***Applied Mathematics*, 2014, 5, 318-326**
Published Online January 2014 (http://www.scirp.org/journal/am)
http://dx.doi.org/10.4236/am.2014.52032

On Metaheuristic Optimization Motivated by the Immune System

Mohammed Fathy Elettreby[1,2*], Elsayd Ahmed[2], Houari Boumedien Khenous[1]
[1]Department of Mathematics, Faculty of Science, King Khalid University, Abha, KSA
[2]Department of Mathematics, Faculty of Science, Mansoura University, Mansoura, Egypt
Email: [*]mohfathy@mans.edu.eg

Received November 16, 2013; revised December 16, 2013; accepted December 23, 2013

Copyright © 2014 Mohammed Fathy Elettreby *et al*. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Mohammed Fathy Elettreby *et al*. All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

ABSTRACT

In this paper, we modify the general-purpose heuristic method called extremal optimization. We compare our results with the results of Boettcher and Percus [1]. Then, some multiobjective optimization problems are solved by using methods motivated by the immune system.

KEYWORDS

Multiobjective Optimization; Extremal Optimization; Immunememory and Metaheuristic

1. Introduction

Multiobjective optimization problems (MOPs) [2] are existing in many situations in nature. Most realistic optimization problems require the simultaneous optimization of more than one objective function. In this case, it is unlikely that the different objectives would be optimized by the same alternative parameter choices. Hence, some trade-off between the criteria is needed to ensure a satisfactory problem. Multiobjective (Multicriteria) optimization has its roots in late 19th century welfare economics, in the works of Edge worth and Pareto. A mathematical description is as follows: finding the vector $\boldsymbol{x}^* = \left(x_1^*, x_2^*, \cdots, x_n^*\right)$ which satisfies the m inequality constraints $g_i\left(\boldsymbol{x}^*\right) \geq 0, i = 1, 2, \cdots, m$, the p equality constraints $h_j\left(\boldsymbol{x}^*\right) = 0, j = 1, 2, \cdots, p$, and minimizing the vector function $F\left(\boldsymbol{x}\right) = \left(f_1\left(\boldsymbol{x}\right), f_2\left(\boldsymbol{x}\right), \cdots, f_k\left(\boldsymbol{x}\right)\right)$, where $\boldsymbol{x} = \left(x_1, x_2, \cdots, x_n\right) \in \Omega$, is the vector of decision variables, each decision variable x_i is bounded by lower and upper limits $l_i \leq x_i \leq u_i, i = 1, 2, \cdots, n$. The space in which the objective vector belongs is called the objective space and the image of the feasible set under F is called *the attained set*.

The scalar concept of *optimality* does not apply directly in the multiobjective setting. A useful replacement is the notion of *Pareto optimality*. So, we have the following definitions related to this notation [3]:

Definition 1 (Pareto dominance) A vector $\boldsymbol{u} = \left(u_1, u_2, \cdots, u_k\right)$ is said to be dominated by another vector $\boldsymbol{v} = \left(v_1, v_2, \cdots, v_k\right)\left(\boldsymbol{u} \prec \boldsymbol{v}\right)$ iff $\forall i \in \left\{1, 2, \cdots, k\right\} : u_i \leq v_i \wedge \left(\exists j \in \left\{1, 2, \cdots, k\right\} : u_j < v_j\right)$.

Definition 2 (Pareto optimality) A solution $\boldsymbol{x} \in \Omega$ is said to be Pareto optimal with respect to Ω iff; there is no $\boldsymbol{y} \in \Omega$ for which $\boldsymbol{v} = F\left(\boldsymbol{y}\right) = \left(f_1\left(\boldsymbol{y}\right), f_2\left(\boldsymbol{y}\right), \cdots, f_k\left(\boldsymbol{y}\right)\right)$ dominate $\boldsymbol{u} = F\left(\boldsymbol{x}\right) = \left(f_1\left(\boldsymbol{x}\right), f_2\left(\boldsymbol{x}\right), \cdots, f_k\left(\boldsymbol{x}\right)\right)$.

Pareto optimal points are also known as *efficient* solutions, *non-dominated*, or *non-inferior* points.

Definition 3 (Pareto optimal set) The Pareto optimal set P_S is the set of Pareto optimal solutions, *i.e.* $P_S = \left\{\boldsymbol{x} \in \Omega \middle| \neg\exists \boldsymbol{y} \in \Omega : F\left(\boldsymbol{y}\right) \prec F\left(\boldsymbol{x}\right)\right\}$.

Definition 4 (Pareto optimal front) The Pareto optimal front P_F is the set of objective functions values corresponding to the solutions in P_S, *i.e.* $P_F = \left\{F\left(\boldsymbol{x}\right) = \left(f_1\left(\boldsymbol{x}\right), f_2\left(\boldsymbol{x}\right), \cdots, f_k\left(\boldsymbol{x}\right)\right) \middle| \boldsymbol{x} \in P_S\right\}$.

