Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms

Janne Koljonen and Jarmo T. Alander
Department of Electrical Engineering and Automation, University of Vaasa
P.O. Box 700, FIN-65101, Vaasa, Finland
jako@uwasa.fi

Abstract

A good selection of the internal parameters of a genetic algorithm may reduce the optimization speed considerably. We studied the effects of two key parameters, namely population size and relative elitism, on the optimization speed and reliability. We found that changing one parameter can be compensated for changing another. By penalizing for the unreliability of the genetic algorithm the optimum combination of the parameters shifted. By using a meta-GA we found relationships between the total population size, relative elitism, and the penalty parameter.
Keywords: genetics algorithms, meta-GA, optimization, GA parameters.

1 Introduction

Genetic algorithm (GA) is the most popular artificial evolutionary algorithm used successfully in difficult optimization and search problems. It consists of a population of trials, a fitness function to map the genotype trials to real numbered phenotypes, and a set of genetic operators to create new trials. GA has several parameters to be selected or tuned, and many alternative implementations for any problem. These include the formalization of the objective function and the genotype-phenotype mapping, selection of genetic operators, selection method, and tuning of parameters such as population size, elitism, mutation rate, etc. (Alander, 1994; Alander, 2002: Darwin, 1998; Koljonen *et al*, 2004, Nordling *et al*, 2004).

The selection of the GA parameters and problem representation has a crucial effect on the GA performance – optimization speed and reliability. In literature, these have been intensively studied both theoretically (Alander, 1992; Alander, 2002; Gao, 2003; Rees *et al*, 1999a; Rees *et al*, 1999b) and empirically (Alander, 1992; Alander, 2002; Rees *et al*, 1999a), yet no final conclusions can be drawn due to the diversity of different kinds genetic algorithms and problem domains.

In this paper, we continue (Alander, 1992; Alander, 2004) to study the effect of the population size and relative elitism on optimization speed and reliability. If tuning GA parameters to the optimal speed, e.g. in mean sense, the probability that the GA becomes stuck without finding a solution within the maximum number of generation may increase. That is why we are also interested in the reliability.

2 Hypotheses and approach

We used an empirical approach to study the effect of population and elite size. The GA was run with two test problems and several parameter combinations. With each configuration the statistics of the GA performance were recorded.

The optimization speed was measured by the number of new individuals created i.e. the number of fitness function calls (trials), inclusive the initial population. The optimization reliability was in turn measured by the fitness of the best individual trial at the end of the optimization (when a solution was found or the maximum number of trials was evaluated). We always minimized and the best fitness was zero if a solution was found and larger than zero otherwise.

2.1 Objective parameters

In our genetic algorithm the population is subdivided into elite and immature population consisting of the latest offspring. The population is sorted by the fitness values of the individuals. The parents of the offspring are randomly drawn from the elite by a uniform probability distribution. Consequently, the individuals in the immature population do not have a chance to be selected as parent; it is merely a place to temporarily place the offspring before sorting.

We studied the effect of the sizes of these subpopulations. The size of the elite is denoted by n_e and the size of the immature population by n_i. The total population size is thus $n_e + n_i$ and relative elitism is defined as $n_e/(n_e + n_i)$. By n_{max} we denote the maximum number of individuals allowed to be created during a GA run.

We presumed that the elite size is more important, because it contains the alleles the offspring inherit. Furthermore, we assumed that by having only a small immature population – i.e. by almost immediately either inserting the offspring into the elite, if fit enough, or rejecting it – we could speed up the GA in some cases. We believed that if good gene combinations were produced, it would be profitable to allow them spreading freely. On the other hand, we understood that it could cause premature convergence if its alleles occupied the population too fast. One can interpret that the smaller n_i the more local the search is.

The other principal parameters were fixed. We used *uniform crossover* (Syswerda, 1989) and mutation rate 0.1 genes mutated per individual.

2.2 Test problems

As a test problem, we decided to use a variation of the popular benchmark problems *one-max*, where the *Hamming distance* to the *all ones* bit string is maximized. In our variation, the target bit string is randomly selected, and the fitness value is the Hamming distance (HD):

$$\mathrm{Hamming}(\mathbf{x},\mathbf{y}) = \sum_{i=0}^{n-1} \left|\mathbf{x}_i - \mathbf{y}_i\right| , \qquad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ are bit vectors, $\mathbf{x}_i$ and $\mathbf{y}_i$ the i^{th} elements of the vectors, and n the dimension of the vectors.

