Feedback-Control Operators for Evolutionary Multiobjective Optimization

Ricardo H.C. Takahashi[1], Frederico G. Guimarães[2], Elizabeth F. Wanner[3], and Eduardo G. Carrano[4]

[1] Universidade Federal de Minas Gerais, Department of Mathematics, Brazil
taka@mat.ufmg.br
[2] Universidade Federal de Ouro Preto, Department of Computer Science, Brazil
frederico.guimaraes@yahoo.com.br
[3] Universidade Federal de Ouro Preto, Department of Mathematics, Brazil
efwanner@iceb.ufop.br
[4] Centro Federal de Educação Tecnológica de Minas Gerais, Brazil
egcarrano@deii.cefetmg.br

Abstract. New operators for Multi-Objective Evolutionary Algorithms (MOEA's) are presented here, including one archive-set reduction procedure and two mutation operators, one of them to be applied on the population and the other one on the archive set. Such operators are based on the assignment of "spheres" to the points in the objective space, with the interpretation of a "representative region". The main contribution of this work is the employment of feedback control principles (PI control) within the archive-set reduction procedure and the archive-set mutation operator, in order to achieve a well-distributed Pareto-set solution sample. An example EMOA is presented, in order to illustrate the effect of the proposed operators. The dynamic effect of the feedback control scheme is shown to explain a high performance of this algorithm in the task of Pareto-set covering.

1 Introduction

Two main concerns are involved in the task of designing of Multi-Objective Evolutionary Algorithms (MOEA's): (i) the "quality" of the Pareto-set estimates that are generated, and (ii) the convergence velocity of the algorithm. The first of such concerns is, in itself, multi-dimensional, and there are not, up to now, any definitive standards for measuring such "quality" [1]. A high-quality solution set, anyway, can be defined as a set of samples that [2]:

– Approach the exact Pareto-set (i.e., should be dominated by a subset of the decision variable space that is as small as possible);
– Cover the whole extension of the Pareto-set (i.e., include samples which are spread along the whole range of the Pareto-set, including the regions near the extremes of such Pareto-set);
– Describe in detail the "body" of the Pareto-set (i.e., these samples are "regularly" spread along the Pareto-set).

Notice that a MOEA can be built without a commitment with the search for a Pareto-set estimate in its whole extension, i.e., without referring to such quality measures, in the cases in which some *a priori* or *on line* decision information is available, allowing the concentration of the search in some sub-regions that are identified as being "of interest".

This paper presents new operators that can be used for designing high-performance MOEA's (in the sense of MOEA'S that produce high-quality Pareto-set estimates, as defined above): the *Sphere-Control operators*. Such operators are based on the raw information about the distances between every pair of solution samples in a set – this motivates the denomination of "sphere" operators. The key concept behind the proposed operators is the usage of a *feedback-control* scheme for the purpose of establishing a dynamic equilibrium associated to the high-quality description of the Pareto-set. This means that while such high-quality description is not attained, there will be measured variables that indicate this fact, carrying the information about what control action should be taken in order to enhance such quality [3]. An instance of such feedback-control scheme is represented in Figure 1. In this figure, the measured variable is the error $e = r - a$, which feeds the Proportional-Integral (PI) controller, which in turn determines the value of the control variable ρ. In the equilibrium, $e = 0$ (which means the desired result of $a = r$).

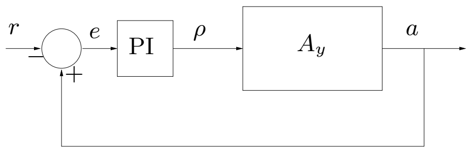

Fig. 1. Diagram of PI feedback control loop for controlling the number of points in the archive set. The variable ρ, the radius of the spheres associated to the points in the archive file, plays the role of the control input variable, while the variable a, the number of points in the archive file, plays the role of the controlled variable. The pre-established reference number of points in the archive, denoted by r, will be attained in the equilibrium, by virtue of the feedback mechanism.

As in other contexts of application of feedback-control techniques, the role of the feedback control scheme is to induce an overall system behavior that presents low sensitivity to variations in the initial conditions and in the algorithm parameter values, delivering rather "stable" results – meaning a high repeatability in the reach of high-quality solution sets [3]. The error variables are defined such that the feedback loop reaches an equilibrium only when a "good description" of the Pareto-set is attained. An unbalanced spread of solutions causes the number of solutions to shrink (by eliminating the more redundant solutions), while the existence of connected areas in the Pareto-set which are not well-covered causes the number of solutions to grow, with the equilibrium being reached only there are no more non-described regions, and the solutions are evenly distributed along the Pareto-set object. The equilibrium itself can be used as a criterion for detecting the end of the task of Pareto-set description, serving as a stop criterion.