The multiobjective problem is almost always solved by combining the multiple objectives into one scalar ob-

[*]Corresponding author.

jective whose solution is a Pareto optimal point for the original MOP. Most algorithms have been developed in the linear framework (*i.e.* linear objectives and linear constraints), but the techniques described below are also applicable to nonlinear problems. Several methods have been proposed to solve continuous multiobjective optimization problems. In the next section, some multiobjective methods are discussed. In the third section, extremal optimization is introduced and two immune motivated optimization algorithms are proposed. In Section 4, we generalize extremal optimization to multiobjective optimization.

2. Some Multiobjective Optimization Methods

Almost every real life problem is multiobjective problem [2,4]. Methods for solving multiobjective optimization problems are mostly intuitive. Here, we list and discuss some of them.

The first method is the ***Minimizing Weighted Sums of Functions***. A standard technique for MOP is to minimize a positively weighted convex sum of the objectives $f_i(x)$, that is, $\min \sum_{i=1}^{n} w_i f_i(x)$ where $w_i \geq 0$ and $\sum_{i=1}^{n} w_i = 1$.

It is known that the minimizer of this combined function is Pareto optimal. It is up to the user to choose appropriate weights. Until recently, considerations of computational expense forced users to restrict themselves toper forming only one such minimization. Newer, more ambitious approaches aim to minimize convex sums of the objectives for various settings of the convex weights, therefore generating various points in the Pareto set. Though computationally more expensive, this approach gives an idea of the shape of the Pareto surface and provides the user with more information about the trade-off among the various objectives. However, this method suffers from two drawbacks. First, the relationship between the vector of weights and the Pareto curve is such that a uniform spread of weight parameters rarely produces a uniform spread of points on the Pareto set. Often, all the points found are clustered in certain parts of the Pareto set with no point in the interesting/middle part" of the set, thereby providing little insight into the shape of the trade-off curve. The second drawback is that non-convex parts of the Pareto set cannot be obtained by minimizing convex combinations of the objectives (note though that non-convex Pareto sets are seldom found in actual applications).

Here another problem is considered. Assume that the set $D = \{d_1, d_2, \cdots, d_n\}$ of decisions weights $w_1, w_2, \cdots, w_n$ reflecting decision-makers' preferences. A matrix $A = (a_{ij}), i, j = 1, 2, \cdots, n$ is obtained after asking the decision maker to quantify the ratio of his/her preferences of one decision over another. In other words, for every pair of decisions d_i, d_j, the term $a_{ij} > 0$ is requested satisfying $a_{ij} \approx \frac{w_i}{w_j}$, *i.e.* instead of giving the weights w_i directly, he/she has a matrix A. This matrix must be a positive reciprocal matrix, *i.e.* $a_{ij} = \frac{1}{a_{ji}} > 0$. For a given positive reciprocal matrix A, different procedures can be followed in order to obtain weights $w_1, w_2, \cdots, w_n$. By Perron-Frobenius theorem [5], the matrix A has a unique positive dominant eigenvalue. But due to the above properties of a_{ij} one can derive the following:

Proposition 1 If $a_{ij} \cdot a_{jk} \cdot a_{ki} = 1, i, j, k = 1, 2, \cdots, n$, then the eigenvalues of A in the case of *n* objectives are *n*; *0* where the zero is repeated $n-1$ times.

Proof: It is direct to see that the characteristic polynomial of the positive reciprocal matrix A satisfies the recurrence relation;

$$P_n(\lambda) = (1-\lambda) P_{n-1}(\lambda) + (-1)^{n-1} (n-1) \lambda^{n-2}, n \geq 2, \tag{1}$$

And $P_2(\lambda) = -\lambda(2-\lambda), P_1(\lambda) = 1-\lambda$.

Using mathematical induction the proposition is proved.

Now, there are two simple methods to determine the weights $w_1, w_2, \cdots, w_n$ given the matrix A [6]. The first method is that the required weights are proportional to the eigenvector corresponding the unique eigenvalue *n* by solving $Aw = nw$, where w is the weight vector. The second method is that it is proportional to the row geometric mean of a_{ij} *i.e.* they are given by; $w_i = \dfrac{\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}}{\sum_{i=1}^{n}\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}}$.

It has been argued that the method of weighted sums of objectives is one of the best methods in multiobjective combinatorial optimization (MOCO) [7]. The reason is that Pareto set for discrete variables is a set of points.

Such set can be obtained via a convex combination of the objectives e.g. the weights method.

The second method is the ***Homotopy Techniques***. Homotopy techniques aim to trace the complete Pareto curve in the bi-objective case ($n = 2$). By tracing the full curve, they overcome the sampling deficiencies of the weighted sum approach [8,9]. The main drawback is that this approach does not generalize to the case of more than two objectives.

The third method is the ***Goal Programming***. In the goal programming approach, we minimize one objective while constraining the remaining objectives to be less than given target values e.g. minimize $f_1(x)$ subject to $f_i(x) \le a_i, i = 2, 3, \cdots, n$, where a_i are parameters to be gradually decreased till no solution is found. This method is especially useful if the user can afford to solve just one optimization problem. However, it is not always easy to choose appropriate goals for the constraints a_i. Goal programming cannot be used to generate the Pareto set effectively, particularly if the number of objectives is greater than two.