This problem is GA-easy because the fitness function directly tells how many bits are correct. The more bits are correct the more probable it is that uniform crossover produces solutions. Hence, the fitness function guides the GA effectively.

The other test fitness function (FF) was square-root distance (SRD) of the phenotype to the target phenotype, rounded down to the nearest integer. By changing the target of the problems, the fitness landscape changes. The subscript indicates the target, e.g. SRD_{127}. One can also select the number of bits and thus change the range of the FF. The SRD fitness function is:

$$fitness = \left\lfloor \sqrt{\left|phenotype - target\right|} \right\rfloor . \qquad (2)$$

3 Optimization speed

We wanted to study how the elite size and relative elitism affect GA optimization speed. We used 16 bit Hamming distance fitness function and 10 bit square-root FF. With the HD and the SRD the GA was allowed to create at maximum 400 and 800 individual, inclusive the initial population, respectively. The GA was run 200 times with the same population size parameters (n_e and n_i), and the statistics of the optimization speed and reliability were recorded.

3.1 Results and discussion

The population size ranges were the following: For the Hamming distance n_e = {8,16,24,32,40,48} and n_i = {1,2,3,…,50}, and for the square-root distance n_e = {8,16,32,64,128} and n_i = {1,2,3,…,100}. The results are shown in Figures 1, 2, and 3.

In Figure 1, it is seen how the optimization speed depends on the population size parameters when the test function was the Hamming distance and the square-root FF. For the HD, the valley representing the optimal parameter combinations is close to the

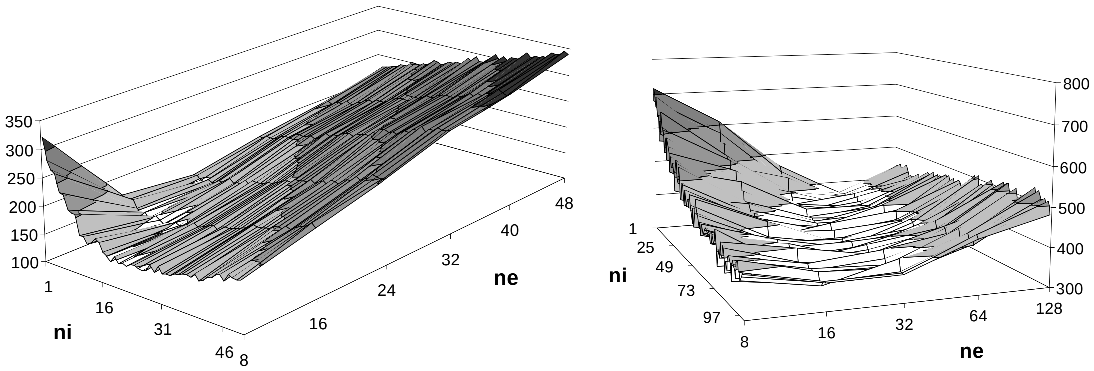

Figure 1: Average number of fitness evaluations with the Hamming distance (left) and the square-root function (right).

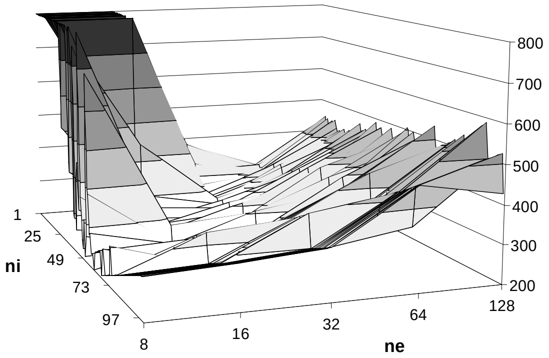

Figure 2: The median value of the optimization speed for SRD is rougher than the average and there is a sharper edge on small population values indicating premature convergence.

origin and narrow. Little changes in population size affect optimization speed a lot. Moreover, n_e and n_i seem to be inter-correlated, i.e. decreasing one can be compensated by increasing the other. The SRD problem was more difficult than the HD, over 300 fitness function calls compared to over 100 with the HD, although it had fewer genes than the HD (12 bits *versus* 16 bits). The optimal population size valley is flatter than with the HD.

