Extension for multi-objective genetic algorithms based on the dynamic population size model

P Kazakov

Information Technologies Department, Bryansk State Technical University, Bryansk, Russia
E-mail: pvk_mail@list.ru

Abstract. The paper describes an original way to improve the efficiency of genetic algorithms for multi-objective optimization using the proposed model for dynamic control of population size. This model is used as the extension (modification) for any evolutionary methods of multi-objective optimization. It is based on using the dynamic population size allows multi-objective genetic algorithms to adapt for search space, to enhance the diversity of population and to increase the number of non-dominated solutions. The model uses two additional parameters – age and lifetime of an individual. The value of the first parameter is equal to the number of generations the individual stays in the population. The age increases after each generation, and if it has exceeded lifetime parameter value, an individual is removed from a population. The special expressions for the lifetime parameter calculation were obtained. An individual’s lifetime depends on its Pareto rank and fitness value. A lifetime increases if ones have improved in comparison with a previous generation. The efficiency of dynamic population size using was studied at solving the test multi-objective optimization problems set with a different complexity. In many cases the algorithms used dynamic population size achieved a better distribution and convergence to the true Pareto-front in comparison with other tested genetic algorithms.

1. Introduction
The population size is one of the main parameters for any evolutionary algorithm. Its value makes an impact on both the speed and accuracy of search: an insufficient population size limits the volume of explored search space, leads to a rapid convergence of an algorithm to locally optimal solutions only; too large – slow down the search, reduces the diversity of population and the effect of a selection operator. The population size selection is particularly important in the work of genetic algorithms for multi-objective optimization (MOGA) [1] since it determines the resulting set and the number of found optimal solutions. An effective population management implies maintaining the balance between the individuals has been found already, coding optimal solutions and individuals, their number further change will be intended to continue the search space exploration. At the same time, the diversity of population should be maintained during an optimization process. It is obviously that different parts of search space are characterized differing from the complexity of topology, required for their exploring a various number of individuals. However, the use in MOGA a fixed population size as usually does
not allow changing the density of individuals’ distribution in the search space without unbalancing in a population and preserving its diversity. This restriction can be removed with a dynamic (self-adapting) population size used in MOGA. Its values depend on the character of search and the features of already found solutions. This extension will expand the functionality of a genetic algorithm and reduce the probability of its convergence to locally optimal solutions.

2. Model of dynamic change of a population size

The model of dynamic change of population a size (Self-Adaptive Population Size SAPS) uses the concept of individual with the age and the fixed lifetime [2]. The age of individual $c_i$, $age(c_i, t), i = 1, \ldots, n_p(t)$, where $n_p(t)$ is the size of population in the generation $t$, same as the number of generations that $c_i$ remains in the population and it depends only on the fitness (suitability) of individual. The extension SAPS does not use a selection operator explicitly, therefore the corresponding individuals remain in the population for more generations. On the one hand, this gives rise to a change in the population size, in particular, increase of the population, on the other, to the conservation of a set of Pareto-optimal solutions [1] in it. At each generation $t$ the size of population is increased to allocate the new individuals on the quantity $n_p(t) = n_p(t) \cdot \rho$, where $\rho \in (0, 1]$ is the reproduction coefficient. In addition to the current age, the LifeTime parameter $LT(c_i, t), i = 1, \ldots, n_p(t)$ is coupled with each individual. It value limits the number of generations for an individual stays in the population. In the SAPS model, the value $LT(c_i, t), i = 1, \ldots, n_p(t)$ is calculated each time after evaluating the fitness of the individual, for which the following functional dependencies have been obtained $\forall c_i, i = 1, \ldots, n_p(t)$:

$$
LT(c_i, t) = \begin{cases} 
\text{age}(c_i, t) + \text{round}\left(\frac{f(c_i, t) - f(c_i, t)_{\text{max}}}{f(c_i, t)_{\text{min}} - f(c_i, t)_{\text{max}}}\right), & \text{if } \text{rank}(c_i, t) = 1 \\
\frac{1 + \max \{\text{rank}(c_i, t)\}}{2} \cdot \left(\text{rank}(c_i, t - 1) \cdot \text{rank}(c_i, t)\right) \cdot \frac{f(c_i, t) - f(c_i, t)_{\text{min}}}{f(c_i, t)_{\text{max}} - f(c_i, t)_{\text{min}}}. & \text{else}
\end{cases}
$$

(1)

There $\text{rank}(c_i, t), i = 1, \ldots, n_p(t)$ is the individual’s rank of Pareto in generation $t$. The non-dominated individuals has the rank equal to one. The value $f(c_i, t), i = 1, \ldots, n_p(t)$ determines the fitness of each individual; it is assumed that its greater value corresponds to the best indicator. The parameters $f(c_i, t)_{\text{min}}, f(c_i, t)_{\text{max}}, i = 1, \ldots, n_p(t)$ represent accordingly the minimum and maximum fitness value among the individuals of generation $t$. For the initial population ($t = 0$) it is set $LT(c_i, t) = \text{rank}(c_i, t), i = 1, \ldots, n_p(t)$.