The fourth method is ***Multilevel Programming***. Multilevel programming is a one-shot optimization technique and is intended to find just one optimal point as opposed to the entire Pareto surface. The first step in multilevel programming involves ordering the objectives in terms of importance. Next, we find the set of points $x \in C$ for which the minimum value of the first objective function $f_1(x)$ is attained. We then find the points in this set that minimize the second most important objective $f_2(x)$. The method proceeds recursively until all objectives have been optimized on successively smaller sets. Multilevel programming is a useful approach if the hierarchical order among the objectives is of prime importance and the user is not interested in the continuous trade-off among the functions. However, problems lower down in the hierarchy become very tightly constrained and often become numerically infeasible, so that the less important objectives have no influence on the final result. Hence, multilevel programming should surely be avoided by users who desire a sensible compromise solution among the various objectives. Also, this method is called ***lexicographic method***.

A famous application is in university admittance where students with highest grades are allowed in any college they choose. The second best group is allowed only the remaining places and so on. The drawback of this method is that two distinct lexicographic optimization with distinct sequences of the same objective functions does not produce the same solution. Also, this method is useful but in some cases it is not applicable.

A fifth method using ***fuzzy logic*** is to study each objective individually and find its maximum and minimum say $f_{\max}(x)$, $f_{\min}(x)$ respectively. Then determine a membership $m_i = \dfrac{f_{i_{\max}}(x) - f_i(x)}{f_{i_{\max}}(x) - f_{i_{\min}}(x)}$. Thus $0 \le m \le 1$. Then apply $\max\min(m_i, i = 1, 2, \cdots, n)$. Again this method is guaranteed to give a Pareto optimal solution. This method is a bit difficult to apply for large number of objectives.

A sixth method is the ***Keeney-Raiffa method*** which uses the product of objective functions to build an equivalent single objective one called Keeney-Raiffa utility function.

The seventh method is ***Normal-Boundary Intersection*** [10,11]. The normal-boundary intersection method (NBI) uses a geometrically intuitive parameterization to produce an even spread of points on the Pareto surface, giving an accurate picture of the whole surface. Even for poorly scaled problems (for which the relative scaling on the objectives are vastly different), the spread of Pareto points remains uniform. Given any point generated by NBI, it is usually possible to find a set of weights such that this point minimizes a weighted sum of objectives, as described above. Similarly, it is usually possible to define a goal programming problem for which the NBI point is a solution. NBI can also handle problems where the Pareto surface is discontinuous or non-smooth, unlike homotopy techniques. Unfortunately, a point generated by NBI may not be a Pareto point if the boundary of the attained set in the objective space containing the Pareto points is non-convex or "folded" (which happens rarely in problems arising from actual applications). NBI requires the individual minimizers of the individual functions at the outset, which can also be viewed as a drawback.

3. Extremal Optimization and the Immune System

In this section, we concentrate more on extremal optimization then relate it to the immune system and generalize it to multiobjective cases.

Extremal Optimization (EO) is an optimization heuristic inspired by the Bak-Sneppen model of self-organized criticality from the field of statistical physics [1,12]. This heuristic was designed initially to address combinatorial optimization problems such as the travelling salesman problem and spin glasses, although the technique has been demonstrated to function in optimization domains. EO was designed as a local search algorithm for combinatorial optimization problems. Unlike genetic algorithms, which work with a population of candidate solutions, EO evolves a single solution and makes local modifications to the worst components. This requires that a suitable representation be selected which permits individual solution components to be assigned a quality measure ("fitness"). This differs from holistic approaches such as ant colony optimization and evolutionary computation that assign equal-fitness to all components of a solution based upon their collective evaluation against an objective function. The algorithm is initialized with an initial solution, which can be constructed randomly, or derived from another search process.

In the Bak-Sneppen model, species are located on the sites of a lattice, and have an associated *fitness* value between 0 and 1. At each time step, the one species with the smallest value (poorest degree of adaptation) is selected for a random update, having its fitness replaced by a new value drawn randomly from a flat distribution on the interval [0; 1]. But the change in fitness of one species impacts the fitness of interrelated species. Therefore, all of the species at neighboring lattice sites have their fitness replaced with new random numbers as well. After a sufficient number of steps, the system reaches a highly correlated state known as self-organized criticality [13].

Extremal optimization is quite similar to the way the immune system (IS) renews its cells [14]. Almost everyday new immune cells are replaced in the blood stream. If within few weeks they were able to recognize antigens (viruses or bacteria) then they are preserved for longer period. Otherwise they are replaced randomly. This dynamics is called extremal dynamics (EO) [15]. It can explain the long range memory of the immune system even without the persistence of antigens. The reason is that if a system evolves according to such dynamics then the annihilation probability for a clone (a type of cells) that has already survived for time *t* is inversely proportional to $t+c$, where *c* is a constant. Boettcher and Percus [1] have used extremal optimization to solve some single objective combinatorial optimization problems. Their algorithm have been modified by [12,16] and used to solve 3-dimensional spin glass and graph coloring problems.