Results with these problems suggest the opposite to the belief (Alander, 2002) that even smaller than optimal values were not harmful to optimization as the slopes away the optima are relatively equally steep in both directions. The reason for this illusion is probably the optimization speed measure (average of the individual created). With small populations there are both good and bad cases. However, as the maximum number of individuals was limited, the poor cases do not affect the mean value significantly. In this case order statistic measures like median indicate the premature convergence tendency better as Figure 2 demonstrates. At small population sizes the slope of the median is very steep. On the

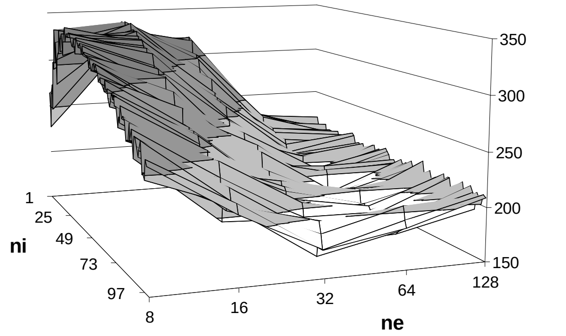

Figure 3: Standard deviation vs. population sizes for SRD.

other hand, the graph of the median is rough at the transition points probably due to the statistical properties of median.

An even earlier indication of the risk of premature convergence is given by standard deviation (Figure 3). Actually, sometimes even before mean performance achieves its minimum, standard deviation begins to grow.

4 Optimization reliability

Optimization speed is only half of the story. Particularly in real time GA applications, both speed and reliability are desired. A solution has to be found within a limited time and it has to be good enough. Sometimes sub-optimal solutions are permitted and a compromise between speed and robustness is done.

4.1 Reliability surfaces

We measured the reliability with the best fitness of a GA run for a limited maximum number of fitness function calls. The best fitness was averaged over 200 GA runs. The population sizes were varied in the same manner as mentioned in section 3.1.

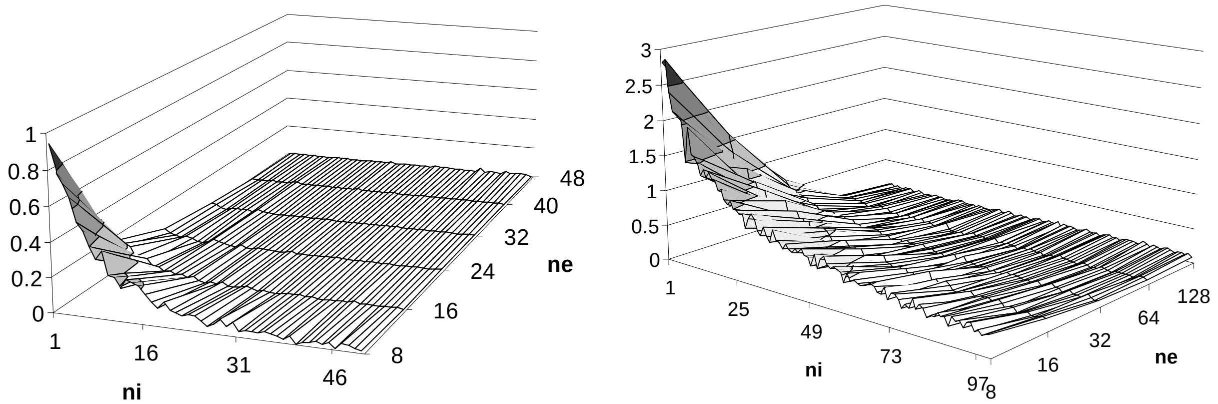

Figure 4: GA optimization reliabilities (mean of the best fitness) of the Hamming distance function (left) and the square-root distance (right) are similar in shape but have different scales.

