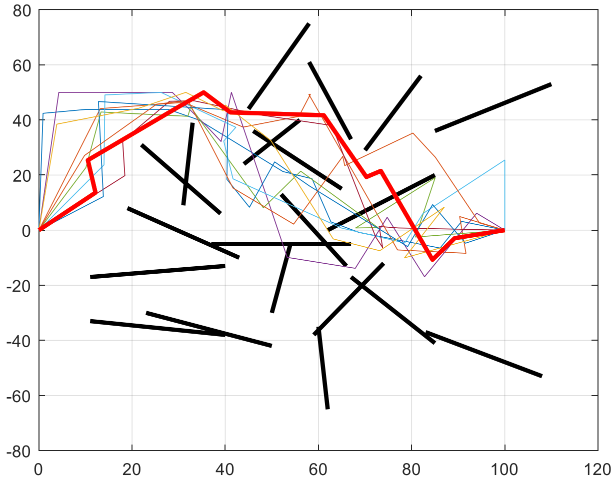

Figure 7. Trajectories results of the GA algorithms (1 population with 250 individuals)

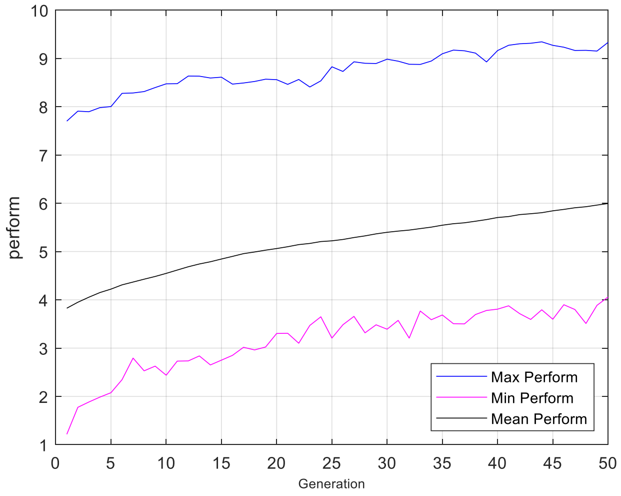

Figure 8. Performance of GA algorithm over generations.

Results are significantly better when the algorithms incorporate the evolutionary gene strategy, as is the case of EGA and MEGA, as shown in figure 11 and figure 13, respectively. There, the best paths solutions found by each population are plotted, where red line is the best solution. All best solutions have fitness values around 11.3.

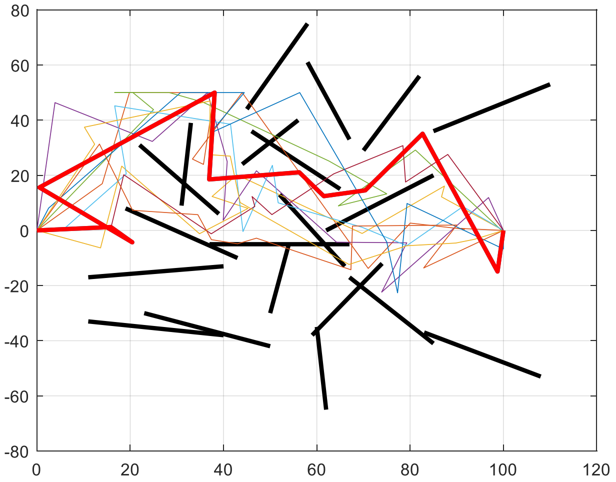

Figure 9. Trajectories results of the MGA algorithm (10 populations each with 25 individuals).

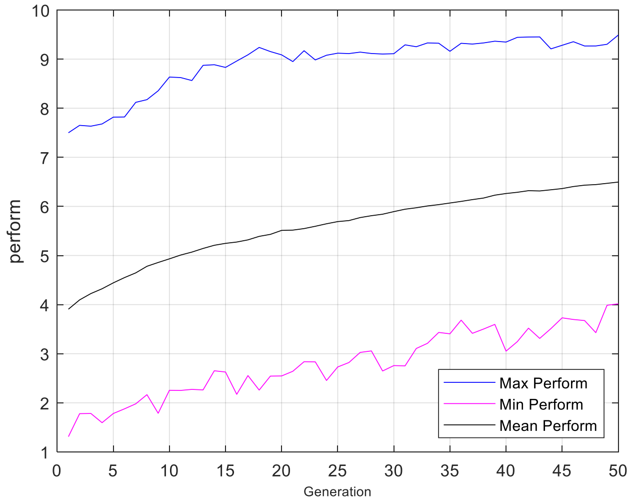

Figure 10. Performance of MGA algorithm over generations.