Specifically, two sphere-control operators are proposed here: a mutation operator that is applied in the archive set, in order to fill eventual gaps in its Pareto-set description – which is called here the *surface-filling mutation*, and an *archive-set reduction* operator which controls the number of non-dominated solutions that are stored. Feedback control mechanisms, based on a switched controller and on a Proportional-Integral (PI) controller, are employed in the surface-filling mutation and in the archive-set reduction operator, in order to enhance the distribution of solutions along the Pareto surface. This motivates the denomination of "sphere-control" operators. These operators are to be employed together, since their effects are complementary, and their dynamic interaction is necessary in order to achieve the desired behavior. Another mutation operator is also defined here, still employing the "sphere" concept, but not employing the feedback-control scheme. This operator is applied in the current population, and resembles the "hypermutation" operation employed in the Artificial Immune System proposed in [4].

The ideas presented here have connection with the ones presented in [5,2] which employ "sphere" operations which are similar to the archive-set reduction operator presented here, yet without any feedback adaptation scheme. The basic idea, both in that references and here, is that a "sphere" means roughly a domain in which the information gained by a solution point in its center would be representative – with no need of further function evaluations inside such sphere. The references [6,7] also employ the concept of "spheres" for construction of an EMOA, with a dual meaning: in that cases, the "sphere" is the domain in which a local search is conducted, with sub-populations assigned to perform searches inside each sphere.

In the specific formulation that is presented here, the proposed operators are structured for continuous-variable spaces. However, the adaptation for discrete-variable problems can be performed directly, provided that some distance metric becomes defined in the discrete-variable space.

An algorithm that instantiates the application of such operators is constructed: the SCMGA (Sphere-Control Multiobjective Genetic Algorithm). Such algorithm is compared with an NSGA-based algorithm and with an SPEA-based algorithm, in order to illustrate the enhancements of the Pareto-set estimates that can be obtained via the proposed approach. The role of the feedback control operators is analyzed, and the results suggest that such operators interact in order to increasingly enhance the description of the Pareto set. In particular, the gaps in the Pareto surface are systematically filled by the proposed operators – leading to surface descriptions of high definition.

2 Multiobjective Genetic Algorithms

Consider $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$ a vector-valued real function. Let $f_i(\cdot)$ denote the *i-th* coordinate of the function in the image space. The multiobjective problems appear from the partial ordering induced by the relation of *dominance*:

$$u \prec v \;\Leftrightarrow\; \begin{cases} f_i(u) \leq f_i(v) \;\forall\; i = 1, \ldots, m \\ \text{ and} \\ \exists\; i \in \{1, \ldots, m\} \text{ such that } f_i(u) < f_i(v) \end{cases} \tag{1}$$

Consider the Pareto-set $\mathcal{P}$ defined by:

$$\mathcal{P} \triangleq \{x^* \in \Omega \mid \;\nexists\; x \in \Omega \text{ such that } x \prec x^*\} \tag{2}$$

in which $x \in \mathbb{R}^n$ is the decision variable vector, and $\Omega \subseteq \mathbb{R}^n$ is the feasible set. A multiobjective optimization problem is defined as the task of generating samples of the set $\mathcal{P}$.

A Multiobjective Genetic Algorithm (MGA) is a genetic algorithm which is intended to produce a set of samples of $\mathcal{P}$. These algorithms can be stated, in general, as:

Algorithm 1. Pseudocode for generic MGA

```
$k \leftarrow 0$
$P_k \leftarrow$ initial_population
$E_k \leftarrow$ evaluate_function($P_k$)
$A_k \leftarrow \emptyset$ % the archive set
$M_k \leftarrow \emptyset$ % the set of points resulting from mutation
$C_k \leftarrow \emptyset$ % the set of points resulting from crossover
while not stop criterion do
    $A_{k+1} \leftarrow$ update_archive($A_k, P_k, E_k$)
    $F_k \leftarrow$ fitness_assignment($P_k, M_k, C_k, E_k$)
    $P_{k+1} \leftarrow$ new_population($A_k, P_k, M_k, C_k, F_k$)
    $M_{k+1} \leftarrow$ mutation($P_{k+1}$)
    $C_{k+1} \leftarrow$ crossover($P_{k+1}$)
    $E_{k+1} \leftarrow$ join( evaluate_function($M_{k+1}, C_{k+1}$), $E_k$)
end while
```

In addition to this "basic" structure, other operators can be added within the main loop. A very common kind of additional operator performs a local search [8]. The algorithm that is to be tested here and the algorithms that are adopted for comparison follow this basic structure. At the end of the execution, the "archive set" A_k contains the algorithm outcome, which constitutes an estimate of the Pareto-set $\mathcal{P}$.

