A Theoretical and Empirical Study on Unbiased Boundary-extended Crossover for Real-valued Representation

Yourim Yoon[1], Yong-Hyuk Kim[2], Alberto Moraglio[3], and Byung-Ro Moon[1]
[1] School of Computer Science & Engineering, Seoul National University
599 Gwanak-ro, Gwanak-gu, Seoul, 151-744, Korea
Email: {yryoon, moon}@soar.snu.ac.kr
[2] Department of Computer Science & Engineering, Kwangwoon University
20 Kwangwoon-ro, Nowon-gu, Seoul, 139-701, Korea
Email: yhdfly@kw.ac.kr
[3] School of Computing, University of Kent, UK
Email: A.Moraglio@kent.ac.uk

May 24, 2011

Abstract

We present a new crossover operator for real-coded genetic algorithms employing a novel methodology to remove the inherent bias of pre-existing crossover operators. This is done by transforming the topology of the hyper-rectangular real space by gluing opposite boundaries and designing a boundary extension method for making the fitness function smooth at the glued boundary. We show the advantages of the proposed crossover by comparing its performance with those of existing ones on test functions that are commonly used in the literature, and a nonlinear regression on a real-world dataset.

Keywords: Real-coded representation, genetic algorithms, crossover bias, quotient metric space.

1 Introduction

Recently many studies on evolutionary algorithms using real-coded encoding have been done. They include ant colony optimization [71], artificial bee colony algorithm [1, 40], evolution strategies (ES) [7], differential evolution [13, 14, 48, 49, 55, 60, 70, 73, 86], particle swarm optimization[10, 33, 39, 47, 50], and so on. In particular, in the field of ES, we can find many studies based on self-adaptive techniques [8, 26, 35, 36, 37, 41, 44, 45, 46, 54, 83].

Many researchers also have concentrated on using real-valued genes in genetic algorithms (GAs), as in [69]. It is reported that, for some problems, real-coded representation and associated techniques outperform conventional binary representation [23, 29, 38, 53, 61, 62, 75, 84]. Several theoretical studies of real-coded GAs have also been performed [24, 30, 42, 65, 66]. The crossover operator is one of the most important operators in GAs and the *genetic repair hypothesis* [7] is introduced to explain why crossover is useful in real-coded evolutionary algorithms.

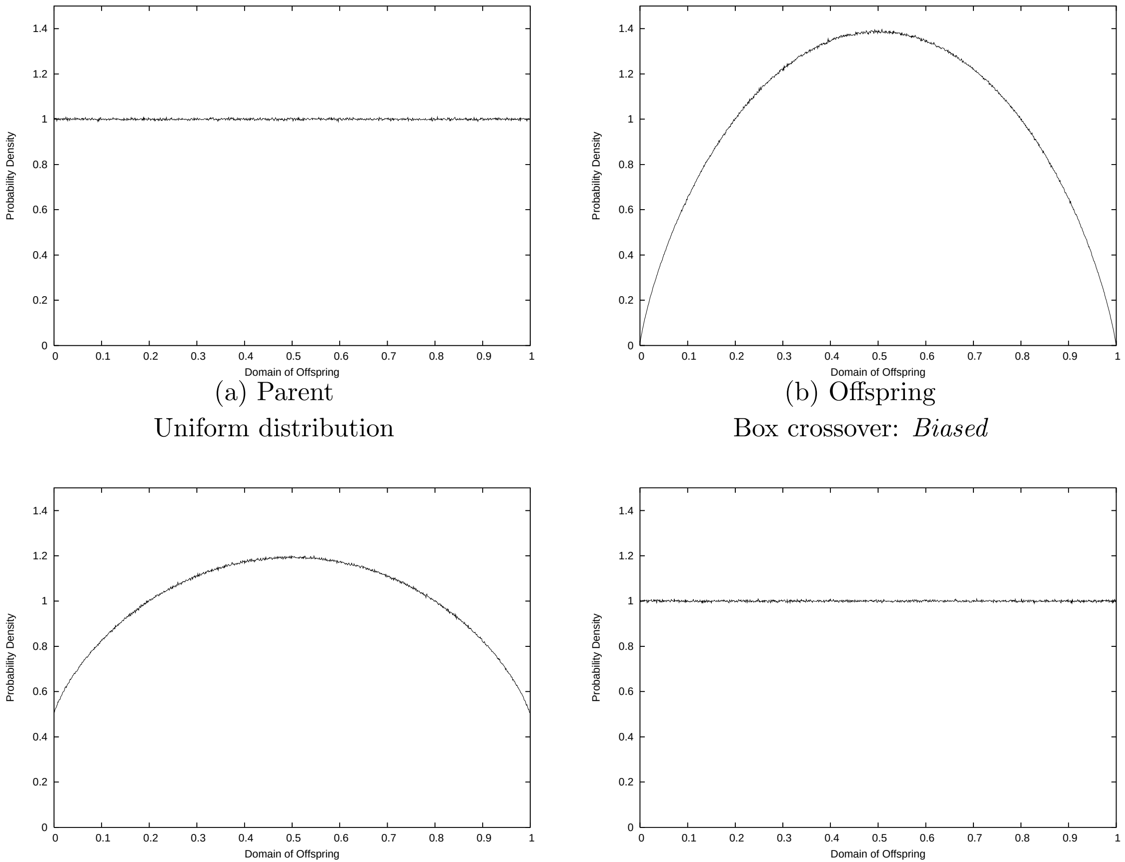
Traditional crossover operators for the real-coded representation are described in [3]. The two main families of traditional crossover operators [58] are discrete crossovers[1] [68] and blend crossovers [56]. Blend crossover operators can be distinguished into line crossovers and box crossovers. Important variations of the last two crossover operators are the extended-line crossover and the extended-box crossover [57].

Recently several new crossovers for the real-coded representation have been designed. Several non-traditional crossover operators for real-coded representation are found in the recent literature. They include SBX (simulated binary crossover) [4, 6, 15, 17, 18, 19], UNDX (unimodal normal distribution crossover) [42, 43, 62, 61], SPX (simplex crossover) [30, 79, 80, 81], PCX (parent-centric crossover) [5, 16], etc [27, 28, 76]. Most of them are complex and based on the specific probability distribution of the offspring (SBX, UNDX, and PCX), self-adaptivity (SBX and UNDX), or multiple parents (UNDX and SPX). Some of them, e.g., include the function of mutation operators. In this paper, we focus on traditional crossover that does not consider the specific probability distribution of the offspring but only what offspring can be generated with a probability greater than zero, given the two parents.

Pre-existing crossovers for the real-coded representation have an inherent bias toward the center of the space. Some boundary extension techniques to reduce crossover bias have been extensively studied [72, 77, 78]. The concept of crossover *bias* first appeared in [22] and it has been extensively used in [72, 78], in which they tried to remove the bias of real-coded crossover heuristically (and theoretically incompletely). In this paper we present the origin of this bias in geometric terms. This bias could be potentially harmful for the search. Therefore, we design a distance naturally suited to the *glued space* and study its corresponding crossover for which the bias completely disappears. Someya and Yamamura [72] have already presented a similar idea related to gluing the opposite boundaries of a given bounded domain. They hybridized mirroring and gluing of the search space: after the boundaries of the search space are extended by *mirroring* the search space [78], they are glued together. Their boundary-gluing idea looks similar to ours, but they simply tried to remove the crossover bias heuristically with no theoretical base. In contrast, we provide a theoretically sound method in which the boundaries of the search space are glued before they are extended. The fitness landscapes on glued spaces may not preserve continuity at the boundaries of the original spaces. We also provide a new boundary extension method of the original space that allows the continuity of the fitness landscape on the corresponding boundary-extended glued space to be preserved. While previous boundary extension methods [72, 77, 78] simply *mirrored* the given fitness landscape into the extended domain without any theoretical basis, our boundary extension is designed to resolve the discontinuity problem occurring when gluing the space with theoretical completeness. Furthermore, the proposed approach is not based on the *mirroring* technique.

The remainder of this paper is organized as follows. In Section 2, we introduce the notion of biased crossover and explain why traditional crossovers based on p-norms are biased. In Section 3, we present a novel crossover defined on the glued space that is unbiased. We discuss a boundary extension method on this glued space to make the fitness function smooth at the boundary of the domain in Section 4. Section 5 hints at the positions of optima when they are randomly located in the domain. Finally, we give some simulation results in Section 6 and draw conclusions in Section 7.

[1] It is also called *dominant crossover.*



(a) Parent
Uniform distribution

(b) Offspring
Box crossover: *Biased*

(c) Offspring
Extended-box crossover (BLX-$\frac{1}{2}$ [21])
Still biased

(d) Offspring
Box crossover in glued space (see Section 3)
Unbiased

Figure 1: Crossover bias in one-dimensional bounded real space

2 Crossover Bias

In this section, we formalize the notion of crossover bias and show that pre-existing crossovers for real-coded GAs have such a bias. In the following sections, we will derive a novel crossover without bias.

Notice that the notion of bias of a crossover operator has different definitions depending upon the underlying representation considered. The bias toward the center of the space considered in real-coded crossovers conceptually differs from the crossover biases on binary strings, which focus on how many bits are passed to the offspring and from which positions, which, in turn conceptually differs from the bias considered in Genetic Programming focusing on bloat. We introduce a geometric notion of *bias* of an operator in a formal and representation-independent manner. This notion is geometric in flavor and generalizes the intuitive notion of bias of an operator in the real-code space to any metric space, so encompassing all combinatorial spaces as well. Let S be the search space and OP be a search operator $z = OP(p_1, p_2, \ldots, p_n)$ where the p_i-s are the parents in S and $z \in S$ is the offspring.

