Approximating Pareto Frontier Using a Hybrid Line Search Approach

Crina Grosan and Ajith Abraham, *Senior member, IEEE*

***Abstract*— The aggregation of objectives in multiple criteria programming is one of the simplest and widely used approach. But it is well known that this technique sometimes fail in different aspects for determining the Pareto frontier. This paper proposes a new approach for multicriteria optimization which aggregates the objective functions and uses a line search method in order to locate an approximate efficient point. Once the first Pareto solution is obtained, a simplified version of the former one is used in the context of Pareto dominance to obtain a set of efficient points, which will assure a thorough distribution of solutions on the Pareto frontier. In the current form, the proposed technique is well suitable for problems having multiple objectives (it is not limited to bi-objective problems) and require the functions to be continuous twice differentiable. In order to assess the effectiveness of this approach, some experiments were performed and compared with two recent well known population-based metaheuristics ParEGO [15] and NSGA II [6]. When compared to ParEGO and NSGA II, the proposed approach not only assures a better convergence to the Pareto frontier but also illustrates a good distribution of solutions. From a computational point of view, both stages of the line search converge within a short time (average about 150 milliseconds for the first stage and about 20 milliseconds for the second stage). Apart from this, the proposed technique is very simple, easy to implement and use to solve multiobjective problems.**

***Index Terms*—Fuzzy controller, line search, multuiobjective optimization, pharmaceutical applications.**

I. Introduction

The field of multicriteria programming abounds in methods dealing with different kind of problems. Nevertheless, there is still space for new approaches, which can better deal with some of the difficulties encountered by the existing approaches. There are two main classes of approaches suitable for multiobjective optimization: scalarization methods and nonscalarizing methods. These approaches convert the Multiobjective Optimization Problem (MOP) into a Single Objective Optimization Problem (SOP), a sequence of SOPs, or into another MOP. There are several scalarization methods reported in the literature: weighted sum approach, weighted t-th power approach, weighted quadratic approach, ε-constraint approach, elastic constraint approach, Benson approach, etc. are some of them [11]. Since the standard weighted sum encounters some difficulties, several other methods have been proposed to overcome the major drawbacks of this method. These include: Compromise Programming [9], Physical Programming [18][19][20][21][22][23][24], Normal Boundary Intersection (NBI) [2][3][4][5], and the Normal Constraint (NC) [25][26] methods. There is also a huge amount of work reported on population-based mataheuristics for MOP [11] [8] [10][31][36]. Comprehensive surveys can be found in [33], [14] [27].

In this paper, we propose a new approach which uses a scalarization of the objectives in a way similar to the weighted *t*-th power approach (where *t* is 2 and the coefficients values are 1). A line search based technique is used to obtain an efficient solution. Starting with this solution, a set of efficient points are further generated, which are widely distributed along the Pareto frontier using again a line search based method but involving Pareto dominance relationship.

Empirical and graphical results and illustrations obtained by the proposed approach are compared with two well known population based metaheuristics namely ParEGO [15] and NSGA II [6].

The paper is structured as follows: in Section 2 the proposed modified Line Search is presented. Numerical experiments considering standard benchmarks are performed in Section 3. A set of 8 multiobjective optimizations problems are considered. Sections 4 and 5 present the application of the proposed approach for solving two practical Multiobjective optimization problems. Conclusions and further research plans are presented in Section 6.

II. Line search generator of Pareto frontier

The line search [12] is a standard and well established optimization technique. The standard line search technique is modified in this paper so that it is able to generate the set of non-dominated solutions for a MOP. The approach proposed is called Line search Generator of Pareto frontier (LGP) and it comprises of two phases: first, the problem is transformed

Crina Grosan is with the Norwegian University of Science and Technology, Trondheim, Norway (corresponding author: phone: +47 73 55 02 67; fax:+47 73 59 27 90; e-mail: crina@ q2s.ntnu.no).

Ajith Abraham is with the Norwegian University of Science and Technology, Trondheim, Norway (ajith.abraham@ieee.org).

into a SOP and a solution is found using a line search based approach. This is called as convergence phase. Second, a set of Pareto solutions are generated starting with the solution obtained at the end of convergence phase. This is called as spreading phase. The convergence and spreading phases are described below.

Consider the MOP formulated as follows:

Let $\Re^m$ and $\Re^n$ be Euclidean vector spaces referred to as the decision space and the objective space. Let $X \subset \Re^m$ be a feasible set and let f be a vector-valued objective function f: $\Re^m \rightarrow \Re^n$ composed of n real-valued objective functions $f=(f_1, f_2,\ldots, f_n)$, where f_k: $\Re^m \rightarrow \Re$, for $k=1,2,\ldots, n$. A MOP is given by:

$\min (f_1(x), f_2(x),\ldots, f_n(x))$,
subject to $x \in X$.

A. Convergence phase

The MOP is transformed into a SOP by aggregating the objectives using an approach similar to the weighted t-th power approach. We consider $t = 2$ and the values of weights equal to 1. The obtained SOP is:

$$\min F = \sum_{i=1}^{n} f_i^2(x)$$

subject to $x \in X$.

A modified line search method is used to find the optimum of this problem. The modification proposed in this paper for the standard line search technique refers to direction and step setting and also the incorporation of a re-start procedure. To fine tune the performance, the first partial derivatives of the function to optimize are also made use of. The proposed modifications refer to:

- the setting of the direction and step
- the re-starting of the line search method.

After a given number of iterations, the process is restarted by reconsidering other arbitrary starting point which is generated by taking into account the result obtained at the end of previous set of iterations.

Direction and step setting

Initially, several experiments were performed in order to set an adequate value for the direction. The standard value +1 or -1 was used and for some functions the value -1 was favourable to obtain good performance. Some experiments were also performed by setting the direction value as being a random number between 0 and 1. It was found that the usage of random number helped to obtain overall very good performance for the entire considered test functions. But usage of the value -1 for direction, obtains almost the same performance similar to that obtained with a random value. So, either of these values (the random one and the value -1) may be used for better performance.

The step is set as follows:

$$\alpha_k = 2 + \frac{3}{2^{2k}+1} \quad (1)$$

where k refers to the iteration number.

The modified *line search* technique is summarized as follows:

```
Line_search()
Set k=1 (Number of iterations)
Repeat
    for i=1 to No of variables
            p_k=random; //or p=-1;
            α_k = 2 + 3/(2^(2k)+1)
            x_i^(k+1) = x_i^k + p_k · α_k
    endfor
    if F(x^(k+1))<F(x^k) then x^(k+1) = x^k .
  k=k+1
Until k=Number of iterations (a priori known).
```

Remarks

- The condition:
- if $F(x^{k+1}) < F(x^k)$ then $x^{k+1} = x^k$
- allows to move to the new generated point only if there is an improvement in the quality of the function.
- Number of iterations for which line search is applied is apriori known and is usually a small number. For the experiments reported in this paper, the number of these iterations was set to 10.
- When restarting the line search method (after the insertion of the re-start technique) the value of the iterations number starts again from 1 (this should not be related to the value of α after the first set of iterations (and after each of the following iterations)).

Several experiments were attempted to set a value for the step, starting with random values (until a point is reached for which the objective function achieves a better value); using a starting value for the step and generating random numbers with Gaussian distribution around this number, etc. As a result of the initial experiments performed, it was decided to use equation (1) to compute the step size. But, of course, there are also several other ways to set this.

Incorporation of re-start procedure

In order to restart the algorithm the result obtained in the

previous set of iterations (denote it by *x*) is taken into account and the steps given below are followed:

For each dimension *i* of the point *x*, the first partial derivative with respect to this dimension is calculated. This means the gradient of the objective function is calculated which is denoted by *g*. Taking this into account, the bounds of the definition domain for each dimension are re-calculated as follows:

$$\text{if } g_i = \frac{\partial F}{\partial x_i} > 0 \text{ then } upper\ bound = x_i;$$

$$\text{if } g_i = \frac{\partial F}{\partial x_i} < 0 \text{ then } lower\ bound = x_i$$

The search process is re-started by re-initializing a new arbitrary point between the newly obtained boundaries.

B. *Spreading phase*

At the end of the convergence phase, a solution is obtained. This solution is considered as an efficient (or Pareto) solution. During this phase and taking into account of the existing solution, more efficient solutions are to be generated so as to have a thorough distribution of all several good solutions along the Pareto frontier. In this respect, the line search technique is made use of to generate one solution at the end of each set of iterations. This procedure is applied several times in order to obtain a larger set of non-dominated solutions. The following steps are repeated in order to obtain one non-dominated solution:

Step 1. A set of nondominated solutions found so far is archived. Let us denote it by *NonS*. Initially, this set will have the size one and will only contain the solution obtained at the end of *convergence phase*.

Step2. We apply line search for one solution and one dimension of this solution at one time. For this:

Step 2.1. A random number *i* between one and |NonS| (|·| denotes the cardinal) is generated. Denote the corresponding solution by $nonS_i$.

Step 2.2. A random number *j* between one and the number of dimensions (the number of decision variables) is generated. Denote this by $nonS_{ij}$.

Step 3. Line search is applied for $nonS_{ij}$.

Step 3.1. Set a random value for *p* between [-0.5, 1].

Step 3.2. Set α (which depends on the problem, on the number of total nondominated solutions which are to be generated, etc.).

Step 3.3. The new obtained solution *new_sol* is identical to $nonS_i$ in all dimensions except dimension *j* which is:

$new\_sol_j = nonS_{ij} + \alpha \cdot p$

Step 3.4. if ($new\_sol_j > upper\ bound$) or ($new\_sol_j < lower\ bound$)

then $new\_sol_j = lower\ bound + \text{random} \cdot (upper\ bound - lower\ bound)$.

Step 4. if $F(new\_sol) > F(nonS_1)$

then discard *new_sol*

else if *new_sol* is nondominated with respect to the set *NonS*

then add *new_sol* to *NonS* and increase the size on *NonS* by 1.

Go to step 2.

Step 5. Stop

These steps are repeated until a set on nondominated solutions of a required size is obtained. In our experiments the size of this set is 100. Note that this procedure it very fast and it takes less than 20 milliseconds to obtain 100 non-dominated solutions.

1) *Estimating the value of α using a Fuzzy Logic Controller*

The performance of the line search algorithm is correlated to directly with its careful selection of α value. The use of fuzzy logic controllers to adapt the α value is useful to improve the performance. An FLC is composed by a knowledge base, that includes the information given by the expert in the form of linguistic control rules, a fuzzification interface, which has the effect of transforming crisp data into fuzzy sets, an inference system, that uses them together with the knowledge base to make inference by means of a reasoning method, and a defuzzification interface, that translates the fuzzy control action thus obtained to a real control action using a defuzzification method. The generic structure of an FLC is shown in Fig. 2.

