A Fast Dynamical Evolutionary Algorithm for Multi-objective Mechanical Component Design

XIUFEN ZOU¹, MINLING WANG¹, MINZHONG LIU², LISHAN KANG²
¹College of Mathematics and Statistics, Wuhan University, Wuhan 430072, China, ²State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, CHINA

Abstract:-In this paper, a fast dynamical multi-objective evolutionary algorithm (DMOEA) based on the principle of the minimal free energy in thermodynamics, was developed to solve mechanical component design problems. Its performance was compared with the ε -constraint method and NSGA-II proposed by Deb et al. Simulation results show that the proposed evolutionary approach produces excellent solutions for mechanical component design problems.

Key-Words: -multi-objective optimization, evolutionary algorithm, mechanical component design.

1 Introduction

Most real-world problems are multi-objective optimization problems (MOPs), and the effectiveness of evolutionary algorithms (EAs) in solving these problems has been widely recognized in recent ten years [1]. After David Schaffer's first study on evolutionary multi-objective optimization (EMO) in the mid of 1980s [2], a number of Pareto-based techniques and elitist algorithms have been proposed in the last decade [3][4], such as Pareto-based ranking procedure (FFGA) [5], niched Pareto genetic algorithm (NPGA) [6], Pareto- archived evolution strategy (PAES) [7], nondominated sorting genetic algorithm (NSGA) [8], and NSGA II [9], the strength Pareto evolutionary algorithm (SPEA) [10], and SPEA2 [11], thermo-dynamical genetic algorithm (TDGA) [12].

Although these techniques performed well in different comparative studies, there is still a large room for improvement as recent studies have shown [9][13][14].

In this paper, we propose a dynamical multi-objective evolutionary algorithm (DMOEA) based on the principle of the minimal free energy in statistical physics, for solving multi-objective optimization problems (MOPs). Two new ideas are introduced in DMOEA. One is a fitness assignment strategy by combining Pareto dominance relation and Gibbs entropy. The other is density distance and the Metropolis criterion for selection of new individuals in each generation.

The paper is structured as follows: Section 2 provides rather detailed descriptions of the proposed algorithm. In Section 3, numerical experiments are conducted, two measurements proposed by Deb [9] are used to compare DMOEA with other three well-known multi-objective evolutionary algorithms (MOEAS): NSGAI, SPEA, PAES for a number of test problems, moreover, the DMOEA is applied to solve the two mechanical component design problems and comparisons with other methods are made. Finally, some conclusions and future work are addressed in Section 4.

2 Description of the DMOEA for Multi-objective Optimization

Without loss of generality, we consider the following multi-objective minimization problem with \( n \) decision variables (parameters), \( M \) objectives and \( k \) constrained conditions:

\[
\begin{align*}
\text{Minimize} & \quad f(x) = (f_1(x), f_2(x), \ldots, f_M(x)) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, k;
\end{align*}
\]

2.1 The New Fitness Assignment Strategy

In statistical physics, the Gibbs distribution models a system in thermo-dynamical equilibrium at a given temperature. Further, it is also known that this distribution minimizes the free energy F defined by

\[
F = -E > -TS
\]
Where $<E>$ is the mean energy of the system, $S$ is the entropy and $T$ is the temperature. The minimization of $F$ means minimization of $<E>$ and maximization of $TS$. It is called “the principle of the minimal free energy”.

