An Enhanced Annealing Genetic Algorithm for Multi-Objective Optimization Problems

Zhong-Yao Zhu
Dept. of Computer Science & Engineering
Chinese University of Hong Kong
Hong Kong

Kwong-Sak Leung
Dept. of Computer Science & Engineering
Chinese University of Hong Kong
Hong Kong

Abstract

In this paper, we present a new algorithm — an Enhanced Annealing Genetic Algorithm for Multi-Objective Optimization problems (MOPs). The algorithm tackles the MOPs by a new quantitative measurement of the Pareto front coverage quality — Coverage Quotient. We then correspondingly design an energy function, a fitness function and a hybridization framework, and manage to achieve both satisfactory results and guaranteed convergence.

1 Introduction

Many real world decision-making problems involve simultaneous optimization of several incommensurable and often competing objectives. Usually, there is no single optimal solution, but rather a set of alternative solutions. These solutions are known as *Pareto-optimal* solutions, to which no other solutions in the search space are superior in all the objectives.

Often, for a multi-objective optimization problem (MOP), a full perception of all the *Pareto-optimal* solutions would be highly desirable (even imperative) to the decision-makers so that a comprehensive and high quality decision can be made. To achieve this, a practical algorithm should have the capability of simultaneously searching a set of *Pareto-optimal* solutions. Among various candidates, Evolutionary Algorithms (EAs) are particularly suitable for this purpose.

In this paper, we present a new algorithm — an Enhanced Annealing Genetic Algorithm (eAGA) for MOPs. The algorithm tackles the MOPs by introducing a quantitative measurement of the Pareto front coverage quality — Coverage Quotient (CQ). Based on CQ we can derive the energy function in the Simulated Annealing Algorithm (SAA) and the fitness function in the Genetic Algorithm (GA). In the algorithm, a population is interpreted as a state in the search space. The proposed algorithm explores the search space from state to state by means of genetic operations (selection, crossover, and mutation) and converges to a minimal energy state containing only *Pareto optima*. Both satisfactory results and guaranteed convergence can be achieved with the eAGA.

The paper is organized as follows: Section 2 gives a brief introduction to multi-objective optimization problems and an overview of the EA implementation. Our algorithm is detailed in section 3, followed by the theoretical analysis in section 4. The simulation results are presented in section 5. The last section is the conclusion and the future work.

2 Background

In this section, the basic concepts of multi-objective optimization problems are introduced. A brief survey of EAs implementation in MOPs and their strength and weakness are briefed as the motivation of our algorithm.

2.1 Multi-objective Optimization

A general multi-objective optimization problem is to optimize a set of objectives subject to some constrains. Mathematically, a MOP may be stated as in (Rao, 1991):

$$\begin{array}{lll} \text{Min/Max} & f_i(x) & i = 1, 2, ..., N \\ \text{Subject to} & g_j(x) \leq 0 & j = 1, 2, ..., J \\ & h_k(x) = 0 & k = 1, 2, ..., K \end{array}$$

where alternative x is a p-dimensional vector having p design or decision variables, $f_i(x)$ $(i = 1, 2, ..., N)$ are the objective functions, and $g_j(x)$ $(j = 1, 2, , ..., J)$ and $h_k(x)$ $(k = 1, 2, ..., K)$ are the constraint functions.

Solutions to the MOP are mathematically characterized in terms of the non-dominated alternatives. Let $F(x) = (f_1(x), f_2(x), ..., f_N(x))$ represent the objective vector. In a minimization problem, for instance, alternative $x^{(1)}$ is said to be partially less than alternative $x^{(2)}$ (denoted by $x^{(1)} \prec x^{(2)}$), if no component of $F(x^{(2)})$ is less than that of $F(x^{(1)})$ and there is at least one component of $F(x^{(2)})$ is strictly greater than that of $F(x^{(1)})$. If $x^{(1)}$ is partially less than $x^{(2)}$, alternative $x^{(1)}$ is said to *dominate* $x^{(2)}$. Any alternative that is not dominated by others is said to be a non-dominated point. Any non-dominated points associated with the MOP are called optimal solutions (more precisely, *Pareto-optimal* solutions or non-dominated solutions) of the MOP. Usually, the image of all *Pareto optimal* solutions in objective space is called the Pareto front.

2.2 Multi-objective Evolutionary Algorithms

According to Zitzler (Zitzler, 1998), the current EA implementation can be categorized as plain aggregation approaches, population-based non-Pareto approaches and Pareto-based approaches.

Plain aggregation approaches combine the objectives into a higher scalar function which is used for fitness calculation; they produce one single solution and require profound domain knowledge which is often not available. Population-based non-Pareto approaches, however, are able to evolve multiple non-dominated solutions in parallel; thereby, the population is mostly monitored for *Pareto optima*. However in contrast to the Pareto-based approaches, they do not make direct use of the concept of Pareto dominance. Such designed algorithms are effective to some extent, but they usually suffer from prematuring to some special areas (Schaffer, 1985). Pareto-based EAs compare solutions according to the $\prec$ relation in order to determine the reproduction probability of each individual. This strategy can meet our requirement well. Nevertheless identifying the $\prec$ relation among individuals usually brings higher running time consumption (Srinivas, 1994).