Definition 1. *We say that a search operator is* unbiased *if, when we choose parents independently and uniformly at random in the search space S, we obtain offspring uniformly at random in the search space S. Formally OP is* unbiased *if independent and identically distributed p_i-s uniformly distributed in S imply that $z = OP(p_1, p_2, \ldots, p_n)$ is uniformly distributed in S.*

The notion of bias so defined can be understood as being the inherent preference of a search operator for specific areas of the search space. This is an important search property of a search operator: an evolutionary algorithm using that operator, without selection, is attracted to the areas the search operator prefers. Arguably, also when selection is present, the operator bias acts as a background force that makes the search keener to go toward the areas preferred by the search operator. This is not necessarily bad if the bias is toward the optimum or an area with high-quality solutions. However, it may negatively affect performance if the bias is toward an area of poor-quality solutions. If we do not know the spatial distribution of the fitness of the problem, we may prefer not to have any *a priori* bias of the search operator, and instead use only the bias of selection, which is informed by the fitness of sampled solutions that constitute empirical knowledge about promising areas obtained in the search, and which is better understood.

Now we investigate the origin of the bias. Box crossover is biased toward the center on the Euclidean space. This is easy to verify experimentally by picking a large number of pairs (ideally infinitely many) of random parents and generating offspring uniformly at random in the boxes identified by the pairs of parents. In the Hamming space, the distribution of the offspring of uniform crossover tends in the limit to be uniform on all space, whereas in the Euclidean space the distribution of the offspring tends to be unevenly distributed on the search space and concentrates toward the center of the space. One way to compensate, but not eliminate, such bias is using extended-line and extended-box crossovers. Figure 1 visualizes the crossover bias in the one-dimensional real space by plotting frequency rates of 10^8 offspring randomly generated by each box-type crossover. As we can see, box crossover is biased toward the center of the domain.[2] We could also observe that extended-box crossover largely reduces the bias but it is still biased toward the center.[3] Another way to compensate for the bias relies on understanding the origin of this bias. So, what is the origin of the different bias of crossover for the Hamming space and for the Euclidean space? The answer relies on the notion of *isotropy* of the underlaying metric space. Informally, a metric space is *isotropic* if all its points are equivalent: every point has the same properties in terms of distance. To define formally the notion of isotropic space, we first need to introduce the notion of isometric spaces.

Definition 2. *Two metric spaces $M = (S, d)$ and $M' = (S', d')$ are* isometric *if there exists a bijective function $g : S \to S'$ that is distance-preserving: so $\forall x, y \in S : d(x, y) = d'(g(x), g(y))$. The mapping g is called* isometry*.*

Definition 3. *A metric space $M = (S, d)$ is isotropic if for any two points $x, y \in S$, there exists an isometry g from M to itself such that $g(x) = y$.*

The Euclidean space, or, more precisely, a bounded hyper-rectangular subspace of the Euclidean space, is not isotropic, because the boundaries introduce an asymmetry on the types of points: the

[2] Appendix A provides the distribution function of offspring for the box crossover and thus formally derives the bias of this crossover.

[3] We can find consistent results with this in [72].

space has boundaries and has a center; hence, each point differs from each other, depending on how far from the boundaries and the center it is. Each point has a special status, hence the space is not isotropic. Conversely, in the Hamming space, each point is the center[4] and simultaneously is a boundary point (when we look at it as a subspace of a bounded vector space on $[0,1]^n$), so each point has the same status and the Hamming space is therefore isotropic. Let us now consider a new metric space obtained by gluing the opposite extremities of the Euclidean hyper-rectangle; in the case of a simple two-dimensional rectangle the space we obtain is the surface of a torus. Notice that the metric associated with this space is no longer the Euclidean metric, because points that were at opposite sides of the rectangle are close to each other in the new space. However, the metric of the glued space can be derived from the Euclidean metric. We present its formal derivation in Section 3. The new space is isotropic, because the boundaries, which are the origin of the inhomogeneity of types of points in the Euclidean space, are no longer there. Intuitively, the crossover based on this new space is unbiased because there cannot be bias toward the center, given that every point is the center of the new space, as in the case of the Hamming space. *It is therefore the isotropy of the space that causes the bias of the search operator to disappear.* We present this idea formally in the following.

Definition 4. *A* distance-based operator *is any search operator whose (conditional) probability distribution depends only on the distances among parents and offspring.*

For example, a search operator OP of arity two is distance-based if there exists a function f such that its conditional distribution probability $Pr\{Z=z|P_1=p_1,P_2=p_2\}=f(d(p_1,p_2),d(p_1,z),d(p_2,z))$.

Theorem 1. *Any distance-based operator OP of arity two based on an isotropic metric space M is unbiased.*

Proof. Let OP have a conditional probability distribution $Pr\{Z=z|P_1=p_1,P_2=p_2\}$. The probability of obtaining $z \in S$ is:

$$Pr\{z\} = \sum_{p_1,p_2\in S} Pr\{p_1,p_2\}\cdot Pr\{Z=z|P_1=p_1,P_2=p_2\}. \tag{1}$$

OP is unbiased if: $Pr\{p_1,p_2\}=\frac{1}{|S|^2}$ implies $Pr\{z\}=\frac{1}{|S|}$ for all $z \in S$. So,

$$\forall z_1,z_2\in S: \sum_{p_1,p_2\in S}\frac{1}{|S|^2}\cdot Pr\{Z_1=z_1|P_1=p_1,P_2=p_2\} = \sum_{p_1,p_2\in S}\frac{1}{|S|^2}\cdot Pr\{Z_2=z_2|P_1=p_1,P_2=p_2\}. \tag{2}$$

Simplifying:

$$\forall z_1,z_2\in S: \sum_{p_1,p_2\in S} Pr\{Z_1=z_1|P_1=p_1,P_2=p_2\} = \sum_{p_1,p_2\in S} Pr\{Z_2=z_2|P_1=p_1,P_2=p_2\}. \tag{3}$$

So, to prove that OP is unbiased we have to prove that the summations in (3) are equal, given the hypothesis that M is an isotropic space and OP is a distance-based operator. Since M is isotropic,

[4] We mean the center of the space X by $\text{argmin}_{x\in X}\sum_{y\in X} d(x,y)$ in the case of discrete space X or $\text{argmin}_{x\in X}\int_{y\in X} d(x,y)$ in the case of continuous space X.

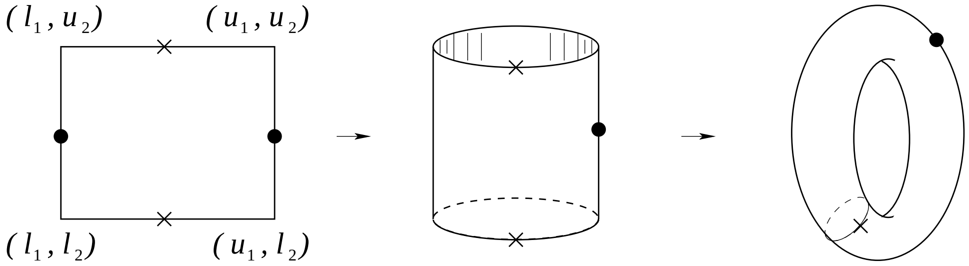


Figure 2: Glued space on $\mathbb{R}^2$. This can be considered as a quotient space.

there exists an isometry g such that $g(z_1) = z_2$. Since g is bijective and the summation is on all pairs of $p_1, p_2 \in S$, we have

$$\sum_{p_1,p_2 \in S} Pr\{Z_2 = z_2 | P_1 = p_1, P_2 = p_2\} = \sum_{p_1,p_2 \in S} Pr\{Z_2 = g(z_1) | P_1 = g(p_1), P_2 = g(p_2)\}. \tag{4}$$

Since OP is distance-based, there exists a function f such that $\forall p_1, p_2 : Pr\{Z_2 = g(z_1) | P_1 = g(p_1), P_2 = g(p_2)\} = f(d(g(p_1), g(p_2)), d(g(p_1), g(z_1)), d(g(p_2), g(z_1)))$. Since g is an isometry, the previous expression equals: $f(d(p_1, p_2), d(p_1, z_1), d(p_2, z_1)) = Pr\{Z_1 = z_1 | P_1 = p_1, P_2 = p_2\}$. This implies that in (3) each summand in the left-hand summation equals a summand in the right-hand summation and vice versa. So the two summations in (3) are equal. □

Theorem 1 can be generalized to all distance-based operators of any arity. In practice, all solution spaces used in search algorithms for real-coded problems are bounded, so all these spaces would give rise to biased crossovers. In the next section, we present a novel crossover for glued spaces, which is bias-free, in a formal and general setting.

3 Crossover in Glued Space

Let the solution space X be $\{x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : l_i \leq x_i < u_i \text{ for each } i\}$ where $l = (l_1, l_2, \ldots, l_n)$ is a lower bound and $u = (u_1, u_2, \ldots, u_n)$ is an upper bound (i.e., the boundaries of the hyper-box that define the domain of the function to optimize). If we apply traditional crossovers on this bounded domain, offspring have a bias toward the center of the space. As discussed in the previous section, one of the methods to eliminate this bias is gluing the boundaries by identifying u_i to l_i for each i, because the glued space obtained is isotropic. Figure 2 shows this glued space for the $\mathbb{R}^2$ case.