Such a statistical framework has been introduced into many fields. Since the minimization of the objective function (convergence towards the Pareto-optimal set) and the maximization of diversity in obtained solutions are two key goals in the multi-objective optimization, the working principle of a MOEA and the principle of finding the minimum free energy state in a thermodynamic system is analogous. In order to plunge a multi-objective optimization problem into such a statistical framework, we combine the rank value $R(i)$ calculated by Pareto-dominance relation with Gibbs entropy $S(i)$ to assign a new fitness $F(i)$ for each individual $i$ in the population, that is

$$F(i) = R(i) - TS(i)$$

(3) where $R(i)$ is the rank value of individual $i$, which is equal to the number of solution $n_i$ that dominates solution $i$ [5]. The rank values can be computed as follows.

$$R(i) = \Omega_i,$$

(4) where $\Omega_i = \{ \bar{x}_j \mid \bar{x}_j < \bar{x}_i, 1 \leq j \leq N, j \neq i \}$. In this way, the $R(i)=0$ corresponds to a non-dominated individual, while a high $Rank(i)$ means that $i$ is dominated by many individuals.

$$S(i) = -p_T(i) \log p_T(i)$$

(5) where $p_T(i) = \frac{1}{Z} \exp(-\frac{R(i)}{T})$ is the analogue of the Gibbs distribution, $Z = \sum_{i=1}^{N} \exp(-\frac{R(i)}{T})$ is called the partition function, and $N$ is the population size.

2.2 Density Estimation Technique

From the expression (3), we easily observe that it is difficult to distinguish the different individuals when their Rank values are equal. Therefore, we use a density estimation technique (which proposed by Deb et al [9]) to compute an individual’s crowding distance. The crowding distance $d(i)$ for individual $i$ is calculated according to the following steps:

1. Sorting the solution set $I$ according to each objective function in ascending order of magnitude.

(2) Noted that $d_m(i)$ is the crowding distance of individual $i$ referring to the $m$-th objective function, then $d_m(1) = d_m(I) = \infty$ (the boundary solutions are assigned an infinite distance value),

$$d_m(i) = \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{m_{\text{max}}} - f_m^{m_{\text{min}}}}, \quad i = 2, 3, \cdots, l-1$$

where $f_m^i$ is the $m$-th objective function, value of the $i$-th individual in the set $I$, $f_m^{\text{max}}$ and $f_m^{\text{min}}$ are the maximum and minimum values of the $m$-th objective function.

(3) Calculating the sum corresponding to all objective functions,

$$d(i) = \sum_{m=1}^{M} d_m(i), \quad M$$

is the number of all objective functions.

A solution with a smaller value of this distance measure is, in some sense, more crowded by other solutions. So we use the crowding distance to correct the expression of fitness value (3):

$$fitness(i) = R(i) - TS_i - d(i)$$

(6) In DMOEA, the fitness values are sorting in increasing order. The individual in population that the fitness value is smallest is called “the best individual”, and the individual in population that the fitness value is largest is called “the worst individual”.

2.3 The New Selection Criterion

In every generation, we always obtain new individuals by genetic operator, but it is worthwhile discussing that we use what kind of way to accept the new individuals, or eliminate the old individuals, and form new population at next generation. Since the DMOEA is based on the thermodynamical principle, we attempt to employ the Metropolis criterion of simulated annealing algorithm (SA)[15] and the crowding distance to guide the select process, that is,

1. If $R(X_{\text{new}}) < R(X_{\text{worse}})$, then $X_{\text{worst}} = X_{\text{new}}$

(2) If $R(X_{\text{new}}) = R(X_{\text{worse}})$ and $d(X_{\text{new}}) > d(X_{\text{worse}})$, then $X_{\text{worst}} = X_{\text{new}}$

(3) else if $\exp(\frac{R_{\text{worst}} - R_{\text{new}}}{T}) > \text{random (0,1)}$, then $X_{\text{worst}} = X_{\text{new}}$

Where $R_{\text{worst}}$ and $R_{\text{new}}$ are respectively the Rank values of the worst individuals and the new individuals.