More comprehensive overviews of EAs in MOPs can be found in (Zitzler, 1999) (Carlos, 1999),

3 The Enhanced Annealing Genetic Algorithm

In this section, we present the new algorithm — an Enhanced Annealing Genetic Algorithm (eAGA) for MOPs. As most of non-dominated sorting GAs, we design our algorithm based on two considerations: 1) the non-dominance of solutions, and 2) the coverage of the Pareto front.

As suggested by its name, the new algorithm is a hybridization of the Simulated Annealing Algorithm (SAA) (Laarhoren, 1989) and the Genetic Algorithm (GA) (Goldberg, 1989). The algorithm tackles the MOPs by introducing a quantitative measurement of the Pareto front coverage quality — Coverage Quotient (CQ). The energy function in SAA and the fitness function in GA are derived from this measurement correspondingly. In the algorithm, a population is interpreted as a state in the search space. The neighborhood relationship between states is defined in terms of their individual discrepancy. The proposed algorithm explores the search space from state to state by means of genetic operations (selection, crossover, and mutation). Ultimately, the exploration converges to a minimal energy state which can be proved to contain only *Pareto optima*. The details are given in the following subsections.

3.1 Uniform Expression of Population

For clarity and simplicity, we define an alternative representation of populations. In this representation, all populations that differ from each other only in the order of individuals are treated as the same and expressed in a unique format.

Suppose the individuals are all expressed as L bits binary strings. Then the individual space is given by $\{0,1\}^L$ and can be isomorphic to the finite-state space:

$$\{0,\ 1, \cdots,\ i, \cdots, 2^L - 1\}$$

where i $(0 \leq i \leq 2^L - 1)$ is the index of individuals (Reeves, 1993). From this point of view, a population Φ with size n can then be represented as $\Phi = (\phi_0\ \phi_1\ \cdots\ \phi_i \cdots\ \phi_{2^L-1})$ where ϕ_i is the number of times of individual i appearing in the population Φ. It is clear that there are at most n nonzero elements in Φ and $\sum_{i=0}^{2^L-1} \phi_i = n$.

The set consisting of all the populations with size n is denoted as S_n. Each population is interpreted as a state in S_n. The neighbors of a state $\Phi = (\phi_0\ \phi_1\ \cdots\ \phi_{2^L-1})$ are defined as all the populations $\Phi' = (\phi'_0\ \phi'_1\ \cdots\ \phi'_{2^L-1})$ which satisfy: $\sum_{i=0}^{2^L-1} |\phi_i - \phi'_i| = 2$. All the neighbors of Φ are denoted as $N(\Phi)$.

3.2 The Pareto Front Coverage Quality Measurement

The Coverage Quotient (CQ) gives a quantitative measurement of the Pareto front coverage quality. It is based on the idea that a good coverage of the Pareto front by a population Φ should minimize the potential that a non-Pareto point is wrongly judged as a Pareto point when compared against Φ. The formal definition is given as follows:

Definition. 1: Given F is a MOP and N is the number of objectives:.

- Let $Pmin_i, Pmax_i$ be the minimum and maximum of all the Pareto points in objective i $(1 \leq i \leq N)$, the hyper-cube

  $$U = [Pmin_1, Pmax_1] \times ... \times [Pmin_N, Pmax_N],$$

  which contains all the Pareto points, is defined as P-Cube of F.

- Given a population $\Phi = \{d_1, d_2, ..., d_n\}$,

  $$\begin{array}{ll} D(d_i) & = [F(d_i)_1, \infty] \times ... \times [F(d_i)_N, \infty], \text{ and} \\ D_n(\Phi) & = D(d_1) \cup D(d_2) \cup ... \cup D(d_n) \end{array} \quad (1)$$

  are defined as the dominating region of individual d_i $(1 \leq i \leq n)$ and dominating region of population Φ respectively.

- The Coverage Quotient (CQ) of Φ is defined as

  $$\mathrm{CQ}(\Phi) = |U| - |U \cap D_n(\Phi)|. \quad (2)$$

- The population Φ is said to be an optimal coverage of the Pareto front if

  $$\mathrm{CQ}(\Phi) = \min_{\Phi'}\{\mathrm{CQ}(\Phi') |\ \Phi' \in S_n\}.$$

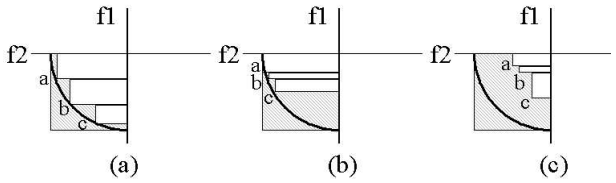

Figure 1: The Coverage Quotient in 3 cases: (a) evenly distributed Pareto points, (b) crowded Pareto points, and (c) non-Pareto points

A graphic illustration of Def.1 is given in Figure 1. Assuming the curve is the Pareto front, the shadowed regions represent the Coverage Quotient in 3 cases respectively.