3 The Sphere Operators

Consider the sets A and P, respectively meaning the archive and the current population:

$$\begin{aligned} A &\triangleq \{\tilde{x}_1, \ldots, \tilde{x}_a\} \\ P &\triangleq \{x_1, \ldots, x_p\} \end{aligned} \tag{3}$$

with $|A| = a$ and $|P| = p$. The images of such sets in the objective space are denoted by:

$$\begin{aligned} A_y &= \{\tilde{y}_1, \ldots, \tilde{y}_a\} \triangleq \{f(\tilde{x}_1), \ldots, f(\tilde{x}_a)\} \\ P_y &= \{y_1, \ldots, y_p\} \triangleq \{f(x_1), \ldots, f(x_p)\} \end{aligned} \tag{4}$$

The remainder of this paper considers that both sets A_y and P_y are normalized according to the extreme values of set A_y, such that the minimal value in any dimension receives the value 0, and the maximal value receives the value 1. This means that some values in P_y can fall outside the range $[0, 1]$. A re-normalization is performed in both sets every time the set A_y is updated.

The main idea behind the "sphere operators" is that if the solution samples regularly cover the set $\mathcal{P}$, they should be located in relation to their nearest neighbors such that the distances to them become of the same order of magnitude. Therefore, a parameter ρ, which has the meaning of a *reference domain radius* for each point, must be defined. This parameter is employed in order to guide the algorithm operations, with the intent to generate points which approximately "represent" the region inside the sphere of such radius centered in that point. Any two neighbors should be separated, therefore, by at least 2ρ. The parameter ρ is dynamically adjusted during the algorithm execution, in order to reach a good dispersion of the sample points along the Pareto-set estimate, considering a reference value of the number of sample points that is to be found.

3.1 Archive-Set Reduction

Consider the set $A = \{x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_a\}$ in which the individual minima of the m objective functions have been put in the first m positions, and the remainder $a - m$ points have been ordered randomly. The set A_y is ordered correspondingly. The archive set is reduced by the Algorithm 2.

Algorithm 2. Pseudocode for Archive-Set Reduction

```
1: $i \leftarrow 1$
2: **while** $|A| > i$ **do**
3:     $A \leftarrow A - \{x_j \mid \|y_i - y_j\| < 2\rho \ , \ i \neq j\}$
4:     $A_y \leftarrow A_y - \{y_j \mid \|y_i - y_j\| < 2\rho \ , \ i \neq j\}$
5:     $i \leftarrow i + 1$
6: **end while**
```

After this operation, there will be no two points in A_y with pairwise distances smaller than 2ρ. This controls the size of the archive set, ensuring that the points will be distributed smoothly. Notice that this operation can be performed directly even in the case of discrete-variable problems, since all information is processed with reference to the objective space.

If this operation was executed with a pre-defined value of ρ, this could cause the size a of set A to become too large or too small, since the exact extension of the set $\mathcal{P}$ is not known *a priori*. This leads to the need of a dynamic adjustment of this parameter.

3.2 Controlling ρ

The adjustment of parameter ρ takes into account a reference number of points that should be stored in the archive set A. Let such reference number be denoted by r, and let the actual number of points in A be denoted by a. The adjustment procedure is described in Algorithm 3. The initial value of ρ is defined such that r spheres of dimension $(m-1)$ occupy a volume equivalent to the unitary simplex of dimension $(m-1)$.

Algorithm 3. Pseudocode for ρ Control

```
 1: if $a > r$ then
 2:     $e \leftarrow (a-r)/r$
 3:     if $e > s$ then
 4:         $\Delta \leftarrow s$
 5:     else
 6:         $\Delta \leftarrow K_p \times e$
 7:     end if
 8:     $\rho \leftarrow (1+\Delta) \times \rho$
 9: else
10:     $\rho \leftarrow K_n \times \rho$
11: end if
```

Default values can be indicated for the control parameters: $K_p = 0.6$, $s = 0.1$, $K_n = 0.9$. This algorithm resembles the PI (proportional-integral) controllers with control signal saturation, which are employed in industrial control systems. A relative error e is calculated in each step. An incremental control action is calculated on the basis of such error. A diagram of such closed-loop feedback control scheme is presented in Figure 1.

For negative errors (the number of archive points is smaller than the reference one), the size of the *reference domain radius*, ρ, must be reduced, in order to allow that more spheres become defined in the next step. In this case, ρ is multiplied by the parameter K_n which must be chosen such that $0 < K_n < 1$. This tends to be the case when the algorithm is starting: there are few points in the archive set, and the radius is continuously reduced.