Definition 5 Extremally driven systems are the systems that updated by identifying an active region of the system and renewing this region whilst leaving the remainder unchanged. The active subsystem is chosen according to some kind of extremal criterion; often it is centered on the location of the minimum of some spatially varying scalar variable.

Consider a system of *n* elements, each element assigned a single scalar variable $x_i, i = 1, 2, \cdots, n$ drawn from the fixed probability distribution function $p(x)$. For every time step, the element with the smallest value in the system is selected and renewed by assigning a new value which is drawn from $p(x)$. It is assumed that no two x_i can take the same value.

Definition 6 For the above system the typical values of x_i increase monotonically in time. This means that any renewed element is likely to have a smaller x_i than the bulk, and hence a shorter than average lifespan until it is again renewed. Corresponding, elements that have not been renewed for sometime will have a longer than average life expectancy. This separation between the shortest and the longest lived elements will become more pronounced as the system evolves and the average x_i in the bulk increases. This phenomenon is called long-time memory.

Proposition 2 Extremally driven systems can generally be expected to exhibit long-term memory [15].

Proof Let $P_t(S)$ be the probability to find the system in a state *S* after *t* updates, where $S = \{x_1, x_2, \cdots, x_N\}$. At the next step $t+1$ only one of the values will change that may be any one of the N elements. Let $S^i = \{x_1, \cdots, \dot{x}_{i-1,i}, x_{i+1}, \cdots, x_N\}$ be the state that x_i is the smallest value. Then $P_{t+1}(S)$ can be given by,

$$P_{t+1}(S) = \sum_{i=1}^{N} p(x_i) \int_{-\infty}^{m_i} P(S^i) \mathrm{d}\dot{x}_i \,, \tag{2}$$

where the function $m_i : R^{N-1} R$ is defined by;

$$m_i = \min\{x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N\}. \tag{3}$$

Using the substitution $u_i = \int_{-\infty}^{x_i} p(z) \mathrm{d}z, 0 \le u_i \le 1$, Equation (2) can be rewritten as;

$$Q_{t+1}(S) = \sum_{i=1}^{N} \int_0^{m_i} Q_t(S^i) \mathrm{d}\grave{u}_i \,, \tag{4}$$

where $Q_t(S(u_i)) \prod_{i=1}^{N} \mathrm{d}u_i = P_t(S(x_i)) \prod_{i=1}^{N} \mathrm{d}x_i$, $S = \{u_1, u_2, \cdots, u_N\}, S^i = \{u_1, \cdots, u_{i-1}, \grave{u}_i, u_{i+1}, \cdots, u_N\}$ and

$m_i = \min\{u_1, \cdots, u_{i-1}, u_{i+1}, \cdots, u_N\}$.

Identity 1 Since $m_i = \min\{u_1, \cdots, u_{i-1}, u_{i+1}, \cdots, u_N\}$ and u_i are independent, then;

$$\int_0^{m_i} m_j^t \mathrm{d}u_i = \begin{cases} m_i^{t+1}, & i = j \\ \dfrac{m_i^{t+1}}{t+1}, & i \neq j \end{cases}. \tag{5}$$

Identity 2 $$\int_{D_i} m_j^t \mathrm{d}V = \begin{cases} \dfrac{(N-1)!(t+1)!}{(N+1)!}, & i = j \\ \dfrac{(N-1)!(t)!}{(N+1)!}, & i \neq j \end{cases}, \tag{6}$$

Where D_i is the domain of space in which u_i is the smallest, and $\mathrm{d}V = \mathrm{d}u_1 \mathrm{d}u_2 \cdots \mathrm{d}u_N = \prod_{k=1}^N \mathrm{d}u_k$.

Now, let us return to Equation (4), using identity (1) and $Q_0 = 1$, we can get;

$$Q_1(S) = \sum_{i=1}^N \int_0^{m_i} Q_0(S^i)\mathrm{d}\grave{u}_i = \sum_{i=1}^N \int_0^{m_i} \mathrm{d}\grave{u}_i = \sum_{i=1}^N mi$$

$$Q_2(S) = \sum_{i=1}^N \int_0^{m_i} Q_1(S^i)\mathrm{d}\grave{u}_i = \sum_{i=1}^N \int_0^{m_i} \sum_{i=1}^N m_i \mathrm{d}\grave{u}_i = \sum_{i=1}^N \int_0^{m_i} (m_1 + m_2 + \cdots + m_N)\mathrm{d}\grave{u}_i$$

$$Q_2(S) = \int_0^{m_1} \sum_{i=1}^N (m_1 + m_2 + \cdots + m_N)\mathrm{d}\grave{u}_1 + \cdots + \int_0^{m_N} \sum_{i=1}^N (m_1 + m_2 + \cdots + m_N)\mathrm{d}\grave{u}_N$$

$$Q_2(S) = \left(\frac{N-1}{2} + 1\right) \sum_{i=1}^N m_i^2 .$$

Then;

$$Q_t(S) = \frac{(N+t-1)!}{(N)!(t)!} \sum_{i=1}^N m_i^t. \tag{7}$$

Since Q_t is symmetric in m_i and therefore in u_i, the probability that any particular element in the system, say u_k, is the smallest at a given time t_w is $\frac{1}{N}$. Let $Q_{t_w+t,t_w}^k(S)$ is the probability to find the system in a state Sat a time $t_w + t$ given that u_k was not the minimum at any of the times $t_w, t_w + 1, \cdots, t_w + t - 1$, then;

$$Q_{t_w+t,t_w}^k(S) = \frac{1}{N-1} \binom{N + t_w + t - 1}{N-1} \sum_{i \neq k}^N m_i^{t_w+t}, t \geq 1. \tag{8}$$