Most present solutions of EGA and MEGA algorithms circumvent the obstacles safely with a minimum length path. Moreover, the genetic diversity of the population also ensures a more global demand for the solution, with each populations providing a good and un-repeated solution. Both methods have a better generational convergence than GA and MGA. At the same time, they have high levels of performance (medium, maximum and minimum) and best final results, with the MEGA method being slightly better (see Figures 12 and 14), while preserving the genetic diversity of populations over the generations. MEGA needs 25 generations to achieve the average performance of value 10 (rising time, *rt*) whereas the EGA algorithm needs 42 generations. These and other values are presented in Table 2.

algorithms.

Algorithm	Performance of BI	Mean performance of BI's	rt (level 10)
GA	9.803	-	>> 50
EGA	10.624	-	>>50
MGA	11.368	11.212	42
MEGA	11.315	11.265	25

Table 2. Main results of GA, EGA, MGA and MEGA

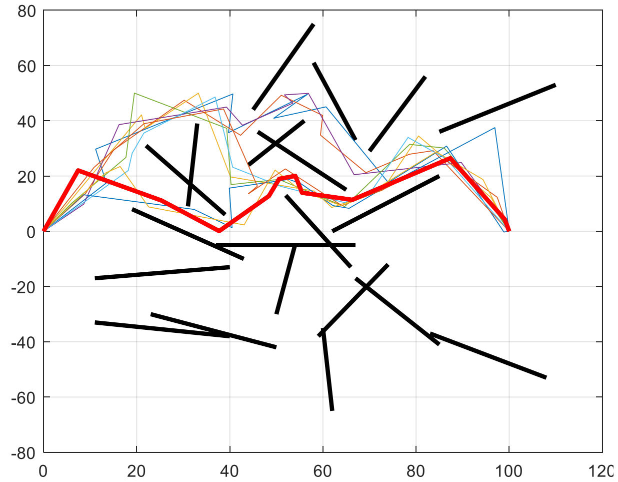

Figure 11. Trajectories results of the EGA algorithms (1 population with 250 individuals)

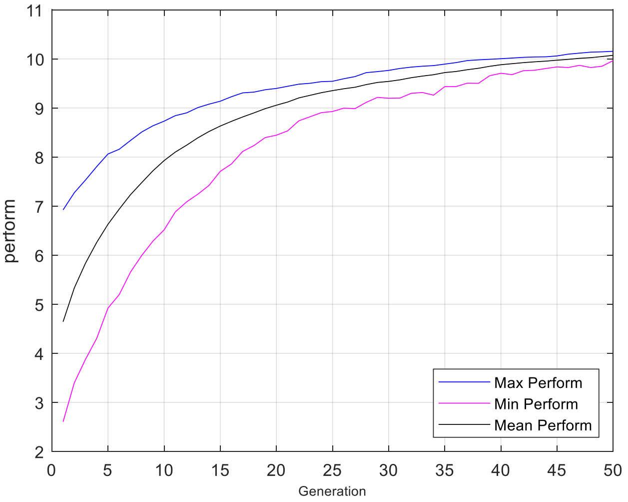

Figure 12. Performance of EGA algorithm over generations.