In Def.1, when we identify an individual's non-dominance by comparing it with population Φ, it can be correctly justified only if it is located inside $D_n(\Phi)$. Thus we can reduce the risk of misjudgment by maximizing the dominating region $D_n(\Phi)$ or equivalently, minimizing the Coverage Quotient $CQ(\Phi)$. However, a direct calculation of CQ from Formulas (1) and (2) is difficult for high dimension objectives. Thus in Def.2, an alternative definition of the Coverage Quotient is presented:

Definition 2: Given F, N, Φ, U and $D(\cdot)$ as in Def.1:

- Let $P(\Phi) = \{d^1, d^2, .., d^{n_p}\} \subset \Phi$ $(n_p \leq n)$ be all the individuals which satisfy;
  1. $d^i \neq d^j$ $(1 \leq i, j \leq n_p)$;
  2. $\forall\ d^i \in P(\Phi),\ \nexists\ d_j \in \Phi$, s.t. $d_j \prec d^i$ $(1 \leq i \leq n_p,\ 1 \leq j \leq n)$.

  The Coverage Quotient (CQ) of Φ is defined as

  $$\mathrm{CQ}(\Phi) = -\sum_{d^i \neq d^j \in P(\Phi)} |(D(d^i) \cup D(d^j)) \cap U|.$$

In Def.2, the calculation of CQ just needs n^2 running time complexity. Meanwhile, with such defined CQ, the proposed algorithm can be guaranteed to converge to a population consisting of only distinct Pareto individuals as proved in section 4.

Correspondingly, we define the energy function of a state Φ as:

$$E(\Phi) = \mathrm{CQ}(\Phi),$$

and the fitness function of an individual $d_i \in \Phi$ $(1 \leq i \leq n)$ as:

$$fit(d_i, \Phi) = e^{\frac{CQ(\Phi \backslash d_i) - MinCQ(\Phi)}{MaxCQ(\Phi) - MinCQ(\Phi)}},$$

in which:

$$\begin{array}{ll} MinCQ(\Phi) & = \min_{d \in \Phi}\{CQ(\Phi \backslash d)\} \\ MaxCQ(\Phi) & = \max_{d \in \Phi}\{CQ(\Phi \backslash d)\} \end{array}$$

3.3 State Transformation (ST) Operation

In the eAGA, the ST operation is the primary search technique. A GA-like evolutionary process is adopted to form the backbone of the ST operation. Furthermore, an additional acceptance procedure is employed to guarantee the global convergence of our algorithm. The framework of this operation is given as follows: For a population $\Phi = \{\phi_0\ \phi_1 ..... \phi_{2^L-1}\}$:

1. Perform Roulette selection on Φ to choose two individuals d_1 and d_2 as the parents.
2. Perform crossover and mutation on d_1 and d_2 to produce two children d_1' and d_2'.

3. Randomly select a parent $d \in \{d_1, d_2\}$ and a child $d' \in \{d'_1, d'_2\}$.
4. Replace d in Φ with d' to form a new population Φ'.
5. Accept Φ' to be the new state at probability $\min\{1, \frac{P(d,\Phi')}{P(d',\Phi)}\}$

in which, $P(k, \Phi)$ is the probability at which individual k ($k \in \{0, 2^L - 1\}$) is produced by performing selection, crossover, and mutation on population Φ. It is calculated as follows:

Given the indices of the nonzero elements in Φ are

$$0 \le p_1 < p_2 < p_3 < ....p_{n'} \le 2^L - 1 \quad (n' \le n),$$

then

$$P(k, \Phi) = \widetilde{F}^T(\Phi)\widetilde{R}(k)\widetilde{F}(\Phi), \tag{3}$$

where $\widetilde{F}(\Phi) = (\widetilde{F}_0(\Phi), ..., \widetilde{F}_{n'}(\Phi)) \in R^{n'}$ with

$$\widetilde{F}_i(\Phi) = \frac{fit(p_i, \Phi)\phi_{p_i}}{\sum_{j=1}^{n'} fit(p_j, \Phi)\phi_{p_j}} \qquad i = 1, ..., n'. \tag{4}$$

and $\widetilde{R}(k) = (\widetilde{R}(k)_{i,j}) = (r_{p_i,p_j}(k)) \in R^{n' \times n'}$.