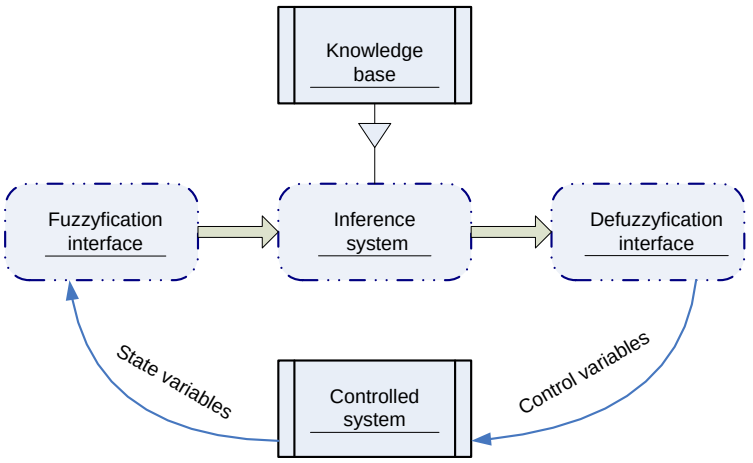

Fig. 1. Generic structure of an FLC

In order to set an adequate value for α so that the solutions will have a good distribution on the Pareto front, we are proceeding as follows:

- Select a sample set of solutions uniform distributed on the Pareto front (denoted by SPS) of size equal to the size of the approximation set obtained by the our approach.
- For each point from the approximation set obtained by our approach identify the closest point in SPS.

- Mark each such identified point from SPS.
- Set the value of distribution indices (D_i) as being equal to the number of marked points from SPS.

Our strategy for updating the α value is to consider the changes of the value of maximum distribution indices (D_{im}) and average distribution indices (D_{ia}) in two continuous iterations. The performance may be measured using two error indices:

$$e_1(t) = \frac{D_{im}(t) - D_{ia}(t)}{D_{im}(t)} \quad (2)$$

$$e_2(t) = \frac{D_{ia}(t) - D_{ia}(t-1)}{D_{im}(t)} \quad (3)$$

Where t is time step,

- $D_{im}(t)$ is the maximum distribution index at iteration t,
- $D_{ia}(t)$ is the average distribution index at iteration t,
- $D_{ia}(t\text{-}1)$ *is the* average distribution index at iteration (t-1).

A two-dimension FLC system is used, in which there are two parameters e_1 and e_2.The membership functions are shown in Fig. 2, where *NL* is Negative large, *NS* is Negative small, *ZE* is Zero, *PS* is Positive small, *PL* is Positive large.

For the controlling the performance, the output α**(***t*) of the fuzzy logic controller is translated using fuzzy *if-then* rules as illustrated in Fig. 3. Center of gravity is used as defuzzification method. Then we use the crisp value to modify the parameters α as follows:

$\alpha(t) = \alpha(t-1) + \Delta\alpha$.

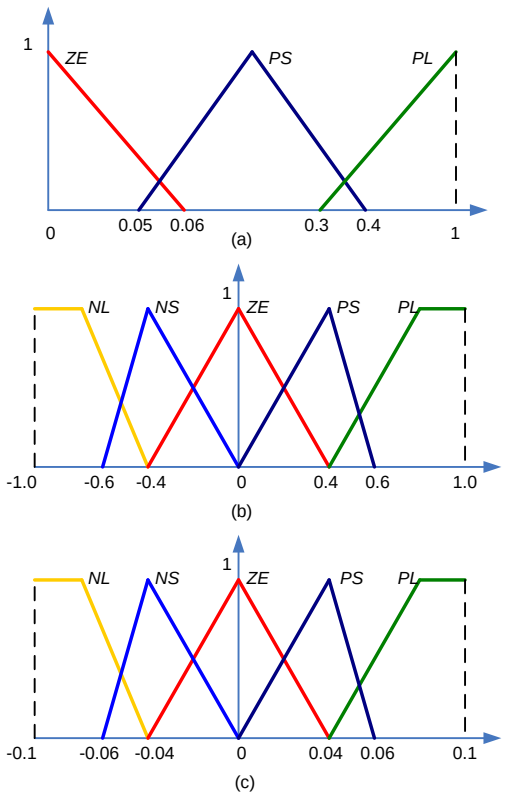

Fig. 2. Membership functions. (a) for e_1; (b) e_2; (c) for Δα.

For applying the procedure described above, the Pareto front it is supposed to be known (and this is the case in all our experiments considered). In Fig. 4, two approximation sets A and B and a sample set of Pareto points (SPS) of size 10 are considered. The value of D_i for the set A is 6 (which means 6 solutions from the SPS are marked) while the value of D_i for the set B is 10. This means set B is obtaining a better distribution on the Pareto front than the set A.

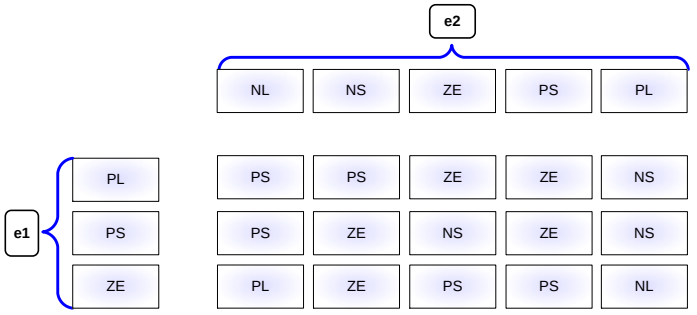

Fig. 3. Fuzzy rules for Δα.

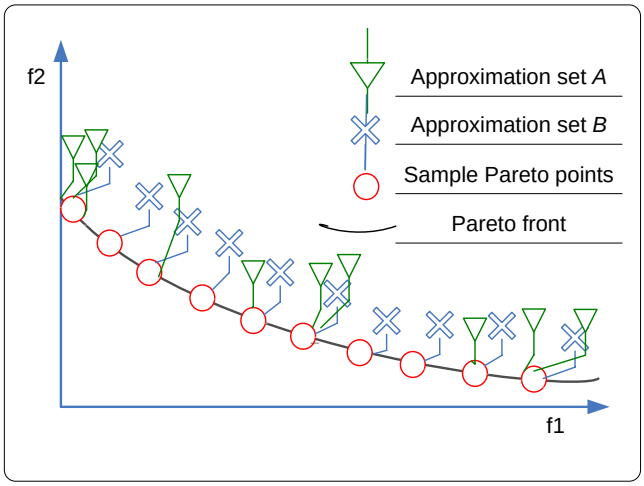

Fig. 4. Illustration of Pareto approximation used for D_i calculation.

Example

An illustrative example for calculation of α is presented in Fig. 5. Suppose we have an approximation set containing 100 solutions. This involves the size of the selected Pareto sample points will be also 100. Let us also suppose that are at the third iteration and we obtained the value 60 for Di for α = 0.1 in the first iteration and value 63 for Di for α = 0.15 in the second iteration. It should be mentioned that the value of α in the first iteration was randomly chosen and for the value of α in the second iteration we had the chance to increase or decrease de current value; so, we increased it to 0.15.

Starting with the third iteration we can apply our fuzzy rules to calculate the next value forα. By using the formulas (2) and (3), the membership functions from Fig. 2 and the fuzzy rules from Fig. 3, the value 0.04 for Δα is obtained. Based on this, the value of α at the iteration 3 will be 0.15 + 0.04 = 0.19.

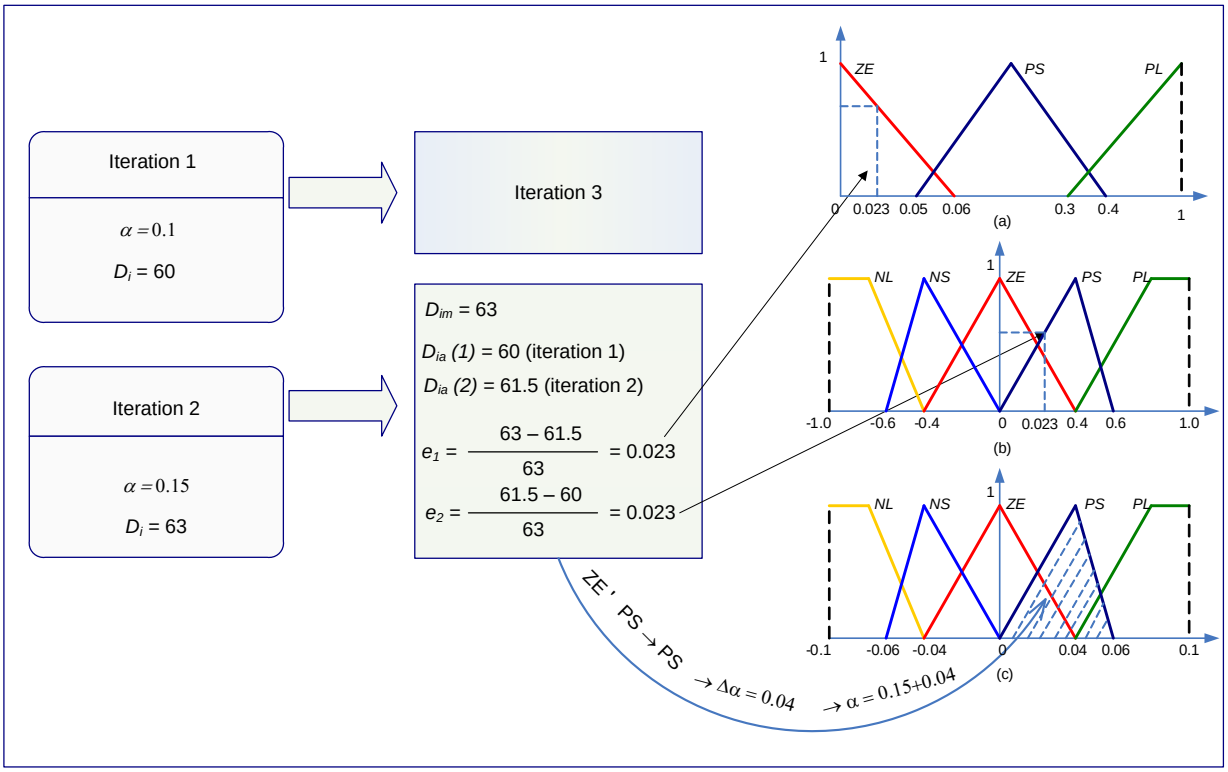

Fig. 5. Example of α calculation.

TABLE I. PARAMETERS USED IN EXPERIMENTS BY PAREGO AND NSGA II. *D* DENOTES THE NUMBER OF DECISION PARAMETER DIMENSIONS.

ParEGO		NSGA II	
Parameter	*Value*	*Parameter*	*Value*
Initial population in latin	$11d - 1$	Population size	20

hypercube			
Total maximum evaluations	250	Maximum generations	13
Number of scalarizing vectors	11 for 2 objectives 15 for 3 objectives	Crossover probability	0.9
Scalarizing function	Augmented Tchebycheff	Real value mutation probability	1/*d*
Internal genetic algorithm evaluations per iteration	200,000	Real value SBX parameter	10
Crossover probability	0.2	Real value mutation parameter	50
Real value mutation probability	1/*d*		
Real value SBX parameter	10		
Real value mutation parameter	50		

III. Experiments and Comparisons

In order to assess the performance of LGP, some experiments were performed using some well known bi-objective and three-objective test functions, which are adapted from [7], [13]. These test functions were also used by the authors of ParEGO [15] and NSGA II [6], which are well known in the computational intelligence community as very efficient techniques for multiobjective optimization. Details about implementation of these two techniques may obtained from [6] and [15]. Parameters used by ParEGO and NSGA II (given in Table I) and the results obtained by these two techniques are adapted from [15].