Formally, the glued space is a *quotient space*. To construct a quotient space which gives an effect equivalent to gluing, the following equivalence relation on $\mathbb{R}^n$ is defined:

Definition 5. *$x \sim y$ if and only if for each $i = 1, 2, \ldots, n$, there exists $a_i \in \mathbb{Z}$ such that $x_i - y_i = a_i(u_i - l_i)$, where $\mathbb{Z}$ is the set of all integers.*

Theorem 2. *The relation $\sim$ is an equivalence relation.*

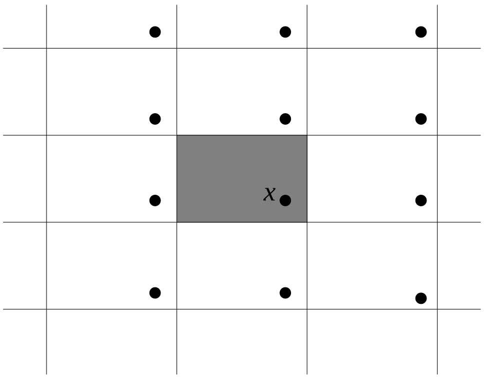


Figure 3: Equivalent class of x on $\mathbb{R}^2$. The shadowed rectangle represents the given bounded domain X. Each rectangle has the same size as X.

Proof. Assume that x, y, and $z \in \mathbb{R}^n$.
(i) *Reflexive*: $x_i - x_i = 0(u_i - l_i)$ for each $i = 1, 2, \ldots, n$. Since $0 \in \mathbb{Z}$, $x \sim x$.
(ii) *Symmetric*: If $x \sim y$, for each $i = 1, 2, \ldots, n$, there exists $a_i \in \mathbb{Z}$ such that $x_i - y_i = a_i(u_i - l_i)$. Then, $y_i - x_i = -(x_i - y_i) = -a_i|u_i - l_i|$ and $-a_i \in \mathbb{Z}$. Hence, $y \sim x$.
(iii) *Transitive*: If $x \sim y$ and $y \sim z$, for each $i = 1, 2, \ldots, n$, there exists $a_i \in \mathbb{Z}$ such that $x_i - y_i = a_i(u_i - l_i)$ and $b_i \in \mathbb{Z}$ such that $y_i - z_i = b_i(u_i - l_i)$. $x_i - z_i = (x_i - y_i) + (y_i - z_i) = a_i(u_i - l_i) + b_i(u_i - l_i) = (a_i + b_i)(u_i - l_i)$. $a_i + b_i \in \mathbb{Z}$ since a_i and $b_i \in \mathbb{Z}$. So, $x \sim z$. □

Let $\langle x \rangle$ be the equivalence class of $x \in \mathbb{R}^n$. In Figure 3, points indicated by bullets are in the same equivalent class on $\mathbb{R}^2$.

X can be considered as a quotient set $\mathbb{R}^n/\sim$ by considering $x \in X$ as $\langle x \rangle \in \mathbb{R}^n/\sim$. However, this gives another topology to the same set. We need to define a distance tailored to this new topology. We define a new distance on $\mathbb{R}^n/\sim$ using the distance on $\mathbb{R}^n$. Let $x, y \in X$.

Definition 6. *Let $x, y \in X$. If d is a metric for $\mathbb{R}^n$,*

$$d_q(x, y) := \min_{x' \in \langle x \rangle, y' \in \langle y \rangle} d(x', y').$$

Theorem 3. *If the metric d is induced by a norm, d_q is a metric for X.*

Proof. Assume that x, y, and $z \in X$.
(i) *Positiveness*: $d_q(x, y) \geq 0$ by the definition of d_q. If $x = y$, $x \in \langle y \rangle$. Then, $0 \leq d_q(x, y) \leq d(x, x) = 0$ and hence $d_q(x, y) = 0$. Now assume that $d_q(x, y) = 0$. There exist $x' \in \langle x \rangle$ and $y' \in \langle y \rangle$ such that $d_q(x, y) = d(x', y') = 0$. Since d is a metric, $x' = y'$. So, $\langle x \rangle = \langle y \rangle$. Therefore $x = y$.
(ii) *Symmetry*: $d_q(x, y) = d(x', y')$ for some $x' \in \langle x \rangle$ and $y' \in \langle y \rangle$. Then, $d_q(x, y) = d(x', y') = d(y', x') \geq d_q(y, x)$. Similarly, $d_q(y, x) \geq d_q(x, y)$.
(iii) *Triangular inequality*: $d_q(x, y) + d_q(y, z) = d(x', y') + d(y'', z'')$ for some $x' \in \langle x \rangle$, $y', y'' \in \langle y \rangle$, and $z'' \in \langle z \rangle$. Since d is induced by a norm, $d(y'', z'') = d(y'' - y'' + y', z'' - y'' + y') = d(y', z'' - y'' + y')$.

Since $y'_i - y''_i = k_i|u_i - l_i|$ for each i, $z'' - y'' + y' \in \langle z \rangle$. Hence,

$$
\begin{aligned}
d_q(x,y) + d_q(y,z) &= d(x',y') + d(y'',z'') && (5)\\
&= d(x',y') + d(y', z'' - y'' + y') && (6)\\
&\geq d(x', z'' - y'' + y') && (7)\\
&\geq d_q(x,y). && (8)
\end{aligned}
$$

□

To calculate the distance between two points x and y using its definition directly, we should consider the distance of any possible pair of points in the classes $\langle x \rangle$ and $\langle y \rangle$ and return the minimum. However, this is impossible in practice, as there are an infinite number of these pairs. Fortunately, we have a practical way to calculate the distance $d_q(x,y)$ using Lemma 1 below, which requires the following definitions to be stated.

Let $x, y \in X$. For each i, let $T_i(y) = \{y_i, y_i + (u_i - l_i), y_i - (u_i - l_i)\}$ and $M_i(y) = \underset{m \in T_i(y)}{\text{argmin}}\{|x_i - m|\}$. $M_i(y)$ is a set because the number of maximizers can be more than one. Let $M(y) = \{(y'_1, y'_2, \ldots, y'_n) \in \mathbb{R}^n : y'_i \in M_i(y) \text{ for each } i\}$.

Lemma 1. *Let x and $y \in X$. If a metric d is induced by p-norm and $y^* \in M(y)$, $d(x',y') \geq d(x,y^*)$ for all $x' \in \langle x \rangle$ and $y' \in \langle y \rangle$, i.e., d_q is well-defined and $d_q(x,y) = d(x,y^*)$. Moreover, $M(y) = \{y' \in \langle y \rangle : d_q(x,y) = d(x,y')\}$.*

Proof. $d(x',y') = (\sum_{i=1}^n |x'_i - y'_i|^p)^{1/p}$.

For each i, $|x'_i - y'_i| = |x'_i - y'_i + x_i - x_i + y_i - y_i| = |(x_i - y_i) + (x'_i - x_i) + (y_i - y'_i)|$. Since $x \sim x'$ and $y \sim y'$, $(x'_i - x_i) + (y_i - y'_i) = k_i(u_i - l_i)$ for some $k_i \in \mathbb{Z}$. Then, $|x'_i - y'_i| = |(x_i - y_i) + k_i(u_i - l_i)|$.

If $k_i \geq 2$, $|(x_i - y_i) + k_i(u_i - l_i)|$ is positive since $x_i - y_i$ cannot be greater than $u_i - l_i$. So, $|x'_i - y'_i| = |(x_i - y_i) + k_i(u_i - l_i)| > |(x_i - y_i) + (u_i - l_i)| = |x_i - \{y_i + (u_i - l_i)\}| \geq |x_i - y^*_i|$. Similarly, in the case that $k_i \leq -2$, $|(x_i - y_i) + k_i(u_i - l_i)|$ is negative. Hence, $|x'_i - y'_i| = |(x_i - y_i) + k_i(u_i - l_i)| > |(x_i - y_i) - (u_i - l_i)| = |x_i - \{y_i - (u_i - l_i)\}| \geq |x_i - y^*_i|$. Finally, if $k_i = 0, 1$, or -1, then $|x'_i - y'_i| = |(x_i - y_i) + k_i(u_i - l_i)| \geq |x_i - y^*_i|$ by the definition of $M(y)$.

So, $d(x',y') = \{\sum_{i=1}^n |x'_i - y'_i|^p\}^{1/p} \geq \{\sum_{i=1}^n |x_i - y^*_i|^p\}^{1/p} = d(x,y^*)$.

Now, we will show that $M(y) = \{y' \in \langle y \rangle : d_q(x,y) = d(x,y')\}$. Suppose that $y' \in \langle y \rangle$ satisfies $d_q(x,y) = d(x,y')$ but $y' \notin M(y)$. Then, there exists a nonempty index set $J = \{j : |x_j - y'_j| \neq |x_j - y^*_j|\}$. For each $j \in J$, $|x_j - y'_j| = |x_j - y_j - k_j(u_j - l_j)|$. If $k_j \geq 2$ or $k_j \leq -2$, then $|x_j - y'_j| > |x_j - y^*_j|$ by the same way as the above. If $k_j = 0, 1$, or -1, then $|x_j - y'_j| \geq |x_j - y^*_j|$. By the assumption, $|x_j - y'_j| \neq |x_j - y^*_j|$ and hence $|x_j - y'_j| > |x_j - y^*_j|$. Therefore, $d(x,y') = (\sum_{i=1}^n |x_i - y'_i|^p)^{1/p} > d(x,y^*)$ and it is a contradiction. □

According to Lemma 1, to calculate $d_q(x,y)$, we need only to find y^* by choosing the minimizer among three elements from $T_i(y)$ for each coordinate and get the Euclidean distance between x and y^*. The segment between x and y on the quotient space is induced by the segment between x and y^* on $\mathbb{R}^n$, as shown as follows for the class of p-norms.