The structure of DMOEA is described as follows:
Procedure DMOEA

Step 1: $t=0$, generate randomly an initial population $P(t) = \{X_1, X_2, \ldots, X_N\}$, $N$ is the population size;
Step 2: Calculate the rank values $\{R_1(t), \ldots, R_N(t)\}$ of all individuals in $P(t)$ according to (4);
Step 3: Save the individuals whose rank values are equal to zero;
Step 4: Calculate the fitness of all individuals according to equation (6), and sort them in increasing order;
Step 5: Repeatedly execute step 6 to step 11 until the termination conditions are satisfied;
Step 6: $t=t+1$;
Step 7: Randomly select $m_1$ individuals to do multi-parent crossover and $m_2$ individuals to mutate, and to generate $n$ new individuals;
Step 8: Compare the new individuals with the worst individuals, and accept the new individuals according to Section 2.3;
Step 9: Calculate the rank values $\{R_1(t), \ldots, R_N(t)\}$ of all individuals in new population $P(t)$ according to (4);
Step 10: Save the individuals whose the rank values are equal to zero;
Step 11: Calculate the fitness of all individuals according to equation (6), sort them in increasing order, and record the worst individuals;
Step 12: Output the all results.

Remark: In step 7, multi-parents crossover originated appeared in Guo T. et al [16]. Randomly select $m$ individual $X'_1, X'_2, \ldots, X'_m$ from population $P(t)$ to form a new individual $s$, $s = \sum_{i=1}^{m} a_i X'_i$, where $a_i$ is a random number in $[-0.5, 1.5]$, and $\sum_{i=1}^{m} a_i = 1$ (Quasi-linear combination can guarantee searching more extensive range, even including the boundary).

3 The Numerical Experiments

3.1 Test Problems

To verify the efficiency and effectiveness of the DMOEA, we have conducted many numerical experiments and compare its performance with several other MOEAs: NSGA II, SPEA, PAES.

(1) Nine Difficult test problems from Deb [3][9]

SCH1, FON, POL, KUR, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6

The following is the two mechanical component design problems;

(2) Two-Bar Truss Design

This problem was originally studied using the $\varepsilon$-constraint method [17], NSGAII [3] and Azarm, S. et al [18], John Eddy et al [19]. The design of truss (see Fig.1) can be written as the following two-objective minimization problem with three variables ($y$ is vertical distance between B and C in m, $x_1$ and $x_2$ is the cross-sectional areas of AC and BC respectively in $m^2$, $f_{\text{volume}}$ is the overall volume of material used, $f_{\text{stress,AC}}$ is the stress in bar AC). The system is subject to three inequality constraints. The first two are constraints on the objective function values, the third constraint limits the stress in bar BC.

$$\text{Minimize } f_{\text{volume}} = x_1 \sqrt{16 + y^2} + x_2 \sqrt{1 + y^2}$$

$$\text{Minimize } f_{\text{stress,AC}} = \frac{20 \sqrt{16 + y^2}}{xy}$$

Subject to

$$g_1(x_1, x_2, y) = f_{\text{volume}} \leq 0.1$$
$$g_2(x_1, x_2, y) = f_{\text{stress,AC}} \leq 100$$
$$g_3(x_1, x_2, y) = \frac{80 \sqrt{1 + y^2}}{xy} \leq 100$$
$$0.0001 \leq x_1 \leq 0.1,$$
$$0.0001 \leq x_2 \leq 0.1,$$
$$1 \leq y \leq 3$$

(3) Gear Train Design

A compound gear train is to be designed to achieve a specific gear ratio between the driver and driven shafts (see Fig.2, is a copy of Fig.293 from Deb [3]). The objective of the gear train design is to find the number of teeth in each of the four gears so as to minimize (a) the error between the obtained gear ratio and a required gear ratio of 1/6.931[20] and (b) the maximum size of any of the four gears. Since the number of teeth must be integers, all four variables are thus strictly integers. By denoting the variable vector $X = (x_1, x_2, x_3, x_4) = (T_d, T_p, T_r, T_f)$, we
write the two-objective optimization problem as follows:

\[
\text{Minimize } f_1(X) = \left(\frac{1}{6.931} - \frac{x_1x_2}{x_3x_4}\right)^2
\]

\[
\text{Minimize } f_2(X) = \max(x_1, x_2, x_3, x_4)
\]

s.t. \(12 \leq x_1, x_2, x_3, x_4 \leq 60, \text{ all } x_i \text{ are integers}\)