In the case of positive errors (the number of archive points becomes greater than the reference one), the size of ρ must be increased, in order to eliminate more points in the operation of archive reduction (which is equivalent to make any point to "represent" a larger sphere around itself). In this case, the increment Δ is proportional to the error, in the case of small errors, and fixed, in the case of large errors, in order to avoid rapid increments of ρ. The need for such saturation is due to stability considerations, in the same sense that appears in the context of industrial control – in this way avoiding the excessive oscillation of the control variable ρ.

It should be noticed that, as the control action over variable ρ is incremental, the net effect has the form of an *integral* control. This integral term in the PI controller is necessary in order to induce an error-less steady behavior in the closed-loop system. Indeed, if a simple P (proportional only) controller were adopted, the steady state value of the controlled variable a would not become equal to the reference value r. This, of course, would lead to the need of *ad-hoc* parameter adjustments in order to obtain a solution set with a pre-established size. In this way, the PI controller structure is the simplest one that fulfills the requirements which are needed here. (Most textbooks on classic control theory, for instance [3], will present a detailed discussion about such effects).

The adoption of such closed-loop control scheme makes the size of ρ to reach an equilibrium point by itself, avoiding the need of an *a priori* knowledge about such parameter, and rendering the multiobjective genetic algorithm robust in relation to this parameter.

Notice that even in the degenerate case of the Pareto-set surface being of dimension less than $m - 1$ (one or more objectives being redundant), the algorithm still works as expected, forming an archive set of Pareto estimates which will have still r elements. The only exception would be in the case of a single non-redundant objective, in which the control variable ρ would shrink up to very small values without obtaining the effect of forming an archive set with r elements. Rigorously, an algorithm should have a stop condition related to the detection of such situation.

3.3 Surface-Filling Mutation

Define the function $v(\cdot)$ as

$$v(y_i) = |\{y_j \mid \|y_i - y_j\| < 3\rho\}|$$

which means the cardinality of the set of points from the archive set A_y which are inside a ball of radius 3ρ around the function argument y_j. Without loss of generality, consider that the set A_y is ordered in increasing order of $v(y_i)$:

$$A_y = \{y_i \mid v(y_i) \leq v(y_{i+1})\}$$

The set A receives the corresponding ordering. The *surface-filling mutation* operator is defined by Algorithm 4 (notice that $|A| = a$).

In Algorithm 4, the matrix Γ performs the tasks of adjusting the mutation to different ranges of the decision variables and introducing correlation between the mutation in different variables, if necessary (in ordinary cases, it can be set as the identity matrix). An important notice about this mutation operator is: it is performed over the *archive* set, instead of being performed over the *current population* set. This operator is followed by a controlled adaptation of mutation radius β, as shown in Algorithm 5.

The idea of the *surface-filling mutation* is to generate mutations in the individuals that have less neighbors in the objective space (i.e., less other points at a distance smaller than 3ρ). These individuals are subject to mutations that are intended either to provide local enhancements or to generate new neighbors that fill the gaps in the description of the Pareto surface by the archive set. However, the radius β that should be employed for the Gaussian mutation is not known *a*

Algorithm 4. Pseudocode for Surface-Filling Mutation

```
 1: $k_1 \leftarrow k_2 \leftarrow k_3 \leftarrow k_4 \leftarrow 0$
 2: for $i = 1$ to $a/3$ do
 3:    Generate $\omega \in \mathbb{R}^n$ with Gaussian distribution (0,1)
 4:    $\hat{x}_i \leftarrow \beta \Gamma \omega + x_i$
 5:    $\hat{y}_i \leftarrow f(\hat{x}_i)$
 6:    if $\hat{y}_i \prec y_i$ then
 7:       $x_i \leftarrow \hat{x}_i$
 8:       $y_i \leftarrow \hat{y}_i$
 9:       $k_1 \leftarrow k_1 + 1$
10:    else if $y_i \prec \hat{y}_i$ then
11:       $k_2 \leftarrow k_2 + 1$
12:    else
13:       $k_3 \leftarrow k_3 + 1$
14:       if $\|y_i - \hat{y}_i\| > \rho$ then
15:          $A \leftarrow A + \hat{x}_i$
16:          $A_y \leftarrow A_y + \hat{y}_i$
17:          $k_4 \leftarrow k_4 + 1$
18:       end if
19:    end if
20: end for
```

Algorithm 5. Pseudocode for Mutation-Radius Control

```
1: if $k_3 > 0$ then
2:    $\alpha = k_4/k_3$
3:    if $\alpha > 0.8$ then
4:       $\beta \leftarrow 0.9 \times \beta$
5:    else if $\alpha < 0.4$ then
6:       $\beta \leftarrow 1.1 \times \beta$
7:    end if
8: end if
```

priori, because the distances that define "neighbors" are measured in the objective space, while the mutation must be performed in the decision variable space. Therefore, a dynamic adaptation is necessary.