A set of 100 non-dominated solutions obtained by LGP, ParEGO, NSGA II is compared in terms of dominance and convergence to the Pareto set. For the first comparison, two indices were computed for each set of two comparisons: number of solution obtained by the first technique which dominate solutions obtained by the second technique and number of solutions obtained by the first technique which are dominated by the solutions obtained by the second technique.

For two sets of *A* and *B* of solutions, which are compared, indices are denoted by *Dominate(A, B)* and *Dominated(A, B)* respectively. Visualization plots are used to illustrate the distribution of solutions on the Pareto frontier.

LGP uses only three parameters:

number of re-starts: 20 (10 for KNO1);

number of iteration per each re-start: 10;

α for the spreading phase (which is set independent for each test function).

Test function KNO1

This test function has two variables and two objectives. It is given by:

$$minimize \quad f_1 = 20 - r \cdot \cos(\phi)$$
$$minimize \quad f_2 = 20 - r \cdot \sin(\phi)$$

where

$$r = 9 - (3\sin\left(\frac{5(x_1 + x_2)^2}{2}\right) + 3\sin(4(x_1 + x_2)) + 5\sin(2(x_1 + x_2) + 2)$$
$$\phi = \frac{\pi(x_1 - x_2 + 3)}{12}$$

The distance from the Pareto front is controlled by *r* and is a function of the sum of the decision variables. The location transverse to the Pareto front is controlled by the difference between the decision variables. Pareto set consists of all pairs whose sum is 4.4116. There are 15 local Pareto fronts and the true Pareto front lies just beyond a local Pareto front which has a larger basin of attraction.

The convergence to the Pareto frontier and the distribution of solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a is depicted in Fig. 6. Different sizes of the objective space are illustrated in order to incorporate all solutions obtained by all techniques. The value of α in the spreading phase which is adapted by the fuzzy controller is 0.62. The behavior of the merit function during the 10 re-starts is depicted in Fig. 7. From the results presented in Table II it can be observed that 7 of the solutions obtained by LGP are dominated by solutions obtained by ParEGO and 2 are dominated by solutions obtained by NSGA II. Solutions obtained by LGP dominate all 100 solutions obtained by both ParEGO and NSGA II. 59 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 42 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

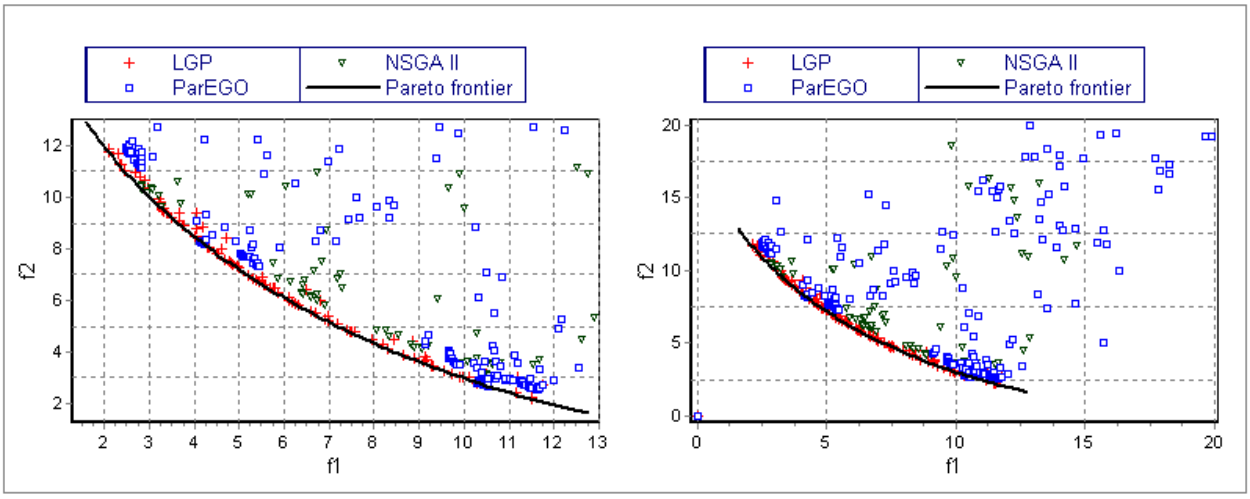

Fig. 6. Distribution of solutions on the Pareto frontier obtained by LGP, ParEGO and NSGA II for test function KNO1.

Test function OKA1

This test function and the test function OKA2 have been proposed in [26]. It is a bi-objective test function having two variables and it is defined as:

$$minimize\ f_1 = x'_1$$

$$minimize\ \ f_2 = \sqrt{2\pi} - \sqrt{|x'_1|} + 2|x'_2 - 3\cos(x'_1) - 3|^{\frac{1}{3}}$$

where

$$x'_1 = \cos\left(\frac{\pi}{12}\right)x_1 - \sin\left(\frac{\pi}{12}\right)x_2$$

$$x'_2 = \sin\left(\frac{\pi}{12}\right)x_1 + \cos\left(\frac{\pi}{12}\right)x_2$$

$$x_1 \in \left[6\sin\left(\frac{\pi}{12}\right),\ 6\sin\left(\frac{\pi}{12}\right) + 2\pi\cdot\cos\left(\frac{\pi}{12}\right)\right]$$

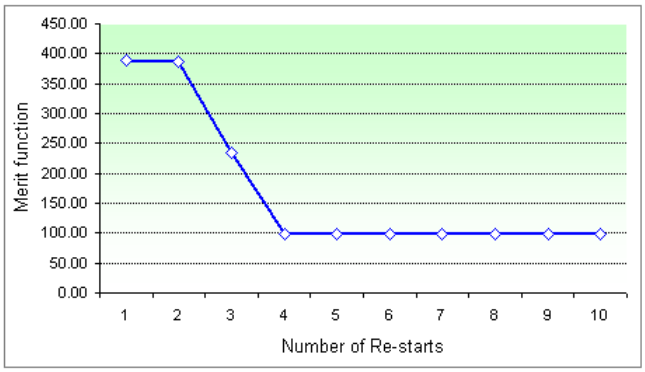

Fig. 7. Behavior of merit function for test function KNO1 during the *convergence phase*.

$$x_2 \in \left[-2\pi\cdot\sin\left(\frac{\pi}{12}\right),\ 6\cos\left(\frac{\pi}{12}\right)\right]$$

The Pareto optimal set lies on the curve $x'_2 = 3\cos(x'_1)+2,\ x'_1 \in [0, 2\pi]$.

The solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a are depicted in Fig. 8. Different sizes of the objective space are illustrated in order to incorporate all solutions obtained by all techniques. The value of α in the spreading phase which is adapted by the fuzzy controller is 0.984. The behavior of the merit function during the 20 re-starts is depicted in Fig. 9. From the results presented in Table III it can be observed that none of the solutions obtained by LGP are dominated by solutions obtained by either ParEGO or NSGA II. Solutions obtained by LGP dominate 83 solutions obtained by both ParEGO and 64 solutions obtained by NSGA II. 77 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 59 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

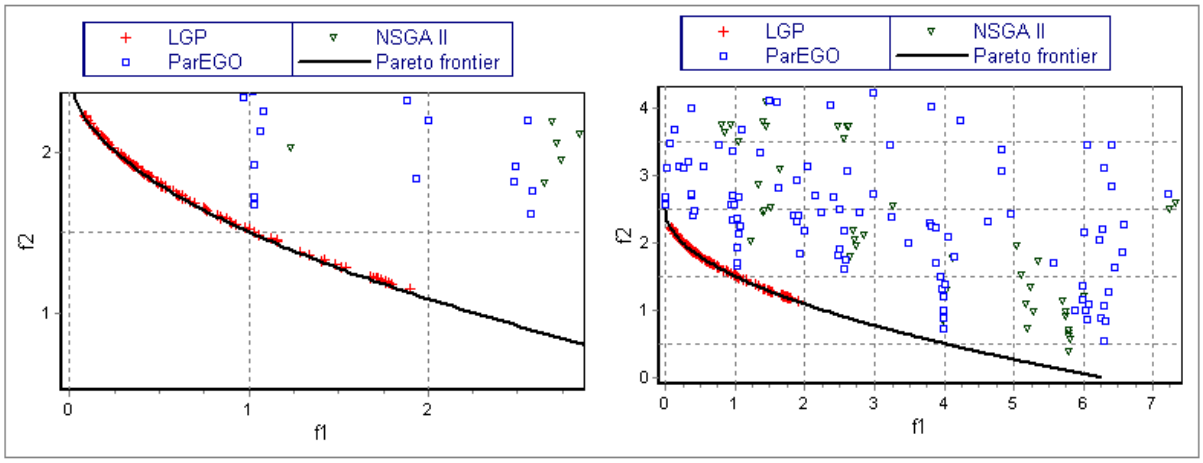

Fig. 8. Distribution of solutions on the Pareto frontier obtained by LGP, ParEGO and NSGA II for test function OKA1.

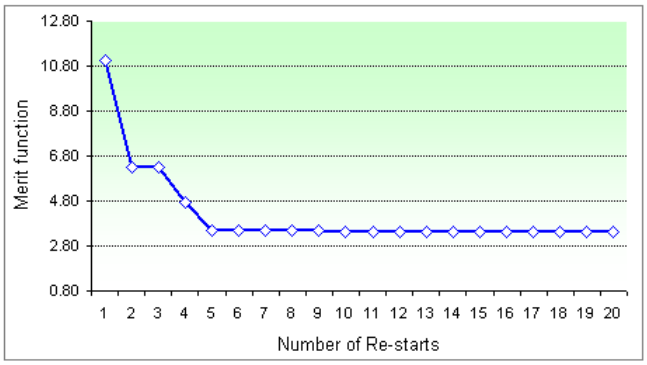

Fig. 9. Behavior of merit function for test function OKA1 during the *convergence phase*.

Test function OKA2

Test function OKA2 is given by:

minimize $f_1 = x_1$

minimize

$$f_2 = 1 - \frac{1}{4\pi^2}(x_1 + \pi)^2 + \left|x_2 - 5\cdot\cos(x_1)\right|^{\frac{1}{3}} + \left|x_3 - 5\cdot\sin(x_1)\right|^{\frac{1}{3}}$$

$x_1 \in [-\pi, \pi]$, $x_2, x_3 \in [-5, 5]$.

The Pareto optima lie on a spiral-shaped curve in the three dimensional variable space.

