15th Triennial World Congress, Barcelona, Spain

EVOLUTIONARY PROGRAMMING BASED ON UNIFORM DESIGN WITH APPLICATION TO MULTIOBJECTIVE OPTIMIZATION [1]

Jihui Zhang * **Marc Bourcerie** * **Jean-Louis Ferrier** *

* *LISA-ISTIA, Université d'Angers, France*

Abstract: Pareto-optimality is one of the important methodes to multiobjective optimization problems. It is desirable to find as much as possible Pareto-optimal solutions, and it is also highly expected to find the ones scattered uniformly over the Pareto frontier such that a variety of compromise solutions can be provided to the decision maker. For this purpose, an evolutionary programming algorithm, called evolutionary programming based on uniform design (UDEP), is proposed to in this paper. Uniform design technique is used to define some fitness functions which can guide the search evenly toward the Pareto frontier. In order to overcome premature and provide as much as possible candidate solution evenly scattered in the whole search space, uniform design technique, variable region search, as well as nicheo technique are used. Uniform design makes it possible to explore the search space evenly, while variable region search and nicheo technique help to keep diversity of the population. Their combination improves the search ability of EP significantly. Many numerical experimental results show the usefulness of the proposed method.

Keywords: evolutionary programming, multiobjective optimization, uniform design, Pareto-optimality, variable region search

1. INTRODUCTION

Multiobjective optimization is with no doubt a very important research topic both for scientists and engineers, not only because of the multiobjective nature of most real-world problems, but also because there are still many open questions in this area. In Operations Research, more than 20 techniques have been developed to deal with functions that have multiple objectives, and many approaches have been suggested, going all the way from a naive combination of objectives into a single one to the use of game theory to coordinate the relative importance of each objective. However, the fuzziness of this area lies on the fact that there is no acceptable definition of "optimum" as in single-objective optimization, and therefore is difficult to even compare results of different methods. Multiobjective optimization problems tend to be characterized by a family of alternatives which must be considered equivalent in the absence of information concerning the relevance of each objective to the others. Multiple solutions arise in even the simplest non-trivial case of two competing unimodal convex objectives. As the number of competing objectives increases and less well-behaved objectives are considered, the problem rapidly becomes increasing complex. During the past several decades, with the increasing research on evolutionary algorithms and their successful application to various problems, it was recongnized that evolutionary algorithms were possibly well-suited to multiobjective optimization. Multiple individuals can search for multiple solution in parallel, eventually taking advantage of any similarities available in the solution set. The ability to

[1] supported by Région des Pays de La Loire, France, Under #00-15144-0

handle complex problems, involving features such as discontinuities, multimodality, disjoint feasible spaces, reinforce the potential effectiveness of evolutionary algorithms in multiobjective optimization. Many research results about evolutionary algorithms with application to multiobjective optimization have been published. Evolutionary algorithms have their own shortcomings too. For example, premature convergence (to a local optimum rather than a global optimum), search speed as well as numerical accuracy are some common problem in almost all the evolutionary algorithms. The key is to keep a proper balance between "exploration" and "exploitation". This is the motivation of this paper.

For multiobjective optimization problems, the "optimal solution" is not unique in most cases, therefore, it is desirable to find as much as possible alternatrive solutions, and it is also highly expected to find the ones scattered uniformly over the Pareto frontier such that a variety of compromise solutions can be provided to the decision maker. This is another motivation of this paper.

2. LITERATURE REVIEW

2.1 *Concepts on Multiobjective Optimization*

The following multiobjective optimization problem is considered:

$$\underset{X\in\Omega}{\text{Minimize}} \quad (f_1(X), f_2(X), \cdots, f_m(X)) \tag{1}$$

where $X = (x_1, x_2, \cdots, x_n) \in \Re^n$, $\Omega \subset \Re^n$ is the feasible solution space defined by the constraints. Usually, it is defined as follows:

$$g_i(X) \leq 0, \quad i = 1, \cdots, k \tag{2}$$

$$h_j(X) = 0, \quad j = 1, \cdots, \ell \tag{3}$$

$$L_i \leq x_i \leq U_i, i = 1, \cdots, n \tag{4}$$

The notion of *Pareto-optimality* is one of the important approaches to multiobjective optimization. The set of possible Pareto-optimal solutions constitutes a *Pareto frontier* in the objective space.