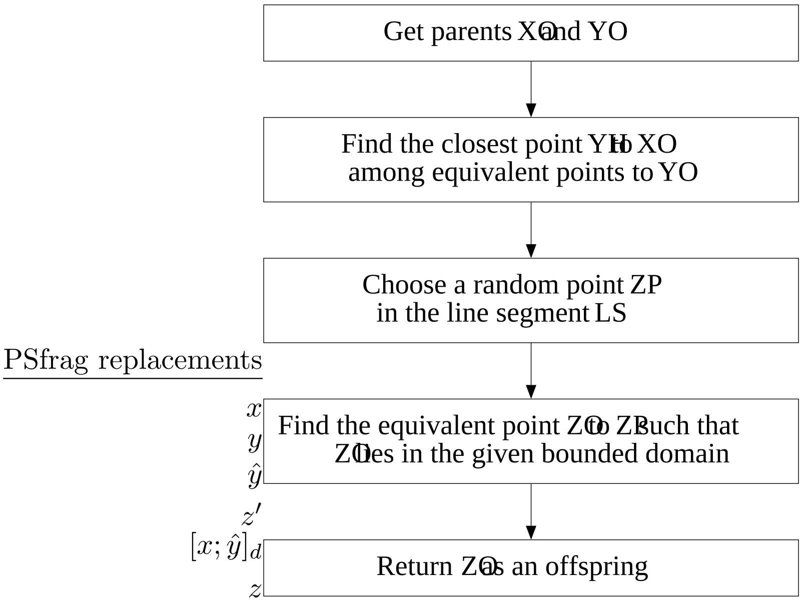


Figure 4: Flowchart of the proposed unbiased crossover

Theorem 4. *If d is induced by the p-norm, the segment $[x;y]_{d_q}$ is the set*

$$\bigcup_{\hat{y}\in M(y)} \{z \in X : z \sim z' \text{ for some } z' \in [x;\hat{y}]_d\},$$

where $[u;v]_d = \{w \in X : d(u,w) + d(w,v) = d(u,v)\}$.

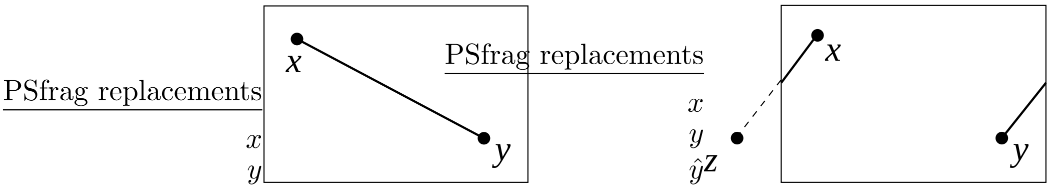
Proof. Let $y^* \in M(y)$ and $z \sim z'$ for some $z' \in [x;y^*]_d$. Then, $d_q(x,z)+d_q(z,y) \leq d(x,z')+d(z',y^*) = d(x,y^*) = d_q(x,y^*)$. Since $d_q(x,z) + d_q(z,y) \geq d_q(z,y)$ by triangular property, $d_q(x,z) + d_q(z,y) = d_q(z,y)$. So, $z \in [x;y]_{d_q}$.

Now, let $z \in [x;y]_{d_q}$. There exist z' such that $d_q(x,z) = d(x,z')$ and y' such that $d_q(z,y) = d(z,y')$ in $\mathbb{R}^n$ by Lemma 1. Let $y^* := y' - z + z'$. Then, $y^* \sim y' \sim y$ and $d_q(x,y) = d_q(x,z) + d_q(z,y) = d(x,z') + d(z,y') = d(x,z') + d(z',y' - z + z') = d(x,z') + d(z',y^*) \geq d(x,y^*)$ since d is a metric. By the definition of d_q, $d_q(x,y) \leq d(x,y^*)$ is true and hence $d_q(x,y) = d(x,y^*)$. This implies $y^* \in M(y)$ and $d(x,z') + d(z',y^*) = d(x,y^*)$. □

Theorem 4 allows us to construct segments in the glued space starting from corresponding segments of the original space. This, in turns, allows us to implement unbiased crossovers on the glued space from the corresponding biased crossovers on the original space. Figures 4 and 5 show the flowchart and the pseudo-code of the unbiased crossover on the glued space, respectively. It takes linear time in the size of the vector. In Figure 6, (a) shows the segment on Euclidean space and (b) shows the segment on glued space (quotient space). Segments may cross the boundaries in the quotient space.

```
$UX(x, y)$
{
    for $i \leftarrow 1$ to $n$
        $T_i \leftarrow \{y_i, y_i + (u_i - l_i), y_i - (u_i - l_i)\}$;
        $M_i \leftarrow \text{argmin}_{m \in T_i} |x_i - m|$;
        $\hat{y}_i \leftarrow$ a random number in $M_i$;
    $z' \leftarrow$ a random vector in $[x; \hat{y}]_d$;
    for $i \leftarrow 1$ to $n$
        find $z_i \in [l_i, u_i]$ by adding some $c \cdot (u_i - l_i)$ to $z'_i$, where $c \in \{-1, 0, 1\}$;
    return $z = (z_1, z_2, \ldots, z_n)$;
}
```

Figure 5: Pseudo-code of the proposed unbiased crossover



(a) $[x; y]_{d_2}$

(b) $[x; y]_{d_q}$

Figure 6: Line segments on Euclidean space and glued space

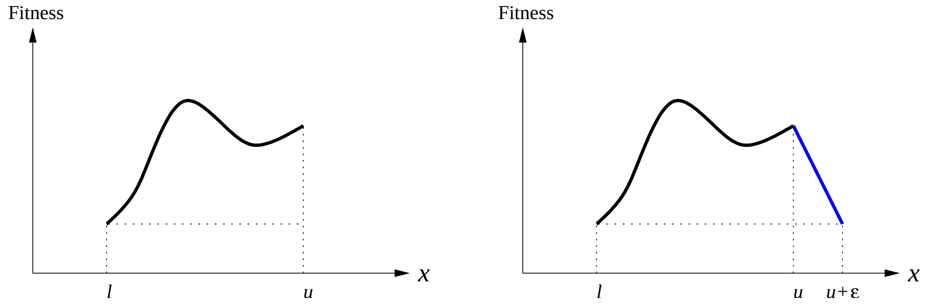
4 Boundary Extension in Glued Space

Whereas using the quotient space makes the corresponding crossover operator unbiased, at the same time it causes an issue. In the quotient space, u_i is identified to l_i, so corresponding points on opposite boundaries are considered equivalent. However, this creates an ambiguity about the fitness of the points on the glued part of the space because each of these points inherit two fitness values from the original space, one from its fitness on the boundary u_i and the other from its fitness from the boundary l_i. So, its fitness is not clearly defined. This issue seems to be easily resolvable using a criterion to disambiguate the fitness on the gluing threshold, for example, by assigning always the fitness of the points from the lower boundary. However, doing this introduces discontinuity in the fitness landscape on the gluing threshold. We show that this discontinuity can be completely removed by appropriately extending the boundaries of the space as follows.

First, let us consider the function defined on a one-dimensional bounded domain. Let X be $\{x \in \mathbb{R} : l \leq x \leq u\}$. For a small real number $\epsilon \in \mathbb{R}$,

$$f_\epsilon(x) := \begin{cases} f(x) & \text{if } l \leq x \leq u \\ \dfrac{x-u}{\epsilon} f(l) + \dfrac{u+\epsilon-x}{\epsilon} f(u) & \text{if } u < x < u+\epsilon. \end{cases}$$

The function f_ϵ connects $f(u)$ and $f(u+\epsilon) = f(l)$ linearly on $[u, u+\epsilon)$ and it has the property that $f_\epsilon|_X = f$. That is, f_ϵ is a continuous extension of f on $[l, u+\epsilon)$. Figure 7 shows this idea. It looks quite natural. We can also apply a similar method to higher dimensional real spaces using a recurrence



(a) $f(x)$ on $[l, u]$ (b) $f_\epsilon(x)$ on $[l, u+\epsilon)$

Figure 7: Boundary extension in one-dimensional bounded domain

relation. In the following, we first consider an analogous boundary extension for a two-dimensional bounded domain and then we tell how it can be generalized to the n-dimensional case.

