# Minimal Genetic Algorithm with Sexual Selection Operators

**Abstract**

Genetic algorithms (GAs) developed out of the field of evolutionary computing in the 20th century, as a method to solve function optimisation problems inspired by Darwinian natural selection. They traditionally involve using the operations of selection, recombination and mutation to maintain and evolve a population of candidate solutions (genotypes) to a problem, iteratively developing a better solution as the genotypes move up the fitness landscape. This traditional approach has proven its strength on a variety of tasks, from setting weights for neural networks [15] to path planning for UAVs [7], however most GA implementations neglect another feature present in most animal evolution, sexual selection. Since evolving, gender has largely remained a feature of most species, seemingly offering survival benefits to these organisms, and so it appears to be a worthwhile feature to model in a GA. This project thus presents and analyses the performance of a novel sexual selection-based genetic algorithm, to study whether sexual selection can be beneficial for genetic algorithm performance, and finds it performs favourably in terms of fitness achieved, population diversity and runtime compared with a non-gendered GA on several problems.

1. **Introduction**

In the field of artificial intelligence, a great deal of research is done into building algorithms for optimisation problems inspired by natural processes, the so-called topic of evolutionary computation, from ant colony optimisations [3], to swarm algorithms such as particle swarm optimisation [10] and bacterial colony optimisation [12], through to evolutionary algorithms, including the branches of genetic algorithms [8] [5] [13], evolutionary programming [6] and evolution strategies [1]. This approach has yielded many fruitful results on optimisation problems, such as the travelling salesman problem [2], setting weights for neural networks [15], and protein structure prediction [16], and continues to be employed in novel ways.

This paper will focus specifically on evolutionary algorithms, and so some further explanation of these techniques is warranted. Evolutionary algorithms take their inspiration from the process of Darwinian natural selection, in which biological systems are in a state of ongoing evolution, adapting to the conditions of the environment which they inhabit, with the more suitable organisms surviving better and passing their genes on to future generations. This leads the organisms to become highly adapted to their environment, which can be viewed through a computational lens as the population iteratively optimising their functionality to solve the problem of survival; herein lies the applicability of modelling natural selection for computational optimisation tasks.

By modelling aspects of natural selection, namely some form of selection based on fitness, mutation of genetic material in produced offspring, and passing of genetic material between parents and offspring, engineers can evolve solutions to problems over many generations using computers, rather than building a solution by hand. Within this evolutionary approach, there exist three techniques: genetic algorithms, evolutionary programming and evolution strategies, which differ mainly in the way they make use of the evolutionary operations, but all involve an iterative population-based search with random variation and selection [6]. This paper researches the first technique, genetic algorithms (GAs); GAs use the operations of crossover of genetic material between parent and offspring, mutation of the genotype, and a selection function based on the fitness of an individual to solve a given problem, and have seen many variations of the basic formula developed in the years since their creation by John Holland in the 1960s [11].

This paper aims to implement and experiment with such a variation on the basic GA by incorporating an element of sexual selection into the algorithm, based upon the work done in [13], and thus aims to answer the question of whether the introduction of sexual selection into a genetic algorithm can be beneficial for performance. Since its creation, sex has persisted in most creatures in the animal kingdom, giving benefits for survival by maintaining diversity of genetic material and specializing the different sexes on different tasks [13]. Furthermore, this paper aims to build a novel version of the sexual GA, by incorporating elements of the microbial GA proposed by [8], to form a more lightweight implementation of the sexual GA, which will be named the minimal sexual GA henceforth. Thus, the contributions of this work are two-fold: firstly, to study the effects of sexual selection on GA performance, both in terms of overall fitness, and diversity of solutions evolved; and secondly to analyse the suitability of a simplified implementation of a sexual GA.

The rest of this paper is structured as follows: section 2 contains a review of the relevant literature, including the sexual genetic algorithm of [13]; section 3 explains the details of the minimal sexual GA implementation, and the associated problems that arose during its construction; section 4 outlines the experiments done to test the hypotheses, and presents the results; section 5 is a discussion of the results, scrutinising how well the goals were achieved, and section 6 provides a brief conclusion.