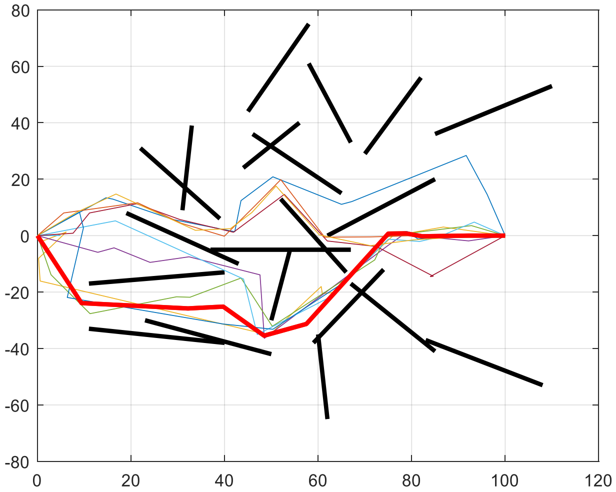

Figure 13. Trajectories results of the MEGA algorithms (10 populations each with 25 individuals)

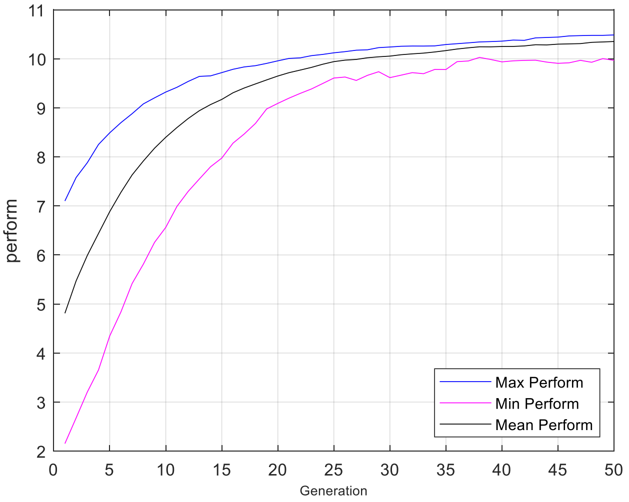

Figure 14. Performance of MEGA algorithm over generations

V. Conclusions

In this paper, a new evolutionary gene strategy was proposed for improved evolutionary algorithm, as well-known AG and MGA algorithms. Essentially, this new strategy gives new field forces to the evolution mechanisms. Its impact is apparent on the improvement of algorithm speed, but mainly on the improvement of quality and numbers of genetics solutions, particularly when applied to solve complex problems.

So, this approach was used on path-planning problems, in a continuous search space. It has shown to be effective in complex and interdependent sub-paths and evolution processes. GCE improved local sub-paths search as sub-processes that catalyze the CCE engine to find an optimal trajectory solution, a task that the standard genetic algorithm is not able to solve.

References

[1] D. Simon. *Evolutionary Optimization Algorithms*, John Wiley & Sons, 2013.

[2] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to Biology, Control, and Artificial Intelligence*, MIT Press, Bradford Books Editions, 1975. (Reprint, ISBN 978-0262581110, 1992)

[3] B.Y. Qu, Y.S. Zhu, Y.C. Jiao, M.Y. Wu, P.N. Suganthan, J.J. Liang, "A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems", *Swarm and Evolutionary Computation*, 38, pp. 1-11, 2018.

[4] I. Zelinka. "A survey on evolutionary algorithms dynamics and its complexity – Mutual relations, past, present and future", *Swarm and Evolutionary Computation*, 25, pp. 2-14, 2015.

[5] T. Ryan Gregory, "Understanding Natural Selection: Essential Concepts and Common Misconceptions", *Evolution: Education and Outreach*, 2(2), pp. 156–175, 2009.

[6] P. Godfrey-Smith, "Conditions for Evolution by Natural Selection", *Journal of Philosophy*, 104, pp. 489-516, 2007.

[7] D. Zeigler, "Natural Selection", in Evolution, David Zeigler (eds), Academic Press, Chapter 2, pp. 9-22, 2014.

[8] V. V. Efremov, "Equilibrium between Genetic Drift and Migration at Various Mutation Rates: Simulation Analysis", *Russian Journal of Genetics*, 41(9), pp. 1055-1058, 2005.

[9] A. Burt and R. Trivers, *Genes in conflict: the biology of selfish genetic elements*, Cambridge, MA: Belknap Press, 2006.

