Adaptive Parameterization of Evolutionary Algorithms Driven by Reproduction and Competition

Mauro Annunziato, Stefano Pizzuti
ENEA – Agency for New technologies, Energy and Environment
'Casaccia' R. C., Via Anguillarese 301, 00060 Rome Italy
Phones: +39-0630484405, +39-0630484411, Fax: +39-0630484811
email:{mauro.annunziato, stefano.pizzuti}@casaccia.enea.it

ABSTRACT: running a genetic algorithm entails setting a number of parameter values. Finding settings that work well on one problem is not a trivial task and a genetic algorithm performance can be severely impacted. Moreover we know that in natural environments population sizes, reproduction and competition rates, change and tend to stabilise around appropriate values according to some environmental factors. This paper deals with a new technique for setting the genetic parameters during the course of a run by *adapting* the population size and the operators rates on the basis of the environmental constrain of maximum population size. In addition genetic operators are seen as alternative reproduction strategies and fighting among individuals is introduced. Finally benchmarks of the proposed strategy on classical optimization problems are shown. The results show that the parameters reach an equilibrium point and that performances on the considered problems are very good.

KEYWORDS: genetic and evolutionary algorithms, dynamic variation of population size, adaptive parameterisation, artificial evolution

INTRODUCTION

We know, Roughgarden (1979) and Song (1988), that in natural environments population sizes of species, together with their reproduction and competition rates, change and tend to stabilise around appropriate values according to some factors such as natural resources and carrying capacity of the ecosystem.
Unfortunately in standard genetic algorithms, Holland (1975), Goldberg (1989), these features, like population size, crossover and mutation probabilities, are stuck, from generation to generation, on a priori defined values. It means that running such an algorithm entails setting the values of such parameters. Finding settings that work well on one problem is not a trivial task; if poor settings are used, a genetic algorithm performance can be severely impacted. It is clear that in this situation we actually have to deal with two optimization problems : the problem itself and the setting of the GA parameters.
Furthermore, the reduction of the number of external parameters of a GA can be seen as a first step towards achieving a problem-dependent self-adaptation of the algorithm.
Within this framework the proposed paper deals with a new solution for dynamically setting the parameters of genetic algorithms during the course of a run.
The main feature of our methodology is variable population size based on free interaction among individuals. To implement such a feature we reconsidered some basic concepts.
First we reviewed the traditional concept of selection in GAs versus that of direct competition among individuals. Second the two classical genetic operators, crossover and mutation, are viewed not so strictly genetic but, in a little more evolutionary way, like two different ways of reproducing.
In this context, as proposed in Jefferson (1995), the metaphor for crossover, mutation and chromosome moves towards *bisexual reproduction*, *mono-sexual reproduction* and *individual* and the general underlying idea is that of moving from the genetic level towards that of artificial evolution of societies and artificial life.
On the base of population size we define reproduction and competition rates. Obviously since population size is variable then such rates are variable too. Reproduction probabilities are defined on the basis of the environmental availability of finite natural resources. Roughly speaking we say that the higher the population size is and the less the resources are. It means that, because of the lack of resources, as the population size increases the competition rate gets higher and the reproduction probabilities decrease.

To test our ideas we used as testbed common instances of the travelling salesman problem (TSP). We wish to underline that the aim of our work is not a study of such a problem, we used this benchmark only to provide individuals with a fitness criterion and to show the effectiveness of the proposed adaptive parameterisation to solve optimization problems.