2.2 *Methods For Multiobjective Optimization*

It includes *Weighted sum approach*, *Goal Programming*, *Goal Attainment*, *The ε-constraint Method*, *Non-aggregating Non-Pareto Approaches*, *Pareto-based Approaches*, etc. Pareto-based approaches is a very important one.

2.3 *Uniform Design*

Experimental design method is a sophisticated branch of statistics. *uniform design* is one important experimental design technique and it has been used in many real application. It was proposed by K.T. Fang and Y. Wang (Hicks, 1981; Fang and Wang, 1994) in 1981 and it also was developed further by other researchers in recent years. The main objective of uniform design is to sample a small set of points from a given set of points such that the sampled points are uniformly scattered.

Suppose there are n factors and q levels per factor. When n and q are given, the uniform design selects q combinations out of q^n possible combinations, such that these q combinations are scattered uniformly over the space of all possible combinations. The selected q combinations are expressed in terms of a *uniform array* $U(n, q) = (U_{i,j})_{q\times n}$, where $U_{i,j}$ is the level of the jth factor in the ith combination.

Uniform array can be constructed as follows. Consider a unit hypercube over an n-dimensional space. It can be denoted, by the set of points in it, as

$$\mathcal{C} = \{(c_1, \cdots, c_n) | 0 \leq c_i \leq 1 \; i = 1, \cdots, n\} \tag{5}$$

Consider any point in $\mathcal{C}$, say $\mathbf{r} = (r_1, r_2, \cdots, r_n)$. A hyper-rectangle is formed between $\mathbf{0}$ and $\mathbf{r}$, and it can be denoted by

$$\mathcal{C}(\mathbf{r}) = \{(c_1, \cdots, c_n) | 0 \leq c_i \leq r_i, \; i = 1, \cdots, n\} \tag{6}$$

A sample of q points are selected such that they are scattered uniformly in the hypercube. Suppose $q(\mathbf{r})$ of these points are in the hyper-rectangle $\mathcal{C}(\mathbf{r})$, then the fraction of points in the hyper-rectangle is $q(\mathbf{r})/q$. The volume of this hyper-rectangle is $\prod_{i=1}^{n} r_i$. The objective of the uniform design is to determine q points such that the following *discrepancy* is minimized:

$$\underset{\mathbf{r}\in\mathcal{C}}{sup} \left| \frac{q(\mathbf{r})}{q} - \prod_{i=1}^{n} r_i \right|. \tag{7}$$

Then these q points in the unit hypercube is mapped to the space with n factors and q levels. If q is a prime and $q > n$, it was proved that $U_{i,j}$ is given by

$$U_{i,j} = (i\sigma^{j-1} \;\; mod \;\; q) + 1, \tag{8}$$

where σ is a parameter (Hicks, 1981).

3. EVOLUTIONARY PROGRAMMING BASED ON UNIFORM DESIGN (UDEP) FOR MULTIOBJECTIVE OPTIMIZATION

At first, we don't consider inequality (or equality) constraints.

3.1 *Evaluation functions*

The value of different objective functions may have different order of magnitude, so it is necessary to normalize the objective functions as follows:

$$h_i(X) = \frac{f_i(X)}{\max_{y\in\psi}\{|f_i(y)|\}} \text{ for } i = 1, 2, \cdots, m, \tag{9}$$

where ψ is a of candidate solutions in the current population. Let D be a prime larger than m. Using uniform design, D evaluation functions can be constructed as follows to guide the search process.

$$\text{fitness}_i(X) = w_{i,1}h_1(X) + \cdots + w_{i,m}h_m(X), \quad \text{for } i = 1, 2, \cdots, D. \tag{10}$$

where,

$$w_{i,j} = \frac{U_{i,j}}{U_{i,1} + U_{i,2} + \cdots + U_{i,m}}, \quad \text{for } i = 1, 2, \cdots, D; j = 1, \cdots, m. \tag{11}$$

and $U_{i,j}$ is the entry element of uniform array $U(m, D) = (U_{i,j})_{D\times m}$.