The Pareto front is given by

$$f_2 = 1 - \frac{1}{4\pi^2}(f_1 + \pi)^2, \quad f_1 \in [-\pi, \pi].$$

The convergence to the Pareto frontier and the distribution of solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a is depicted in Fig. 10. The value 1 is considered for α for this test function. The behavior of the merit function during the 20 re-starts is depicted in Fig. 11. From the results presented in Table IV it can be observed that 28 of the solutions obtained by LGP are dominated by solutions obtained by ParEGO while 37 solutions obtained by ParEGO and 41 solutions obtained by NSGA II are dominated by solutions obtained by LGP. 31 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 69 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

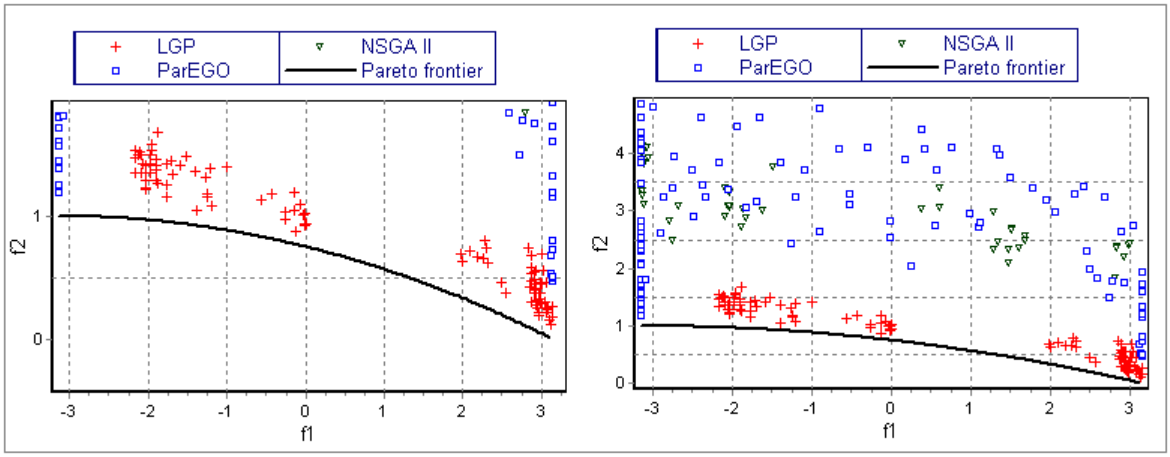

Fig. 10. Distribution of solutions on the Pareto frontier obtained by LGP, ParEGO and NSGA II for test function OKA2.

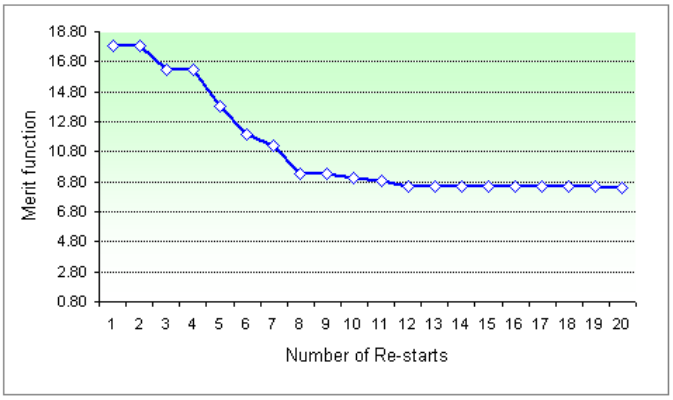

Fig. 11. Behavior of merit function for test function OKA2 during the *convergence phase*.

Test function VLMOP2

This test function has been proposed in [35]. It is a bi-objective problem having scalable number decision variables. We use 2 variables.

$$minimize\ f_1 = 1 - \exp\left(\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right)$$

$$minimize\ f_2 = 1 - \exp\left(\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$$

$x_1, x_2 \in [-2, 2]$, n=2.

The Pareto front is concave and the Pareto optima lie on the diagonal passing from $\left(-\frac{1}{\sqrt{n}}, -\frac{1}{\sqrt{n}}\right)$ to $\left(\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right)$ in the decision variable space.

The convergence to the Pareto frontier and the distribution of solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a is depicted in Fig. 12. The value of α in the spreading phase which is adapted by the fuzzy controller is 0.968. The behavior of the merit function during the 20 re-starts is depicted in Fig. 13. From the results presented in Table V it can be observed that 6 of the solutions obtained by LGP are dominated by solutions obtained by ParEGO. Solutions obtained by LGP dominate 49 solutions obtained by ParEGO and 75 solutions obtained by NSGA II. 75 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 37 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

Test function VLMOP3

Test function VLMOP3 has been proposed by Veldhuizen and Lamont in [35] and consists of three objective functions of two variables given by:

$$\text{minimize } f_1 = 0.5\left(x^2 + y^2\right) + \sin\left(x^2 + y^2\right)$$

$$\text{minimize } f_2 = \frac{(3x - 2y + 4)^2}{8} + \frac{(x - y + 1)^2}{27} + 15$$

$$\text{minimize } f_2 = \frac{1}{x^2 + y^2 + 1} - 1.1 \cdot \exp\left(-x^2 - y^2\right)$$

x, y ∈ [-3, 3].

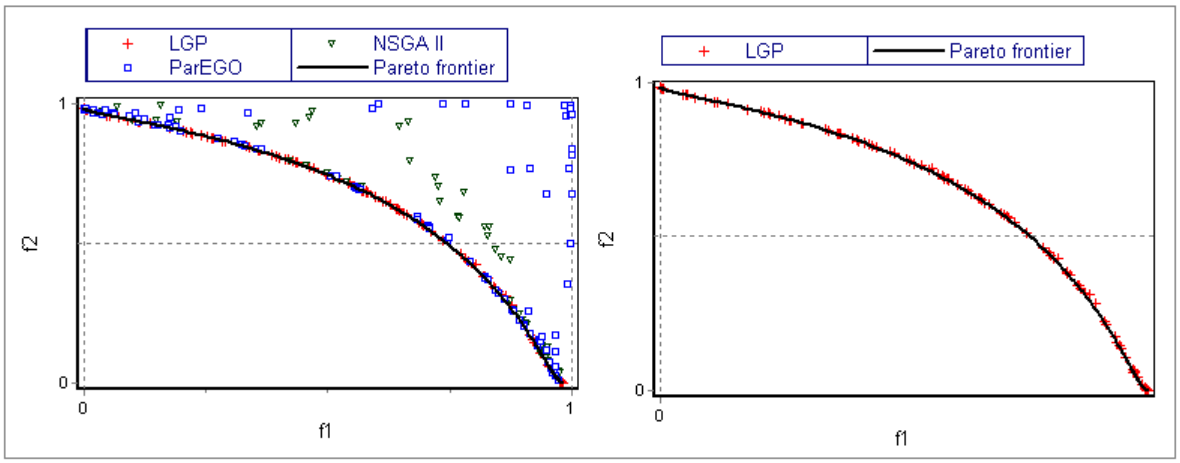

Fig. 12. Distribution of solutions on the Pareto frontier obtained by LGP, ParEGO and NSGA II for test function VLMOP2.

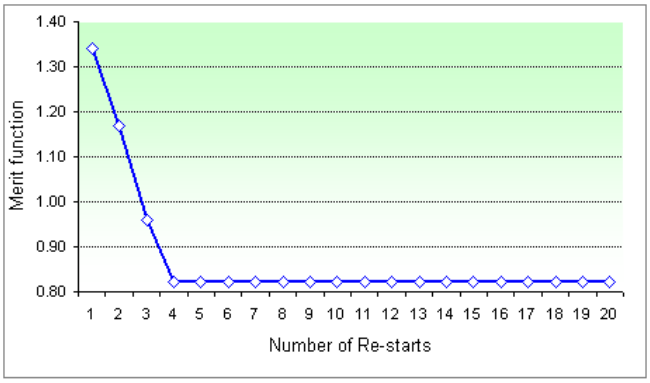

Fig. 13. Behavior of merit function for test function VLMOP2 during the *convergence phase*.

This test function has a disconnected Pareto optimal set and the Pareto optimal front is a courve following a convoluted path through objective space.

The convergence to the Pareto frontier and the distribution of solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a is depicted in Fig. 14. The value of α in the spreading phase which is adapted by the fuzzy controller is 0.95. Pareto front obtained by LGP considering a set of 1000 solutions is depicted in Fig. 15. The behavior of the merit function during the 20 re-starts is depicted in Fig. 16. From the results presented in Table VI it can be observed that 6 of the solutions obtained by LGP are dominated by solutions obtained by ParEGO. Solutions obtained by LGP dominate 49 solutions obtained by ParEGO and 75 solutions obtained by NSGA II. 75 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 37 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

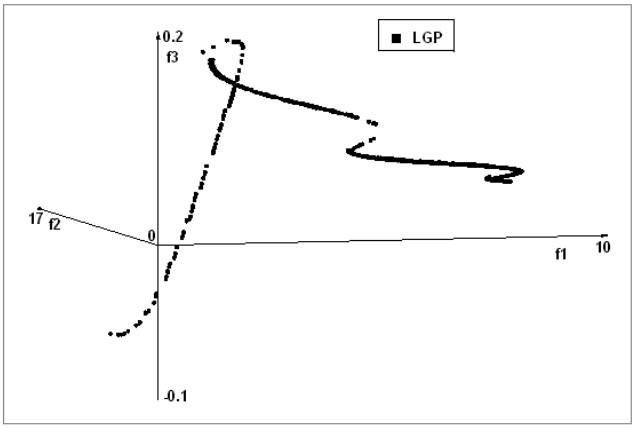

Fig. 15. Pareto front obtained by LGP for VLMOP3 test problem with a final set of 1000 solutions.

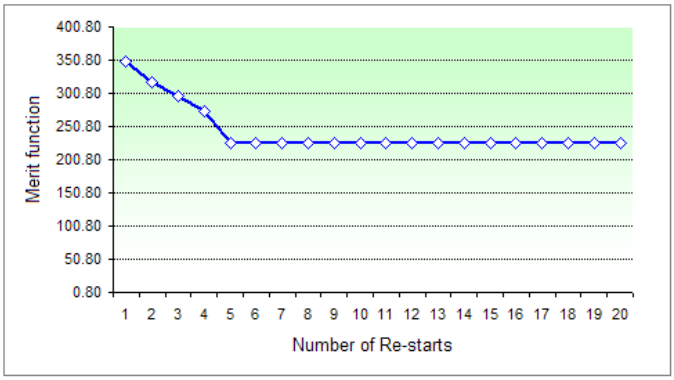

Fig. 16. Behavior of merit function for test function VLMOP3 during the *convergence phase*.

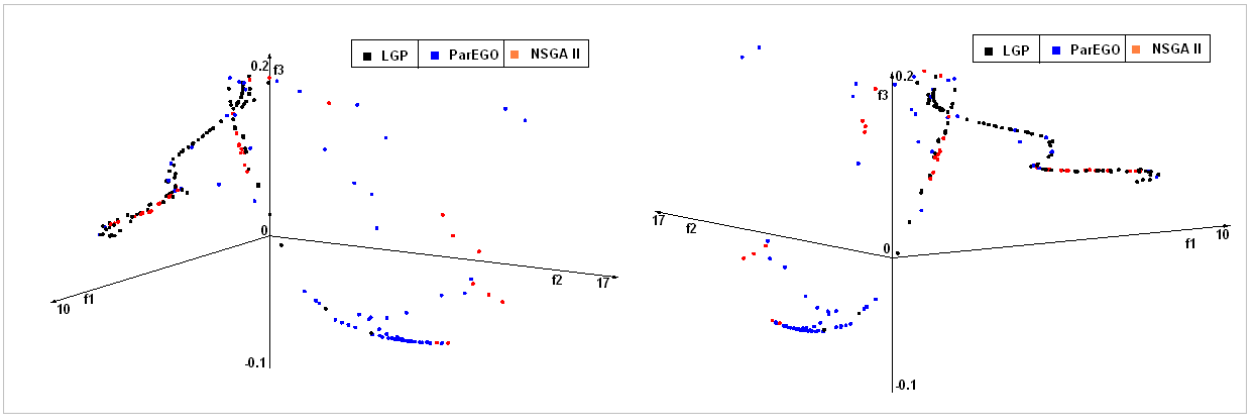

Fig. 14. Pareto front obtained by LGP, ParEGO and NSGA II for the test function VLMOP3.