The corresponding probability that u_k is the smallest, denoted by q_{t_w+t,t_w}^k can be calculated by integrated Equation (8) over D_k using Equation (6);

$$q_{t_w+t,t_w}^k = \frac{1}{N + t_w + t}, \tag{9}$$

which independent of k also $Q_{t_w+t,t_w}^k(S)$. Then the probability of an element being renewed decreases with the time since it was last renewed according to $q_{t_w+t,t_w}^k \sim t^{-1}$, also $q_{t_w+t,t_w}^k \sim t_w^{-1}$.

$R(t)$ is defined as the probability that a randomly chosen element i has the same value of x_i at time t that it had at $t = 0$, then $R(0) = 1$ and $R(1) = \frac{N}{N-1}$. For $t \geq 2$ observe that $R(t)$ only decrease when an element is renewed for the first time, so $R(t+1) = (1 - q_{t,0}^k) R(t)$ hence from Equation (9);

$$R(t) = R(1) \prod_{s=1}^{t-1} (1 - q_{s,0}^k). \tag{10}$$

Since $q_{s,0}^k = \frac{1}{N+s}$, $(1 - q_{s,0}^k) = \frac{N+s-1}{N+s}$, then

$$R(t) = R(1) \prod_{s=1}^{t-1} (1 - q_{s,0}^k) = \frac{N-1}{N+t-1}. \tag{11}$$

The slow decay of $R(t)$ shows that a significant proportion of the system will remain in its initial state until arbitrarily late times, already suggesting some form of long-term memory.

The existence of aging can be most clearly expressed in terms of the two-time correlation function $C(t_w+t,t_w)$ between the state of the system at times t_w and t_w+t. The probability that a randomly chosen element has the same value of x_i at times t_w and t_w+t is $C(t_w+t,t_w)$. Note that $C(t_w,t_w)=1$, $C(t_w+t,t_w)=\frac{N-1}{N}$ where $C(t,0)=R(t)$,

$$C(t_w+t,t_w)=C(t_w+1,t_w)\prod_{s=t_w+1}^{t_w+t-1}\left(1-q_{s,t_w}^{k}\right)$$

$$C(t_w+t,t_w)=\frac{N-1}{N}\left(\frac{N+t_w}{N+t_w+t-1}\right),t\geq 1.$$

After a short transient this scales as $C(t_w+t,t_w)\approx\frac{N-1}{N}\left(1+\frac{t}{t_w}\right),\frac{t_w}{N}\gg 1$.

That *t* and t_w only appear in the ratio $\frac{t}{t_w}$ is what we mean by aging. Aging indicates the existence of some form of long-term memory. Finally, as *N* tends to infinity we get $R(\tau)=(1+\tau)^{-1}$ and $C(\tau_w+\tau,\tau_w)=\frac{\tau_w+1}{\tau_w+\tau+1}$ where $\tau\equiv\frac{1}{N}$. This can explain the long term memory of the immune system in the absence of antigens.

Nature has been created in a fascinating way. As we learn more, we found that it is more efficient to imitate it. Recently, this approach has been applied to optimization [1]. The author's considered the spin glass problem. They assign a spin variable $x_i\in\{-1,1\}$ for each site $i,i\in\{1,2,\cdots,n\}$. Each site is connected to each nearest neighbors j via a bond variable $J_{ij}\in\{-1,1\}$, assigned at random. The goal is to find the configuration $S=\{x_1,x_2,\cdots,x_n\}$, to approximately minimize the cost function or Hamiltonian;

$$C(S)=H(x)=-\frac{1}{2}\sum\sum_{\langle i,j\rangle}J_{ij}x_ix_j. \tag{12}$$

Also, they assigned a fitness f_i to each spin variable x_i by the relation $f_i=x_i\left(\frac{1}{2}\sum_j J_{ij}x_j\right)$. Then the one with the lowest fitness is removed and replaced with another one at random. It has been applied to the both, spin glass and graph coloring problems. The extremal optimization algorithm is as follows:

Choose $\{x_i,i=1,2,\cdots,n\}$ randomly to form an initial configuration S, set $S_{best}=S$. Repeat the following steps as desired,

1) Find the fitness f_i of the variable x_i for all $i=1,2,\cdots,n$.

2) Find the site i^* with the lowest fitness (*i.e.* $f_{i^*}\leq f_i$) and choose another configuration S' randomly such that x_{i^*} is replaced by another state.

3) If $H(S')<H(S)$ then $S_{best}=S'$, else $S_{best}=S$.