$r_{i,j}(k)$ $(i, j = p_1, \ldots, p_{n'})$ is the probability at which individual k is produced from the crossover and mutation of the individuals i and j. It is computed by means of the transformation:

$$r_{i,j}(k) = r_{i\oplus k, j\oplus k}(0) \tag{5}$$

and

$$\begin{aligned} r_{i,j}(0) \quad &= \tfrac{1}{2}\sum_{k=0}^{L} \tfrac{R_c}{L+1}((1-R_m)^{L-H(m_1(i,j,k))}R_m^{H(m_1(i,j,k))} \\ &+(1-R_m)^{L-H(m_2(i,j,k))}R_m^{H(m_2(i,j,k))}) \\ &+\tfrac{1}{2}(1-R_c)((1-R_m)^{L-H(i)}R_m^{H(i)} \\ &+(1-R_m)^{L-H(j)}R_m^{H(j)}) \end{aligned}$$

where

$$\begin{aligned} m_1(i,j,k) &= i \otimes \overline{(2^k-1)} \oplus j \otimes \overline{(2^k-1)}, \\ m_2(i,j,k) &= i \otimes \overline{(2^k-1)} \oplus j \otimes (2^k-1), \quad 0 \le k \le L, \end{aligned}$$

with R_c being the crossover rate; R_m being the mutation rate; $\oplus$ being the *exclusive-or* operator; $\otimes$ being the *logical-and* operator; $-$ being the *inverse* operator; and H(m) being the Hamming distance between individuals m and 0 (Vose, 1991) (Vose, 1995).

3.4 P-Cube Approximation

Notice that identifying P-Cube requires pre-knowledge of all the *Pareto optima*, which is usually unavailable in most cases. In the eAGA, a dynamic estimation approach is adopted which can approximate the P-Cube as the algorithm proceeds. In the algorithm, two arrays $Bmin_i$ and $Bmax_i$ and $2N$ binary strings $Imin_i$ and $Imax_i$ $(1 \le i \le N)$ are maintained. The $Bmin_i$ and $Bmax_i$ record the minimal and maximal values in the ith objective of all the Pareto points ever found and the $Imin_i$ and $Imax_i$ $(1 \le i \le N)$ record the individuals which attain $Bmin_i$ and $Bmax_i$ in the ith objective respectively. For $Imin_i/Imax_i$, it is replaced by a new Pareto point d iff

- $d \prec Imin_i/Imax_i$; or
- d and $Imin_i/Imax_i$ are non-dominated to each other, but $F(d)_i < Bmin_i/F(d)_i > Bmax_i$.

(Note: / denotes or).
Once an $Imin_i/Imax_i$ is modified, $Bmax$ and $Bmin$ must be updated correspondingly by:

$$\begin{array}{lll} Bmin_i & = \min & \{F(Imin_k)_i, F(Imax_k)_i |\ k = 1, ..., N\} \\ Bmax_i & = \max & \{F(Imin_k)_i, F(Imax_k)_i |\ k = 1, ..., N\} \\ & & (1 \le i \le N.) \end{array} \tag{6}$$

It is proved in section 4 that this estimation can approximate and eventually converge to the P-Cube as the algorithm progresses.

3.5 Algorithm Framework

The overall framework of our algorithm is given as follows:

1. Initialization
   - **1.1** Select an initial population Φ
   - **1.2** Select an initial temperature T
   - **1.3** Select an annealing function $S(T) = \alpha \cdot T$
   - **1.4** Perform non-dominated sorting on Φ and fill the $Bmin$, $Bmax$, $Imin$ and $Imax$
2. State Transformation
   - **2.1** Perform Roulette selection on Φ to choose two individuals d_1 and d_2 as the parents.
   - **2.2** Perform crossover and mutation on d_1 and d_2 to produce two children d'_1 and d'_2.
   - **2.3** Randomly select a parent $d \in \{d_1, d_2\}$ and a child $d' \in \{d'_1, d'_2\}$.
   - **2.4** Replace d in Φ with d' to form a new population Φ'.
   - **2.5** Update the $Bmin$, $Bmax$, $Imin$ and $Imax$ with individual d'.
   - **2.6** Accept Φ' to be the new state at probability

     $$\min(1, \frac{P(d,\Phi')}{P(d',\Phi)})\min\{1, e\frac{E(\Phi) - E(\Phi')}{T}\}$$

   - **2.7** If Φ' is accepted, then $\Phi := \Phi'$,
3. $T := S(T)$

4. If stop criterion is not met, then goto 2.1

5. Exit.

Note: α is an annealing parameter which satisfies $0 < \alpha < 1$.

4 Convergence Analysis

In this section, the theoretical analysis of the proposed algorithm is presented.