Test function DTLZ1a

The test function DTLZ1a is a two objective test function and has 6 variables [15]. It is given by:

$$minimize \quad f_1 = \frac{1}{2} x_1 (1+g)$$

$$minimize \quad f_2 = \frac{1}{2}(1-x_1)(1+g)$$

$$g = 100\left[5 + \sum_{i=2}^{6}\left((x_i - 0.5)^2 - \cos\left(2\pi(x_i - 0.5)\right)\right)\right]$$

$x_i \in [0, 1]$, $i=1, \ldots, n$, $n=6$.

The Pareto set for this function consists of all solutions where all by the first decision variables are equal to 0.5 and the first decision variable may take any value between 0 and 1.

For this test function, the value of α for the spreading phase which is adapted by the fuzzy controller is 0.01. The convergence to the Pareto frontier and the distribution of solutions obtained by LGP, ParEGO and NSGA II for the test function DTLZ1a is depicted in Fig. 17. Different sizes of the objective space are illustrated in order to incorporate all solutions obtained by all techniques. It is obvious that LGP assure a very good convergence and distribution for this function. The convergence of the merit function during the 20 re-starts is depicted in Fig. 18. From the results presented in Table VII it can be observed that none of the solutions obtained by LGP are dominated neither by ParEGO or by NSGA II, while solutions obtained by LGP dominate all 100 solutions obtained by ParEGO and NSGA II. 91 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 75 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

Test function DTLZ4a

Test function DTLZ4a has three objective functions and 8 decision variables and is given by:

$$minimize \quad f_1 = (1+g)\cos\left(\frac{x_1^{100}\pi}{2}\right)\cos\left(\frac{x_2^{100}\pi}{2}\right)$$

$$minimize \quad f_2 = (1+g)\cos\left(\frac{x_1^{100}\pi}{2}\right)\sin\left(\frac{x_2^{100}\pi}{2}\right)$$

$$minimize \quad f_3 = (1+g)\sin\left(\frac{x_1^{100}\pi}{2}\right)$$

$$g = \sum_{i=3}^{8}(x_i - 0.5)^2$$

$x_i \in [0, 1]$, $i=1, \ldots, n$, $n=8$.

The Pareto front is 1/8 of the unit sphere centered in origin. The Pareto optimal set consist of all solutions but the first two decision variables are equal to 0.5 and the first two decision variables may take any value between 0 and 1.

For test function DTLZ4a the value of α in the spreading phase which is adapted by the fuzzy controller is 0.2. The distribution of solutions on the Pareto frontier and the convergence to the Pareto frontier for all the three algorithms is depicted in Fig. 19. The distribution on the Pareto frontier obtained for different other values of α is depicted in Fig. 21. With a higher number of nondominated solutions (1,000) LGP assure a better coverage of the Pareto frontier (as evident from Fig. 22). The convergence of the merit function is depicted in Fig. 20.

From Fig. 19 it can be observed that, compared to ParEGO and NSGA II, LGP is assuring a very good convergence. The latter two approaches are not converging very well with the parameters used.

As evident from Table VIII none of the solutions obtained by LGP are dominated neither by ParEGO or by NSGA II while solutions obtained by LGP dominate all 100 solutions obtained by ParEGO and NSGA II. 98 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO while 60 of the solutions obtained by ParEGO are dominated by solutions obtained by NSGA II.

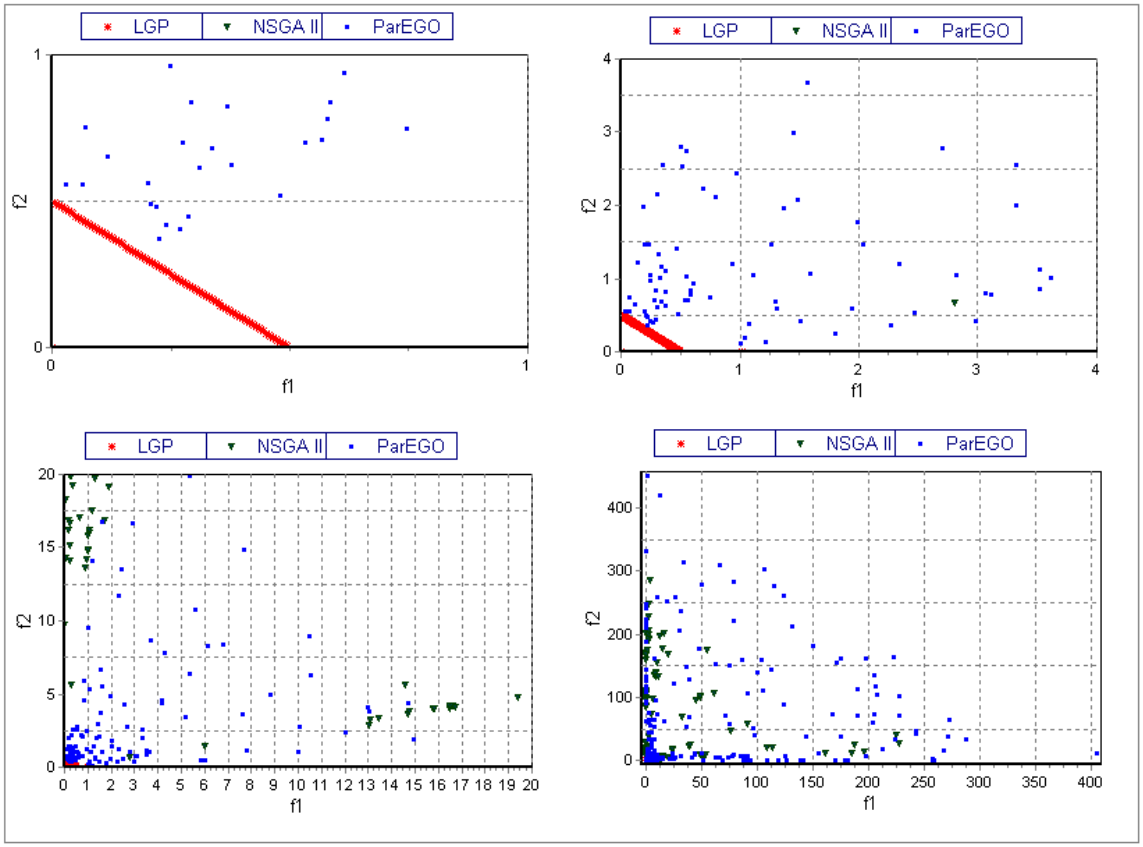

Fig. 17. Distribution of solutions on the Pareto frontier obtained by LGP, ParEGO and NSGA II for test function DTLZ1

.

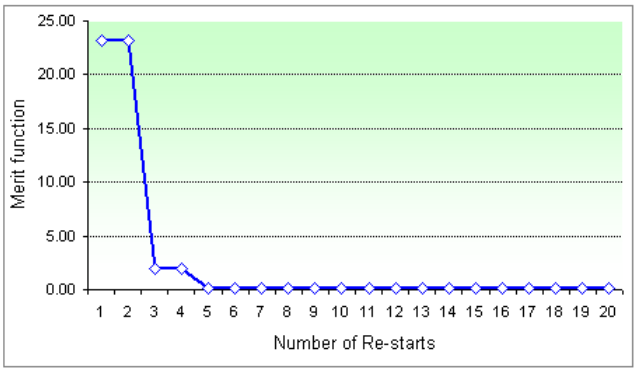

Fig. 18. Behavior of merit function for test function DTLZ1a during the *convergence phase*.

Test function DTLZ7a

This test function has 3 objectives and 8 decision variables and it is given by:

minimize $f_1=x_1$
minimize $f_2=x_2$
minimize $f_3=(1+g)h$

$$g = 1+\frac{9}{6}\sum_{i=3}^{8} x_i$$

$$h = 3-\sum_{i=1}^{2}\left[\frac{f_i}{1+g}\left(1+\sin\left(3\pi f_i\right)\right)\right]$$

$x_i \in [0, 1]$, $i=1, \ldots, n$, $n=8$.

The Pareto front has four discontinuous regions and the Pareto set consists of all solutions where all by the first two decision variables are equal to 0.
The test function DTLZ7a has 4 discontinuous Pareto regions. LGP is able to converge very well and it is able to spread into the all four disconnected Pareto regions from a single starting point. The value of α which is adapted by the fuzzy controller is 0.99, but there is not much difference between different values of α as in the case of DTLZ4a test function. The test function DTLZ7a has 4 discontinuous Pareto regions. LGP is able to converge very well and it is able to spread into the all four disconnected Pareto regions from a single starting point. The value of α used is 1, but there is not much difference between different values of α as in the case of DTLZ4a test function. Fig. 23 depicts the distribution of LGP solutions for two different values of α. As evident from Fig. 24, both ParEGO and NSGA II ar far from the Pareto front in terms of convergence. 44 of the solutions obtained by LGP dominates solutions obtained by ParEGO while 15 solutions obtained by ParEGO dominates solutions obtained by LGP. 91 solutions obtained by NSGA II are dominated by solutions obtained by LGP and 20 solutions obtained by LGP are dominated by solutions obtained by

NSGA II. 39 solutions obtained by ParEGO are dominated by solutions obtained by NSGA II while 33 of the solutions obtained by NSGA II are dominated by solutions obtained by ParEGO (as evident from Table IX). The convergence of the merit function is depicted in Fig. 25.

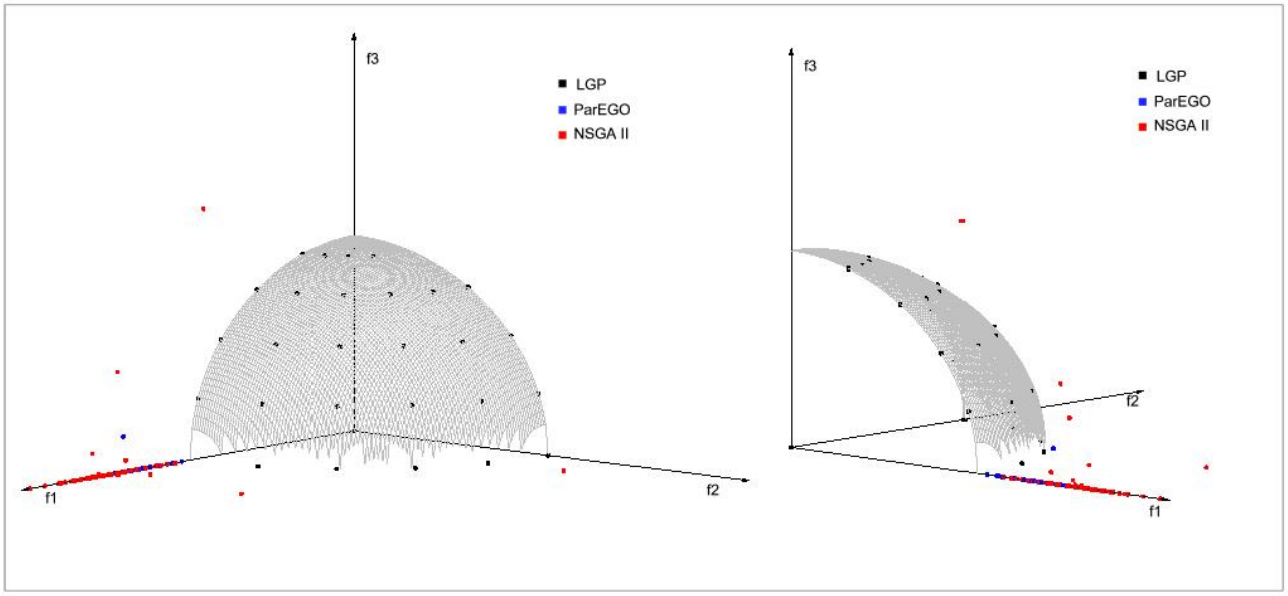

Fig. 19. Convergence to the Pareto frontier and distribution of solutions obtained by LGP, ParEGO and NSGA II on the Pareto frontier for test function DTLZ4a (view from different angles).