A too small β would cause the mutated individuals to become near the original ones, leading to distances (in the objective space) smaller than ρ. The dynamic control of β is built in order to produce a reference proportion of new non-dominated individuals outside the sphere of radius ρ around the original ones. In the instance case shown above, if less than 40% of the new non-dominated individuals are outside such sphere, the mutation radius β is increased by 10%. On the other hand, if more than 80% of such individuals become outside such sphere, the radius is reduced to 90% of its value; this is performed in order to guarantee the local search nature of the *surface filling mutation* operator.

Once more, the adaptation of parameter β employs a feedback control scheme. In this case, a switched control action is performed over variable β.

This mechanism gives rise to a local search procedure that performs perturbation steps that are both local and not too small, leading to an enhanced efficiency in the search.

3.4 Inverse-Fitness Mutation

Another mutation operator, to be applied over the *current population* set, is also defined: the *inverse-fitness mutation*. This is a simple Gaussian mutation:

$$\hat{x} \leftarrow x + \eta \Gamma \omega$$

in which $\omega \in \mathbb{R}^n$ is obtained from a Gaussian distribution (0,1). The radius η is variable: each individual has its own radius, that depends on its fitness value. The idea is to rank the fitness of all individuals in the population, and assign an η to each individual as a linear function of its position in the ranking, with the worst individual receiving an $\eta = \sigma$, where σ is the search radius employed in the generation of the initial population. After this, the values of η that are smaller than 2% of σ are changed to that value. Such mutation is similar to the one proposed in [4], in the context of Artificial Immunological Algorithms.

4 An Instance of EMOA: The SCMGA

In order to evaluate the *sphere-control* operators, an instance of multiobjective genetic algorithm is proposed here, using such operators along with some other well-known operators. This algorithm instance is called here the SCMGA, standing for *Sphere-Control Multiobjective Genetic Algorithm.*

The additional operators that are needed are defined as:

- Initial population: generated from a Gaussian distribution, with a given search radius σ, around a given center x_0;
- Crossover: the real-biased scheme presented in [9] is employed here[1];
- Pareto-ranking fitness assignment: the scheme employed by MOGA [10,11] is employed here;
- Selection: a binary tournament is employed, over a set of individuals composed by the old population plus the individuals generated via the mutation and crossover operations;

[1] The real-biased crossover scheme is stated as follows: (i) Consider the parent individuals A and B. Without loss of generality, consider that the fitness of individual A is better than the fitness of individual B (the operator is to be applied after the selection, when the fitness values of all individuals are known). (ii) Take a parametrized line segment that passes over A and B, such that A is parametrized by 0.1 and B is parametrized by 0.9. (iii) Generate two random numbers ϕ_1 and ϕ_2, both from an uniform distribution in the interval $[0, 1]$. (iv) Take $\phi_3 = \phi_1 \times \phi_2$. Clearly, ϕ_3 has a quadratic distribution over the same interval, with greater density values near 0, and smaller density values near 1. (v) One offspring individual is taken as the point parametrized by ϕ_3 (which means that this point has a greater probability of being near A, the best parent, than of being near B). (vi) The other offspring is obtained as an uniform-distribution sample of the same segment.

This algorithm, featuring also all *sphere-control* operators, is employed here with the purpose of performing some preliminary tests on the proposed new operators.

5 Results

5.1 Establishing the SCMGA

A preliminary test is conducted for the purpose of verifying if the proposed algorithm gives Pareto-set estimates that are compatible with the currently employed reference algorithms. For this purpose, algorithms and benchmark functions which are available at the PISA platform[2] are employed here. The functions to be employed are the Kursawe function with 3 and with 8 variables (denoted respectively by Kur-3 and Kur-8), and the Quagliarella & Vicini function with 3 variables (denoted by QV-3), all described in [12]. These particular functions have been chosen because they are the only options in PISA which comply with the requirements of (i) being of continuous-variables, and (ii) not depending strongly on the definition of "bounding boxes" for the decision variables (this last requirement is due to the particularly simple implementation of SCMGA used here, which has not been constructed for dealing with such bounding box constraints).

The reference algorithms are the Nondominated Sorting GA II (NSGA-II) presented in [13], the Strength Pareto Evolutionary Algorithm 2 (SPEA-2) presented in [12], the Indicator-Based Evolutionary Algorithm (IBEA) presented in [14] and the Hypervolume Estimation Algorithm for Multiobjective Optimization (HypE) presented in [15]. All those algorithms have been run as available from PISA.

All algorithms, for all test functions, have been executed with population size of 300 individuals, number of parent individuals and number of offspring individuals per generation both equal to 150, and other specific parameters of each algorithm as defined by default in PISA. The SCMGA has been run with population of 150 individuals and the reference size of the archive set equal to 300. The algorithms have been assigned 30000 function evaluations in all cases.