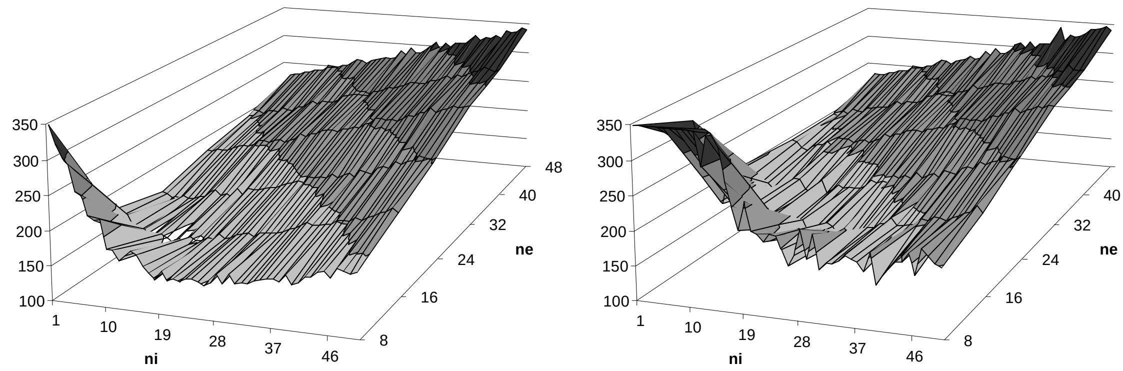

Figure 5: The effect of penalty parameter. The minimum with *a* = 50 (left) is lower and closer to the origin than with *a* = 750 (right). The test fitness function was the Hamming distance.

The optimization reliability surfaces are shown in Figure 4. As presumed, the larger the populations the more reliable the optimization. Full reliability can almost be achieved for both test functions.

However, with speed-optimal population sizes the reliability is not complete. Henceforth, a compromise may be desirable.

4.2 Cost function of speed and reliability

To make a compromise between speed and reliability we need some criterion. Let us define a cost function ε that rewards for speed and penalizes for unreliability. A simple cost function is a linear combination of speed and reliability with a penalty parameter *a*:

$$\varepsilon(a) = fitnessevaluations + a \cdot bestfit\,. \quad (3)$$

4.3 Cost function surfaces

The cost function has only one tunable parameter, whose selection is up to the designer of the GA. The effect of the penalty parameter, i.e. the cost function surface with respect to population sizes, is demonstrated in Figure 5.

The conclusion is that increasing the penalty parameter makes the surface steeper at small population sizes and moves the cost function minimum to larger populations. Because the fitness value is stochastic, it is hard to localize the population size optima with respect to different penalty parameter values. To overcome this difficulty we took the advantage of using another GA, called meta-GA, to find the optima.

5 Meta-GA optimization

The cost function enables to compromise between optimization speed and reliability. If we increase penalty parameter *a*, we emphasize more reliability. We showed in the previous section that changing *a* may move the population size optimum. Next we minimize the cost function with a meta-GA.

5.1 Method

Meta-GA is a method to optimize the GA parameters by another GA. In other words, the problem of selecting parameters is now another optimization problem, which is stochastic due to the stochastic nature of the GA (Mansfield, 1990; Alander, 2002; Bäck *et al*, 2002). In the literature, many variations of meta-optimization have been proposed. Sometimes another GA can be avoided as the GA itself optimizes its parameters (See e.g. Hartono *et al*, 2004).

A meta-GA is similar to a regular GA, but the individuals are GAs, too (Figure 6). The meta-GA runs as many GAs, i.e. the actual optimization problems, as its population size is. The fitness function is

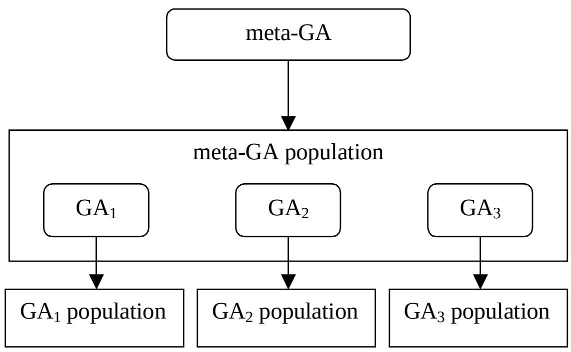

Figure 6: The principle: Meta-GA optimizes the parameters of the actual GAs.

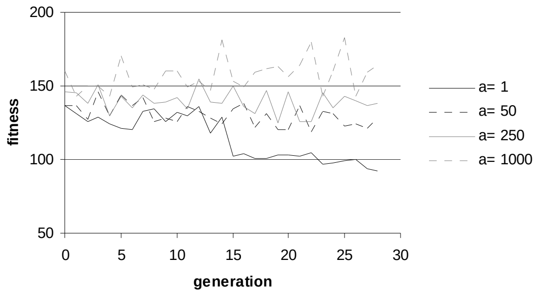

Figure 7: Fitness of the meta-GA ε(a) vs. generation of the meta-GA. The stochastic nature of the fitness function is visible.