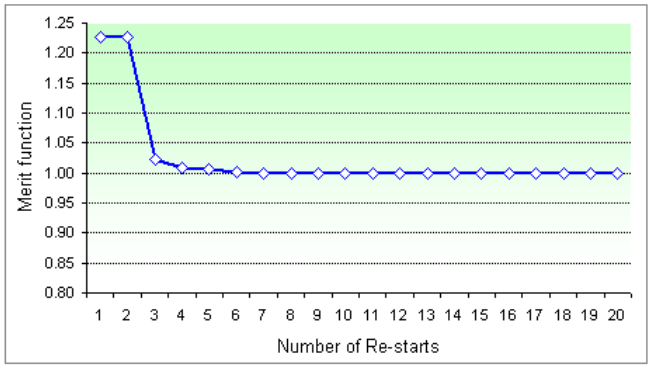

Fig. 20. Behavior of merit function for test function DTLZ4 during the *convergence phase*.

As evident from the graphical representation of the results obtained by all the three techniques, ParEGO and NSGA II and not always providing a very good convergence to the Pareto front. This means that they require a higher number of generations (which, in turn, involves a higher computational time) in order to assure a good convergence to the true Pareto front. LGP converge very fast and require less than 200 mS (milliseconds) to obtain a set of Pareto solutions. Also, the convergence is very good and not all 20 re-starts are required in the convergence phase (for test functions KNO1 and VLMOP3 4 re-starts are enough while for test functions DTLZ1, DTLZ4 and OKA1 5 restarts assure the convergence to the first Pareto optimal solution.

IV. Investigation of LGP performances for optimization of a flow injection system for determining hydroquinone

This problem requires optimizing the determination of hydroquinone by using a flow injection system with amperometric detection. There are three factors that affect the analytical signal:

the carrier solution flow rate, Φ (mL min^{-1}),

the conditioning cell potential, E_a (mV),

the working electrode potential, EL_1 (mV), of the analytical cell.

The experimental conditions taken are:

central values equal to 0.75 mL/min for the flow rate Φ,

−100 mV for the conditioning cell potential, E_a,

175 mV for the working electrode potential, EL_1, with steps of 0.25 mL/min, 50 and 125 mV.

The task is to find experimental conditions for the three factors so that:

- maximize the signal size estimated as the average value (the peak height of the fiagram in μA) of 5 signals obtained under the same experimental conditions
- minimize the relative variability estimated as the coefficient of variation of the signal, measured as a percentage.

It is known that, in general, when increasing the size of an analytical signal, its variability is also increased, thus the two objectives are conflicting. More details about this problem and about the optimization conditions can be found in [29][32]. The authors estimated that the peak height should be below 1.89 μA and the coefficient of variation below 1.5%.

TABLE II. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION KNO1.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	7	59	NSGA II	2	42
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	7	2	ParEGO	100	42	NSGA II	100	59

TABLE III. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION OKA1.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	83	64	ParEGO	0	77	NSGA II	0	59
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	83	59	NSGA II	64	77

TABLE IV. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION OKA2.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	37	41	ParEGO	28	31	NSGA II	0	69
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	28	0	ParEGO	37	69	NSGA II	41	31

TABLE V. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION VLMOP2.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	49	75	ParEGO	6	75	NSGA II	0	37
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	6	0	ParEGO	49	37	NSGA II	75	75

TABLE VI. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION VLMOP3.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	44	91	ParEGO	15	33	NSGA II	20	39
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	15	20	ParEGO	44	39	NSGA II	91	33

TABLE VII. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION DTLZ1A.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	0	75	NSGA II	0	91
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	100	91	NSGA II	100	75

TABLE VIII. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION DTLZ4A.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	0	98	NSGA II	0	60
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	100	60	NSGA II	100	98

TABLE IX. THE DOMINANCE BETWEEN SOLUTIONS OBTAINED BY LGP, PAREGO AND NSGA II FOR TEST FUNCTION DTLZ7A.

Dominate	ParEGO	NSGA II	***Dominate***	LGP	NSGA II	***Dominate***	LGP	ParEGO
LGP	100	100	ParEGO	0	97	NSGA II	0	13
Dominated	ParEGO	NSGA II	***Dominated***	LGP	NSGA II	***Dominated***	LGP	ParEGO
LGP	0	0	ParEGO	100	13	NSGA II	100	97

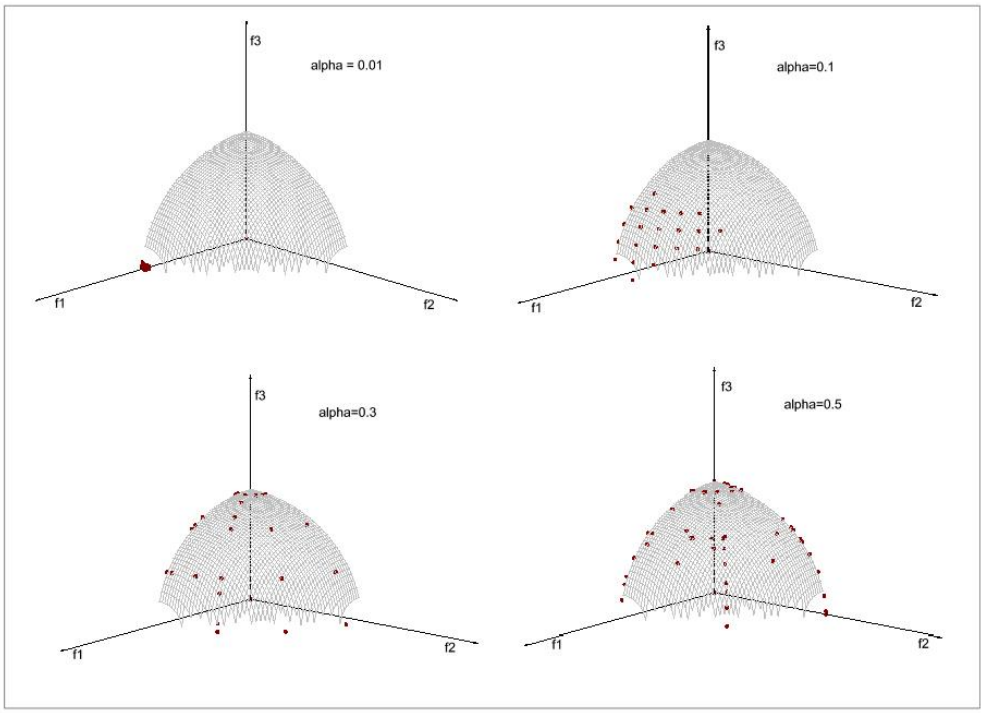

Fig. 21. Distribution of solutions on the Pareto frontier obtained by LGP by considering different values for α in the spreading phase for test function DTLZ4a.

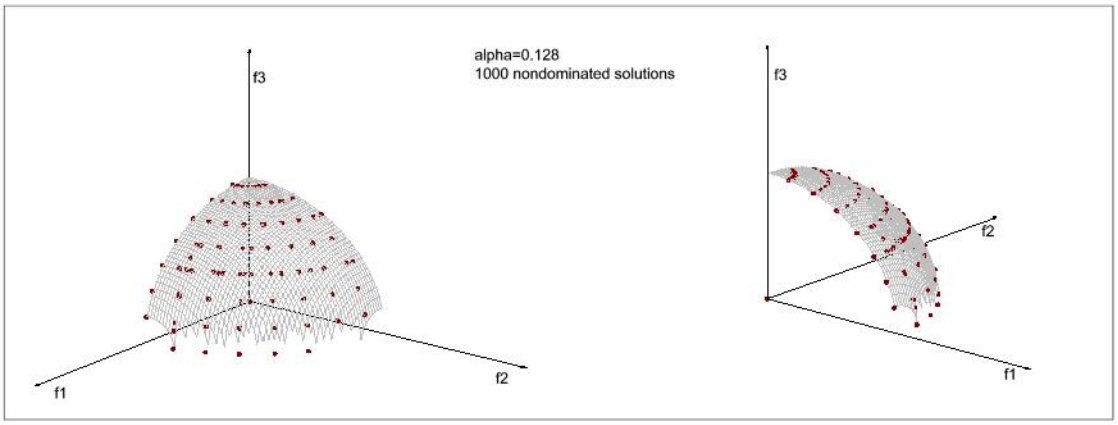

Fig. 22. Distribution of solutions obtained by LGP for DTLZ4 test function considering a set of 1,000 nondominated solutions (view from different angles).

By using the notations:

$\Phi \rightarrow x_1$
$E_a \rightarrow x_2$
$EL_1 \rightarrow x_3$

the optimization problem can be formulated as follows:

maximize the peak height (μA):

$$f_1 = 1.51 + 0.25x_1 - 0.05x_2 + 0.06x_3 - 0.03x_1^2 + 0.02x_2^2 + 0.02x_3^2 + 0.03x_1x_2 + 0.07x_1x_3 - 0.02x_2x_3$$

minimize the Coefficient of variation (%):

$$f_2 = 0.70 - 0.41x_1 - 0.13x_2 - 0.23x_3 + 0.44x_1^2 - 0.07x_2^2 + 0.13x_3^2 + 0.24x_1x_2 + 0.08x_1x_3 - 0.14x_2x_3$$

The mathematical analysis of the models provides the stationary point (0.49, −0.57, −3.04) for the peak height and (0.55, -0.46, 0.47) for the coefficient of variation. The stationary point of the model for the peak height is outside the experimental domain whereas that of the model for the coefficient of variation is inside it. However, both of them are saddle points. We are applying LGP for solving this optimization problem by considering 10 re-starts and 10 iterations per each restart.

Fig. 26 shows the Pareto-optimal front (consisting of 1000 solutions) for the peak height (objective f_1) and the

coefficient of variation (objective f_2): when the peak height increases, the coefficient of variation also increases and vice versa. The value of α adapted by the fuzzy controller in the spreading phase is 0.54.

The corresponding level of factors in the experimental domain is depicted in Fig. 27. In the graphical representation x_1 corresponds to the carrier solution flow rate Φ, x_2 corresponds to the conditioning cell potential E_a and x_3 corresponds to the working electrode potential EL_1 of the analytical cell.