Fig. 1 A two-bar truss. This is a reprint of Fig. 3 from John Eddy et al [19]

Fig. 2 A compound gear train

### 3.2 Results and Discussion

In order to compare with NSGA II, SPEA, PAES, we use the two performance metrics defined in [9] to test DMOEA, and the setting of parameters is basically the same as that in [9] (see Table 1). The algorithm has been coded in C language and implemented on a Pentium PC 500MHz in double precision arithmetic. For each problem, 20 runs with different random seeds have been carried out.

Table 2 shows the mean and variance of the convergence metric obtained by four algorithms: DMOEA, NSGA-II (real-coded), SPEA, and PAES. DMOEA is able to converge better in all problems except in POL, where NSGA-II found better convergence. In all cases with DMOEA, the variance in 20 runs is also smaller (In the table, the zero variance means that the variance is less than \(10^{-6}\)).

Table 3 shows the mean and variance of the diversity metric obtained by all four algorithms. DMOEA performs better than NSGA-II (real-coded), SPEA and PAES in all test problems except in KUR, where NSGA-II found better diversity.

Fig. 3 shows the obtained non-dominated solutions by DMOEA for the two-bar truss problem. The figure shows that DMOEA can obtain widely spread solutions when compared to the \(\varepsilon\)-constraint method and NSGAII (see ref. [3])

Table 4 shows the extreme solutions (marked as ‘EE’ and ‘DD’) obtained by DMOEA for the gear train design, which indicates the power of the DMOEA in finding a wider spread of solutions when compared to the NSGA II (the extreme solutions is marked as ‘E’ and ‘D’ [3]). What is also important is that all solutions of the NSGA II (including ‘EE’ and ‘DD’) have been found in just one simulation run of the DMOEA.

#### Table 1 Parameter settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>(N=100)</td>
</tr>
<tr>
<td>Crossover</td>
<td>Multi-parent crossover, the crossover probability is 0.1 (different from [9])</td>
</tr>
<tr>
<td>Mutate</td>
<td>uniform mutate, the mutate probability is (1/n) (where (n) is the number of decision variables)</td>
</tr>
<tr>
<td>The maximum generation</td>
<td>(g=250) (25000 function evaluations)</td>
</tr>
<tr>
<td>Temperature</td>
<td>(T=10000).</td>
</tr>
</tbody>
</table>

#### Table 2 The mean \(\bar{\gamma}\) (first row) and variance \(\sigma^2\) (second row) of the convergence metric

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SCH1</th>
<th>FON</th>
<th>POL</th>
<th>KUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMOEA</td>
<td>0.000408</td>
<td>0.002245</td>
<td>0.034703</td>
<td>0.022262</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.000043</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.003891</td>
<td>0.001931</td>
<td>0.015553</td>
<td>0.020964</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.000001</td>
<td>0</td>
<td>0.00018</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.003403</td>
<td>0.125692</td>
<td>0.037812</td>
<td>0.045617</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.000038</td>
<td>0</td>
<td>0.00005</td>
</tr>
<tr>
<td>PAES</td>
<td>0.001313</td>
<td>0.151263</td>
<td>0.03864</td>
<td>0.057323</td>
</tr>
<tr>
<td></td>
<td>0.000003</td>
<td>0.000905</td>
<td>0.00431</td>
<td>0.011989</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMOEA</td>
<td>0.000150</td>
<td>0.000119</td>
<td>0.000493</td>
<td>0.000290</td>
<td>0.000520</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000001</td>
<td>0</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.033482</td>
<td>0.072391</td>
<td>0.114500</td>
<td>0.513053</td>
<td>0.296564</td>
</tr>
<tr>
<td></td>
<td>0.004750</td>
<td>0.031689</td>
<td>0.007940</td>
<td>0.118460</td>
<td>0.013135</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.001799</td>
<td>0.001339</td>
<td>0.047517</td>
<td>7.340299</td>
<td>0.221138</td>
</tr>
<tr>
<td></td>
<td>0.000001</td>
<td>0</td>
<td>0.000047</td>
<td>6.572516</td>
<td>0.000449</td>
</tr>
<tr>
<td>PAES</td>
<td>0.082085</td>
<td>0.126276</td>
<td>0.023872</td>
<td>0.854816</td>
<td>0.085469</td>
</tr>
<tr>
<td></td>
<td>0.008679</td>
<td>0.036877</td>
<td>0</td>
<td>0.527238</td>
<td>0.066664</td>
</tr>
</tbody>
</table>
4 Conclusions and Future Work