The comparisons have been performed in the following way. Each algorithm is executed once over each problem. For each problem, the 300 solutions of the five algorithms, are pooled in a single set (the combined Pareto-set estimate), and a non-dominance algorithm is executed over this pool set. After that, the number of solutions coming from each algorithm in the pool are counted, and the results are presented in Table 1.

A visual comparison, which cannot be shown here due to space limitation, also indicates that SCMGA produces solution sets that are well-distributed. As long as such results are "typical" (they are similar in several executions), it seems reasonable to conduct further studies about the SCMGA. We advise the reader that due to the limited number of tests that has been performed so far,

[2] http://www.tik.ee.ethz.ch/pisa/

Table 1. Results of different algorithms on benchmark problems, in a single execution

	pool size	NSGA-II	SPEA-2	IBEA	HypE	SCMGA
Kur-3	376	22	17	27	24	286
Kur-8	288	0	0	0	0	288
QV-3	1049	171	178	184	228	288

the only conclusion that can be drawn is that SCMGA produces results that are compatible with the ones produced by other standard algorithms. Further conclusions with the meaning of a comparison will need much more tests.

5.2 The Structure of Control Action

An experiment has been conducted with the SCMGA, with the same parameters above, in the 3D Kur problem, allowing 160000 function evaluations (which have been performed in 498 generations). Figure 2 presents the evolution of the number of points in the archive set, $|A|$, along with the evolution of the size of the sphere radius ρ which controls such set size. The reference adopted for $|A|$ is 300 points (i.e., the control subroutine will try to reach this reference and stay on it). Interesting observations can be drawn from Figure 2. It should be noticed that the number of points grows in the first phase of the algorithm time evolution. As long as the number of points is smaller than the reference in this

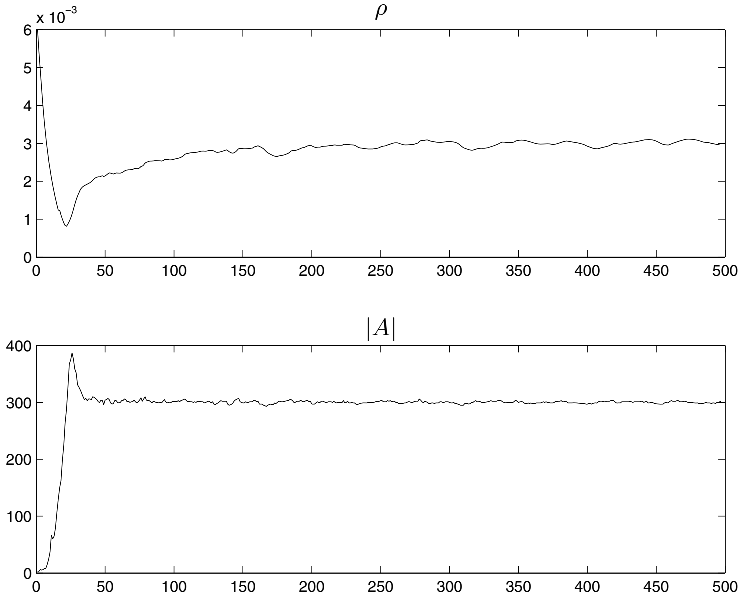

Fig. 2. The evolution of the size of the sphere radius, ρ (upper figure); and of the number of points in the archive set A, denoted by $|A|$ (lower figure), with the number of SCMGA generations

first phase, the control action results in a continuous decreasing of the radius ρ. At the generation 21, the number of points becomes greater than the reference value; the control variable ρ begins to grow. However, as its value has possibly been set to a value smaller than the "equilibrium" one, the controlled variable $|A|$ continues to grow, giving rise to an "overshoot" effect. Further increments in ρ lead the controlled variable $|A|$ to track the reference value nearly at the generation 30. In all generations after that one, $|A|$ presents a small oscillation around the reference. It is important to notice the behavior of the control variable ρ after the moment that $|A|$ reaches the reference value: although $|A|$ becomes nearly constant up to the end of algorithm execution, ρ presents a trend of slow growth from generation 30 up to generation 150. This is related to an adjustment of the samples y_i in A_y, which become increasingly more well-distributed along the Pareto-set of the problem. This leads to a greater "smallest" distance among any two points in A_y. After generation 150, up to the end, the value of control variable ρ becomes approximately constant, with a small oscillation around this "equilibrium".