RELATED WORK

The idea that one should adapt one's genetic algorithm during the course of a run is not a new one.
Shaefer (1986) has built an approach to numerical optimization around this idea in his ARGOT program. The thing adapted in ARGOT is the representation itself, and pursuing this idea has led Shaefer to some interesting results.
Shaffer (1987) has investigated encoding crossover points on chromosomes, so that information controlling crossover locations evolves during run.
Whitley (1987) has investigated the adaptation of the reproductive fitness of a parent based on performance of its offspring. The idea described here is that one adapts the probability that a genetic operator will be applied in reproduction based on the performance of *the operator's* offspring.
Davis (1989) proposed that an adaptive mechanism should alter the probability of applying an operator in proportion to the observed performance of the individuals created by that operator in the course of a run. Such a mechanism rewards the operator that produces a good child and the operator setting the stage for this production because purely local measures of performance were seen to be not sufficient.
In Cartwright (1991) the use of information gathered from the problem's data space, before the algorithm begins, is suggested to assist in the choice of parameters.
Bäck (1992) presented an approach in which a basic idea from *Evolution Strategies* (ESs) is transferred to GAs. Mutation rates, instead of being handled as a global constant external parameters, are changed into endogenous items which are adapting during the research process. In this work experimental results indicate that environment-dependent *self-adaption* of appropriate settings for the mutation rate is possible even for GAs.
One of the widest studies on the topic is that of DeJong (1993) in which the author proposed a classification of the control parameter setting strategies.
Fernandes (1999) presents niGAVaPS, a genetic algorithm that mimics some mechanisms of evolution in natural environment populations. niGAVaPS tunes the population size parameter according to the state of the search process and adapts the reproduction rate as function of the current population size. Each chromosome has a lifetime, which corresponds to the number of generations in which it will remain in the population. This value is calculated according to the chromosome fitness and population characteristics at the moment of its creation.
Other studies can be found in Davis (1991) and Schaffer (1989).
In this framework it has to be mentioned that the particular subject which has been the most extensively studied is that of dynamically adjusting the population size during one EA run.
In his classical paper DeJong (1975), studied, among other GA aspects, the optimal population size for a set of numerical functions, experimenting with values ranging from 50 to 100.
Grefenstette (1986) used a meta-level GA to control the parameters of another GA, finding population size values between 30 and 80, but his results were only slightly better than DeJong's.
Goldberg (1989b) stated that a small initial population size can lead to premature convergence, since there are few schemata to process in the initial population. On the other hand, a large population results in long computational time to get improvements, imposing a large computational cost per generation.
This assertion was also sustained by an extensive empirical study, Schaffer (1989), on the effects of changes in control parameters on the (on-line) performance of genetic algorithms on function minimization problems. The authors concluded that a large population can achieve a large sampling of search space, although imposing a large cost per generation, and noticing that the search operators can also reach the schemata not present in the initial population.
Moreover Goldberg (1989b), described a reasoning on the real-time Rate of Schema Processing (how many vs. in how much time), which should be maximized. Results suggest that small population sizes should be selected for serial GA implementations and large population sizes for perfectly parallel GA implementations. Small populations process fewer schemata, faster, and converge faster than large populations, where diversity and slow convergence prevails.
According to Reeves (1993), the choice for an initial population size should not be a random process, and points to a relation with alphabet cardinality. He presents a formula for the probability that at least an allele being present at each locus in an initial random generated population, thus indirectly supplying a possible way to estimate initial population size, related to desired initial population diversity and search space coverage.
A technique for dynamically adjusting the population size with respect to probability of selection error, based on Goldberg (1991), is presented in Smith (1993).
Michalewicz (1994) presented a genetic algorithm, GAVaPS – the Genetic Algorithm with Varying Population Size, that introduces the concept of *age* or *lifetime* of the chromosome, namely the number of generations that a chromosome

stays “alive”, which in turn replace a need for a selection mechanism. Population size increase is obtained only by reproduction events, in which all individuals have equal chance of being chosen. The algorithm proceeds in a generational manner increasing at each time step each individual’s *age*. When an individual’s age exceeds its (previously fixed) lifetime, it is removed from the population. This concept of age replaces the selection concept, in the sense that there is no replacement of individuals other that their lifetime limit imposes.
In Schlierkamp-Voosen (1996), a mechanism of competing sub-populations with variable size is presented, which is dynamically adjusted according to certain criteria. There is a quality criterion for each group, as well as a gain criterion, which dictates the amount of change in the group’s size. The mechanism is conceived making all groups, except the best, to have its size decreased at each size change by the same percentage. The size of the best group is the only one to increase.
In Hinterding (1996) an experiment with a self-adaptive (with respect to mutation probability), steady-state GA is presented, in which three sub-populations have their sizes adjusted at regular intervals - *epochs*, defined in terms of number of genome evaluations - according to the current state of the search. The idea is to maintain the three groups apart in their sizes, so that each of the groups may search with different strategies, and also to maximize the performance of the group with the midmost size. The criteria used for varying the sizes is (best of sub-population) fitness diversity.
Costa (1999) proposed an empirical comparative study of evolutionary algorithms in which a classification framework for dynamical control of population size in EAs is proposed. In this work the authors showed that, when no previous information exists, choosing a dynamic random variation control strategy for the population size is a reasonable choice, outperforming blind choices for the fixed settings.
Although it has been recognized by the evolutionary computation community that population size plays an important role in the success of the problem solving process, there is still limited understanding of the effects and merits of dynamically adapting this parameter DeJong (1993) and this subject is still an open research topic (see, for instance, Grefenstette (1986), Schlierkamp-Voosen (1996), Balazs (1999)).