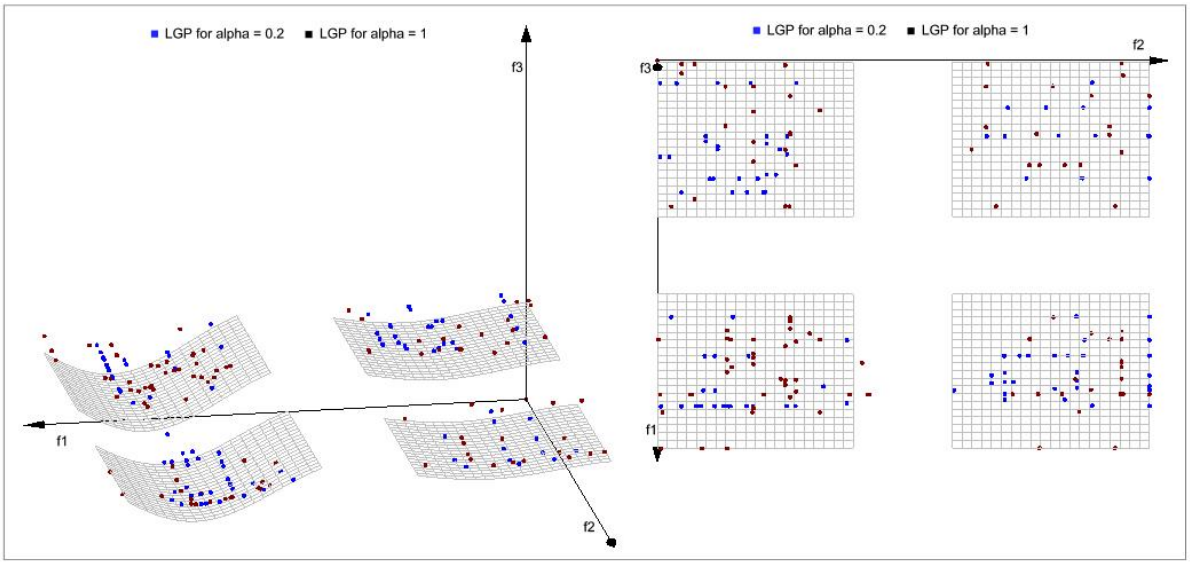

Fig. 23. Distribution of solutions on the Pareto frontier obtained by LGP by considering different values for α in the spreading phase for test function DTLZ7a.

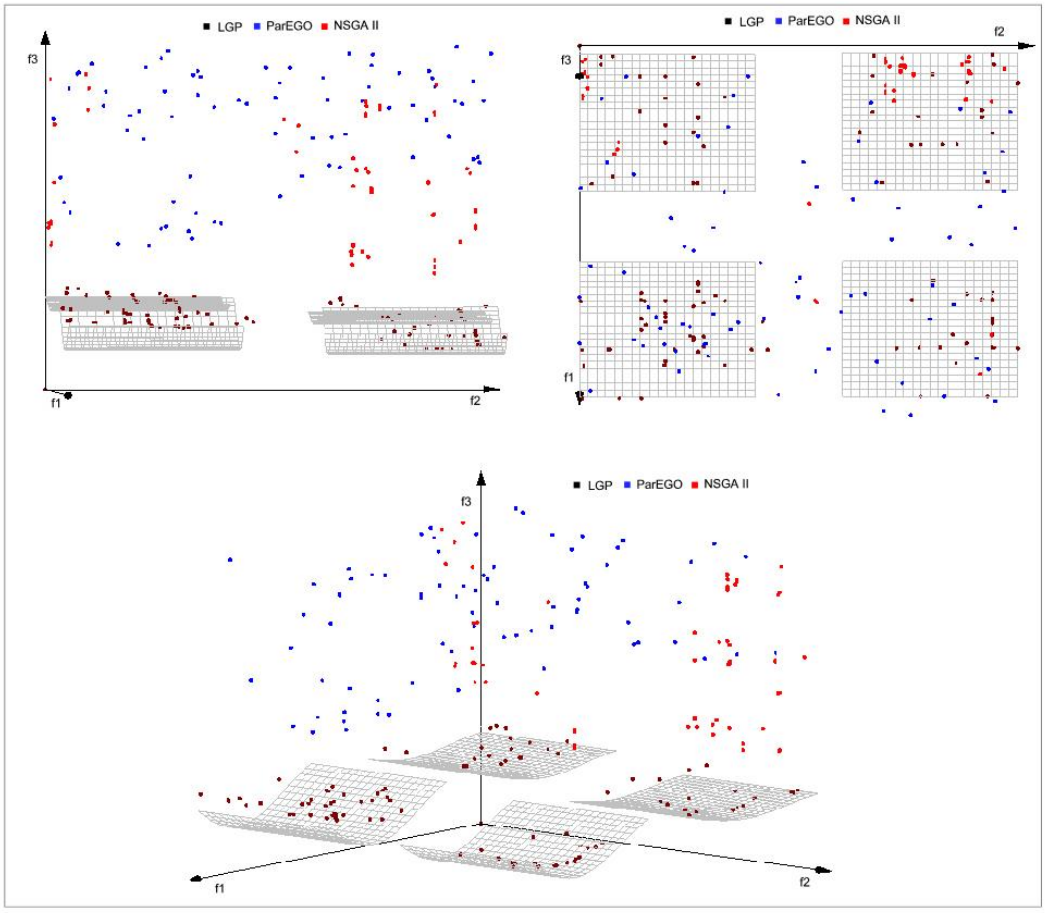

Fig. 24. Convergence to the Pareto frontier and distribution of solutions obtained by LGP, ParEGO and NSGA II on the Pareto frontier for test function DTLZ7a (view from different angles).

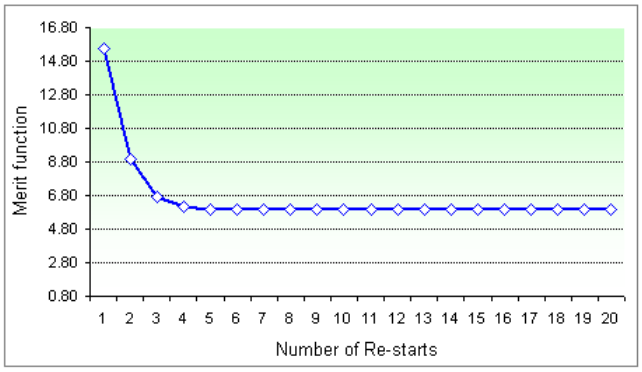

Fig. 25. Behavior of merit function for test function DTLZ7 during the *convergence phase*.

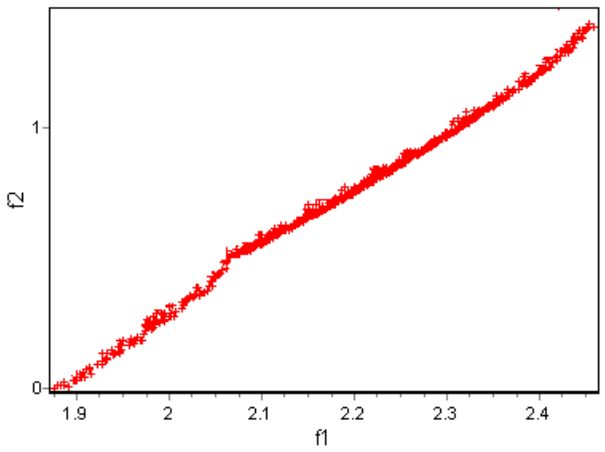

Fig. 26. Pareto optimal front obtained by LGP for the optimization of a flow injection system for determining hydroquinone.

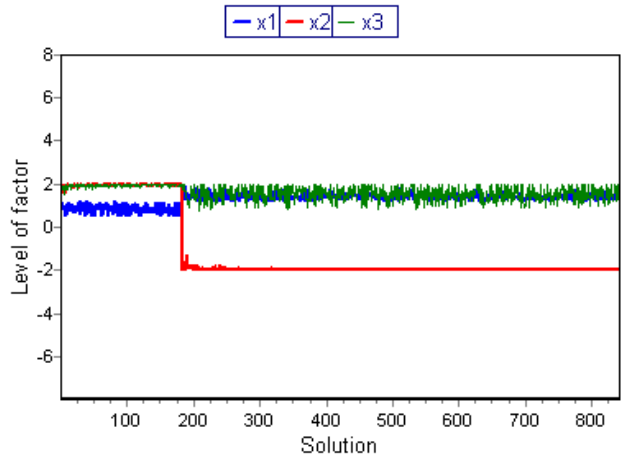

Fig. 27. The level of factors obtained by LGP: x_1 corresponds to Φ, x_2 corresponds to E_a and x_3 corresponds to EL_1.

The results obtained by LGP can the possibility to choose the desired solution based on the expected values for the peak height and the coefficient of variation. Some authors consider that peak heights below 1.5 μA are unacceptable and values above 1.8 μA are good. Also, they look for values for coefficient of variation below 1.5% [29]. Desirability values are defined to increase (or decrease) linearly between the two limits. Given all the required conditions for solving this problem, we generated an approximation of the Pareto front and set so that the user can select depending on further preferences.

V. LGP FOR THE OPTIMIZATION OF THE FORMULATION OF AN ORAL SOLUTION

The application studied in this section comes from pharmaceutics and it is described in detail in [17] and [29]. The problem refers to the formulation of a solution of slightly soluble drug mainly depending on the percentage of surfactant (polysorbate 80), propylene glycol (%) and invert sugar medium (mL). The central values of the factors that define the experimental domain are 4.0% of polysorbate 80, 20% of propylene glycol and 55 mL of sucrose invert medium with steps 0.3%, 3% and 6 mL respectively. With a central composite design (spherical and with α=1.68), two response surfaces (quadratic models) are fitted for the turbidity (ppm) and cloud point (°C) of the resulting solutions.

For fitting the model for the cloud point two experimental points had to be removed with abnormal residuals that produce the regression model to be non significant. These points were, in codified variables, (0, −1.68, 0) and (0, 1.68, 0).

Based on some experimental results, the authors of [29] reduced the size of variation of the second factor (propylene glycol) to [−1.3, 1.3].

The goal of the fitting of the response surfaces is to find experimental conditions to reduce the turbidity and to increase the cloud point.

By using the notations:

polysorbate 80→x_1
propylene glycol →x_2
invert sugar medium →x_3

The optimization problem can be formulated as follows:

maximize the turbidity:

$$f_1 = 3.04 - 0.59x_1 - 0.15x_2 + 0.12x_3 + 0.24x_1^2 + 0.49x_2^2 - 0.06x_3^2 + 0.05x_1x_2 - 0.40x_1x_3 - 0.65x_2x_3$$

minimize the cloud point:

$$f_2 = 72.22 + 0.84x_1 + 3.03x_2 - 5.58x_3 + 0.63x_1^2 + 1.03x_2^2 - 0.68x_3^2 - 0.15x_1x_2 + 0.92x_1x_3 - 0.72x_2x_3$$

The analysis of the quadratic models (the standard mathematical methodology) provides the stationary or

critical points (points where gradient vector is equal to the null vector) which are (0.82, −0.23, −0.51) for turbidity and (0.81, −2.24, −2.37) for the cloud point. The stationary point for turbidity is inside the experimental domain and that of the cloud point is outside it. In any case, both of them are saddle points (nor maximum nor minimum). As mentioned in [29], according to the generalization (theorem 2, page 16 in [1]) of the Weierstrass Extreme Value Theorem (e.g. theorem 3.9, page 57 in [30]), both continuous functions in the experimental domain (which is a compact set) attain its maximum and minimum values at points within the compact region; therefore in this case in the boundary of the experimental domain. This theorem, although guaranteeing that there are extreme points in the experimental domain for both models, does not provide any indication about the experimental conditions (the level of the factors) to obtain such extreme points, that means that we do not have any idea about how to move the experimental conditions for minimizing turbidity and maximizing cloud point [29].