Figure 3 presents the evolution of the size of the surface-filling mutation radius, β, along with the evolution of the proportion of successful (not too close to the original point) attempts of mutation. The control variable β is switched in order to keep the value of the controlled variable k_4/k_3 between the values 0.4 and 0.8. It can be seen that, at the beginning of algorithm execution, most of successful mutations occur at large distances (the proportion k_4/k_3 is nearly equal to 1). In order to guarantee that the mutation operator performs a local

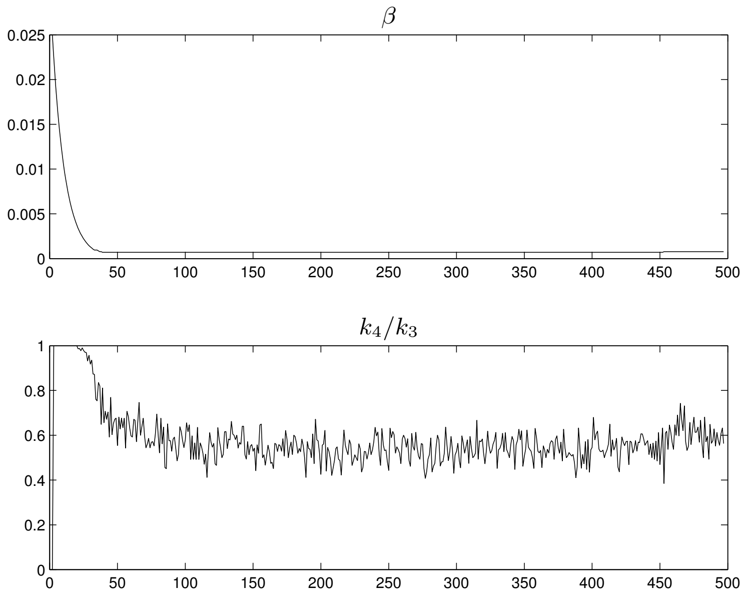

Fig. 3. The size of the mutation radius, β (upper figure); and the proportion k_4/k_3 of successful (not too close) attempts of mutation (lower figure)

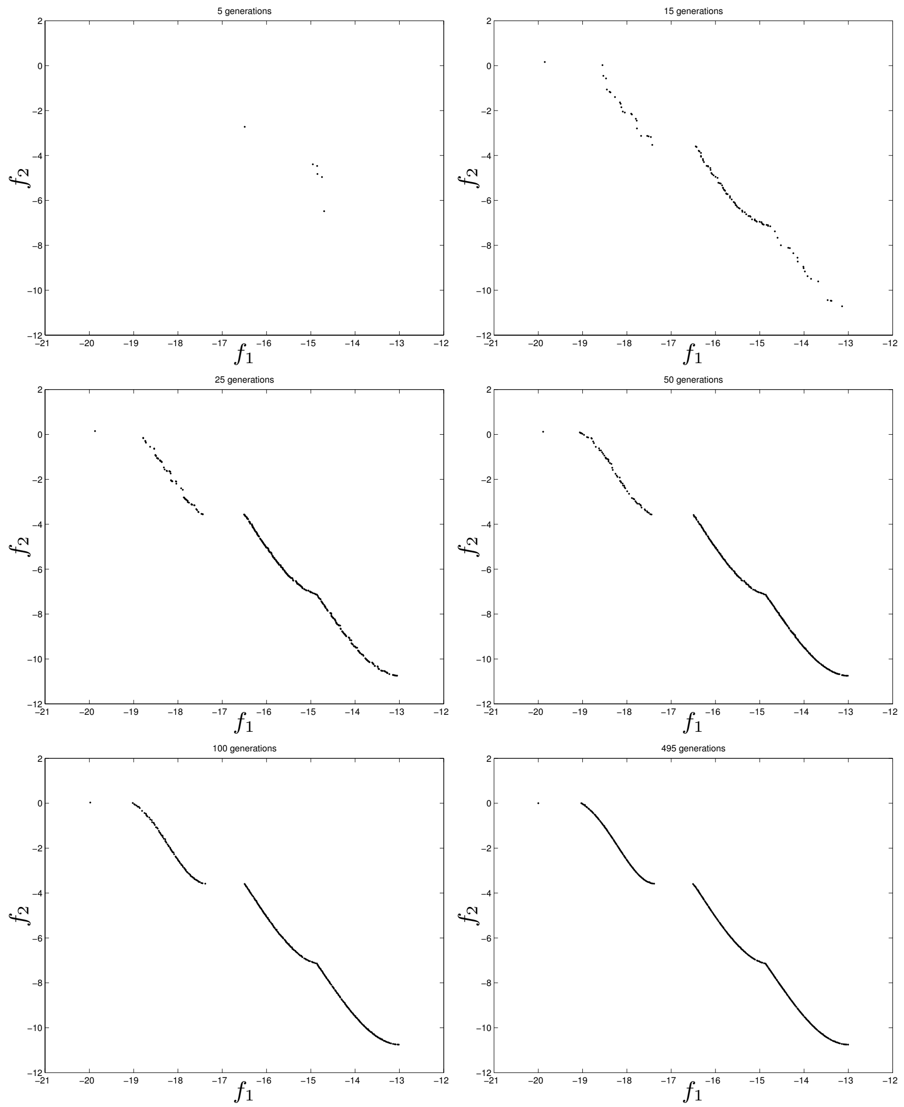

Fig. 4. Evolution of the archive-set image A_y. Top-left: 5 generations. Top-right: 15 generations. Middle-left: 25 generations. Middle-right: 50 generations. Bottom-left: 100 generations. Bottom-right: 495 generations.