Let X be $\{(x_1, x_2) \in \mathbb{R}^2 : l_1 \le x_1 \le u_1, l_2 \le x_2 \le u_2\}$. If $x \in X$, we just take the fitness value of $f(x)$. However, if $x_1 > u_1$ or $x_2 > u_2$, we have to define the fitness value at that point. There are two natural candidate functions we can consider to obtain a continuous extension of $f(x)$ in the bi-dimensional case, as follows. In the case that $l_1 \le x_1 < u_1 + \epsilon_1$ and $l_2 \le x_2 \le u_2$,

$$f_1(x_1, x_2) := \begin{cases} f(x_1, x_2) & \text{if } l_1 \le x_1 \le u_1 \\ \dfrac{x_1 - u_1}{\epsilon_1} f(l_1, x_2) + \dfrac{u_1 + \epsilon_1 - x_1}{\epsilon_1} f(u_1, x_2) & \text{if } u_1 < x_1 < u_1 + \epsilon_1. \end{cases}$$

In the case that $l_1 \le x_1 \le u_1$ and $l_2 \le x_2 < u_2 + \epsilon_2$,

$$f_2(x_1, x_2) := \begin{cases} f(x_1, x_2) & \text{if } l_2 \le x_2 \le u_2 \\ \dfrac{x_2 - u_2}{\epsilon_2} f(x_1, l_2) + \dfrac{u_2 + \epsilon_2 - x_2}{\epsilon_2} f(x_1, u_2) & \text{if } u_2 < x_2 < u_2 + \epsilon_2. \end{cases}$$

If $l_1 \le x_1 < u_1 + \epsilon_1$ and $l_2 \le x_2 < u_2 + \epsilon_2$, f_ϵ may be defined in two ways. One consists of applying f_1 first and then applying f_2, and the other consists of applying f_2 first and then applying f_1. That is,

$$f_\epsilon^1(x_1, x_2) := \begin{cases} f_2(x_1, x_2) & \text{if } l_1 \le x_1 \le u_1 \\ \dfrac{x_1 - u_1}{\epsilon_1} f_2(l_1, x_2) + \dfrac{u_1 + \epsilon_1 - x_1}{\epsilon_1} f_2(u_1, x_2) & \text{if } u_1 < x_1 < u_1 + \epsilon_1 \end{cases}$$

or

$$f_\epsilon^2(x_1, x_2) := \begin{cases} f_1(x_1, x_2) & \text{if } l_2 \le x_2 \le u_2 \\ \dfrac{x_2 - u_2}{\epsilon_2} f_1(x_1, l_2) + \dfrac{u_2 + \epsilon_2 - x_2}{\epsilon_2} f_1(x_1, u_2) & \text{if } u_2 < x_2 < u_2 + \epsilon_2. \end{cases}$$

The following theorem shows that both methods give rise to the same extension.

Theorem 5. *$f_\epsilon^1(x_1, x_2) = f_\epsilon^2(x_1, x_2)$ on $[l_1, u_1 + \epsilon_1) \times [l_2, u_2 + \epsilon_2)$.*

Proof. We consider the following four sub-domains that partition the extended domain $[l_1, u_1 + \epsilon_1) \times [l_2, u_2 + \epsilon_2)$.
Case 1: If $l_1 \le x_1 \le u_1$ and $l_2 \le x_2 \le u_2$,

$$f_\epsilon^1(x_1, x_2) = f_\epsilon^2(x_1, x_2) = f(x_1, x_2). \tag{9}$$

Case 2: If $l_1 \leq x_1 \leq u_1$ and $u_2 < x_2 < u_2 + \epsilon_2$,

$$f_\epsilon^1(x_1, x_2) = f_\epsilon^2(x_1, x_2) = \frac{x_2 - u_2}{\epsilon_2} f(x_1, l_2) + \frac{u_2 + \epsilon_2 - x_2}{\epsilon_2} f(x_1, u_2). \tag{10}$$

Case 3: If $u_1 < x_1 < u_1 + \epsilon_1$ and $l_2 \leq x_2 \leq u_2$,

$$f_\epsilon^1(x_1, x_2) = f_\epsilon^2(x_1, x_2) = \frac{x_1 - u_1}{\epsilon_1} f(l_1, x_2) + \frac{u_1 + \epsilon_1 - x_1}{\epsilon_1} f(u_1, x_2). \tag{11}$$

Case 4: If $u_1 < x_1 < u_1 + \epsilon_1$ and $u_2 < x_2 < u_2 + \epsilon_2$,

$$\begin{aligned} f_\epsilon^1(x_1, x_2) = {} & \frac{x_1 - u_1}{\epsilon_1} \left(\frac{x_2 - u_2}{\epsilon_2} f(l_1, l_2) + \frac{u_2 + \epsilon_2 - x_2}{\epsilon_2} f(l_1, u_2) \right) \\ & + \frac{u_1 + \epsilon_1 - x_1}{\epsilon_1} \left(\frac{x_2 - u_2}{\epsilon_2} f(u_1, l_2) + \frac{u_2 + \epsilon_2 - x_2}{\epsilon_2} f(u_1, u_2) \right). \qquad (12) \\ f_\epsilon^2(x_1, x_2) = {} & \frac{x_2 - u_2}{\epsilon_2} \left(\frac{x_1 - u_1}{\epsilon_1} f(l_1, l_2) + \frac{u_1 + \epsilon_1 - x_1}{\epsilon_1} f(u_1, l_2) \right) \\ & + \frac{u_2 + \epsilon_2 - x_2}{\epsilon_2} \left(\frac{x_1 - u_1}{\epsilon_1} f(l_1, u_2) + \frac{u_1 + \epsilon_1 - x_1}{\epsilon_1} f(u_1, u_2) \right). \qquad (13) \\ \Rightarrow f_\epsilon^1(x_1, x_2) = {} & f_\epsilon^2(x_1, x_2). \end{aligned}$$

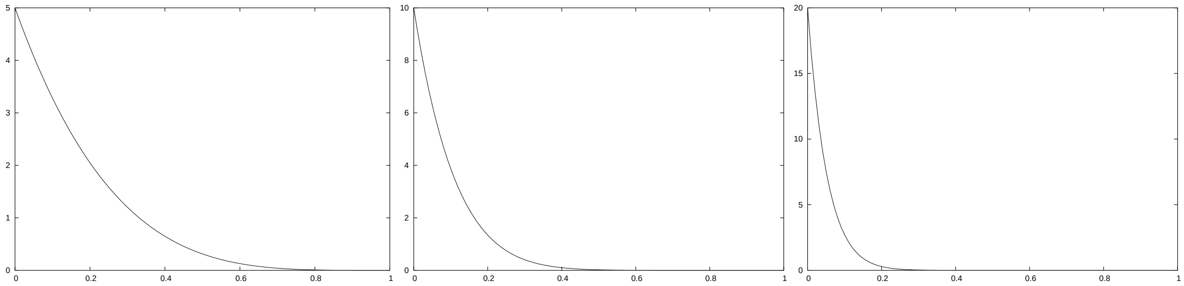
□

For a general n-dimensional space, we proceed in a similar way. We define f_k by induction as follows.

- $k = 0$: We set f_0 to f.
- $1 \leq k \leq n$: If $l_i \leq x_i < u_i + \epsilon_i$ for $1 \leq i \leq k$ and $l_i \leq x_i \leq u_i$ for $k+1 \leq i \leq n$,

$$f_k(x_1, x_2, \ldots, x_n) := \begin{cases} f_{k-1}(x_1, x_2, \ldots, x_n) & \text{if } l_k \leq x_k \leq u_k \\ \frac{x_k - u_k}{\epsilon_k} f_{k-1}(x_1, \ldots, x_{k-1}, l_k, x_{k+1}, \ldots, x_n) \\ \quad + \frac{u_k + \epsilon_k - x_k}{\epsilon_k} f_{k-1}(x_1, \ldots, x_{k-1}, u_k, x_{k+1}, \ldots, x_n) & \text{if } u_k < x_k < u_k + \epsilon_k. \end{cases}$$

Then, every f_k is well-defined, and we choose f_n as the final continuous extension, i.e., $f_\epsilon := f_n$. Thus, for any function and any dimension of its domain, we can always get a well-defined extended function that maintains its continuity on the boundary-extended quotient space.



(a) $n = 5$ (b) $n = 10$ (c) $n = 20$

Figure 8: Probability density function $f(t)$ of $d(X, \partial D)$ according to dimension

5 Distribution of Positions of Randomly-Located Optima

Before presenting experiments, we present in this section a rather counter-intuitive theoretical result that may motivate further the use of bias-free search operators in high-dimensional spaces. In particular, we investigate the distribution of the positions of the optima of a class of functions with respect to its distance from the boundary. In these functions, we assume the optima to be located uniformly at random in each uni-dimensional projection of the domain of the function. This result holds for all Minkowski distances.

Let n be the dimension of the domain space $D = [-1, 1]^n$. We assume that $X = (X_1, X_2, \ldots, X_n)$ is a random variable to indicate the position of the optimum, and X_is are independent and identically-distributed random variables such that $X_i \sim U([-1, 1])$. Then, the cumulative distribution function for $d(X, \partial D)$, any Minkowski distance from the boundary, is in the following.

$$\begin{aligned} Pr(d(X,\partial D) \leq t) &= Pr(\max_{1\leq i\leq n} |X_i| \geq 1-t) && (14)\\ &= 1 - Pr(\max_{1\leq i\leq n} |X_i| < 1-t) && (15)\\ &= 1 - \prod_{i=1}^{n} Pr(|X_i| < 1-t) && (16)\\ &= 1-(1-t)^n, \qquad 0 \leq t \leq 1. && (17) \end{aligned}$$

The probability density function $f(t)$ of $d(X, \partial D)$ becomes

$$\begin{aligned} f(t) &= \frac{d}{dt} Pr(d(X,\partial D) \leq t) && (18)\\ &= n(1-t)^{n-1}. && (19) \end{aligned}$$

Figure 8 shows the probability density functions $f(t)$s of $d(X, \partial D)$ for the dimensions 5, 10, and 20. We can approximate the probability that the optimum is in the neighborhood of the center or the boundary of the domain, as follows. We know that $B_{d_1}(x; r) \subseteq B_{d_2}(x; r) \subseteq B_{d_\infty}(x; r)$, where $B_d(x; r) = \{z \in D : d(x, z) \leq r\}$. Then, the probabilities of the optimum being in $B_d(\mathbf{0}; \epsilon)$ are always less than or equal to that being in $B_{d_\infty}(\mathbf{0}; \epsilon)$. The latter becomes exactly $Pr(d(X, \partial D) \geq 1 - \epsilon) = 1 - Pr(d(X, \partial D) < 1 - \epsilon) = \epsilon^n$. We can also easily obtain that the probability of the optimum being in the ϵ-neighborhood of the boundary is $Pr(d(X, \partial D) \leq \epsilon) = 1 - (1-\epsilon)^n$. For some dimensions and a small $\epsilon = 0.002$, we can get the probabilities as follows:

where the probability of the optimum being	$n = 5$	$n = 10$	$n = 20$	$n = 2500$
the neighborhood of the center	$\leq 10^{-13}$	$\leq 10^{-26}$	$\leq 10^{-53}$	$\leq 10^{-6749}$
the neighborhood of the boundary	0.01	0.02	0.04	0.99

We can see that the probabilities of the optimum being in the neighborhood of the center are nearly 0, but the probabilities being in the neighborhood of the boundary increase as the dimension, and surprisingly, it becomes very close to 1 for sufficiently large dimensions.