From the mathematical experiments performed in [17] and [29] it is obvious that there are no experimental conditions for the three factors that simultaneously provide the minimum value for turbidity and the maximum value for cloud point. The pairs of values which can be obtained is in the Pareto-optimal front estimated for these two functions depicted in Fig. 24 (the value of α in the spreading phase adapted by the fuzzy controller is 0.982).

In the Pareto-optimal front, turbidity ranges from 1.11 to 4.38 ppm and the cloud point from 64.7 to 83.8 °C, but it is clearly observed in Fig. 28 how an increase in cloud point is linked to an increase in turbidity and if we want to decrease turbidity, cloud point is also decreased.

The solutions obtained by LGP are depicted in Fig. 29. Knowing the experimental conditions (inside the experimental domain) that estimate values optimal in one or the other response the user can choose according to the needs (for instance, solutions for which the turbidity is above some limits and the cloud point is below some limits and vice versa).

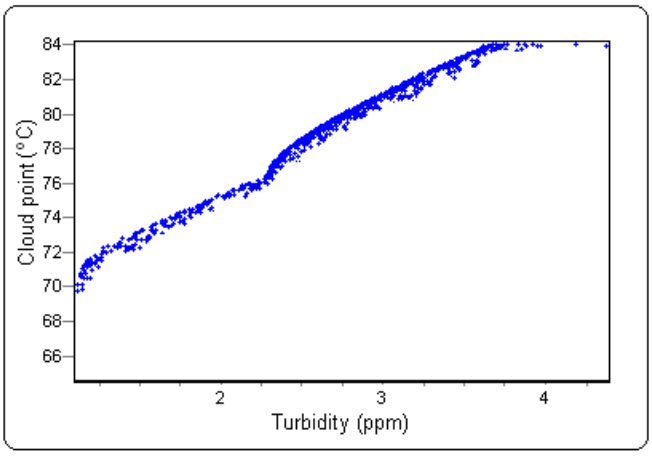

Fig. 28. Pareto front obtained by LGP for turbidity and cloud point conditions.

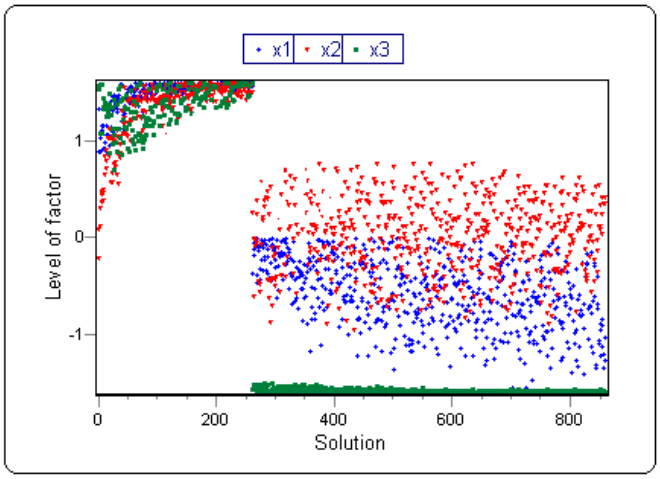

Fig. 29. Solutions (in the variable space) obtained by LGP for the optimization of an oral solution application.

VI. Conclusions

The paper proposes a new approach for multiobjective optimization which uses an aggregation of objectives and transforms the MOP into a SOP. A line search based technique is applied in order to obtain one solution. Starting from this solution a simplified version of the initial line search is used in order to generate solutions with a well distribution on the Pareto frontier. Numerical experiments performed show that the proposed approach is able to converge very fast and provide a very good distribution (even for discontinuous Pareto frontier) while compared with state of the art population based metaheuristics such as ParEGO and NSGA II.

Compared to NSGA II and ParEGO, LGP has only few parameters to adjust. It is computationally inexpensive, taking less than 200 milliseconds to generate a set of nondominated solutions well distributed on the Pareto frontier.

We also considered two practical problems for which we have applied LGP. The first problem refers to the optimization of a flow injection system for determining hydroquinone. LGP is able to generate a very good approximation of the Pareto front and provide results which clearly fulfill all the problem requirements. The second application is from the pharmaceutical design and requires the optimization of an oral solution. LGP is obtaining a good Pareto front offering to the user the possibility to select from a wide range of solutions.

The only inconvenience is that LGP involves first partial derivatives which makes it be restricted to a class of problems which are continuous twice differentiable. But almost all practical engineering design problems are continuous differentiable.

One of the further work ideas is to extend LGP to deal with constraint multiobjective optimization problems.

REFERENCES

[1] W. Cheney, Analysis for Applied Mathematics, Graduate texts in Mathematics 208, Springer-Verlag New York, Inc., 2001

[2] Das, I. and Dennis, J., A Closer Look at Drawbacks of Minimizing Weighted Sums of Objective for Pareto Set Generation in Multicriteria Optimization Problems, Structural Optimization, 14, pp. 63–69, 1997.

[3] Das, I., An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection. In: Short Paper Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization 2, pp. 411–413, 1999.

[4] Das I, Dennis JE, Normal Boundary Intersection: A new method for generating the Pareto surface in multicriteria optimization problems. SIAM Journal of Optimization, 8(3), pp. 631-657, 1998.

[5] Das I, On characterizing the `knee' of the Pareto curve based on Normal-Boundary Intersection. Structural Optimization, 18(2/3), pp. 107-115, 1999.

[6] Deb, K., Pratap. A, Agarwal, S., and Meyarivan, T., A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transaction on Evolutionary Computation, 6(2), 181-197, 2002.

[7] Deb, K., Thiele, L., Laumanns, M. and Zitzler, E., Scalable multi-objective optimization test problems. Proceedings of the Congress on Evolutionary Computation (CEC-2002). (Honolulu, USA). pp. 825-830, 2002.

[8] Deb, K., Multiobjective Optimization Using Evolutionary Algorithms, Series in Systems and Optimization, John Wiley and Sons, Chichester, England, 2001.

[9] Chen, W., Wiecek, M. M., and Zhang, J., Quality Utility – A Compromise Programming Approach to Robust Design, Journal of Mechanical Design, 121, pp. 179–187, 1999

[10] Ehrgott, M., and Gandibleux, X., Multiobjective Combinatorial Optimization, Multiple-Criteria Optimization: State of the Art Annotated Bibliographic Surveys, Kluwer Academic Publishers, Boston, Massachusetts, 2002.

[11] Ehrgott, M., and Wiecek, MM., Mutiobjective Programming, In Multiple Criteria Decision Analisys: State of the art surveys, J. Figueira et al. (Eds.), International Series in Operations Research & Management Science, Volume 78, Springer New York, 2005.

[12] Gould, N., An introduction to algorithms for continuous optimization, Oxford University Computing Laboratory Notes, 2006.

[13] Huband, S., Hingston, P., Barone, L., and While, L., A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit, IEEE Transactions on Evolutionary Computation, 10(5), pp. 477-506, 2006.

[14] J. Jahn, Vector Optimization: Theory, Applications, and Extensions, Springer-Verlag, Berlin, 2004.

[15] Knowles, J., ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation, 10 (1), 50-66, 2006.

[16] J.D. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, TIK Report No. 214, Institute fur Technische Informatik und Kommunikationsnetze ETH Zurich, 2005.

[17] G. Lewis, D. Mathieu, R. Phan-Tan-Luu, Pharmaceutical Experimental Design, Marcel Decker, New York, 2000.

[18] Messac, A. and Mattson, C. A., Generating Well-Distributed Sets of Pareto Points for Engineering Design using Physical Programming, Optimization and Engineering, 3, pp. 431–450, 2002.

[19] Messac A, Physical programming: Effective optimization for computational design, AIAA Journal, vol. 34, no. 1, pp. 149–158, 1996.

[20] Messac A, Melachrinoudis E, Sukam CP, Mathematical and Pragmatic Perspectives of Physical Programming, AIAA Journal, 39 (5), pp. 885-893, 2001.

[21] Messac A, Mattson C A (2002), Generating Well-Distributed Sets of Pareto Points for Engineering Design using Physical Programming, Optimization and Engineering, Kluwer Publishers, 3(4), pp. 431-450, 2002.

[22] Messac A, Ismail-Yahaya, A Multiobjective Robust Design using Physical Programming, Structural and Multidisciplinary Optimization, Journal of the International Society of Structural and Multidisciplinary Optimization (ISSMO), Springer, 23(5), pp. 357-371, 2002.

[23] Messac A, Batayneh WM, Ismail-Yahaya A, Production Planning Optimization with Physical Programming, Engineering Optimization, Taylor and Francis Publisher, 34(4), pp. 323-340, 2002.

[24] Messac A, Martinez M, Simpson T, Effective Product Family Design Using Physical Programming, Engineering Optimization, 34(3), pp. 245-261, 2002.

[25] Messac, A., Ismail-Yahaya, A., and Mattson, C. A., The Normalized Normal Constraint Method for Generating the Pareto Frontier, Structural and Multidisciplinary Optimization, 25(2), pp. 86–98, 2003.

[26] Messac A, Mattson CA, Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier, AIAA Journal, 42(10), pp. 2101-2111, 2004.

[27] K.M. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Norwell, 1999.

[28] Okabe, T., Jin, Y., Olhofer, M., and Sendhoff, B., On test functions for evolutionary Multiobjective optimization, In Proceedings of 8th Conference on Parallel Problems Solving from Nature, Springer Verlag Germany, pp. 792-802, 2004.

[29] M.C. Ortiz, L. Sarabia, A. Herrero, M.S. Sánchez, Vectorial optimization as a methodogical alternative to desirability function, Chemometrics and Intelligent Laboratory Systems, 83, pp. 157–168, 2006.

[30] M.H. Protter, Basic Elements of Real Analysis, Undergraduate Texts in Mathematics, Springer-Verlag New York, Inc., 1998.

[31] Resende MGC,de Sousa JP (Eds), Metaheuristics: Computer Decision-Making, Kluwer Academic Publishers, The Nederlands, 2004.

[32] M. E. Rueda, L. A. Sarabia, A. Herrero and M. C. Ortiz , Optimisation of a flow injection system with electrochemical detection using the desirability function: Application to the determination of hydroquinone in cosmetics, Analytica Chimica Acta 479, pp. 173-184, 2003.

[33] S. Ruzika, and M. M. Wiecek, Approximation Methods in Multiobjective Programming , Journal of Optimization Theory and Applications: Vol. 126, No. 3, pp. 473–501, 2005

[34] S. Sayın, Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming, Math. Program., Ser. A, 87, pp. 543–560, 2000.

[35] Veldhuizen, D.A.V. Lamont, G.B., Multiobjective Evolutionary Algorithm Test Suites, In Proceedings of ACM Symposium on Applied Computing, J. Caroll et al. (Eds.), San Antonio, Texas, pp. 351-357, 1999.

[36] C. Grosan, A. Abraham, A new approach for solving nonlinear equations systems, IEEE Transactions on Systems, Man and Cybernetics Part A (in press).