Thus, the individual’s lifetime depends only on its Pareto rank and fitness value. The solutions presented by individuals with Pareto rank equal to one are optimal for this population. Therefore, for such individuals the lifetime is calculated by the separate expression (the first expression of the system (1)). According to it, the each non-dominated individual with fitness above the average necessarily passes on to the next generation. At the determining the lifetime for the rest of individuals along with fitness the change in the rank of Pareto in comparison with the previous generation is taken into account. If the rank has improved (decreased), then the corresponding individuals can increase the time of their stay in the population and, thus, the probability of participation in the crossover operator.
With the use of various indicators of quality of a solution, for example, the features of its location on the Pareto front, the density of its neighborhood on it with other points, etc. As a result, the individuals with the same rank, but different suitability can have a differing lifetime. Thus, in the next generation mainly pass those individuals with the rank of Pareto is equal to one, or it has improved, and the value of their fitness at that above average. Also, the proposed model for calculating the lifetime of individuals has a sufficiently low computational complexity, does not use special variables and functions that are not relative to a basic MOGA. This is an important factor for MOGA, since the parameter \( LT(c_i, t) \) must be calculated for every individual on the each generation. After calculating the value \( LT(c_i, t) \), the population is reduced by removing all individuals with \( age(c_i, t) > LT(c_i, t) \) from it.

Thus, the work of a general multi-objective genetic algorithm with the dynamic population size consists in the following steps.

Step 1. \( t = 0 \). Create the initial population with a size \( n_p(t) \) and to calculate the rank of Pareto for every individual. \( \forall c_i, i = 1, ..., n_p(t) : LT(c_i, t) = rank(c_i, t) ; age(c_i, t) = 0 \).

Step 2. \( t = t + 1 \). Modify the population size on value \( n_p(t) = n_p(t) \cdot \rho \); \( n_p(t) = n_p(t) + n_p(t) \).

Step 3. Apply the genetic operators of MOGA to individuals of a population. Augment the age of «old» individuals and set zero age to all new individuals.

\[ \forall c_i, i = 1, ..., n_p(t) - n_p(t) : age(c_i, t) = age(c_i, t) + 1 , \forall c_k, k = n_p(t) - n_p(t) + 1, ..., n_p(t) : age(c_k, t) = 0 \].

Step 4. Calculate the lifetime for every individual, using the system of equations above.

Step 5. Modify the population size by the value \( n_p(t) = \left( | \{ c_k, k = 1, ..., n_p(t) \} | age(c_k, t) > LT(c_k, t) \right) ; \)

\[ n_p(t) = n_p(t) - \tilde{n}_p(t) \].

Step 6. If the stop algorithm condition was fulfilled, then finish. Else go to step 2.

The proposed method of dynamic population size control [3] can be used in any MOGA. In this case, the main control parameter (reproduction coefficient) of SAPS will have an important influence on the Pareto set obtaining accuracy. Its value can be determined experimentally depending on the features of the solved problem.

3. Selection of the reproduction coefficient value

The series of experiments were made to analyze the effect of various values of reproduction coefficient on changes in the size and composition of the population. The most well-known genetic algorithm for multi-objective optimization NSGA-II [4] with modification SAPS (NSGA-II+SAPS) was used for solving the test problem DTLZ1 [5] in the experiments. The number of criteria \( m \) was given equal to two, the set of various values of parameter \( \rho = \{0.1, 0.3, 0.5, 0.8\} \) was analyzed. The same size of initial population \( n_p(t = 0) = 100 \), the number of generations as well as the values of probabilities of crossover and mutation operators were set for an each case. In the process of the Pareto set obtaining the time history of population size (Fig. 1), number (Fig. 2) and proportion of non-dominated solutions were evaluated. The analysis of results made it possible to draw the following conclusions.