6 Simulation

We conducted experiments on well-known problems of function optimization [25, 31] and a nonlinear regression problem on a real-world dataset. The genetic framework in our experiments is based on Tsutsui and Goldberg [78], a recent study on real-coded GAs. We used the same test functions for our experiments.[5]

6.1 Test Functions and Experimental Methodology

Table 1 shows the test functions we used for the experiments. F_{Sphere} and $F_{\text{Rastrigin}}$ have the optima at the center of the domain. $F_{\text{M-Sphere}}$ and $F_{\text{M-Rastrigin}}$ were modified so that their optima are located just at the boundary of the domain. F_3 has the optimum at the boundary of the domain and F_{Schwefel} has it near the boundary. We used the resolution value Δ to determine whether the optimal solution was found as in [78]. We defined the successful detection of the solution as being within Δ range of the actual optimum point. If the actual optimum point is $(x_1^*, x_2^*, \ldots, x_n^*)$, any solution in the set $\{(x_1, x_2, \ldots, x_n) \ : \ x_i \in [x_i^* - \Delta/2, x_i^* + \Delta/2]\}$ is considered as the optimum.

We mainly followed the genetic framework by Tsutsui and Goldberg [78]. Its basic evolutionary model is quite similar to that of CHC [21] and $(\mu + \lambda)$-ES [7]. Figure 9 depicts its flowchart. Let the population size be N. A collection of $N/2$ pairs is randomly composed, and crossover and mutation are applied to each pair, generating $N/2$ offspring. Parents and newly generated offspring are ranked and the best N individuals among them are selected for the population in the next generation. The population size was 400 for all experiments, as in [78]. If the population has no change during $n \times r \times (1.0 - r)$ generations, it is reinitialized except for the best individual. Here, r is a divergence rate and we set it to 0.25 as in [21]. The proposed GA terminates when it finds the global optimum.

For crossover, we used various crossover operators: extended-box crossover, box crossover, quotient box crossover, and boundary-extended quotient box crossover. We used BLX-α ($\alpha = 1/2$) of [21] as extended-box crossover. We used the same extension rate for boundary extension. We set ϵ_i to $0.1 \times (u_i - l_i)$.[6]

After crossover, we either mutate the offspring, as in [78], or do not. Mutation was applied only to show the potential of the proposed methodology comparing it with the work of [78]. We used two

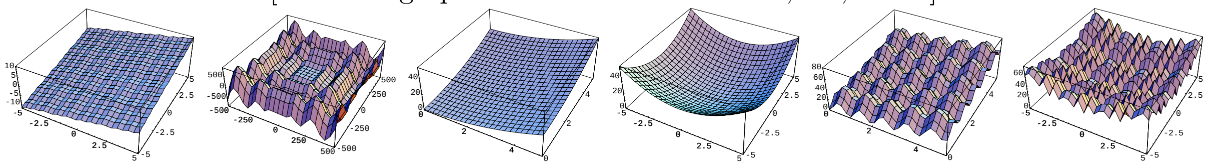
[5] A number of studies have been based on real-coded representation. We adopted the work of Tsutsui and Goldberg [78], who presented an idea called boundary extension, the most similar to ours. We used the same experimental setting as theirs, to fairly compare with the proposed methodology.

[6] In our preliminary test, the size of the boundary extension seldom affected the performance.

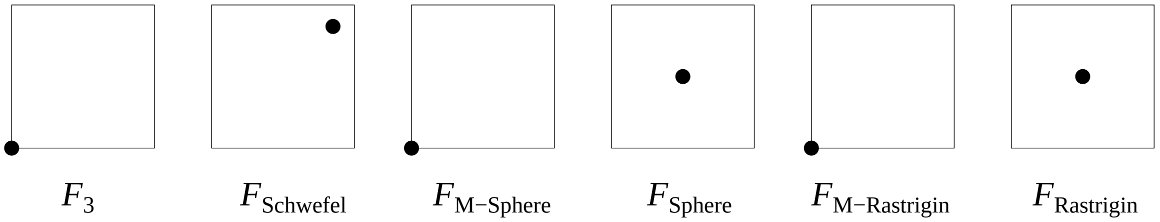
Table 1: Test Functions

Function	n	Range of x_i: $[l_i, u_i]$	Optimum	Δ
$F_3 = \sum_{i=1}^{n} \lfloor x_i \rfloor$	5	$[-5.12, 5.11]$	$x_i \in [-5.12, -5.0)$ for each i	0.01
$F_{\text{Schwefel}} = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	10	$[-512, 511]$	$(420.968746, \ldots, 420.968746)$	1.0
$F_{\text{M-Sphere}} = \sum_{i=1}^{n} {x_i}^2$	20	$[0.0, 5.11]$	$(0, 0, \ldots, 0)$	0.01
$F_{\text{Sphere}} = \sum_{i=1}^{n} {x_i}^2$	20	$[-5.12, 5.11]$	$(0, 0, \ldots, 0)$	0.01
$F_{\text{M-Rastrigin}} = \sum_{i=1}^{n} \left({x_i}^2 - 10\cos(2\pi x_i) + 10\right)$	20	$[0.0, 5.11]$	$(0, 0, \ldots, 0)$	0.01
$F_{\text{Rastrigin}} = \sum_{i=1}^{n} \left({x_i}^2 - 10\cos(2\pi x_i) + 10\right)$	20	$[-5.12, 5.11]$	$(0, 0, \ldots, 0)$	0.01

[Function graphs for 2-dimensional cases, i.e., $n = 2$]

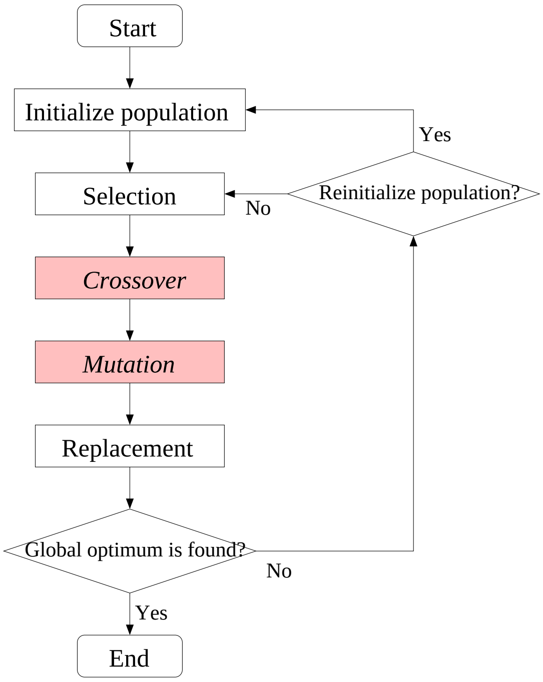


[Relative position of the optimum for each function]



different operators when we applied mutation; Type I and Type II mutation. They are all quite simple mutation operators, originating from [6] and [78]. Both mutations are consecutively applied all the time; Type II mutation after Type I mutation. Figure 10 reports these mutations. Type I mutation is a simple dynamic Gaussian mutation inspired from [6]. It is designed to provide traditional box crossover with a similar effect to extended-box crossover. It depends on the distance between parents and, as population converges, the strength of Type I mutation approaches zero. Type II mutation is a simple static Gaussian mutation, the same as in [78]. Different mutation rates were applied for each crossover type.[7] From our preliminary test for tuning mutation rates, we chose the best performed one for each type of crossover. Table 2 shows the best mutation rates for two crossover types. Mutation

[7] In our preliminary test, the performance of crossovers largely depended on the used mutation rates. We tried to use the best performed mutation rates according to crossover type to make a fair comparison.



(We will change only *crossover* and *mutation* for comparison.)

Figure 9: Flowchart of CHC

Table 2: Mutation Rates for Each Type of Crossover

Crossover type	Type I mutation	Type II mutation
Extended-box crossover	0	$0.05/k$
Box crossovers	$0.50/k$	$0.10/k$

$k = 1 + \lfloor numberOfGenerations/100 \rfloor$.

rates decrease as the number of generations increases.

6.2 Results

We present results for the number of function evaluations employed to find the optimum. Table 3 shows the results from 30 runs. As a measure of performance, we used the percentage gap $100 \times (output - best)/best$,[8] where *best* means the best value obtained in our experiments and previous literature [78] for the instance. The smaller the value is, the smaller the difference from the optimum is, i.e., the smaller, the better.