3.2 *Generation of the initial population*

Uniform design technique is used to generate the initial population such that they are uniformly distributed over the search space. If the solution space was large, it is necessary to divide the search space into some smaller subspaces if the search is very large.

Let $[\mathbf{l}, \mathbf{u}] = [(\mathbf{l}(1), \mathbf{u}(1)); \cdots; (\mathbf{l}(n), \mathbf{u}(n))]$ denote the whole search space. The following method can be used to divide $[\mathbf{l}, \mathbf{u}]$ into s disjoint subspaces, where s can take the value of $2, 2^2$, and 2^3 etc.

Algorithm 1. Search space division

Step 1: Let $\mathbf{l}' = \mathbf{l}$ and $\mathbf{u}' = \mathbf{u}$. Repeat the following computation $\log_2 s$ times: select the rth dimension such that

$$\mathbf{u}'(r) - \mathbf{l}'(r) = \max_{1\le i\le n}\{\mathbf{l}'(i) - \mathbf{u}'(i)\},$$

and then let

$$\mathbf{u}'(r) \Leftarrow \frac{\mathbf{u}'(r) + \mathbf{l}'(r)}{2}.$$

Step 2: Compute $\Delta_i = \mathbf{u}'(i) - \mathbf{l}'(i)$ and $N_i = \frac{\mathbf{u}'(i)-\mathbf{l}'(i)}{\Delta_i}$ for $i = 1, 2, \cdots, n$. Then compute the subspace $[\mathbf{l}^k, \mathbf{u}^k]$ for all $1 \le j_i \le N_i$ and $1 \le i \le n$ as follows:

$$\begin{cases} \mathbf{l}^k = \mathbf{l} + \big((j_1 - 1)\Delta_1, (j_2 - 1)\Delta_2, \cdots, (j_n - 1)\Delta_n\big) \\ \mathbf{u}^k = \mathbf{l} + (j_1\Delta_1, j_2\Delta_2, \cdots, j_n\Delta_n) \end{cases}$$

where $k = \sum_{i=1}^{n-1}(j_i - 1)\prod_{\ell=i+1}^{n} N_\ell + j_n$.

After the solution space is divided into s subspaces, Q_0 points are selected from each subspace, and totaly sQ_0 (here Q_0 is a prime) points are sampled and they compose the initial population. Consider any subspace, say the kth subspace, and it is denoted by

$$[\mathbf{l}^k, \mathbf{u}^k] = [(\mathbf{l}^k(1), \cdots, \mathbf{l}^k(n)), (\mathbf{u}^k(1), \cdots, \mathbf{u}^k(n))].$$

in this subspace, the domain $[\mathbf{l}^k(i), \mathbf{u}^k(i)]$ of x_i is quantized into Q_0 levels $\alpha^k_{i,1}, \alpha^k_{i,2}, \cdots, \alpha^k_{i,Q_0}$, where $\alpha^k_{i,j}$ is given by

$$\alpha^k_{i,j} = \mathbf{l}^k(i) + (j - 1)\frac{\mathbf{u}^k(i) - \mathbf{l}^k(i)}{Q_0 - 1} \quad (i = 1, 2, \cdots, n; j = 1, 2, \cdots, Q_0) \tag{12}$$

After this quantization, there are Q_0^n points for sampling in this subspace. The following Q_0 points can be selected, using uniform array $U(n, Q_0)$, from them for testing:

$$\begin{cases} (\alpha^k_{1,U_{1,1}}, \alpha^k_{2,U_{1,2}}, \cdots, \alpha^k_{n,U_{1,n}}), \\ (\alpha^k_{1,U_{2,1}}, \alpha^k_{2,U_{2,2}}, \cdots, \alpha^k_{n,U_{2,n}}), \\ \vdots \\ (\alpha^k_{1,U_{Q_0,1}}, \alpha^k_{2,U_{Q_0,2}}, \cdots, \alpha^k_{n,U_{Q_0,n}}). \end{cases} \tag{13}$$

In a same manner, Q_0 points are selected from each of the other subspaces, then totaly sQ_0 points are selected. These sQ_0 points compose the initial population.