THE ADAPTIVE EVOLUTIONARY ALGORITHM : FROM GENETICS TOWARDS ARTIFICIAL EVOLUTION OF POPULATIONS

Life and evolution in nature are organised at least into four fundamental levels of structure, Jefferson (1995) : the molecular level, the cellular level, the organism level and the population level. At present the tendency is to study the molecular level through wet-bench lab techniques (wetware), the cellular and population levels with software simulations and the organism level with hardware (robotic) experiments.
What we propose is a population level based evolutionary algorithm for optimization problems. The aim underlying our approach is not to recreate nature as it is, but it is to move from classical genetic algorithms towards artificial evolution of populations composed by individuals able to meet and interact.
In this context when two individuals meet then they can interact in two ways : by reproducing (bisexual reproduction) or by fighting for natural resources (the stronger kills the weaker), otherwise the current individual can differentiate (mono-sexual reproduction). Like in standard GAs all these features are probabilistic, the difference is that probabilities are adaptive instead of being a priori fixed.
The main environmental constrain driving the adaptation mechanism is the maximum population size, namely the resources of the ecosystem, and it is the only parameter to be set.

```
Procedure AdaptiveEA {

        Initialize()

        While( not termination condition ) {

                evaluate()

                getTwoIndividuals();
                if(meeting){
                        if(reproduction)
                                biSexsualReproduction(); //Xover
                        else
                                fight(); // the weaker is removed from population
                }
                else
                        monoSexsualReproduction(); // mutation
        }

}
```

Figure 1: Algorithm

ADAPTATION RULES

The main feature of our algorithm is the adaptability of the parameters. Population size is limited by the environmental limit and its dynamics are determined by the reproduction and competition rules among individuals. These two adaptive rates are defined as :

$$Pr = 1 - \frac{Cp}{Mp} \quad [1]$$

where Pr is the reproduction rate, Cp is the current population size and Mp is the maximum population size.

$$Pc = 1 - Pr \quad [2]$$

where Pc is the competition rate and Pr is the reproduction rate.

Roughly speaking we are imposing the rule that if the population density is low then the reproduction rate is high and the competition rate is low. Vice versa, because of the lack of environmental resources, the competition rate gets higher and the reproduction rate decreases.

INITIALIZATION

Traditionally in GAs the initialisation of the population is performed on an a priori fixed number of individuals and it is totally random. In this strategy the initial size of the population is a random number, limited by the maximum size, of individuals and a few of them are initialized with a greedy strategy which depends on the fitness criterion. This is done because small rates of the population initialised in such a way improve speed convergence towards good solutions keeping at the same time good diversity.

SELECTION AND MEETING

In classic Genetic Algorithms, before crossover, the selection stage is performed, Holland (1975). This step is meant to model the natural feature that the fitter an individual is and the higher the probability to survive is. In the traditional approach this is implemented as a roulette wheel with slots weighted in proportion to the fitness values of the individuals. Through such a wheel a linear search is performed Goldberg (1989) or an intermediate population Whitley (1993) is created.

In our evolutionary algorithm we replaced this stage with the meeting concept. At each iteration we pick the i^{th} individual of the population, for i from 1 to population size, up and then we randomly look for a second individual. The meeting probability is thus defined as the population density

$$Pm = \frac{Cp}{Mp} \quad [3]$$

where Cp is the current population size and Mp is the maximum population size.

If someone is met then interaction will start, else mono-sexual reproduction of the current individual might occur.

In this way everyone has a chance of mating and bio-diversity is enhanced, computational time is reduced because the $O(n)$, where n is the population size, selection routine based on roulette wheel is removed and memory is saved because the intermediate population is not needed anymore.

CROSSOVER AND BISEXUAL REPRODUCTION

Crossover is that operator which allows two chromosomes to mutually exchange their genetic equipment giving rise to new sons. In traditional Genetic Algorithms when two chromosome meet then mating is performed with a probability a priori defined. If it happened then the two offsprings resulting from the crossover operator would replace their parents, for instance if they were fitter, in the population, else nothing happens and both originals individuals would survive in the population. This mechanism insures the population size to be constant.

The modification we introduce moves towards bi*sexual reproduction*: mating is performed according to the adaptive rate *Pr* [1], and if it occurred then the resulting sons would not replace their parents, they would simply be added to the population. In this situation population increases of two new elements.

This is much more natural than the genetic crossover because when two individuals mate and have sons these, generally, do not kill their parents as they bear.

When the population reaches its maximum limit then reproduction gets destructive in the sense that sons will replace their parents if fitter. This ensures, although this limit, the evolution of the specie.

COMPETITION

Competition starts according to the adaptive rate *Pc* [2]. It means that when two individuals meet and they do not mate then they fight for survival, the stronger kills the weaker and this one is kicked from the population off. This is essential to let the population not to explode in size because in this way the population size decreases of one unit.
This simple strategy mimics in a more nature-like way than the “roulette wheel selection” the evolutionary concept of the *survival of the fittest* because we leave the environment to determine the survivals.