On the test functions whose optima are located at the boundary of the domain (F_3, $F_{\text{M-Sphere}}$, and $F_{\text{M-Rastrigin}}$), boundary-extended quotient box crossover was overall better than any other crossover - the results of [78], extended-box crossover, and traditional box crossover. Overall, quotient box

[8] The *percentage gap* is a well-known measure of performance in the field of operations research, e.g., [11, 67].

```
// $x$ and $y$ are parents.
TypeI-Mutation($x, y, z, p$)
{
        **for** $i \leftarrow 1$ to $n$
                **if** a random number from $[0, 1]$ is less than mutation rate $p$
                    $z_i \leftarrow z_i + N(0, |x_i - y_i|)$;
        **return** $z$;
}

TypeII-Mutation($z, p$)
{
        **for** $i \leftarrow 1$ to $n$
                **if** a random number from $[0, 1]$ is less than mutation rate $p$
                    $z_i \leftarrow z_i + N(0, (u_i - l_i)/2)$;
        **return** $z$;
}
```

Figure 10: Used mutation operators

Table 3: Number of Function Evaluations to Find the Optimum

Function	F_3	F_{Schwefel}	$F_{\text{M-Sphere}}$	F_{Sphere}	$F_{\text{M-Rastrigin}}$	$F_{\text{Rastrigin}}$	Total
Crossover	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	ave
Best results from [78]	6520.6	32243.0	32094.9	37113.6	129488.9	101852.2	
	(150.5)	(1347.1)	(113.1)	(189.4)	(2306.0)	(3027.8)	
%-gap	423.0%	7.6%	**0.0**%	**0.0**%	345.4%	37.1%	135.5%
Extended-box crossover	8320.0	45980.0	57073.3	40166.7	303866.7	107513.3	
(BLX-1/2)	(176.8)	(1141.4)	(167.2)	(151.2)	(2138.9)	(4662.3)	
%-gap	567.4%	53.5%	77.8%	8.2%	945.2%	44.8%	282.8%
Box crossover	2986.7	74213.3	40833.3	43693.3	50486.7	74266.7	
	(117.0)	(1766.5)	(129.3)	(378.6)	(417.1)	(1483.8)	
%-gap	139.6%	147.8%	27.2%	17.7%	73.7%	**0.0**	67.7%
Quotient box crossover	2866.7	29953.3	40306.7	44333.3	39906.7	81413.3	
	(120.9)	(263.3)	(154.3)	(347.6)	(208.0)	(2510.4)	
%-gap	129.9%	**0.0**%	25.6%	19.5%	37.3%	9.6%	37.0%
Boundary-extended	1246.7	32886.7	34326.7	45513.3	29073.3	88426.7	
quotient box crossover	(57.9)	(514.0)	(143.4)	(426.2)	(213.8)	(8181.7)	
%-gap	**0.0**%	9.8%	7.0%	22.6%	**0.0**%	19.1%	9.8%
p-value in ANOVA test†	8.31e−75	3.75e−44	2.00e−116	4.39e−20	7.00e−142	2.12e−8	

Results from $n = 30$ runs. The smaller, the better. (Ave: average, σ: standard deviation.)
%-gap means the percentage difference-ratio 100 × (output − best_result)/best_result.
† From one-way ANOVA test among the results of four types of box crossovers.
Small p-values (e.g., less than 0.01) indicate the results of tested crossovers have different averages.

crossover performed better than traditional crossovers, and boundary-extended quotient box crossover performed better than quotient box crossover for those functions. To make sure our experimental results are statistically significant, we performed one-way ANOVA test among the results of four types of box crossovers on each function. All the p-values from ANOVA tests are very small (much less than 0.01). That is, the results of tested crossovers have different averages (see the last row of

Table 4: Qualities of Boundary-extended Quotient Box Xover on F_{Schwefel} w/ Different Extension Rates

Extension rate ϵ_i	$0.025 \times (u_i - l_i)$	$0.05 \times (u_i - l_i)$	$0.1 \times (u_i - l_i)$	$0.2 \times (u_i - l_i)$	$0.4 \times (u_i - l_i)$
# of evaluations: Ave($\sigma/\sqrt{n}$)	30333.3 (316.4)	31320.0 (369.7)	32886.7 (514.0)	38460.0 (898.8)	117186.6 (8845.8)
Extended volume / original volume	$1.025^{10} \approx 1.3$	$1.05^{10} \approx 1.6$	$1.1^{10} \approx 2.6$	$1.2^{10} \approx 6.2$	$1.4^{10} \approx 28.9$

Results from $n = 30$ runs. The smaller, the better. (Ave: the average, σ: the standard deviation.)

Table 3).

Quotient box crossover performed the best on the test function F_{Schwefel} whose optimum is located near the boundary of the domain. Quotient box crossover and boundary-extended quotient box crossover were better than traditional crossovers for that function. However, boundary-extended box crossover was worse than quotient box crossover. This seems to be caused by the boundary extension having little influence since the optimum is not at the boundary of the domain. This phenomenon arises from the boundary extension increasing the size of the search domain. Table 4 shows how the solution quality of boundary-extended quotient box crossover on the function F_{Schwefel} changes for different extension rates. As mentioned above, extension rate hardly affected performance in most cases. However, in the case of F_{Schwefel}, as the extension rate is smaller, we could obtain better performance.

We also present the results of reference tests without crossover or mutation. We conducted additional experiments to examine the impact of crossover bias on performance and the interaction between crossover and mutation. They include GAs without crossover and those without mutation. In the GAs only with mutation, we used Type II mutation of Figure 10 and five different mutation rates that decrease according to generations. Table 5 shows the results from 30 runs. As a measure of performance for these tests, we used the function values of the best solutions after the fixed number of function evaluations, because they rarely found the optimum. To test function F_3 fairly, we terminated the genetic search after exactly 2,000 function evaluations. For the tests of the other functions, we used exactly 50,000 function evaluations.

On the tests without crossover, we could obtain quite good qualities, but in most cases, improved results were obtained when we combine crossover together with mutation, though the results on $F_{\mathrm{Rastrigin}}$ always became worse. On the tests without mutation, we could check that the crossover bias of traditional crossovers largely affects their performance. When the optimum is located at the center of the domain, traditional box crossover could find better solutions than quotient box crossover and its boundary-extended version. However, when the optimum is located at or near the boundary of the domain, these operators performed quite poorly. In the latter case, since quotient box crossover has no crossover bias toward the center, it performed better than traditional box crossover. Boundary-extended quotient box crossover further improved on it. In summary, only when the optimum is located at the center of the domain, traditional box crossover performed best as we expected from its crossover bias.

We conducted additional experiments on eight additional test functions chosen from [74]. The additional functions are shown in the top of Table 6. The first four functions (Shifted Sphere, Shifted Schwefel, Shifted Rotated Elliptic, and Shifted Schwefel2) are unimodal. The others (Shifted Rosenbrock, Shifted Rotated Griewank, Shifted Rastrigin, and Shifted Rotated Weierstrass) are multimodal. All the functions have the optima near the middle of the center and the boundary of the domain. These

Table 5: Results of Reference Tests without Crossover or Mutation

	Function	F_3	F_{Schwefel}	$F_{\text{M-Sphere}}$	F_{Sphere}	$F_{\text{M-Rastrigin}}$	$F_{\text{Rastrigin}}$
Crossover	Type	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)	Ave($\frac{\sigma}{\sqrt{n}}$)
Function value of the optimum		−30.00	−4189.83	0.00	0.00	0.00	0.00
Mutation only	rate $0.05/k$	−22.72	−4164.74	0.69	0.74	19.46	19.17
		(0.21)	(2.63)	(0.04)	(0.04)	(0.55)	(0.53)
	rate $0.10/k$	−23.32	−4174.86	0.50	0.73	17.75	18.61
		(0.20)	(1.66)	(0.03)	(0.04)	(0.48)	(0.51)
	rate $0.20/k$	−24.70	−4175.23	0.64	1.24	20.56	25.44
		(0.20)	(1.25)	(0.03)	(0.05)	(0.59)	(0.56)
	rate $0.50/k$	−25.37	−4148.51	2.04	4.57	39.75	51.46
		(0.15)	(2.59)	(0.06)	(0.15)	(0.84)	(1.00)
	rate $1.00/k$	−24.64	−3999.54	6.34	12.43	73.35	90.36
		(0.17)	(11.06)	(0.21)	(0.36)	(1.19)	(1.11)
Extended-box crossover	w/o mutation	−23.00	−4166.14	0.00	0.00	22.93	26.43
(BLX-1/2)		(0.18)	(7.92)	(0.00)	(0.00)	(0.52)	(2.27)
	w/ mutation	−24.63	−4186.00	0.00	0.00	19.75	54.83
		(0.14)	(0.98)	(0.00)	(0.00)	(0.35)	(2.10)
Box crossover	w/o mutation	−21.08	−2537.27	46.28	0.01	83.36	5.13
		(0.18)	(39.46)	(0.81)	(0.00)	(0.36)	(0.43)
	w/ mutation	−28.37	−4024.81	0.00	0.00	0.08	11.60
		(0.14)	(33.23)	(0.00)	(0.00)	(0.00)	(1.77)
Quotient box crossover	w/o mutation	−25.50	−4125.13	19.17	0.12	57.11	15.61
		(0.06)	(15.66)	(0.27)	(0.01)	(1.30)	(0.82)
	w/ mutation	−28.77	−4189.83	0.00	0.00	0.00	16.89
		(0.15)	(0.00)	(0.00)	(0.00)	(0.00)	(1.89)
Boundary-extended	w/o mutation	−26.00	−4149.70	16.49	0.19	33.14	20.04
quotient box crossover		(0.05)	(13.80)	(0.30)	(0.02)	(1.10)	(0.83)
	w/ mutation	−30.00	−4189.83	0.00	0.00	0.00	16.07
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(2.00)
# of function evaluations		2000	50000	50000	50000	50000	50000

Results from $n = 30$ runs. The smaller, the better. (Ave: the average, σ: the standard deviation.)
$k = 1 + \lfloor \mathit{numberOfGenerations}/100 \rfloor$.

functions are suitable for examining the impacts of crossover bias on performance. Table 6 shows the results of GAs only with crossovers. As a measure of performance for these tests, we used the function values of the best solutions after a fixed number of function evaluations. The values are from 30 runs using exactly 100,000 function evaluations. Quotient crossovers performed better than original ones in all the cases except the function of Shifted Rotated Weierstrass. In many cases, the boundary extension improved the performance further. In particular, this is clear in the case of the function of Shifted Rotated Griewank. It may be due to the Shifted Rotated Weierstrass having the optimum much closer to the center than other functions and the Shifted Rotated Griewank having the optimum farthest from the center (see the second-to-rightmost column of Table 6). To support our experimental results, we also conducted one-way ANOVA test among the results of three types of box crossovers on each function. All the p-values from ANOVA tests are very small (much less than 0.01), indicating that the results of the tested crossovers have different averages (see the rightmost column of Table 6).