1. **Literature Review**

The microbial GA

This variant of genetic algorithms was proposed by Harvey [8] as a minimal version of the genetic algorithm, still utilising the standard mechanics of mutation, recombination and selection, but in a barebones implementation. It achieves minimalism in a few ways; chiefly, it takes inspiration from the steady state GA [Whitley,’89] by producing one offspring at a time from two parents out of the population, rather than forming a whole new population out of the previous generation, as traditional GAs do. This negates the need to maintain two population lists to copy between. Along with this, it employs tournament selection, in which the fitter of 2 randomly chosen individuals wins and passes its genetic material onto the loser via crossover, instead of the usual roulette wheel selection. All this effectively serves to collapse multiple operations done in regular GAs into one, thus giving it its minimalism. The ‘microbial’ in its name refers to its similarity with bacterial conjugation, ‘where segments of DNA are transferred between two members of a population’ [8].

Sexual selection with competitive/cooperative operators for genetic algorithms

This paper by Sanchez-Velazco and Bullinaria [13] forms the basis of my work. It proposes a genetic algorithm including a concrete implementation of sexual selection, with a distinction between two genders, which may be working towards different tasks; notions of competitive and cooperative fitness, which govern fitness of the males and females respectively; and differing mutation rates in each gender, as is seen in nature. The authors also propose different crossover operators to accommodate the sexual aspect. The algorithm is shown to perform well experimentally, offering improvements over the standard GA for a number of test criteria.

Sexual Selection Mechanism for Agent-Based Evolutionary Computation

This version of a GA with sexual selection aspects, proposed by Drezewski [4], differs from the above in that there is a notion of neighbourhoods within the populations of males and females. At each time step, the males search for females for mating within their neighbourhood, and the female only accepts the mating if the given male is within the same basin of attraction for local minimum that she resides in. There is no notion of cooperative fitness, and the acceptance decision for whether two individuals mate is based upon Euclidean distance between the individuals. The authors find that the algorithm produces a good population diversity, with many local minima found on multi-modal functions; my research will similarly attempt to show whether my microbial sexual GA produces good diversity of solutions.

1. **Implementation**

To start my implementation of the minimal sexual GA, I first built an implementation of the microbial GA in Python; this was a simple process due to the minimal nature of the algorithm, requiring only a few lines of code to build. The microbial GA implementation works as follows: firstly, a population of bit-string individuals is randomly created, and the fitness of each individual is calculated, according to whatever the fitness function is; then, it loops for the number of generations specified, and on each loop it does n = size of the population tournament selections, performing crossover and mutation on the offspring, and replacing each tournament loser with its offspring. The best fitness of the population is stored at each generation.

Following this, the next step was to implement a GA with sexual selection operators. I decided to base my sexual GA on the paper in [13] as this approach allowed for optimisation of a single objective function, or multiple objective functions, and I found this greater adaptability desirable.

As mentioned in the literature review, this GA incorporates the following key factors: 2 distinctive populations, one of females and one of males, with females in the set and males in the set ; a notion of competitive fitness within gender (), which is effectively the value of the objective function for an individual, and is used directly to score the males’ fitness; a notion of cooperative fitness between genders as seen in equation 1, providing a symbiotic optimisation for ‘survival’ by the two genders – this fitness score governs the female selection, and is made up of a weighted combination of the competitive fitness of a female individual, the difference in fitness score of the chosen male and the female’s son (called ), and an age factor (called ), which serves to model nature’s link between age and fertility, using a simple triangle function with width and maximum fertility age ; different crossover operations for the male and female offspring, as shown in figure 1; and finally differing mutation rates between the two genders, which is also observed in nature. The algorithm produces two individuals, one of each gender, to maintain population size.

(1)

(2)

(3)

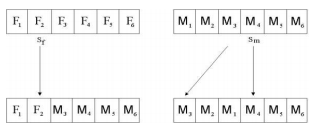


Figure - basic single point crossover for male offspring (left); reversing the mother's genetic material and crossing over for female offspring (right)

My implementation of the regular sexual GA in Python thus consists of firstly initialising 2 populations of individuals, one of females and one of males; then, a way to keep track of all the individuals’ fitness is required, and for the females the indices of the children (for calculating ) and their own age (for calculating ). I therefore build a dictionary of attributes for each of the genders (strictly speaking the male population could just have an array for fitness, but I build a dictionary for the males too for consistency and for ease of extending if desired), containing the fitness scores for the males, and the fitness, children and age for the females. Then, the algorithm loops over the number of required generations, doing, for as many individuals as there are in the population, a roulette wheel selection to obtain a male, and then the cooperative fitness of each female must be calculated taking the chosen male’s fitness into account, which can be used to then do a roulette-wheel selection of a female. Next, the crossover operations are performed as described above, followed by mutation, with different rates for each sex (the overall mutation rate adds up to the same rate as the microbial GA for consistency), to form a male and a female offspring, and finally the attributes are updated, including setting the female child’s age to zero, and incrementing all the female ages each generation. The offspring then replace the least fit individuals in each population, forming a new population each generation.