MUTATION AND MONO-SEXUAL REPRODUCTION

Traditionally mutation is that operator which gets one chromosome and randomly changes the alleles of one or more genes. Generally this operator is performed with a probability a priori defined and if it occurred then the mutated chromosome would replace the original one. In this case the population size is preserved because the new solution destroys the original one.
Another point of view is that of considering mutation as non-destructive moving towards the concept of *mono-sexual* reproduction. It is performed according to the adaptive reproduction rate *Pr* [1] and when it occurs an individual first clones itself and then mutates. The mutated individual doesn’t replace the original one, it is simply added to the population and the population size increases of one unit.
Like bisexual reproduction when population reaches its maximum value then we perform mutation by replacing the original individual with the mutated one if fitter.

TERMINATION

The algorithm ends when a maximum number of iterations are reached.

GLOBAL POPULATION DYNAMICS

The formula underlying our algorithm which describes the population grow rate is defined in such a way :

$$N(t+1) = N(t) + 2P_mP_rN(t) - P_m(1-P_r)N(t) + (1 - P_m)N(t)P_r \qquad [4]$$

Where N(t+1), N(t) are the population size at time t+1, t, P_m is the meeting probability [3], P_r is the reproduction probability [1]. The term $2P_mP_rN(t)$ refers to the bisexual reproduction step, $P_m(1-P_r)N(t)$ refers to the competition stage and $(1 - P_m)N(t)P_r$ refers to mono-sexual reproduction case. By developing the calculations then [4] reduces to

$$N(t+1) = 2N(t)\left(1 - \left(\frac{N(t)}{Mp}\right)^2\right) \qquad [5]$$

Where Mp is the maximum population size.
Dividing [5] with Mp then we get

$$X(t+1) = 2X(t)(1 - X(t)^2) \qquad [6]$$

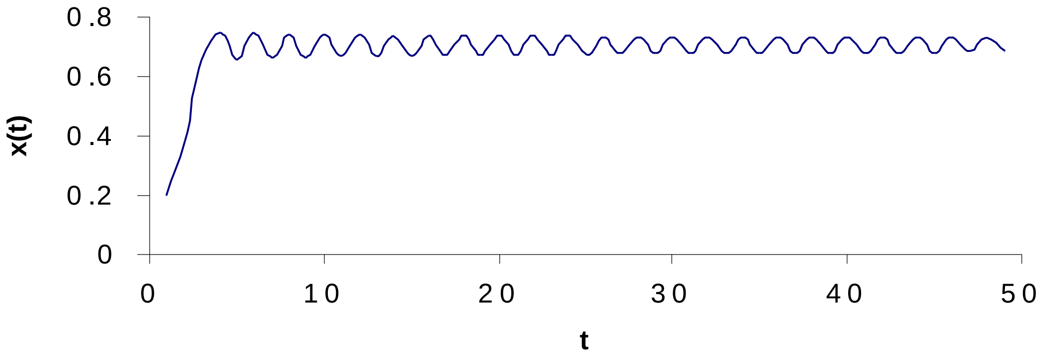

Figure 2: Theoretical behaviour of population dynamics according to equation [6]

Finally we wish to close this section remarking the main differences with the related work of the last fifteen years. The introduction of competition is maybe the most relevant one. In fact it allows us to replace the classic genetic selection stage with the new meeting concept and selection is left to the environmental dynamics. Individuals die because of fighting.
Population size is not constrained by any resizing scheme, it is freely determined by the balance of the underlying environmental interactions (reproduction and competition).
Adaptation rules of the operators are driven by population density within an environment with finite natural resources.

APPLICATION TO OPTIMIZATION PROBLEMS : THE TRAVELLING SALESMAN PROBLEM

The Travelling Salesman Problem (TSP) is one of the most famous and studied problems. Its formulation is: “given *n* towns find the minimal tour such that each town, except the first, is visited exactly once”. It is known to be a NP hard problem and it is often used as benchmark for algorithms and approaches. In the following paragraphs we describe how the genetic features of our algorithm have been customized to solve TSP. We wish to underline that we did not use the best operators for this problem because our task is not that of studying this specific problem, we are only interested in demonstrating the effectiveness of our approach to solve optimization problems and in providing a fitness criterion to allow us to analyse the dynamics of the algorithm.

CODING

We enumerate the *n* towns with integer numbers so that one possible solution is a permutation of *n* numbers. This represents the sequence to follow considering the first gene as the following of the last. In this way we define the length of the genotype as *n*, the number of towns, and we code the i-th gene as an integer *j*, between 1 and *n*, representing that at the i-th step town *j* is visited.