1. There is no appreciable increase in the number of population for the small values of reproduction coefficient (\( \rho < 0.3 \)). It decreases at the beginning, then grows, but reaches and exceeds its initial size only starting from the second half of generations. At that the share of non-dominated solutions in the population is relatively high (more than 30%) and does not almost change throughout the search. This allows concluding about its rapid stagnation due to the lack of diversity of population is contained mainly the same, constantly cloning individuals.
2. The population shows an increase in the number of individuals with the growth of reproduction coefficient ($\rho \geq 0.3$) in the first third of generations. When $\rho = 0.5$ is reached the population and the number of non-dominated solutions in it quickly doubles. In the last third of generations the number and proportion of non-dominated solutions for $\rho = 0.3$ practically does not change, it indicates the convergence of algorithm in large. In comparison in this time interval the change in the number and proportion of non-dominated solutions at $\rho = 0.5$ is still liable to appreciable variations, it does not allow making a final conclusion about the convergence of the algorithm to the preset number of generations.

3. Further increase in the value of the reproduction coefficient ($\rho > 0.5$) result in a rapid population growth at the very beginning of the MOGA working. Afterwards, the dynamics of population size change and the number of non-dominated solutions in it takes a chaotic character. This is typically for the exclusive use of a mutation operator to obtain new members of the population. The ratio of the population size and the number of non-dominated solutions in the population is mostly random. It allows drawing a conclusion about essential problems of the convergence of MOGA with such values of the reproduction coefficient and not only in the solved problem situation.

As a result it was found that the best balance between the population growth rate and the number of non-dominated solutions in it for the solved test problem is achieved when $\rho \in [0.2,\ldots,0.4]$. At the same time, the solving of the considered problem with the same conditions of experiments, but with many criteria ($m = 8$) showed the validity of use of the reproduction coefficient $\rho > 0.5$ ($\rho = 0.7$). This can be explicated by a non-linear increase in the filling rate of population with non-dominated solutions under the number of optimized criteria rises. Thereby, at the optimization process time an increase of population diversity is needed, only the possibility of rapid population growth allows maintaining a balance between the explored and unexplored areas of the search space.

4. Study of efficiency of the multi-objective genetic algorithms extension based on the dynamic population size model
The series of experiments were made to evaluate the efficiency of the proposed model of self-adaptation of a population size. They consisted in solving the set of test multi-objective optimization
problems (DTLZ1, DTLZ2, DTLZ3, DTLZ6) [5] with differing number of criteria. This allowed evaluating the impact of SAPS on the preservation of the efficiency of basic MOGA on rise of complexity of the solving problems. The most well-known algorithms SPEA2 (Strength Pareto Evolutionary Algorithm) [6] and NSGA-II (Non-dominated Sorting Genetic Algorithm) [4] as basic MOGA were used.

The results of multi-objective optimization were evaluated by several quality indicators. The set of indicators [7] we used to calculate them, it allows evaluating various properties of the obtained by MOGA Pareto set in the spaces of variables or criteria: $I_{ONVG}$ (Overall Non-dominated Vector Generation) determines the obtained Pareto set count; $I_S$ (Spacing) is used to evaluate the distribution uniformity of solutions along the Pareto front; $I_{DE}$ (Dimensions Extent) allows evaluating the maximum length of the Pareto front for the each dimension; $I_{GD}$ (General Distance) allows evaluating the degree of closeness between the obtained and the desired true Pareto fronts; $I_{OTC}$ (Overall Time Computing) is intended for evaluating the time efficiency of MOGA on the Pareto set obtaining.

The values of control parameters were experimentally selected for each MOGA: the sizes of population and Pareto-archive, the number of generations and the probabilities of crossover and mutation operators. The values for all algorithms were the same and depended on the solving problems [3]. The solving results of all problems are summarized in the table 1. The values of all indicators are averaged over the number of MOGA runs. The values of the indicators that have been improved using the SAPS extension were printed in bold.