The weakness of this approach is that focusing only on the worst fitness can lead to narrow deterministic process. To overcome this weakness, the authors replaced step (2) of the above algorithm by ranking the sites in an ascending order according to their fitness. Then the replaced site is chosen randomly according to the probability distribution $p_k\propto k^{-\tau}$.

We proposed two modifications. The first immune motivated optimization IMOP1 replaces the second step (2) by the following,

1) Find the fitness f_i of the variable x_i for all $i=1,2,\cdots,n$.

2) Replace randomly the sites with lowest 5% fitness (not just the one with the lowest fitness).

3) If $H(S')<H(S)$ then $S_{best}=S'$, else $S_{best}=S$.

This gives a significantly higher fitness than the previous deterministic one (at the same time) but still it has the previous drawback of falling in local minima. The second immune motivated optimization algorithm IMOP2 which preserves the advantage of the above modification is to define α_i by;

$$\alpha_i = \left(\frac{f_{\max} - f_i}{f_{\max} - f_{\min}} \right)^{\tau}, \tag{13}$$

Where $f_{\max}\left(f_{\min}\right)$ is the maximum (minimum) possible fitness and $\tau > 0$. The second step in the above algorithm IMOP1 is now replaced by the following,

1) Find the fitness f_i of the variable x_i for all $i = 1, 2, \cdots, n$.

2) For all $i = 1, 2, \cdots, n$, if $RND < \alpha_i$ then replace x_i.

3) If $H\left(S'\right) < H\left(S\right)$ then $S_{best} = S'$, else $S_{best} = S$.

where *RND* is a uniformly distributed random number. This algorithm has the following advantages,

1) It has a better chance of avoiding getting stuck in local minima.

2) It does not require the ranking of all fitness at each time step as the modification of Boettcher and Percus requires. This saves significant time especially for large number of sites.

We reconsidered the spin glass problem (that Boettcher studied in his paper [1]) and apply our immune motivated optimization algorithm IMOP2. We graphed the average energies obtained by Boettcher and that obtained by our algorithm for the ±*J* spin glass in $d = 3$ as a function of *τ*. For each $n\left(L^d = 216, 3434, 512, 729\right)$, where *L* is the length of the cube, 10 instances were chosen. For each instance 10 runs were performed stating from different initial conditions at each τ. The results were averaged over the number of runs and over the number of instances. We can see from the figures (from **Figures 1(a)** to **(d)** for $n = 216$, $n = 343$, $n = 512$ and $n = 729$) and from the results of Boettcher that we get better minimum values than them in all case. Also, we can see from **Figures 2** and **3**, that we reach to better minimum values in the same time that Boettcher's takes. This means that our approach is faster than the ordinary EO.

In **Table 1**, we compare the modified EO approximations to the average ground-state energy per spin $e_3\left(n\right)$ of the ±*J* spin glass in $d = 3$ with EO results using Extremal optimization. For each size $n = L^d$ we have studied a large number I of instances. Also shown is the average time t (in seconds) needed for EO to find the presumed ground state. We note that the new results are better and faster than the other three results.

4. Extremal Optimization as a Multiobjective Metaheuristic

Generalizing extremal optimization to multiobjective optimization is done by defining the weighted fitness at