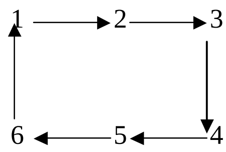

Figure 3: Example of tsp coding for 123456

INITIALIZATION

Given *n* the initial population size and *m* the number of towns then *n-m* individuals are initialized randomly while *m* are initialized according this simple greedy strategy : start from each of the different towns and then choose the nearest town not visited yet until all towns are visited.

BISEXUAL REPRODUCTION

The crossover operator is implemented in a very simple way. We take one, randomly chosen, crossover point *p*, with 0<*p*<*n-1*, then two new individuals are generated. Each inherits the genes from 0 to *p-1* directly from one of the two parents and then the remaining genes are filled with the towns not visited yet in the order of the other parent.

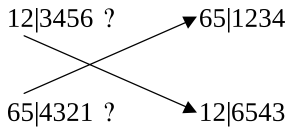

Figure 4: Bisexual reproduction example

MONO-SEXUAL REPRODUCTION

Because of the integer coding of the solution and to preserve the constrain that each town must be visited exactly once we simply implemented the mutation operator by swapping the alleles of two randomly chosen genes (ex. **4**23**1**56 ? ? **1**23**4**56)

FITNESS

As fitness function we adopted the sum of the Euclidean distance between the connected towns.

EXPERIMENTAL RESULTS

In the following part we rep ort the experimental results we curried out on three classical problems, see Reinelt, Oliver (1987), Eilon (1969), : Oliver30 (30 towns), Eilon51(50 towns) and Eilon76(75 towns).
The following tables compare the experimental results of best known real and integer, in brackets, solutions with different techniques and show results on the average behaviour of the adaptive parameters.

TOWNS	**GREEDY**	**SIMPLE G.A.**	**ADAPTIVE E.A.**	**BEST KNOWN**
30	473.32 (469)	425.94 (423)	**423.74 (420)**	**423.74 (420)**
50	505.77 (503)	443.98 (441)	428.98 (427)	**427.86 (425)**
75	612.65 (605)	568.95 (563)	553.16 (546)	**542.31 (535)**

Table I:. Comparisons of the best results among different strategies

From this table it is possible to notice that the Adaptive EA performs pretty well. The solutions found are very close to the best ones, the worst performance is that of 75 towns in which it is achieved a result which is about 4% worse than the best. Moreover we wish to remark the comparison with the simple genetic algorithm because it has been curried out using the same implementation of the operators used for the adaptive strategy. From this comparison it is clear the improvement of the solutions found.

Max Pop. Size	**1000**	**3000**	**5000**	**7000**	**10000**
Average pop. size	730	2189	3643	5087	7293
density	0.73	0.729	0.729	0.727	0.729

Table II:. Average results on population size

Max Pop. Size	**1000**	**3000**	**5000**	**7000**	**10000**
Bisexual rate (%)	30.43	30.4	31.03	31.7	30.96
Monosex rate (%)	9.68	9.67	9.52	9.28	9.58
Compet rate (%)	40.07	40.06	40.53	40.98	40.52

Table III:. Average results of the adaptive rates as function of the max population size

From these two tables one remark is needed. From table II we can notice that the population density, defined as the ratio between the average and the maximum population size, is almost constant and it ra nges in the interval 0.727-0.73. This value can be considered, see also figure 5, the equilibrium point around which the population oscillates. This result is of particular interest because it represents the *natural* constant balance of the different interactions, reproduction and competition, of individuals dealing with finite environmental resources. Moreover the equilibrium point tends to the average, as in the theoretical case of equation [6], and the experimental dynamics are very similar to those expec ted in figure 2. The slight differences can be ascribed to the random generator.
From table III and the following figures we can notice the same behaviour. All the parameters tend to stabilize near an equilibrium point, the average value, around which they oscillate.