The minimal sexual GA is then a hybridisation of the two implementations above. It is largely similar to the regular sexual GA implementation, with a few key alterations to align it with the microbial GA’s minimal approach. Tournament selection, as used in the microbial GA, is employed to choose a male individual, and a female individual. This works by firstly choosing two random individuals from the male population, comparing their fitness, and setting a loser and a winner; then, two individuals from the female population are randomly chosen, their cooperative fitnesses computed, and a winner and loser are again set. Crucially, this negates the need, as seen in the regular sexual GA, to compute the fitness of the entire population of females for each iteration, massively reducing the overheads. Furthermore, as done in the microbial GA, the indices of the losing male and female are retained, and the offspring formed replace these losing individuals, providing the free elitism within each gender group.

In addition, I also included an ‘incest’ parameter to allow or prevent children mating with their parents, as a potential avenue to further maintain genetic diversity.

The final implementation of the minimal sexual GA requires 13 fewer lines of code than the regular sexual GA, thus succeeding immediately in one sense of increased minimalism; the experiments in the next section analyse the time performance increase over the regular sexual GA to confirm its minimal status, and analyse its overall fitness performance to ensure it remains a viable genetic algorithm.

1. **Experimental results**

To test the overall suitability of the minimal sexual GA and thus answer the question of whether sexual selection can be beneficial for GA performance, the following requirements will be analysed:

1. The minimalist requirement: is the timing performance of the minimal sexual GA better than the standard sexual GA?
2. Fitness performance requirement: does the minimal sexual GA perform to a similar level to both the regular sexual GA and the microbial GA?
3. Overall population quality requirement: does the minimal sexual GA produce a greater amount of good solutions to the problems?

For the experiments, the same parameters were used in the minimal and sexual GA, with a male mutation rate of 1/number of genes, a female mutation rate of 0.5/number of genes, and weights in the cooperative fitness function of equation 1 of , and in the age scaling function of equation 3, and . Thus, the mutation rate of the microbial GA was set to 1.5/number of genes, to match the overall mutation rate of the sexual algorithms.

The first problem used to test the performance of the minimal sexual GA, is the so-called pub problem, created by our lecturer, Chris Buckley, of Sussex University. In this simplistic problem, there is a party host who is trying to optimise their party arrangements. However, many of their friends do not really get on, so they think it best to host two parties, and try to obtain the optimal split of guests between the two parties so as to maximise positive interactions, and minimise negative ones. Thus, a genotype is encoded using 1s and -1s, with each gene indicating whether an individual is attending party 1 or 2, and the fitness function calculates the total positive interaction, using an ‘interaction matrix’ which specifies whether guest A likes guest B, with a 1 indicating A and B like each other, -1 indicating they don’t; equation 4 presents the fitness function ( is the -th individual’s genotype). The algorithms were run 100 times, using a randomly initialised interaction matrix each time, with 20 people (genotype length of 20); results are presented below.

(4)