Lemma 1: For MOP F, suppose $Bmin_i^g$ and $Bmax_i^g$ $(1 \le i \le N)$ are minimal and maximal values of all Pareto points in the ith objective; $Imin_i^g$ and $Imax_i^g \in \{0,1\}^L$ $(1 \le i \le N)$ are the individuals which attain $Bmin_i^g$ and $Bmax_i^g$ in the ith objective respectively. Given mutation rate R_m is nonzero, the estimations $Bmin_i$ and $Bmax_i$ will converge to $Bmin_i^g$ and $Bmax_i^g$ $(1 \le i \le N)$ as the algorithm progresses.

Proof: We prove the Lemma in two steps:

1. We will prove that $Imin_i^g$ $(1 \le i \le N)$ will be found as the algorithm proceeds. Let p_i^n be the probability that $Imin_i^g$ is not found in iteration n. It is obvious that $p_i^n \le 1 - R_m^L < 1$. Then the probability P_i^n that $Imin_i^g$ $(1 \le i \le N)$ is not found in the first n iteration is:

$$P_i^n = \prod_{j=1}^{n} p_i^j < (1 - R_m)^n \longrightarrow 0 \ (n \to \infty).$$

2. We will prove that once an $Imin_i^g$ $(1 \le i \le N)$ is found, it cannot be replaced by other individuals. Suppose $Imin_i^g$ is found, then an individual d will replace it iff

   Case 1: $d \prec Imin_i^g$; or

   Case 2: d and $Imin_i^g$ are non-dominated to each other, but $F(d)_i < Bmin_i^g$.

   Because $Imin_i$ is a Pareto point, Case 1) would not happen. If Case 2) happens, this means that there exists another non-dominated individual d' satisfies: $F(d')_i < Bmin_i^g$ which contradicts with the assumption.

The $Imax_i^g$ $(1 \le i \le N)$ case can be proved in the same way.

Combining 1, 2 and Formula (6), we finish the proof of Lemma 1.

Lemma 2: Given population $\Phi = \{d_1, d_2, ..., d_n\}$ is the optimal coverage in Def.1, then d_i $(1 \le i \le n)$ are all Pareto individuals. (Suppose there exist more than n Pareto individuals).

Proof: (By contradiction). Suppose d_k $(1 \le k \le n)$ is not a Pareto individual. Then there exists a Pareto individual d which dominates d_k.

1. If $d \in \Phi$, then replace d_k with a Pareto individual $d' \notin \Phi$ to form a new population Φ'. It is easy to see that:
$$E(\Phi) > E(\Phi'),$$
which contradicts with the definition of optimal coverage.

2. If $d \notin \Phi$, then replace d_k with d to form a new population Φ'. It is the same as in 1 that:
$$E(\Phi) > E(\Phi'),$$
which contradicts with the definition of optimal coverage.

This finishes our proof of Lemma 2.

Lemma 3: Given population $\Phi = \{d_1, d_2, ..., d_n\}$ is the optimal coverage in Def.2, then d_i $(1 \le i \le n)$ are all distinct Pareto individuals. (Suppose there exist more than n Pareto individuals).

Proof: The proof of Lemma 3 is similar to that of Lemma 2 with minor modification.

Theorem : As T $\to$ 0, the eAGA converges to a population consisting of only distinct Pareto individuals.

Proof: It is clear that for any fixed temperature T, as the population evolves, the eAGA defines a homogeneous finite state population Markov chain. Let M be the number of total populations. Then the probability transition matrix of the Markov chain can be expressed as (Iosifescu, 1980): For states Φ^i and Φ^j,

$$P_{i,j}(T) = \begin{cases} G_{i,j}(T)A_{i,j}(T) & j \ne i \\ 1 - \sum\limits_{l=1, l\ne i}^{M} G_{i,l}(T)A_{i,l}(T) & j = i \end{cases}. \quad (7)$$

Here

$$G_{i,j}(T) = \begin{cases} P(\Phi^i, \Phi^j) min\{1, \frac{P(d,\Phi^i)}{P(d',\Phi^j)}\} & \Phi^j \in N(\Phi^i) \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

and $$A_{i,j}(T) = min\{1, e^{(\frac{E(\Phi^i) - E(\Phi^j)}{T})}\}, \quad (9)$$

in which: $d \in \Phi^j$ and $d \notin \Phi^i$; $d' \in \Phi^i$ and $d' \notin \Phi^j$.

From Formula 8, we can see that $G_{i,j}(T) = G_{j,i}(T)$.