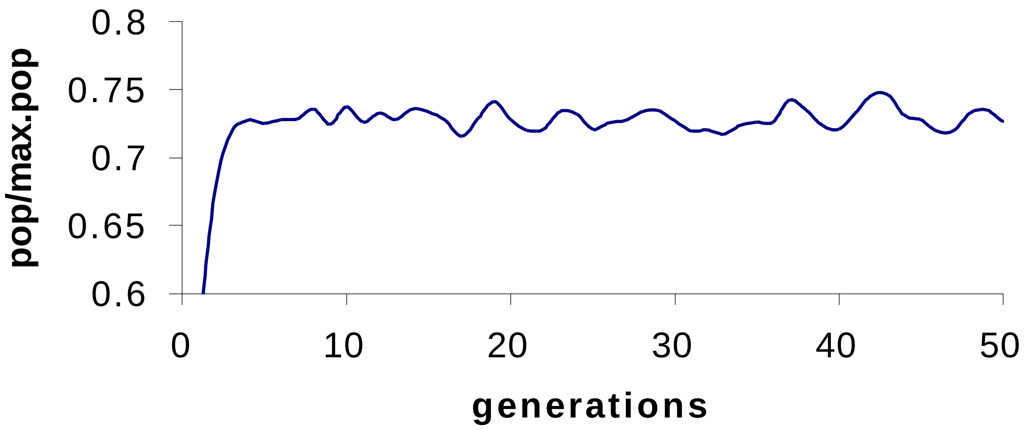

Figure 5: Example of density population dynamics

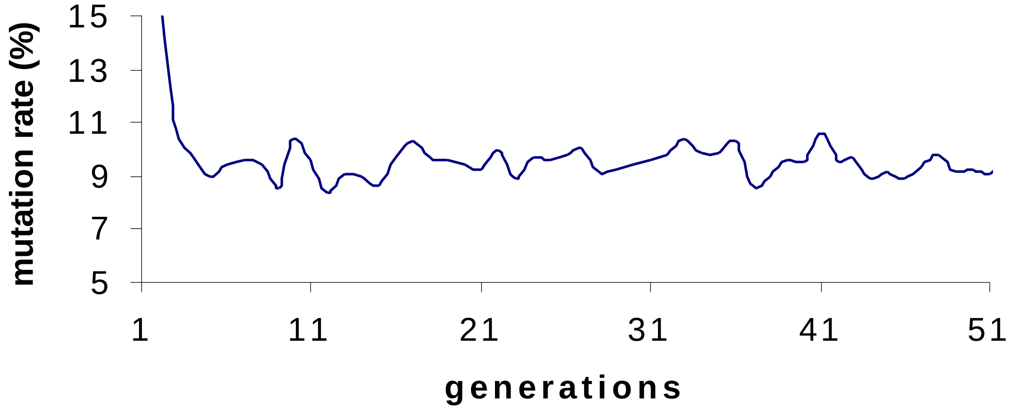

Figure 6: Example of mono-sexual rate dynamics

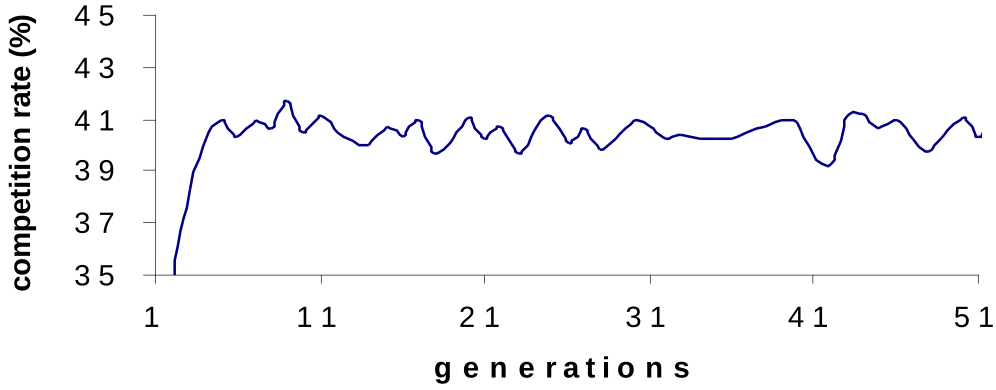

Figure 7: Example of competition rate dynamics

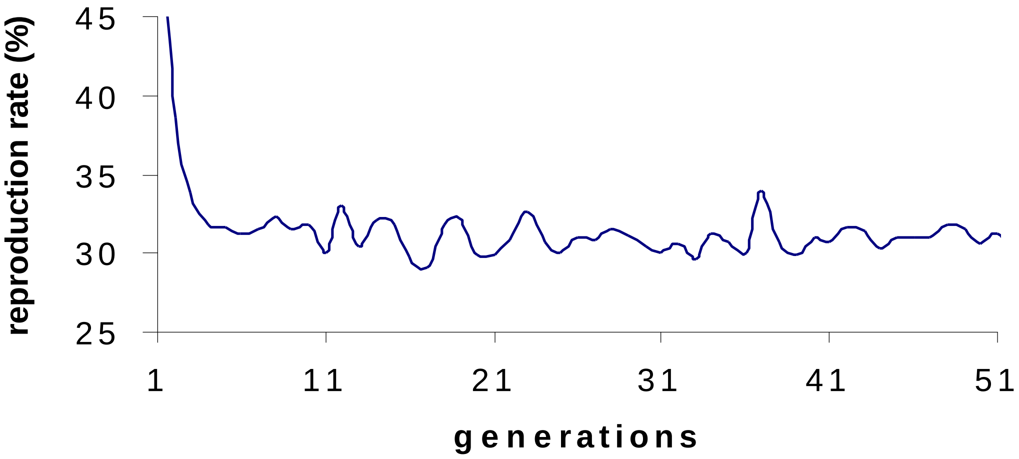

Figure 8: Example of bisexual rate dynamics

Finally we have to say that the equilibrium points around which the parameters te nd to stabilize are not dependent on the initial population size which is randomly chosen.