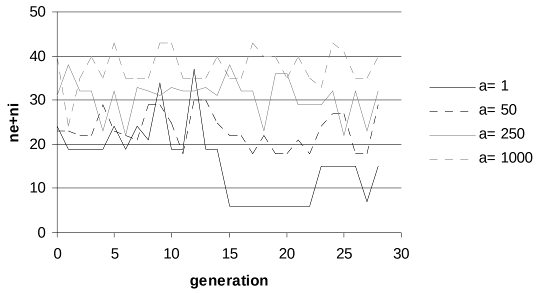

Figure 8: Total population size during optimization. The converged levels with different values of a are clearly separated and in the expected order.

defined by measuring GA performance: now we use the cost function $\epsilon(a)$, which is stochastic.

5.2 Results with Hamming distance

To demonstrate the stochastic nature of the meta-GA fitness function (cost function ϵ) we have plotted the meta-fitness against the meta-GA generation (Figure 7) with four different values of a. The n_e and n_i fluctuated heavily during the meta-GA optimization, because the optimum is a flatter valley. Figure 8 shows that the optimal total population size has less fluctuation. During the last ten generations the decrease in fitness was small. Hence we decided to use the average of the last ten meta-GA generations when determining the optimal population sizes with respect the penalty parameter a.

The optimal elite and immature population sizes are plotted against a in Figure 9. The optimal elite size seems to increase linearly and the immature population size perhaps more or less logarithmically with increasing a. However, these conclusions may be too daring considering the small sample size and the internal variation.

The optimal total population size and relative elitism vs. a are shown in Figure 10. It indicates quite strong logarithmic dependence (R^2 = 94 %).

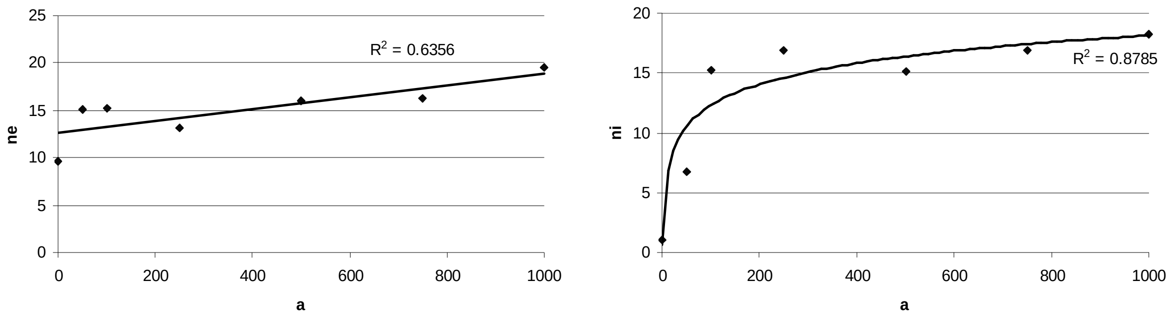

Figure 9: The optimal n_e (left) and n_i vs. penalty parameter. The relationships are quite linear and logarithmic, respectively.

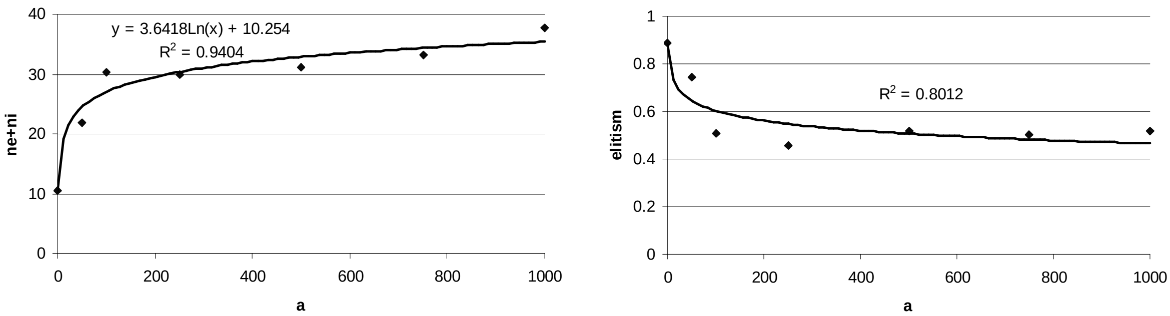

Figure 10: The optimal total population size (left) depends logarithmically on the penalty parameter. Relative elitism vs. the penalty parameter (right).