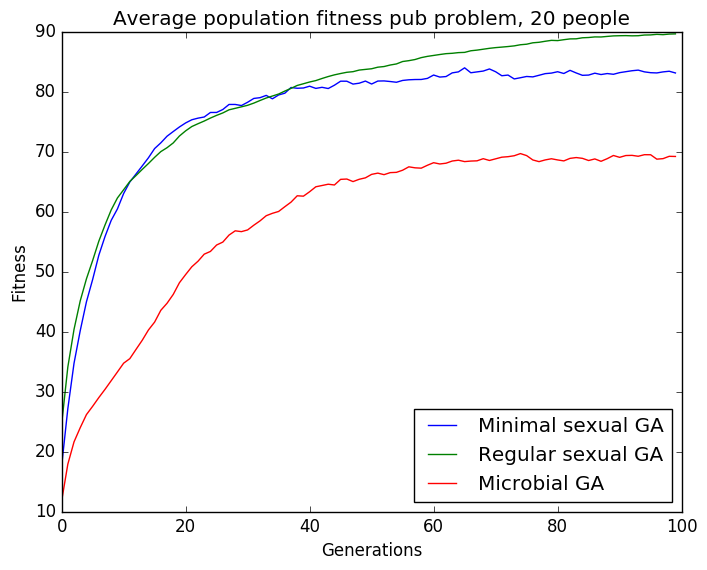
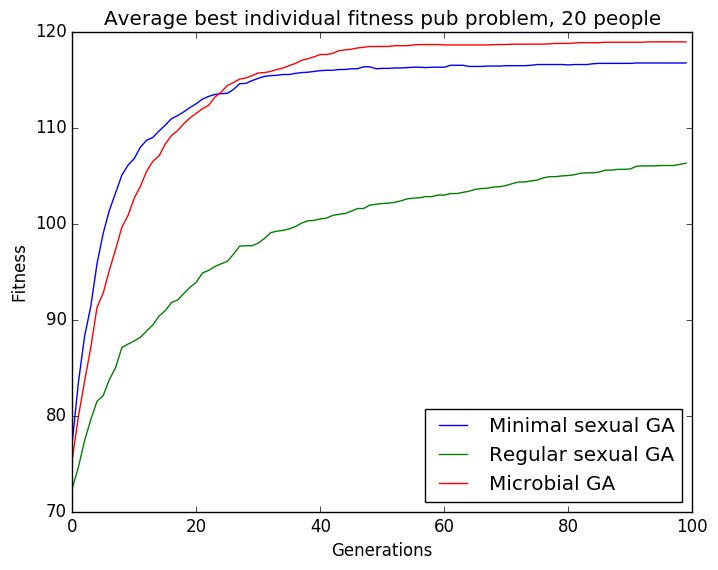


Figure 2 - pub problem over 100 runs, (left) average best individual fitness; (right) fitness average of population

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | Min Sexual GA | Reg Sexual GA | Microbial GA |
| **Avg timings for 20 runs (s)** | 9.121 +/- 0.397 | 32.017 +/- 0.743 | 9.083 +/- 0.188 |

Table 1 - average time taken (seconds) by each algorithm doing 20 runs

We can see from figure 2, that the minimal sexual GA greatly outperforms the regular sexual GA on average best individual fitness, and has comparable performance with the microbial GA, however the microbial GA does slightly outperform the minimal sexual GA interestingly. We can also see that the microbial GA and minimal sexual GA’s fitness increases steeply, reaching a value near to their maximum after only about 30 generations, showing the benefit of employing a tournament selection approach; overall, this means the fitness performance requirement appears to be met. As for the average fitness of the population, which gives an indication of the quality of the whole population, both sexual GAs greatly outperform the microbial GA, as one would expect, and each of them evolve an overall good population quickly. Thus the overall population quality requirement also appears to be met. Finally, the timing performance in table 1 shows that the minimal sexual GA vastly outperforms the regular sexual GA, and even has similar timing performance to the microbial GA. Thus, the minimalist requirement is certainly met.

Next, the algorithms were tested on a more concrete problem to confirm the behaviour, the 0-1 knapsack problem [9]. In this problem, there is a knapsack with a given capacity, and a list of items (one of each item), each with a specified benefit and weight; the goal is to fill the knapsack as to maximise the total benefit, while ensuring the total weight is below the capacity. Equation 5 specifies the fitness function, where is the benefit of item , and is 1 or 0 depending on whether item is present. The algorithms were again run 100 times, using 50 items, and randomly initialised benefit and weight values (each between 1 and 30); results are below.