From Formula 9, it follows that whenever $E(\Phi^i) \le E(\Phi^j) \le E(\Phi^k)$,

$$A_{i,k}(T) = min\{1, e^{(\frac{E(\Phi^i) - E(\Phi^k)}{T})}\} = A_{i,j}(T)A_{j,k}(T)$$

and whenever $E(\Phi^i) \le E(\Phi^j)$, $0 \le A_{i,j}(T) \le 1$ and

$$\lim_{T\to 0} e^{(\frac{E(\Phi^i) - E(\Phi^j)}{T})} = 0.$$

Accordingly, $\lim_{T\to 0} A_{i,j}(T) = 0$. By the Folklore's lemma (Laarhoren, 1989), the stationary distribution $q(T)$ of the Markov chain exists and satisfies

$$\lim_{T\to 0} q_i(T) = \frac{1}{|S_{opt}|}\chi_{S_{opt}}(i), \tag{10}$$

in which $S_{opt} = \{\Phi | E(\Phi) \text{ attains the minimum}\}$, and

$$\chi_{S_{opt}}(i) = \begin{cases} 1 & i \in S_{opt} \\ 0 & \text{otherwise.} \end{cases}$$

from Formula 10, $\lim_{T\to 0} \Phi(T) \in S_{opt}$ follows.

This finishes the proof of the Theorem.

5 Simulation

In this section, we show the effectiveness and efficiency of the eAGA in a set of simulations. The test problems are given as follows:

$$\begin{aligned} Min.\ T_1(x) &= (f_1(x), f_2(x)) \\ f_1(x) &= 1 - exp(-\textstyle\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2) \\ f_1(x) &= 1 - exp(-\textstyle\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2) \\ \text{where} &\quad -4 \le x_i \le 4,\ n = 3 \end{aligned}$$

$$\begin{aligned} Min.\ T_2(x) &= (f_1(x), f_2(x)) \\ f_1(x) &= x_1 \\ f_2(x) &= g(x)h(f_1(x), g(x)) \\ g(x) &= 1 + 9 \times \textstyle\sum_{i=2}^{n} x_i/(n-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \\ \text{where} &\quad 0 \le x_i \le 1,\ n = 30 \end{aligned}$$

$$\begin{aligned} Min.\ T_3(x) &= (f_1(x), f_2(x)) \\ f_1(x) &= x_1 \\ f_2(x) &= g(x)h(f_1(x), g(x)) \\ g(x) &= 1 + 9 \times \textstyle\sum_{i=2}^{n} x_i/(n-1) \\ h(f_1, g) &= 1 - (f_1/g)^2 \\ \text{where} &\quad 0 \le x_i \le 1,\ n = 30 \end{aligned}$$

$$\begin{aligned} Min.\ T_4(x) &= (f_1(x), f_2(x)) \\ f_1(x) &= x_1 \\ f_2(x) &= g(x)h(f_1(x), g(x)) \\ g(x) &= 1 + 9 \times \textstyle\sum_{i=2}^{n} x_i/(n-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_i) \\ \text{where} &\quad 0 \le x_i \le 1,\ n = 30 \end{aligned}$$

The simulations are carried out to verify: 1) the efficiency of the eAGA, and 2) the effectiveness of the Coverage Quotient in both definitions. For these purposes, the execution results of eAGA-I, eAGA-II, SPEA (Zitzler, 1999) and NSGA (Srinivas, 1994) are compared in terms of: 1) the non-dominance of resulting solutions, and 2) the coverage of the Pareto front, which are two main considerations in most of the current EA-based MOP algorithms. (**Note**: eAGA-I and eAGA-II are abbreviations for the eAGA with CQ defined by Def.1 and Def.2 respectively).

To make the comparisons fair, the algorithms are executed 30 times on each of the test problems. In each run, all algorithms begin from the same initial population. The final results are taken as the average of these 30 runs. All algorithms are executed for the same length of time.

Independent of the algorithms and the test problems, each simulation is carried out using the following parameters:

Population size	:	100
Crossover rate	:	0.8
Mutation rate	:	0.01
Individual length	:	12
Niching parameters σ_{share}	:	0.48862
Elitist population size	:	100

Particularly, in the eAGA we take:

Initial Temperature T	:	1000
Annealing Parameter α	:	0.97

In Figures 2 to 5, the Pareto fronts achieved by the different algorithms are displayed.