CONCLUSIONS AND FUTURE WORK

In this paper an adaptive evolutionary algorithm is proposed with the aim of moving towards simulations of natural evolution without external parameter settings. The key idea of the proposed strategy is the concept of adaptive parameterisation driven by reproduction, competition and dynamic population.
To achieve this task the classic genetic operators, crossover and mutation, are seen as two different ways of generating new sons, without replacing the parents, and real competition among individuals is introduced.
Dynamic population, obviously, removes the constrain of population size. This is important because such a parameter is the most studied and the one that mainly affects the result of traditional genetic algorithms. In the proposed approach population size is determined at each generation by the balance between reproduction and competition.
The main environmental constrain driving the adaptation mechanism is the maximum population size, namely the resources of the ecosystem, it is the only parameter to be set and reproduction and competition rates are defined as function of it.
To study the parameters dynamics and to test the effectiveness of the adaptive EA to solve optimization problems we applied our methodology to classic benchmarks: the 30, 50 and 75 towns travelling salesman problems. The experimental results have shown a great capability to find very good solutions, at most 4% worse than the best known solutions. However our main task is not that of studying such a particular problem. The most interesting results are those concerning the parameters dynamics. In fact such a study shows that the algorithm finds itself the parame ters values out by, as result of the underlying interactions, oscillating around equilibrium points which are independent on the starting conditions (initial population size). Moreover it is of particular interest that all the parameters tend to stabilize around almost the same values which does not depend on the maximum population size.
In this situation we achieved our main task : to curry out an adaptive evolutionary algorithm which does not depend on external parameter settings.
The line of research for future work is that which stresses the simulation of natural evolution. For example sex and age may be introduced in the individuals, new reproduction and competition strategies can be developed and different adaptive rules can be studied. Moreover it is possible to go further by introducing artificial life concepts. For instance physical space can be introduced and adaptive rates, instead of being global environmental parameters, can be defined inside the individuals as endogenous features.

ACKNOLEDGEMENT

Authors wish to thank SOPIN S.p.A. – Company for Computer Science for its financial support which allowed us to conduct such a study.

REFERENCES

Bäck, T. , 1992, "Self-Adaptation in Genetic Algorithms", Proceedings of the First European Conference on A rtificial Life (263-271), MIT Press, Cambridge, Massachusetts.

Bäck, T. , 1996, "Evolutionary Algorithms in Theory and Practice", Oxford University Press

Balazs, M.E. , Ritchter, D. L., 1999, "A Genetic Algorithm with Dynamical Population: Experimental Re sults". Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference, pp 25-30. Morgan Kaufmann.

Cartwright, H. M., Mott, G. F., 1991, "Looking Around: Using Clues from the Data Space to Guide Genetic Algorithm Searches", Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications (108-114), Morgan Kaufmann

Costa, J.C. , Tavares, R. , Rosa, A. , 1999, "Experimental Study on Random Variation of Population Size", Proceedings IEEE SMC, vol VI, pp 607-612, Tokyo

Davis, L. , 1991, "Handbook of Genetic Algorithms", Van Nostrand Reinhold

Davis, L. , 1989, "Adapting operator probabilities in genetic algorithms", Proceedings of the Third International Conference on Genetic Algorithms and Their Applications (61 -69), Morgan Kaufmann

Davis, L. , 1991, "Handbook of Genetic Algorithms", Van Nostrand Reinhold

DeJong, K. , 1975, "An analysis of the behavior of a class of genetic adaptive systems", (Doctoral dissertation, University of Michigan), Dissertation Abstracts International 36(10), 5140B, (University Microfilms No. 76-9381)

DeJong, K., Spears W. , 1993, "On The State of Evolutionary Computation", Proceedings of the Fifth International Conference on Genetic Algorithms and Their Applications (618 -623), Morgan Kaufmann

Eiben, A. E., Hinterding, R., Michalewicz, Z. , 1998, "Parameter Control in Evolutionary Algorithms", Technical Report 98-07, Department of Computer Science, Leiden University, The Netherlands

Eilon, S., Watson-Gandy, C.T.D., Christofides, N. , 1969, "Distribution management: mathematical modeling and practical analysis", Operational Research Quarterly, 20, 37-53

Fernandes, C., Tavares, R., Rosa, A., 1999, "niGAVaPS – Outbreeding in Genetic Algorithms" presented at Evonet Summer School 99 , Antwerp(Belgium)