(5)

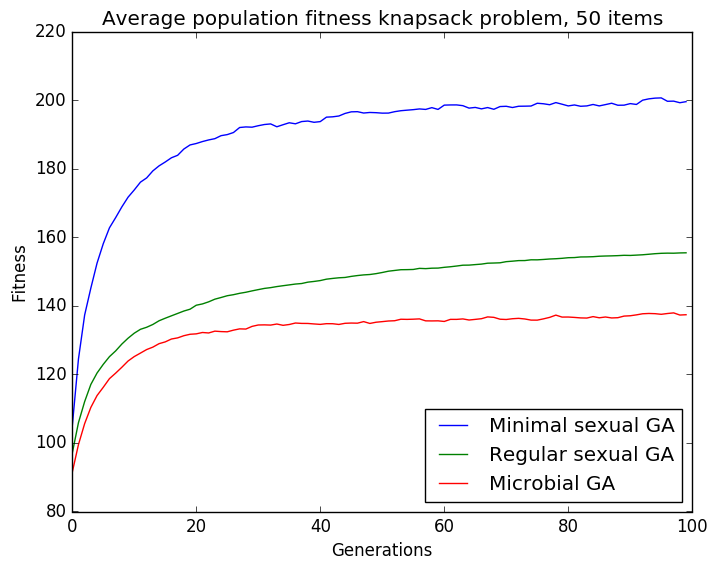
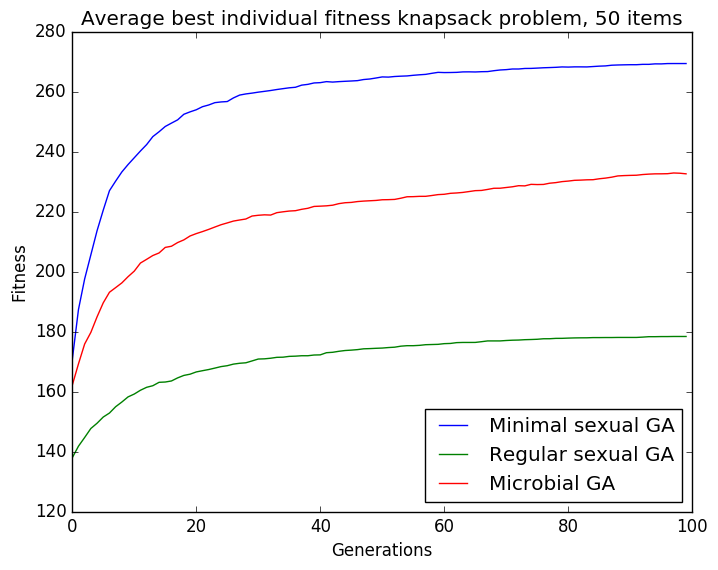


Figure 3 - knapsack problem over 100 runs, (left) average best individual fitness; (right) fitness average of population

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | Min Sexual GA | Reg Sexual GA | Microbial GA |
| **Avg timings for 20 runs (s)** | 34.927+/- 1.438 | 71.050+/- 3.498 | 26.718 +/- 0.909 |

Table - average time taken (seconds) by each algorithm doing 20 runs

Once again, the average best individual fitness for the minimal sexual GA solidly outperforms the other 2 algorithms as seen in figure 3, and again the regular sexual GA actually performs the worst; clearly the tournament selection approach offers further benefits than just minimalism, at least for this problem. Furthermore, the average fitness of the population was far greater, and we can see again that the sexual selection component generally improves overall population fitness. The weaker population fitness of the regular sexual GA also implies a greater adaptability of the minimal sexual GA to a wider variety of problems. Finally, table 2 shows the improved timing performance of the minimal sexual GA over the regular; there is a 50% decrease in runtime, and only a about a 30% increase in run time over the microbial GA. Thus, these results serve to confirm the test criteria, of minimalism, fitness performance, and overall population quality.

1. **Discussion**

From the experimental results, it looks promising that the minimal sexual GA provides a worthwhile alternative to the microbial GA, with comparable fitness performance across the board. In the pub problem, the microbial GA does outperform the minimal sexual GA however in terms of best individual’s fitness, which could be due to the simplicity of the problem; the sexual GA performs a more radical mixing of genotypes when creating offspring, which makes it better at exploring the whole problem space, but in the pub problem perhaps the fitness landscape is simple enough that more rigidly maintaining the genetic material of good individuals across generations is beneficial. This is an avenue that could be explored by further research.

The good average population fitness performance of the minimal sexual GA provides a good indication that the algorithm is working as expected. The maintenance of 2 populations effectively allows the male population to strive straight towards optimisation of the objective function, while the female population ‘incubates’ alternative solutions, mixing them back in with the male population periodically, and thus spreading the population throughout the problem search space. In contrast, the microbial GA just aims towards the objective function, thus giving it a greater tendency to find local minima.