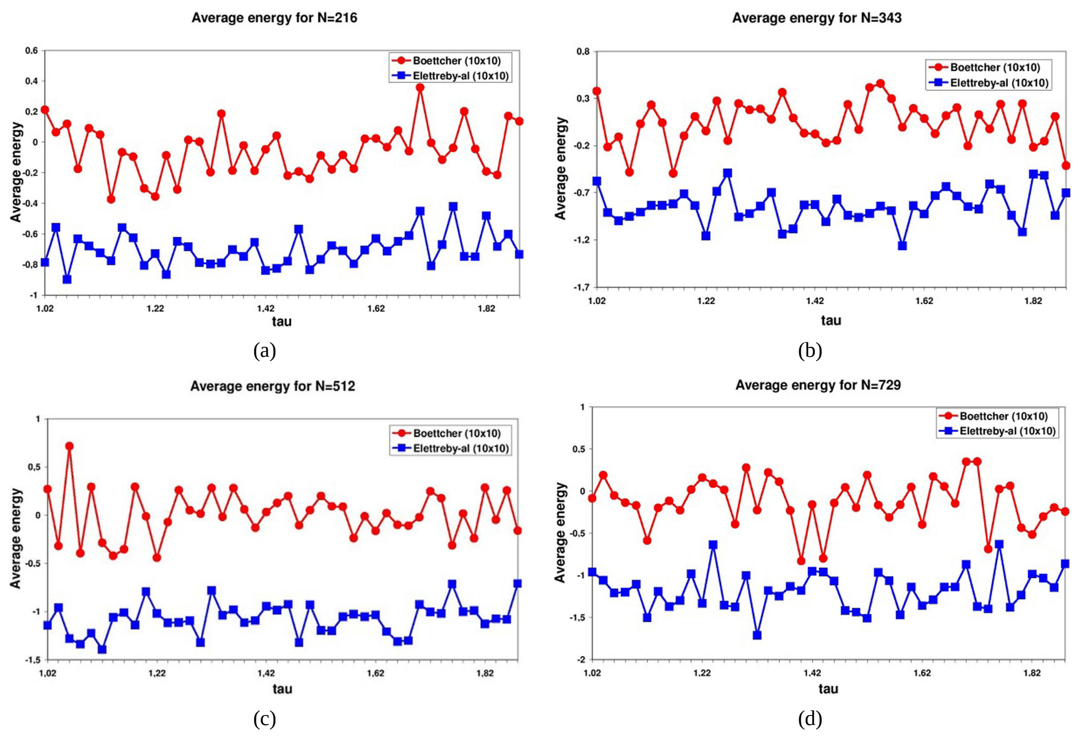

Figure 1. **(a-d) Plot of the average energies obtained by EO (red) and modified EO forthe ±*J* spin glass in *d* = 3 as a function of *τ* for size *n* =216, 343, 512, 729.**

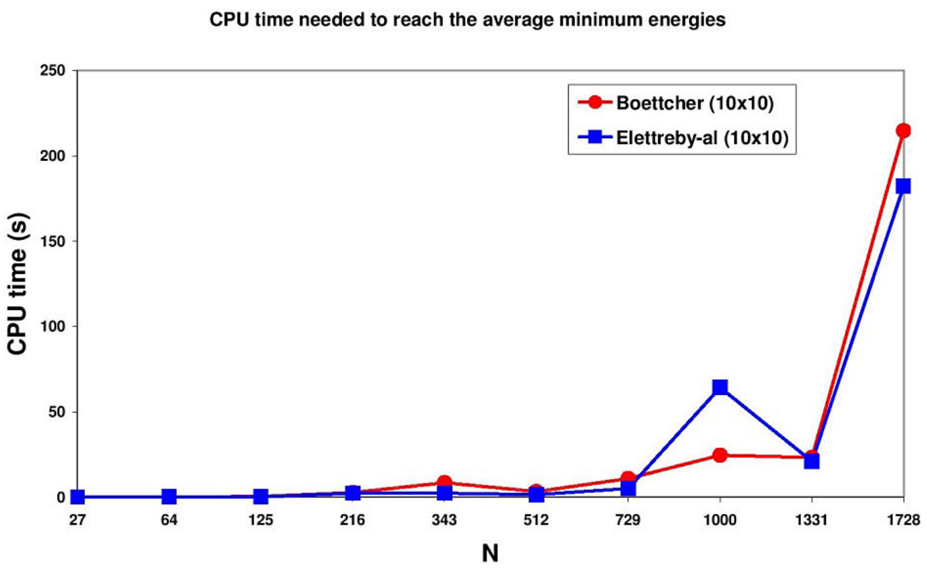

Figure 2. This figure shows the time needed to reach the minimal average energy for each L. The red line is our results using modified EO algorithm and the blue one is the results using.

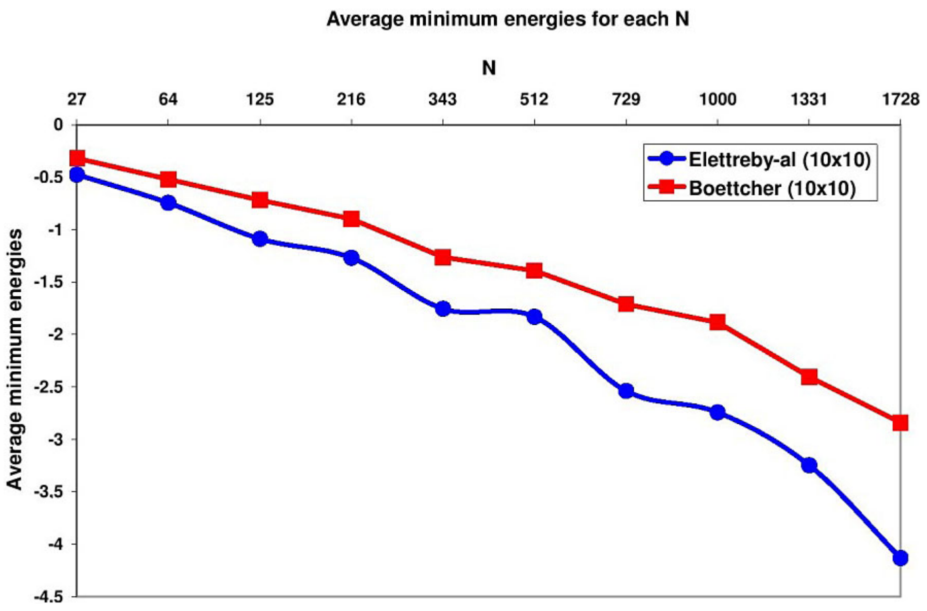

Figure 3. This figure shows the minimal average energy for each L. The red line is our results using modified EO algorithm and the blue one is the results using.

Table 1. Shows a comparison between the modified EO results, EO results.