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

Goldberg, D. E. , 1989b, "Sizing populations for serial and parallel genetic algorithms", Proceedings of the Third International Conference on Genetic Algorithms and Their Applications (70-79), Morgan Kaufmann

Goldberg, D. E., Deb, K., Clark, J. H. , 1991, "Genetic Algorithms, Noise, and the Sizing of Populations", IlliGAL Report No. 91010, Department of General Engineering, University of Illinois

Goldberg, D. E., 1990, "A note on Boltzmann Tournament Selection for Genetic Algorithms and Population -Oriented Simulated Annealing", TCGA Report 90003, Dept. Engineering Mechanics, University of Alabama

Grefenstette, J.J., Gopal R., Rosmaita B.J., Van Gucht D. , 1985, "Genetic Algorithms for the Travelling Salesman Problem", Proc. of the an International Conference on Genetic Algorithms and their Applications, 160 -168

Grefenstette, J.J. ,1986, "Optimization of Control Parameters for Genetic Algorithms". IEEE Transactions on Systems, Man and Cybernetics, SMC-16(1), (122-128)

Hinterding, R., Michalewicz, Z., Peachey, T. C. , 1996, "Self-adaptive Genetic Algorithm for Numeric Functions", Proceedings of the 4th Conference on Parallel Problem Solving from Nature (420-429), in Lecture Notes from Computer Science, Springer-Verlag

Holland, J. H. , 1975, "Adaptation in natural and artificial systems", MIT Press, Cambridge, Massachusetts

Holland, J. H. , 1993, "Royal Road Functions", Genetic Algorithms (GAList Internet mailing list) Digest, volume 7 issue 22, August 12, 1993, Article dated August 7

Jefferson, D., Taylor, C. , 1995, "Artificial Life as a Tool for Biological Inquiry, in Artificial Life: an Overview", edited by C. G. Langton, MIT press, 1-10

Langton, C.G. , 1987, "Artificial Life", Proceedings of an Interdisciplinary Workshop on Synthesis and Simulation of Living Systems, Addison Wesley, Los Alamos, New Mexico

Michalewicz, Z., 1994, "Genetic Algorithms + Data Structures = Evolution Programs" (second, extended edition). Springer-Verlag.

Oliver, I., Smith, D., Holland, J.H. , 1987, "A study of permutation crossover operators on the travelling salesman problem", Proc. of the 2nd International Conference on Genetic Algorithms, J. J. Grefenstette ed., Lawrence Erlbaum, Hillsdale (NJ), 224-230

Reeves, C. R. , 1993, "Using Genetic Algorithms With Small Populations", Proceedings of the Fifth International Conference on Genetic Algorithms and Their Applications (92 -99), Morgan Kaufmann

Reinelt, G. "TheTSPLIB", http://www.iwr.uni -heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html

Roughgarden, J. , 1979, "Theory of Population Genetics and Evolutionary Ecology". Prentice-Hall

Schaffer, J.D., Caruana, R.A., Eshelman, L.J., Das R., 1989, "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization". Proceedings of the Third International Conference on Genetic Algorithms and Their Applications (51 -60), Morgan Kaufmann

Schlierkamp-Voosen, D., Mühlenbein, H., 1996, "Adaptation of Population Sizes by Competing Subpopulations". Proceeding of the 1996 IEEE International Conference on Evolutionary Computation. IEEE catalog number 96TH8114.

Smith, R. , 1993, "Adaptively Resizing Populations: An Algorithm and Analysis", Proceedings of the Fifth International Conference on Genetic Algorithms and Their Applications (653), Morgan Kaufmann

Song, J., Yu, J. , 1988, "Population system control". Springer-Verlag

Whitley, D. , 1987, "Using Reproductive Evaluation to Improve Genetic Search and Heuristic Discovery" in *Genetic Algorithms and their Applications*, John J. Grefenstette, editor. Lawrence Erlbaum Associates

Whitley, D. , 1993, "A Genetic Algorithm Tutorial", Tech. Rep. CS -93-103, Colorado State University

Whitley, D., Starkweather, T., Fuquay, D. , 1989, "Scheduling problems and travelling salesman: the genetic edge recombination operator", Proc. of the 3rd International Conference on Genetic Algorithms, J.D. Shaffer ed. , San Mateo (CA), 133-140

Shaefer, C.G., 1986, "The ARGOT Strategy: Adaptive Representation Genetic Optimizer Technique", in Grefenstette , J. "Genetic Algorithms and their Applications", Lawrence Erlbaum Associates

Shaffer, D. , Morishima, A. , 1987, "An Adaptive Crossover Distribution Mechanism for Genetic Algorithms", in Grefenstette, J. "Genetic Algorithms and their Applications", Lawrence Erlbaum Associates