3.3 *Selection and Mutation*

Each one of the functions in (10) can be used to evaluate the individuals in the current population and select the best $\lfloor N/D \rfloor$ individuals as offsprings. Then $D\lfloor N/D \rfloor$ offsprings are generated. If $D\lfloor N/D \rfloor < N$, then other $N - D\lfloor N/D \rfloor < N$ offsprings can be generated randomly in the search space. The selection and mutation process are directed by the following rules:

Rule 1:
if $fitness_i(X_i(k)) < fitness_i(X_i(k-1))$ **then**
$X_i(k)' = X_i(k)$;
$dir^j_i(k) = sign(x^j_i(k) - x^j_i(k-1))$ and
$age(i) = 1$;
else if $fitness_i(X_i(k)) > fitness_i(X_i(k-1))$ **then**
A local search procedure (as discussed later) is executed, and a middle solution, X^{mid}, is obtained.
if $fitness_i(X^{mid}) < fitness_i(X_i(k-1))$ **then**
$X_i(k)' = X^{mid}$
$dir^j_i(k) = sign(x^{mid}_i - x^j_i(k-1))$ and
$age(i) = 1$;
else
$age(i) := age(i) + 1$.
end if
else
$age(i) := age(i) + 1$.
end if

where, $i = 1, 2, \cdots, N; j = 1, 2, \cdots, n$ for each i. $sign(\cdot)$ denotes the symbolic function.

Rule 2:
if $age(i) = 1$ **then**
$\sigma(i) = rem(k_1 \cdot fitness_i(X_i(k)), \gamma d)$ and
$x^j_i(k)' = x^j_i(k) + dir^j_i \cdot |N(0, \sigma(i))|$;
else

$\sigma(i) = rem(k_2 \cdot fitness_i(X_i(k)) \cdot age(i), \gamma d)$ and
$x_i^j(k)' = x_i^j(k) + N(0, \sigma(i))$.
end if

where $rem(\cdot,\cdot)$ is the remainder function. $\gamma \in (0,1)$, and d is the length of the corresponding dimension in the corresponding subspace. $k_1 > 0$ and $k_2 > 0$ are control parameters. Usually, k_1 takes a larger value, and k_2 takes a smaller value. $i = 1, 2, \cdots, N; j = 1, 2, \cdots, n$ for each i.

3.4 *Local Search*

Consider an arbitrary, say the kth, generation of solutions. Suppose one candidate solution, say the ith ($i \in \{1, 2, \cdots, N\}$) one, is worse than its predecessor, i.e. it satisfies that $fitness_i(X_i(k)) > fitness_i(X_i(k-1))$, then a local search procedure is executed in order to improve it.

Let

$$\underline{v}(j) = min\big(X_i^j(k), X_i^j(k-1)\big), \qquad (14)$$

$$\overline{v}(j) = max\big(X_i^j(k), X_i^j(k-1)\big), \qquad (15)$$

$$(j = 1, 2, \cdots, n).$$

Then $\underline{v}$ and $\overline{v}$ compose an n-dimensional search space (if $\exists$ some $j \in \{1, 2, \cdots, n\}$ such that $\underline{v}(j) = \overline{v}(j)$, then let $\overline{v}(j) := \overline{v}(j) + \epsilon$, where $\epsilon > 0$ is a small positive real number). A temporary population with size Q_1 (where Q_1 is a prime) is generated using the procedure discussed in section 3.2, then let X^{mid} be the best one in this temporary population.

3.5 *Keeping Population Diversity*

Let $r > 0$ be a given small real number (e.g. 0.5), the following procedure will be executed after the generation of each offspring population.

```
for i = 2 : N
  for j = 1 : i − 1
    while ||X_i − X_j|| < r
      for k = 1 : n
        X(j, k) := X(j, k) + λ ∗ N(0, 1);
      end
    end
  end
end
```

where λ is a small positive real number (e.g. 0.05), $\mathcal{N}(0,1)$ is a standard Normal random variable.