L	$e_3(n)$	T	EO	t
3	−0.373353	0.05	−0.161658	0.03
4	−0.526250	0.02	−0.227500	0.05
5	−0.861333	0.17	−0.325571	0.049
6	−0.881816	2.35	−0.518475	2.71
7	−1.086920	2.37	−0.563167	8.47
8	−1.474320	1.49	−0.516188	3.3
9	−1.797780	5.11	−0.700000	10.94
10	−2.171540	64.23	−0.867413	24.57
11	−2.319440	20.97	−0.792750	23.31
12	−2.511710	182.22	−1.210750	214.72

each site by,

$$wf_i = \sum_{l=1}^{Q} w_l f_{i_l}, \text{ where } 0 \le w_l \le 1, \sum_{l=1}^{Q} w_l = 1. \quad (14)$$

Then we apply the extremal optimization algorithm.

Now, we comment on extremal optimization as a metaheuristic optimization procedure.

Definition 7. A metaheuristic [17] is an iterative process which guides a heuristic by learning strategies to find a near optimal solution.

Metaheuristics are characterized by:

1) They give approximate solutions.

2) They include mechanisms to avoid being trapped into locally optimal solutions.

3) They are not a specific problem.

4) They include diversified and intensified mechanisms for an efficient search of solutions.

Extremal optimization has all these properties. The intensification and diversification mechanisms are included in the random number in the algorithm IMOP2. If the random number is large, a large number of sites are going to change their state hence diversification occurs. If the random number is small, only a small number of sites will change their state hence intensification occurs.

REFERENCES

[1] S. Boettcher and A. Percus, “Optimization with Extremal Dynamics,” *Physical Review Letters*, Vol. 86, No. 23, 2001, pp. 5211-5214.

[2] Y. Collette and P. Siarry, “Multi-Objective Optimization: Principles and Case Studies,” Springer, New York, 2003.

[3] M. R. Chen and Y. Z. Lu, “A Novel Elitist Multiobjective Optimization Algorithm: Multiobjective Extremal Optimization,” *European Journal of Operational Research*, Vol. 188, No. 3, 2008, pp. 637-651. http://dx.doi.org/10.1016/j.ejor.2007.05.008

[4] M. Ehrgott, “Multi-Criteria Optimization,” Springer, New York, 2005.

[5] N. F. Britton, “Essential Mathematical Biology,” Springer, New York, 2003. http://dx.doi.org/10.1007/978-1-4471-0049-2

[6] R. Blanquero, E. Carrizosa and E. Conde, “Inferring Efficient Weights from Pairwise Comparison Matrices,” *Mathematical Methods of Operations Research*, Vol. 64, No. 2, 2006, pp. 271-284. http://dx.doi.org/10.1007/s00186-006-0077-1

[7] I. Das, “Applicability of Existing Continuous Methods in Determining Pareto Set for a Nonlinear Mixed Integer Multicriteria Optimization Problem,” 8*th AIAA Symposium on Multidiscipline*, *Analysis and Optimization*, Longbeach CA, 2000. http://dx.doi.org/10.2514/6.2000-4894

[8] J. R. Rao and P. Y. Papalambros, “A Non-Linear Programming Continuation Strategy for One Parameter Design Optimization Problems,” In: B. Ravani, Ed., *Advances in Design Automation Montreal*, Vol. 19, No. 2, Que, Canada, 17-21 September 1989, ASME, New York, pp. 77-89.

[9] J. Rakowska, R. T. Haftka and L. T. Watson, “Tracing the Efficient Curve for Multi-Objective Control-Structure Optimization,” *Computing Systems in Engineering*, Vol. 2, No. 6, 1991, pp. 461-471. http://dx.doi.org/10.1016/0956-0521(91)90049-B

[10] I. Das and J. E. Dennis, “Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems,” *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 631-657. http://dx.doi.org/10.1137/S1052623496307510

[11] I. Das, “Nonlinear Multicriteria Optimization and Robust Optimality,” Ph.D., Department of Computational and Applied Mathematics, Rice University, Houston, 1997.

[12] E. Ahmed and M. El-Alem, “Immune Motivated Optimization,” *International Journal of Theoretical Physics*, Vol. 41, No. 5, 2002, pp. 985-990. http://dx.doi.org/10.1023/A:1015705512186

[13] P. Bak and K. Sneppen, “Punctuated Equilibrium and Criticality in a Simple Model of Evolution,” *Physical Review Letters*, Vol. 71, No. 24, 1993, pp. 40834086. http://dx.doi.org/10.1103/PhysRevLett.71.4083

[14] I. Roitt and P. Delves, “Essential Immunology,” Blackwell Publishers, 2001.

[15] D. Head, “Extremal Driving as a Mechanism for Generating Long-Term Memory,” *Journal of Physics A*: *Mathematical and General*, Vol. 33, No. 42, 2000, p. L387. http://dx.doi.org/10.1088/0305-4470/33/42/102

[16] E. Ahmed and M. F. Elettreby, “On Combinatorial Optimization Motivated by Biology,” *Applied Mathematics and Computation*, Vol. 172, No. 1, 2006, p. 4048. http://dx.doi.org/10.1016/j.amc.2005.01.122

[17] C. Blum and A. Roli, “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,” *ACM Computing Survey*, Vol. 35, No. 3, 2003, pp. 268-308.