In addition, the observed weaker performance of the regular sexual GA in terms of best individual fitness merits further investigation. The implication is that roulette wheel selection creates a weaker algorithm, as other research supports [17], however it would be worthwhile to do further experiments to verify this is the case in the regular sexual GA.

If I had had more time, I would have done experiments on more problems, including problems with real number encodings, as I had problems trying to implement such problems; furthermore I would experiment with a variety of parameter sets, and see how this influenced performance, including an analysis of the benefit of including the incest parameter in the minimal sexual GA.

1. **Conclusion**

This paper proposed a novel genetic algorithm, based upon the microbial GA, and a GA with sexual selection. It implemented a hybrid form of these two algorithms, and found the preliminary results to be favourable in terms of runtime, best individual fitness and average population fitness performance on several test problems. It has been an insightful and fruitful project undertaking, and I hope to do further research into the benefits of modelling sex in evolutionary computation in future, including further experimentation with the minimal sexual GA on different parameter sets.

**References**

1. *Beyer, Hans-Georg; Schwefel, Hans-Paul.* Evolution Strategies, *2002. Natural Computing, p3-52. University of Dortmund, Germany.*
2. *Cong, Shuang; Jia, Yajun; Deng, Ke.* Particle Swarm and Ant Colony Algorithms and Their Applications in Chinese Traveling Salesman Problem*, 2010. New Achievements in Evolutionary Computation,* *Peter Korosec (Ed.).* *Vukovar, Croatia.*
3. *Dorigo, Marco; Stutzle, Thomas*. Ant Colony Optimization, *2004. Scituate, MA, USA.*
4. *Drezewski, Rafał; Cetnarowicz, Krzysztof.* Sexual Selection Mechanism for Agent-Based Evolutionary Computation*, 2007. Department of Computer Science, AGH University of Science and Technology. Krakow, Poland.*
5. *Falkenauer, Emanuel.*Genetic Algorithms and Grouping Problems, *1997. John Wiley & Sons Ltd. Chichester, England.*
6. *Fogel, D.B.* An Overview of Evolutionary Programming, 1999. *Evolutionary Algorithms, Davis L.D., De Jong K., Vose M.D., Whitley L.D. (eds). The IMA Volumes in Mathematics and its Applications, vol 111. Springer, New York, NY.*
7. *Guang, Xiao; Wei Fu, Xiao; Qing Chen, Da.* A Genetic-Algorithm-Based Approach to UAV Path Planning Problem*, 2005. School of Electronics and Information, Northwestern Polytechnical University.*
8. *Harvey, Inman.* The Microbial Genetic Algorithm*, 1996. Submitted as a Letter to Evolutionary Computation. Brighton, UK.*
9. *Hristakeva, Maya; Shrestha, Dipti. (2018).* Solving the 0/1 Knapsack Problem with Genetic Algorithms*. University of California, California, USA.*
10. *Kenedy, J; Eberhart, R.* Particle Swarm Optimization, *1995. IEEE International Conference on Neural Networks. Perth, WA, Australia.*
11. *Mitchell, Melanie.* Genetic Algorithms: An Overview*, 1995. Santa Fe Institute, New Mexico, USA.*
12. *Niu, Ben; Wang, Hong.* Bacterial Colony Optimization, *2012. Discrete Dynamics in Nature and Society. China.*
13. *Sanchez-Velazco, Jose; Bullinaria, John A.* Gendered Selection Strategies in Genetic Algorithms for Optimization*, 2003. School of Computer Science, The University of Birmingham. Birmingham, UK.*
14. *von Neumann, John.* First Draft of a Report on the EDVAC, 1945. *Moore School of Electrical Engineering, University of Pennsylvania, USA.*
15. *Whitley, Darrell.* Genetic Algorithms and Neural Networks*, 1995. Genetic Algorithms in Engineering and Computer Science, p191- 201. Colorado, USA.*
16. *Wang, Bing; Zhang, Xiang.* Evolutionary Computation Applications in Current Bioinformatics*, 2010. New Achievements in Evolutionary Computation, Peter Korosec (Ed.). Vukovar, Croatia.*
17. *Zhong, Jinghu; Hu, Xiaomin; Zhang, Jun; Gu, Min.* Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms*, 2005.*International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet. Vienna, Austria.