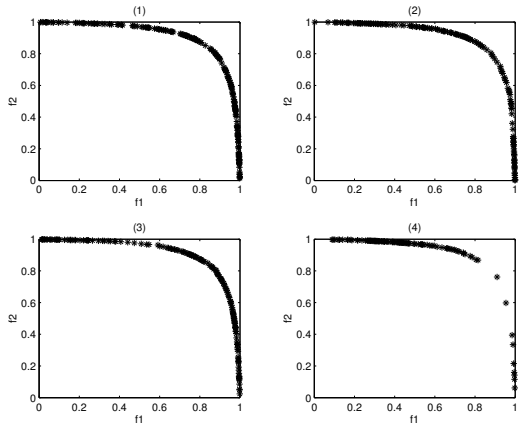

Figure 2: Test function $T1$

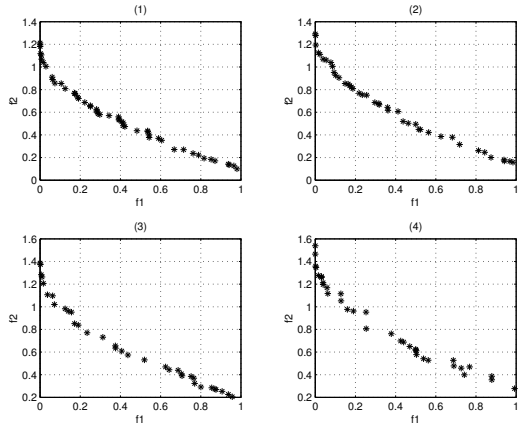

Figure 3: Test function $T2$

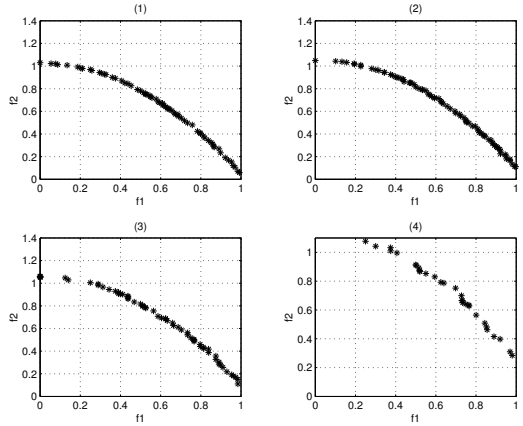

Figure 4: Test function $T3$

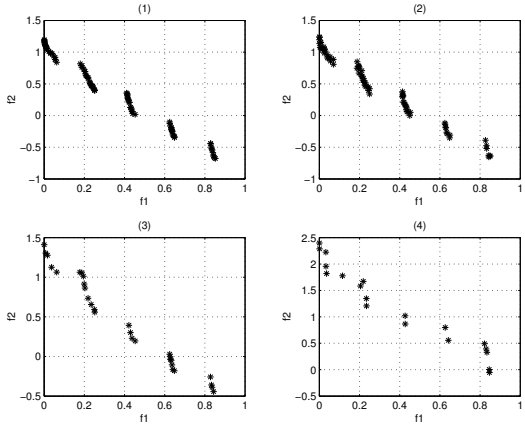

Figure 5: Test function $T4$
(1): eAGA-I (2): eAGA-II (3): SPEA (4): NSGA

It can be observed from the figures that eAGA-I has the best performance in all the four test problems. In T2, T3, and T4, eAGA-II can yield similar performance as eAGA-I. However, in T1, SPEA slightly outperforms eAGA-II which misses some *Pareto optima* close to the boundaries.

In measuring the non-dominance of results, we adopt a quantitative metric, *C metric*, presented in (Zitzler, 1999). To identify the non-dominance of a solution, we need to compare it with all the other individuals in the search space. This is definitely unrealistic. Thus, instead of identifying the absolute non-dominance of a solution, *C metric* compares the non-dominance relationship between the outcomes of two algorithms. Given a pair of algorithms A_1 and A_2, *C metric* estimates the non-dominance of A_1 by calculating the percentage of the solutions of A_1 which are dominated by those of A_2. Mathematically, *C metric* is defined as follows:

Let $X', X'' \subseteq X$ be two sets of decision vectors. The function C maps the ordered pair (X', X'') to the interval [0,1]:

$$C(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X' : a' \prec a''\}|}{|X''|}.$$

The value $C(X', X'') = 1$ means that all solutions in X'' are dominated by or equal to solutions in X'. The opposite, $C(X', X'') = 0$ represents the situation that none of the solutions in X'' are covered by the set X'. Note that both $C(X', X'')$ and $C(X'', X')$ have to be considered, since $C(X', X'')$ is not necessarily equal to 1-$C(X'', X')$.

The comparison results of *C metric* are given in Table 1. For each ordered algorithm pair, there are 30 C values according to the 30 runs performed. Each C value is computed on the basis of the non-dominated sets achieved by the pair of algorithms with the same initial population. The final result is taken as the average of these 30 C values.

Table 1: Comparison of $C(X', X'')$ and $C(X'', X')$, in which X'' is the outcomes of eAGA-I and X' is the outcomes of eAGA-II, SPEA, and NSGA respectively.

$C(x', x'')$	$T1$	$T2$	$T3$	$T4$
eAGA-II	0.1791	0.3492	0.2646	0.4751
SPEA	0.2388	0.1384	0.4101	0.2214
NSGA	0.1194	0.0776	0.1124	0.1041

$C(x'', x')$	$T1$	$T2$	$T3$	$T4$
eAGA-II	0.3024	0.7152	0.5490	0.6964
SPEA	0.2928	0.8412	0.4483	0.7108
NSGA	0.7137	0.9927	0.8334	0.9174

In measuring the coverage of the Pareto front, we adopted the metric presented by (Deb, 2000). This metric is based on the consecutive distances among the non-dominated solutions. The non-dominated solutions are compared with a uniform distribution and the deviation is computed as follows: Given a set of non-dominated solutions P,

$$\Delta = \sum_{i=1}^{|P|} \frac{|d_i - \overline{d}|}{|P|},$$

in which, d_i is the Euclidean distance between two consecutive solutions in P in the phenotype space and $\overline{d}$ is the average of all the d_is. In order to ensure that this calculation takes into account of the spread of solutions in the entire region of the Pareto front, the boundary solutions are included in P. In our implementation, the boundary solutions are the individuals which attend minimum in at least one objective function.