| Problem | Indicator | SPEA2 | NSGA-II | SPEA2 +SAPS | NSGA-II +SAPS |
|---------|-----------|-------|---------|-------------|---------------|
| DTLZ1 ($m = 2$) | $I_{ONVG}$ | 28 | 35 | 28 | 66 |
| | $I_{GD}$ | 0.071 | 0.067 | **0.068** | **0.062** |
| | $I_S$ | 0.186 | 0.223 | 0.188 | **0.214** |
| | $I_{DE}$ | 0.971 | 0.964 | 0.968 | **0.973** |
| | $I_{OTC}$ (s) | 6.3 | 4.8 | 11.4 | 8.6 |
| | $I_{ONVG}$ | 113 | 138 | **113** | **179** |
| | $I_{GD}$ | 5.732 | 6.124 | **5.441** | 6.132 |
| | $I_S$ | 0.133 | 0.124 | **0.121** | 0.125 |
| | $I_{DE}$ | 1.714 | 1.847 | **1.793** | **1.892** |
| | $I_{OTC}$ (s) | 191.4 | 31.6 | 267.8 | 48.5 |
| | $I_{ONVG}$ | 188 | 207 | **208** | **236** |
| DTLZ2 ($m = 4$) | $I_{ONVG}$ | 226.953 | 310.603 | 231.762 | 314.547 |
| | $I_{GD}$ | 0.328 | 0.287 | **0.297** | **0.276** |
| | $I_S$ | 2.137 | 2.312 | **2.167** | **2.376** |
| | $I_{DE}$ | 6217.3 | 698.7 | 10342.7 | 1095.4 |
| | $I_{OTC}$ (s) | 319 | 371 | **360** | **401** |
| DTLZ3 ($m = 6$) | $I_{ONVG}$ | 16.623 | 11.237 | **14.547** | **11.142** |
| | $I_{GD}$ | 0.251 | 0.201 | **0.247** | 0.196 |
| | $I_S$ | 1.862 | 1.913 | **2.034** | **2.173** |
| | $I_{OTC}$ (s) | 76784.3 | 2843.8 | 128132.5 | 4546.4 |
As it can be seen from the table, the main effect of using SAPS extension is related with an increase of the number of found Pareto-optimal solutions in all cases. In turn, this has influenced the improvement of the values of several other indicators. In particular, \( I_{DE} \) is intended for evaluate the extension of Pareto front. In some cases the results on the indicators \( I_{GD}, I_{S} \) have improved. This is probably caused by the increase of Pareto front filling density by a larger number of solutions in comparison with the original SPEA2 and NSGA-II. At the same time, none of the modified versions of these MOGA have not showed the best results in all problems. However in problems (DTLZ1, DTLZ6) the algorithm NSGA-II+SAPS fully exceeded SPEA2+SAPS. Also the both versions of NSGA-II were faster than its rival in the all experiments. Moreover the variation in time between SPEA2 and NSGA-II becomes significant with the increase in the number of criteria. Taken together this lets make the conclusion about the preferred using the proposed model of dynamic population size change in NSGA-II.

5. Conclusion
The proposed method of dynamic control of a population size of MOGA in the multi-objective optimization process enables not only to maintain the diversity of a population by permanent updating the lifetime of individuals, but also to increase the number of found Pareto-optimal solutions. This is especially important when their number is limited to the primary set population size. Thus the use of the created SAPS extension in MOGA SPEA2 and NSGA-II have enabled to attain an increase in the number of found Pareto-optimal solutions in combination with an improvement in the quality of the related Pareto front. The time complexity growth over an integration of the mechanisms of dynamic population control in MOGA depends on the complexity of a topology of criteria space and the selection of values for the single additional control parameter reproduction coefficient.

References
[1] Deb K 2009 Multi-Objective Optimization Using Evolutionary Algorithms (Hoboken: Wiley)
[2] Michalewicz Z 1996 Genetic Algorithms + Data Structures = Evolution Programs (Berlin: Springer-Verlag)
[3] Averchenkov V and Kazakov P 2014 The new manner for improving of the evolutionary methods for a multi-objective optimization Herald of Computer and Information Technologies 9 pp 23-29
[4] Deb K 2002 A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II IEEE Transactions on Evolutionary Computation 6(2) pp 182–197
[5] Deb K, Thiele L, Laumanns M and Zitzler E 2005 Scalable Test Problems for Evolutionary Multi-Objective Optimization Evolutionary Multiobjective Optimization Theoretical Advances and Applications (New York: Springer) pp 105-145
[6] Zitzler E, Laumanns M and Thiele L 2002 SPEA2: Improving the Strength Pareto Evolutionary Algorithm Proceedings of the EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems pp 95–100
[7] Zitzler E, Thiele L, Laumanns M, Fonseca C and Grunert da Fonseca V 2003 Performance Assessment of Multiobjective Optimizers: An Analysis and Review IEEE Transactions on Evolutionary Computation 7(2) pp 117–132