search, the mutation radius β is reduced: this causes the controlled variable k_4/k_3 to reduce. Both the control variable β and the controlled variable k_4/k_3 reach an equilibrium from the generation 50 up to the end of algorithm execution. It should be noticed that the controlled variable reaches a fast oscillatory motion around the value 0.6, which is the center of the reference band.

The gradual evolution of the archive set image A_y is depicted in the sequence of frames in Figure 4.

It can be seen that, in the first 5 generations, very few archive points have been generated. A delineation of an estimation of the Pareto-front starts to appear near generation 15. The lower portion of the Pareto-front becomes well-delineated nearly at generation 25, and becomes "completed" nearly at generation 50. The upper portion of the front, however, becomes completed only nearby generation 100. From generation 100 to generation 495, there is no visually perceptible enhancement in the front.

These observations should be compared with the former analysis, which indicated roughly that the variables $|A|$ and β become stable before generation 50, the variable k_4/k_3 reaches equilibrium after generation 50 and before generation 100, and variable ρ becomes in equilibrium nearby generation 100. A very interesting conjecture to be launched is: the equilibrium of such variables seems to be related to the completion of Pareto-set description.

6 Conclusion

The algorithm SCMGA, constructed with the proposed sphere-control operators, has shown effectiveness in finding a well-defined and stable Pareto-set estimate. The algorithm seems to be endowed with a capability of guiding the search toward the "less-defined" regions of the current estimate, until producing a complete estimate. The dynamic interaction between the algorithm internal variables that is induced by a feedback-control scheme seems to play an essential role in constituting such property.

Beyond the specific operators and the algorithm instance that have been presented here, the authors believe that the main contribution of this paper is to introduce the idea of using feedback-control principles for the design of evolutionary computation algorithms. The main concern in using such principles is to define, in a meaningful way, the control variables and the controlled variables of the feedback scheme.

References

1. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Trans. on Evolutionary Computation 7(2), 117–132 (2003)
2. Silva, V.L.S., Wanner, E.F., Cerqueira, S.A.A.G., Takahashi, R.H.C.: A new performance metric for multiobjective optimization: The integrated sphere counting. In: Proc. IEEE Congress on Evolutionary Computation, Singapore (2007)
3. Ogata, K.: Modern Control Engineering, 4th edn. Prentice-Hall, Englewood Cliffs (2001)
4. de Castro, L.N., Timmis, J.: An artificial immune network for multimodal function optimization. In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation, vol. 1, pp. 699–704 (2002)
5. Takahashi, R.H.C., Palhares, R.M., Dutra, D.A., Gonalves, L.P.S.: Estimation of Pareto sets in the mix H_2/H_{inf} control problem. International Journal of Systems Science 35(1), 55–67 (2004)

6. Bui, L.T., Deb, K., Abbass, H.A., Essam, D.: Interleaving guidance in evolutionary multi-objective optimization. Journal of Computer Science and Technology 23(1), 44–63 (2008)
7. Bui, L.T., Abbass, H.A., Essam, D.: Local models – an approach to distributed multi-objective optimization. In: Computational Optimization and Applications (2007) (to appear, published online in 2007), doi:10.1007/s10589-007-9119-8
8. Wanner, E.F., Guimaraes, F.G., Takahashi, R.H.C., Fleming, P.J.: Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria. Evolutionary Computation 16(2), 185–224 (2008)
9. Takahashi, R.H.C., Vasconcellos, J.A., Ramirez, J.A., Krahenbuhl, L.: A multiobjective methodology for evaluation genetic operators. IEEE Trans. on Magnetics 39, 1321–1324 (2003)
10. Fonseca, C.M., Fleming, P.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the 5th International Conference: Genetic Algorithms, San Mateo, USA, pp. 416–427 (1993)
11. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 7(3), 205–230 (1995)
12. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Tech. Rep. 103 (2001)
13. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
14. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)
15. Bader, J., Zitzler, E.: HypE: Fast Hypervolume-Based Multiobjective Search Using Monte Carlo Sampling. Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, TIK Report 286 (October 2006)