In addition, some part (e.g. 20% of the population) of the candidate solutions are generated using variable region search, i.e., they are generated uniformly from a real subspace of $[L, R]$.

3.6 *Non-dominated Solutions*

The final step in each generation is to record the non-dominated solutions. Suppose the non-dominated solutions found before generation i is $Non\_dom(i-1)$, then it is updated in the ith generation as follows: all candidate solutions in set $Non\_dom(i-1)$ as well as that in the new generation are evaluated according to the definition of Pareto-optimal condition, then $Non\_dom(i)$ is composed of the candidate solutions with no dominance relation between each pair of them.

3.7 *Termination Condition*

In this paper, a fixed generation number is used as the termination condition of the procedure.

3.8 *Handling of Inequality/Equality Constraints*

In this paper, the inequality (quality) constraints (2) and (3) are treated using the traditional penalty method. For this purpose, the following penalty functions are defined.

$$F_i(X) = f_i(X) + \sum_{j=1}^{k} p_j\{[g_j(X)]_+\}^\alpha + \sum_{j=1}^{\ell} q_j |h_j(X)|^\alpha, \quad \text{for } i = 1, \cdots, m \qquad (16)$$

where p_i ($i = 1, \cdots, k$), q_j ($j = 1, \cdots, \ell$) and α are positive penalty factors. The $F_i(\ )$ is used in instead of $f_i(\ )$ in the previous subsections.

4. NUMERICAL EXPERIMENTS

Problem 1 (Y.B. Yun and Tanino, 2001)

$$\min_{x_1,x_2} \ (f_1(x_1,x_2), f_2(x_1,x_2)) = (x_1, x_2)$$
$$s.t.$$
$$(x_1-2)^2 + (x_2-2)^2 - 4 \le 0$$
$$x_1 \ge 0, \quad x_2 \ge 0$$

Problem 2 (Y.B. Yun and Tanino, 2001)

$$\min_{x_1,x_2} \ (f_1(x_1,x_2), f_2(x_1,x_2)) = (2x_1 - x_2, -x_1)$$
$$s.t.$$
$$(x_1-1)^3 + x_2 \le 0$$
$$x_1 \ge 0, \quad x_2 \ge 0$$

Problem 3 (Y.B. Yun and Tanino, 2001)

$$\min_{x_1,x_2} \ (f_1(x_1,x_2), f_2(x_1,x_2)) = (x_1, x_2)$$
$$s.t.$$
$$x_1^3 - 3x_1 - x_2 \le 0$$
$$x_1 \ge -1, \quad x_2 \le 2$$

Problem 4 (A.D. Belegundu and Salagame, 1994)

$$\min_{x_1,x_2} \ (f_1(x_1,x_2), f_2(x_1,x_2)) = (-2x_1 + x_2, 2x_1 + x_2)$$
$$s.t.$$
$$-x_1 + x_2 - 1 \le 0$$
$$x_1 + x_2 - 7 \le 0$$
$$0 \le x_1 \le 5, \quad 0 \le x_2 \le 3$$

Problem 5 (J.Fernando and Verdegay, 1996)

$$\underset{X}{\text{Min}}\ (f_1(X), f_2(X)) = (5x_1 + 3x_2, 2x_1 + 8x_2)$$
$$s.t.$$
$$x_1 + 4x_2 - 100 \le 0$$
$$3x_1 + 2x_2 - 150 \le 0$$
$$-5x_1 - 3x_2 + 200 \le 0$$
$$-2x_1 - 8x_2 + 75 \le 0$$
$$x_1 \ge 0, \quad x_2 \ge 0$$

Problem 6 (Shigeru, 1997)

$$\underset{x_1, x_2}{\text{Minimize}}\ \ (f_1(x_1, x_2), f_2(x_1, x_2)) = (x_1, x_2)$$
$$s.t.$$
$$x_1^2 + x_2^2 - 1 \le 0$$
$$0 \le x_1 \le 1, \quad 0 \le x_2 \le 1$$