In this paper, we have presented DMOEA, a high performance evolutionary algorithm for multi-objective optimization problems that employs a new fitness assignment strategy and a new accepting criterion based on the principle of the minimal free energy. We have conducted experiments by computer simulation to explore the applicability of DMOEA to the complex problems of multiobjective optimization. Our computational experience shows that the proposed DMOEA can obtain very promising results. In addition, the procedure can be easily extended to solve other real-world application problems with different objectives. Therefore in the future, the applicability of DMOEA to more difficult and complex types of real-world problems, including the combinatorial optimization problems must be studied.

Acknowledgement

We would like to thank professor Kalyanmoy Deb in Indian Institute of Technology, India for warmly explaining our questions, also thank Mr. Naoki Mori in Osaka Prefecture University, Japan for provide us their valuable papers.

References:


Table 3 The mean $\Delta$ (first row) and variance $\sigma^2$ (second row) of the diversity metric

| Algorithm | SCH1 | FON | POL | KUR | DMOEA | SCH1 | FON | POL | KUR | NSGAII (Real-coded) | SCH1 | FON | POL | KUR | SPEA | SCH1 | FON | POL | KUR | PAES | SCH1 | FON | POL | KUR |
|-----------|------|-----|-----|-----|-------|------|-----|-----|-----|-----|-------------------|------|-----|-----|-----|------|------|-----|-----|-----|------|------|-----|-----|-----|-----|------|-----|-----|-----|-----|
| DMOEA     | 0.202006 | 0.351841 | 0.272671 | 0.580337 | - | 0.000416 | 0.000345 | 0.002868 | 0.000992 | - | 0.477899 | 0.378065 | 0.452150 | 0.411477 | - | 0.003471 | 0.000639 | 0.002868 | 0.000992 | - | 0.004372 | 0.005546 | 0.008475 | 0.002619 | - | 1.063288 | 1.162528 | 1.02007 | 1.079838 | - | 0.002868 | 0.008945 | 0.0 | 0.013772 | - |
| NSGAII    | 0.909917 | 0.919882 | 0.969957 | 0.969957 | - | 0.950715 | 0.950715 | 0.950715 | 0.950715 | - | 0.909917 | 0.919882 | 0.969957 | 0.969957 | - | 0.909917 | 0.919882 | 0.969957 | 0.969957 | - | 0.909917 | 0.919882 | 0.969957 | 0.969957 | - | 0.909917 | 0.919882 | 0.969957 | 0.969957 | - | 0.909917 | 0.919882 | 0.969957 | 0.969957 | - |

Fig.3 Optimized solutions obtained using DMOEA for the two-bar truss problem

Table 4 Obtained extreme solutions by DMOEA for the gear train design

<table>
<thead>
<tr>
<th>Solution x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>Max.diameter(cm)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE</td>
<td>15</td>
<td>15</td>
<td>39</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>E</td>
<td>12</td>
<td>12</td>
<td>27</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>D</td>
<td>12</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>DD</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>