[10] J.C. Avise, "1976 Selfish Genes", In *Conceptual Break-throughs in evolutionary Genetics*, John C. Avise, (eds), Academic Press, Chapter 48, pp. 101-102, 2014. ISBN 9780124201668.

[11] S. Okasha, "Population Genetics", in *The Stanford Encyclopedia of Philosophy*, (Fall 2015 Edition), Edward N. Zalta (ed.), 2015.

[12] R.M. Sibly, R.N. Curnow, "Evolution of discrimination in populations at equilibrium between selfishness and altruism", *Journal of Theoretical Biology*, 313, pp. 162-171, 2012.

[13] H. Demsetz, "Seemingly altruistic behavior: selfish genes or cooperative organisms", *Journal of Bioeconomics*, 11(3), pp. 211-221, 2009.

[14] K. Kleiner, "The selfish gene that learned to cooperate", *New Scientist*, 191(2564), pp. 13, 2006.

[15] P. Salgado, G. Igrejas and P. Afonso, "Hybrid PSO-cubic spline for autonomous robots optimal trajectory planning", In *Proceedings of the INES 2017 - 21st International Conference on Intelligent Engineering Systems*, pp. 131-136, Oct. 20-23, Larnaca, Cyprus, 2017.

[16] P. Salgado, G. Igrejas and P. Afonso, "Evolutionary Based on Selfish and Altruism Strategies - An Approach to Path Planning Problems", In *Proceedings of the IEEE International Conference on Intelligent Systems (IS)*, Madeira, Portugal, 2018.

[17] H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, H. Zhou, "Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey", *Swarm and Evolutionary Computation*, Vol. 44, pp. 365-387, 2019. ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2018.04.011.

[18] O. Kramer, *Genetic Algorithm Essentials*, Studies in Computational Intelligence, Springer, 2017.

[19] G. Wu, R. Mallipeddi, P. N. Suganthan, "Ensemble strategies for population-based optimization algorithms – A survey", *Swarm and Evolutionary Computation*, Vol. 44, pp. 695-711, 2019. ISSN 2210-6502, https://doi.org/10.1016/j.swevo.2018.08.015.

[20] S. Karakatič, V. Podgorelec, "A survey of genetic algorithms for solving multi depot vehicle routing problem", *Applied Soft Computing*, Vol. 27, pp. 519-532, 2015. ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2014.11.005.

[21] C. Lamini, S. Benhlima, A. Elbekri, "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning", *Procedia Computer Science*, Vol. 127, pp. 180-189, 2018. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2018.01.113.

Author Biographies

Paulo Salgado - Received the B.A.Sc. degree in electric and computer engineering, the M.A.Sc. degree in electric and computer engineering, from the Faculty of Engineering, University of Porto, Portugal, in 1989 and 1994. He received the Ph.D. degree in electric and computer engineering at 1999 from Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal.

He is currently University Associate Professor in the Department of Engineering, School of Science and Technology, University of Trás-os-Montes e Alto Douro, Vila Rea, Portugal. He is a research member of CITAB (Centre for the Research and Technology of Agro-Environmental and Biological Sciences), Portugal. His current research interests include soft-computing, intelligent control, pattern recognition, fuzzy logic, neural networks and time series analysis techniques from a systems design engineering and theorical perspectives. He is the author or coauthor of four books, 6 journal papers, as well as many conference articles.

Paulo Afonso - Received the B.A.Sc. degree in electric and computer engineering, from FEUP (Faculty of Engineering) of Porto University, Portugal, in 1989. He received the Ph.D. degree in chemical engineering, in the simulation and process control domain in 1999 from the Universidade de Coimbra, FCTUC (Faculdade de Ciências e Tecnologia), Portugal.

He is currently a Coordinator Professor in the ESTGA - Escola Superior de Tecnologia e Gestão de Águeda, Universidade de Aveiro, Portugal. He is research member of IT - Instituto de Telecomunicações, Aveiro, Portugal.

His current research interests include robotics, electronic instrumentation, soft-computing, intelligent control, pattern recognition, fuzzy logic, neural networks and time series analysis techniques from a systems design engineering and theoretical perspectives. He is the author or coauthor of one book, 7 journal papers, as well as many conference articles.