The deviation measure Δ of these consecutive distances is then calculated for each run. An average of these deviations over 30 runs is calculated as the measure ($\overline{\Delta}$) for comparing different algorithms. Thus, it is clear that an algorithm having a smaller ($\overline{\Delta}$) is better, in terms of its ability to widely and evenly spread solutions in the Pareto front.

Table 2 shows the average deviation, ($\overline{\Delta}$) in all the test problems.

Table 2: Comparison of average deviation $\overline{\Delta}$ obtained using eAGA-I, eAGA-II, SPEA, and NSGA.

	$T1$	$T2$	$T3$	$T4$
eAGA-I	0.0064	0.0175	0.0093	0.0222
eAGA-II	0.0072	0.0211	0.0101	0.0338
SPEA	0.0069	0.0261	0.0144	0.0482
NSGA	0.0174	0.0318	0.0295	0.0765

The quantitative comparison in Tables 1 and 2 conforms with our observation in Figures 2 to 5. In all the four test problems, eAGA-I is observed to have the best performance in both the non-dominance of solutions and the coverage of the Pareto front. In T2 and T4, the results of eAGA-I can cover more than 70% of those of SPEA. Nevertheless, the results of SPEA can only cover less than 23% of the those of eAGA-I. Similar performances are yielded by eAGA-I and SPEA in T2 and T3. But the results of eAGA-I have much more even distribution along the Pareto front as shown in Table 2. As eAGA-II, in T2 and T3, eAGA-I outperforms it by covering more than 53% of its results. Meanwhile, it can only cover less than 35% of those of eAGA-I. In T1 and T4, eAGA-I still has better performance, even the superiority is not so remarkable. In measuring the coverage of the Pareto front, eAGA-I and eAGA-II outperform SPEA and NSGA in most of the cases. However, in T1 and T4, eAGA-II fails to find the *Pareto optima* in the regions near the boundaries. We believe that this failure is caused by the limitation of Def.2 which assigns less reproduction potential to the individuals in these regions. We acknowledge the existence of such limitation and will focus our attention to improve this weakness in the future work.

6 Conclusion

In this paper, we have presented an Enhanced Annealing Genetic Algorithm (eAGA) for Multi-Objective optimization problems. We have also proved its convergence. On four difficult test problems borrowed from the literatures, it is found that the proposed eAGA-I and eAGA-II outperform SPEA and NSGA — two well known multi-objective EAs in the explicit goals of the non-dominance of the solutions and the coverage of the Pareto front. With the properties of high effectiveness and superior performance, the eAGA should find increasing attention and applications in the near future.

Acknowledgments

This research is partially supported by a Hong Kong Government RGC Earmarked Grant, Ref. No. CUHK 4212/01E.

References

Carlos, A. C. (1999), A Comprehensive Survey of Evolutionary-based Multi-objective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269-308.

Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T (2000), A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. *KanGAL Reports* No. 200001, Indian Institute of Technology Kanpur.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. MA: Addison-wesley.

Iosifescu, M. (1980), *Finite Markov Processes and Their Application*, New York: John Wiley & Sons.

Laarhoren, P. J. M. Van., & Aarts, E. H. L. (1989). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers.

Rao, S. S., (1991), *Optimization Theory and Application*, New Delhi: Wiley Eastern Limited.

Reeves, C. R. (1993). *Modern Heuristic Techniques for Combinatorial Problem*. Halsted Press, John Wiley & Sons.

Schaffer, J.D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In J.J. Grefenstette (Eds.), *Genetic algorithms and their applications: Proceedings of the First International Conference on Genetic Algorithms*, 93-100.

Srinivas, N., & Deb, K. (1994). Multi-objective Optimization using Non-dominated Sorting in Genetic Algorithms. *Evolutionary Computation*. 2(3):221-248.

Vose, M. D.,& Liepins, G. E. (1991), Punctured equilibria in genetic search, *Complex system*, Vol.5:31-44.

Vose, M. D. (1995), Modeling simple genetic algorithms. *Evolutionary computation*, 33(4):453–472.

Zitzler, E., & Thiele, L. (1998), Multi-objective Optimization Using Evolutionary Algorithm — A Comparative Case Study. *Parallel Problem Solving from Nature V*:292-301, Netherlands: Springer.

Zitzler, E., & Thiele, L. (1999), Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *Evolutionary Computation*, 3(4): 257-271.