During all the experiments, the following parameters are used. The population size is fixed as 200. The generation number is fixed as 300. Subspace number is $S = 32$. The factor number is equivalent to the dimension of variable X. $D = 31$, $Q_0 = 31$, $\gamma = 0.5$, $Q_1 = 31$, $\lambda = 0.05$, $r = 0.5$, $k_1 = 10$, $k_2 = 0.1$, $p_j = 1$ (for $j = 1, \cdots, k$), $q_i = 1$ (for $i = 1, \cdots, \ell$), $\alpha(t) = 10 \times (-0.05t)$ (t is the generation number).

For each benchmark problem, simulation is repeated for 20 times. The non-dominated solutions are recorded. For all test problems, UDEP has a good performance in terms of both search speed and solution accuracy. As an illustration, the search performance of UDEP for problem 1 is shown in figures 1–3. The real Pareto frontier is shown in bold curves, while the non-dominated solutions are marked using circles. From these figures, it is obvious that UDEP has a very satisfactory performance.

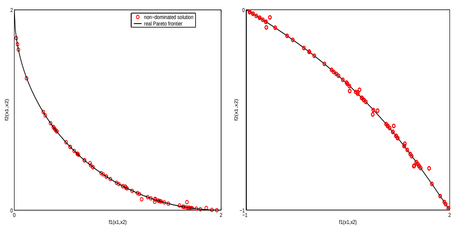

Fig. 1. UDEP results for problem 1 and 2

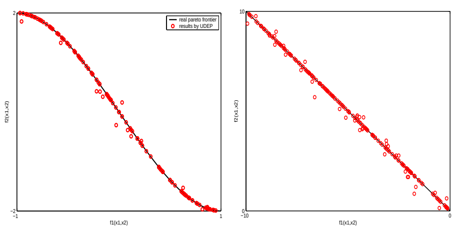

Fig. 2. UDEP results for problem 3 and 4

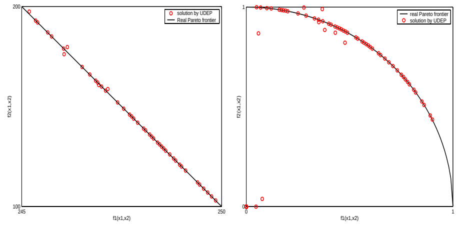

Fig. 3. UDEP results for problem 5 and 6

5. CONCLUSIONS

An evolutionary programming algorithm, called evolutionary programming based on uniform design (UDEP), is proposed to in this paper. Uniform design makes it possible to explore the search space evenly, while variable region search and nicheo technique help to keep diversity of the population. Their combination improves the search ability of EP significantly. Many numerical experimental results show the usefulness of the proposed method.

6. REFERENCES

A.D. Belegundu, D.V. Murthy and R.R. Salagame (1994). Multiobjective optimization of laminate
ceramic compsites using genetic algorithms. In: *proc. of Fifth AIAA/USAF/NASA Symp. on Multidisciplinary Analysis and Optimization*. pp. 1015–1022. Florida.

Fang, K.T. and Y. Wang (1994). *Number-Theoretic Methods in Statistics*. Chapman & Hall. London.

Hicks, C.R. (1981). A note on uniform distribution and experimental design. *KEXUE TONGBAO* pp. 485–489.

J.Fernando and J.L. Verdegay (1996). Interval multiobjective solid transportation problem via genetic algorithms. In: *proc. of the 6th conf. on information and management of uncertainty in knowledge-based systems (IPMU'96)*. pp. 787–792.

Shigeru, O. (1997). Pareto genetic algorithm for areodynamic design using the navier-stokes equations. In: *Genetic algorithms and evolution strategy in engineering and computer science: recent advances and industrial applications* (D. Quagliarella, Ed.). pp. 245–266. John Wiley & Sons. Chichester.

Y.B. Yun, H. Nakayama and T. Tanino (2001). Generation of efficient frontiers in multiobjective optimization problems by generalized data envelopment analysis. *European Journal of Operational Research* **129**, 586–595.