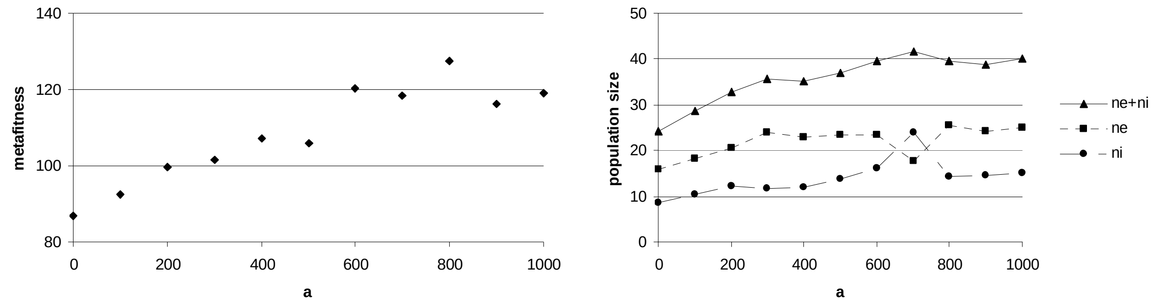

Figure 11: With the easiest square-root function (SRD_0) the penalty factor has an almost linear effect on the meta-fitness (left). Optimal population sizes with respect to *a* with the SRD_0 (right).

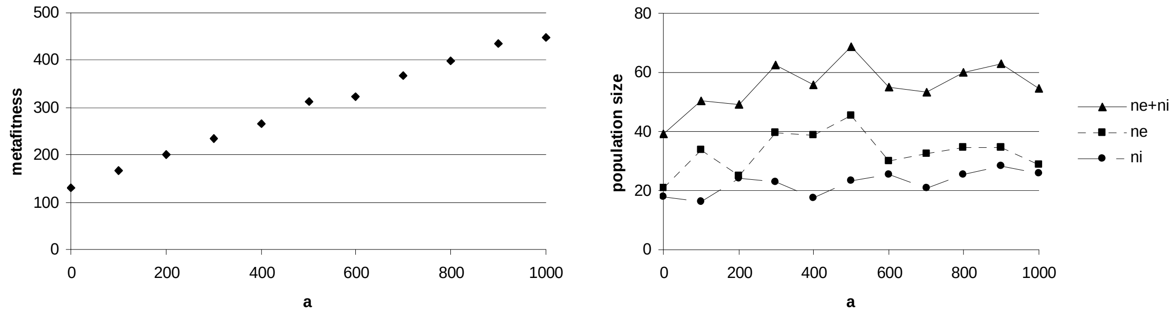

Figure 12: Meta-fitness vs. penalty factor with SRD_{127} (left). Optimal population sizes with SRD_{127} (right).

This result is more reliable as the total population size was found to vary less during optimization (see Figure 10). Yet only qualitative conclusions should be drawn: total population size should be increased logarithmically and elite size almost linearly when increasing *a*. The relative elitism should be in turn decreased, which corresponds to more global search.

5.3 Results with square-root distance

The same procedure of meta-GA optimization was also carried out with the other test problem, square-root distance. Now we used two fixed targets, the GA easiest one (SRD_0) and the GA hardest one (SRD_{127}) of the SRD family, in order eliminate the

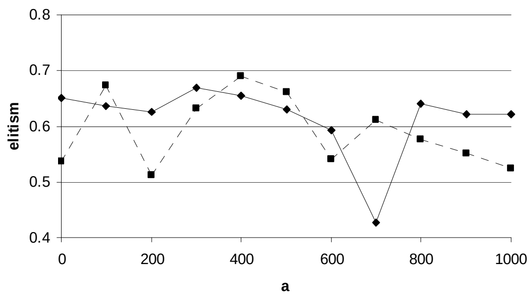

Figure 13: Relative elitism vs. penalty factor with SRD_0 (solid) and SRD_{127} (dotted line)

variation between different targets without the need for impractical number of runs.

The meta-fitness depends almost linearly on the penalty parameter with both targets (see Figures 11 and 12). We see also from the meta-fitness that the SRD_{127} is more GA difficult than the SRD_0. We can conclude, as for the SRD_0, that the total population size must be increased quite logarithmically when increasing penalty parameter *a*. The same holds for the population size of the SRD_{127} given in Figure 12. Moreover, we can see that a larger population is required for optimal GA performance for the easy SRD_0 than for the difficult SRD_{127}.

Finally, the relative elitism is plotted against the penalty parameter *a* in Figure 13. Despite high variance we can conclude that the tendency is downward as *a* increases i.e. to a more global search. Furthermore, it suggests higher elitism (more local search) for the easier SRD_0 than for the more difficult SRD_{127}. However, variation is great and no statistically significant conclusion can be drawn.

6 Conclusions

The population size is a key parameter of a genetic algorithm. We have introduced two population size parameters: elite size and relative elitism. We studied their influence on the optimization speed and reliability. The results confirmed the common belief that decreasing population size increases optimization speed to a certain point, after which premature convergence slows the optimization speed down. The optimization reliability in turn usually increases monotonically with increasing population size.

The optimization speed and reliability were combined into a cost function that penalizes for unreliability. The optimal population parameters were searched with a meta-GA approach with a set of penalty parameters. We found that the optimal population size increases and relative elitism decreases (corresponding to more global search) with increasing penalty parameter i.e. higher reliability requirement. However, the results should be considered only qualitatively due to high variance. Moreover, results from the two test functions, Hamming distance and a discrete square-root distance, indicate that the optimization speed and reliability landscapes vs. population size parameters have different shapes. The same holds partly on the optimal population sizes with respect to reliability requirements. As a conclusion, qualitative thumb-of-rules can be given but not precise quantitative ones.

Acknowledgements

Janne Koljonen would like to thank the Research Institute for Technology at the University of Vaasa for the financial support.

References

J.T. Alander. On optimal population size of genetic algorithms. In: *Proceedings of Computer Systems and Software Engineering, 6th Annual European Computer Conference*, 65–70, 1992, IEEE Computer Society, IEEE Computer Society Press.

J.T. Alander. An Indexed Bibliography of Genetic Algorithms: Theory and Comparisons. University of Vaasa, 1994. URL: ftp://ftp.uwasa.fi/cs/report94-1.

J.T. Alander. Potentials of Genetic Algorithms, TEKES, 2002.

J.T. Alander. On the basic genetic algorithm parameters – an empirical study. In: *Proceedings of the 11th Finnish Artificial Intelligence Symposium* (STeP2004), **3:** 19–32, 2004. Finnish Artificial Intelligence Society (FAIS), Vantaa (Finland).

T. Bäck, D.B. Fogel, Z. Michalewicz (eds.). *Evolutionary Computation* 2, 2002, ISBN: 0750306653.

C. Darwin. *On the Origin of Species*. Reprint of the 1st Edition, published Nov. 1859, Wordsworth Limit, 1998, Great Britain. ISBN: 1853267805.

Y. Gao. Population size and sampling complexity in genetic algorithms. In: *Proceedings of the Bird of a Feather Workshops* (GECCO03), 178–181, 2003.

P. Hartono, S. Hashimoto, M. Wahde. Labeled-GA with adaptive mutation rate, In: *Proc. IEEE CEC*, 1851–1858, 2004.

J. Koljonen, J. Lappalainen, J.T. Alander, A. Backman. LEDall – adaptive LED lighting system. In: *Proceedings of the 11th Finnish Artificial Intelligence Symposium* (STeP2004), **3**: 115–125, 2004. Finnish Artificial Intelligence Society (FAIS), Vantaa (Finland).

R.A. Mansfield. *Genetic Algorithms*, University of Wales, 1990.

T.E.M. Nordling, J. Koljonen, J.T. Alander, P. Geladi. Genetic algorithms as a tool for wavelength selection. In: *Proceedings of the 11th Finnish Artificial Intelligence Symposium* (STeP2004), **3**: 99–113, 2004. Finnish Artificial Intelligence Society (FAIS), Vantaa (Finland).

J. Rees, G.J. Koehler. An investigation of GA performance results for different cardinality alphabets. In: *IMA Volumes in Mathematics and its Applications Proceedings from the IMA Workshop on Evolutionary Algorithms*, 191–206, 1999a, Springer, New York.

J. Rees, G.J. Koehler. A new direction in group support system theory: an evolutionary approach to group decision-making. 1999b, URL: http://citeseer.ist.psu.edu/rees99new.html.

G. Syswerda. Uniform crossover in genetic algorithms. In: *Proceedings of the Third International Conference on Genetic Algorithms*, 2–9, 1989.