From the results of Table 3, Table 5, and Table 6, we can state that boundary-extended quotient box crossover and quotient box crossover are suitable for the functions whose optima are located at or near the boundary of the domain. Boundary-extended quotient box crossover and quotient box crossover did not perform well for the functions whose optima are located at the center of the

Table 6: Additional Test Functions and Results without Mutation

Function	n	Range of x_i: $[l_i, u_i]$
Shifted Sphere $= \sum_{i=1}^{n} (x_i - o_i)^2 - 450$	30	$[-100, 100]$
Shifted Schwefel $= \sum_{i=1}^{n} (\sum_{j=1}^{i} (x_j - o_j))^2 - 450$	30	$[-100, 100]$
Shifted Rotated Elliptic $= \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} z_i^2 - 450$	30	$[-100, 100]$
Shifted Schwefel2 $= (1 + 0.4\lvert N(0,1)\rvert) \sum_{i=1}^{n} (\sum_{j=1}^{i} (x_j - o_j))^2 - 450$	30	$[-100, 100]$
Shifted Rosenbrock $= \sum_{i=1}^{n-1} \big(100((x_i - o_i + 1)^2 - (x_{i+1} - o_{i+1} + 1))^2 + (x_i - o_i)^2\big) + 390$	30	$[-100, 100]$
Shifted Rotated Griewank $= \sum_{i=1}^{n} z_i^2/4000 - \prod_{i=1}^{n} \cos(z_i/\sqrt{i}) + 1 - 180$	30	$[-600, 600]$
Shifted Rastrigin $= \sum_{i=1}^{n} \big((x_i - o_i)^2 - 10\cos(2\pi(x_i - o_i)) + 10\big) - 330$	30	$[-5, 5]$
Shifted Rotated Weierstrass $= \sum_{i=1}^{n} \sum_{k=0}^{20} \frac{1}{2^k} \cos(2\pi 3^k (z_i + \frac{1}{2})) - n \sum_{k=0}^{20} \frac{1}{2^k} \cos(\pi 3^k) + 90$	30	$[-0.5, 0.5]$

$\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o})M$, where M is an orthogonal matrix.
$\boldsymbol{o} = (o_1, o_2, \ldots, o_n)$ is the optimal solution.

		Quotient	Boundary-extended	Position of	p-value in
	Box crossover	box crossover	quotient box crossover	optimum†	ANOVA test‡
Function	Ave $(\sigma/\sqrt{n})$	Ave $(\sigma/\sqrt{n})$	Ave $(\sigma/\sqrt{n})$	$\frac{d(\text{center},\text{optimum})}{E_X(d(\text{center},X))}$	
Shifted Sphere	2.13e+4 (2.20e+2)	5.67e+3 (1.65e+2)	4.35e+3 (1.64e+2)	0.95	2.94e−82
Shifted Schwefel	2.27e+4 (2.05e+2)	1.52e+4 (3.58e+2)	1.77e+4 (3.75e+2)	0.94	1.69e−27
Shifted Rotated Elliptic	1.72e+8 (4.94e+6)	4.89e+7 (1.63e+6)	5.16e+7 (1.71e+6)	0.86	1.13e−42
Shifted Schwefel2	2.44e+4 (1.76e+2)	1.55e+4 (3.56e+2)	1.82e+4 (3.45e+2)	0.94	2.81e−33
Shifted Rosenbrock	4.52e+9 (9.59e+7)	1.19e+9 (4.17e+7)	1.02e+9 (3.40e+7)	0.96	6.12e−59
Shifted Rotated Griewank	9.32e+2 (1.41e+1)	2.22e+2 (8.71e+0)	1.73e+2 (5.93e+0)	1.15	5.60e−73
Shifted Rastrigin	−1.88e+2 (1.06e+0)	−2.53e+2 (1.60e+0)	−2.57e+2 (1.53e+0)	0.96	4.00e−55
Shifted Rotated Weierstrass	1.15e+2 (2.22e−1)	1.29e+2 (2.59e−1)	1.17e+2 (7.71e−1)	0.76	7.32e−40

Results from $n = 30$ runs. The smaller, the better. (Ave: the average, σ: the standard deviation.)
The number of function evaluations was 100,000 for all tests.
† The smaller the value is, the closer to the center the optimum is. (d is the Euclidean distance.)
‡ From one-way ANOVA test among the results of three types of box crossovers.
Small p-values (e.g., less than 0.01) indicate the results of tested crossovers have different averages.

domain. However, they were not too bad and the averages over %-gaps for all test functions (see the right-most column of Table 3) were better than other traditional crossovers. In general, it is unlikely that the optima are at the center of the domain for real-world applications.[9] Moreover, when we assume that the optimum is randomly located in the domain, the probability of the optimum being in the neighborhood of the boundary is much larger than the probability being near the center, and surprisingly it becomes quite close to 1 when the domain dimension is sufficiently large (see Section 5). So we can conclude that the proposed crossover is a good choice.

[9] Most of the real-world problems are constrained [47, 64], and most of the constrained problems have optima on the boundaries.

Table 7: Results on Nonlinear Regression on Real-world Dataset

	Extended-box crossover		Box crossover		Quotient box crossover		Boundary-extended quotient box crossover		p value in ANOVA test†
Sum of	Ave	$\sigma/\sqrt{n}$	Ave	$\sigma/\sqrt{n}$	Ave	$\sigma/\sqrt{n}$	Ave	$\sigma/\sqrt{n}$	
squared errors	36.51	1.38	1.18	0.036	0.15	0.0052	0.041	0.0012	1.15e−72

Results from $n = 30$ runs. The smaller, the better.
(Ave: the average, σ: the standard deviation.)
† From one-way ANOVA test among the results of four types of box crossovers.
Small p-value (e.g., less than 0.01) indicates that the results of tested crossovers have different averages.

To demonstrate how the proposed methodologies work on real-world applications, we conducted a nonlinear regression [32] on a real-world dataset. We used the *forest fires* dataset first introduced in [59] and provided by the UCI repository of machine learning databases [12], which is widely used in data-mining research [9, 52, 63, 85]. The number m of real-valued input attributes is 8 and every sample has its real-valued burn area. The number s of samples is 517. The tested nonlinear regression model is as follows:

$$\hat{y} = c_0 + \sum_{i=1}^{m} c_i x_i + \sum_{i<j} c_{ij} x_i x_j.$$

Therefore, the number of coefficients to predict becomes 37 ($= 1+m+m(m-1)/2$). We set the range of each coefficient to [0,100]. As a measure of performance for these tests, we used the sum of squared errors ($\sum_{j=1}^{s}(y(j) - \hat{y}(j))^2$) of the best function obtained after the fixed number of evaluations. The values are from 30 runs where each run uses exactly 100,000 evaluations. Table 7 shows the results of GAs with the same genetic parameters only varying crossover and mutation. We can see that quotient box crossover outperformed extended-box and box crossovers. Boundary-extended quotient box crossover improved more than the quotient one. We can guess that the optimum of this real-world problem is located at or near the boundary of the domain. The results show the potential of the proposed methodologies on real-world problems.

7 Conclusions

We emphasize the following aspects of our research contribution. First, we have shown that traditional crossovers specified for the Minkowski metrics are biased operators: they produce offspring toward the center of the space with higher probability from uniformly distributed parents. We have explained that the origin of this bias is due to these spaces being non-isotropic: not all points are fully-symmetric in relation to the distance.

Second, we have shown how to completely remove the crossover bias by transforming Minkowski spaces. They can be made isotropic by gluing their opposite sides together and considering the distances associated with these glued spaces. We have then studied formally and in full generality the unbiased crossovers associated with these new spaces.

Third, the fitness functions on glued spaces may present discontinuity at the boundaries of the original spaces. Using a recursive method, we extended the boundary of the original space so that the continuity of the fitness function on the corresponding boundary-extended glued space is preserved.

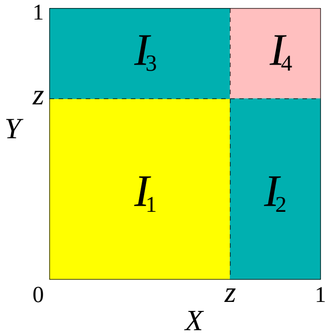


Figure 11: Sub-domains of parents when offspring z is given

Finally, we performed extensive experimentation including a real-world problem, and we could show that crossovers defined on glued space perform well on the functions whose optima are not near the center of the domain. We could also show that, on the functions whose optima are at the boundary of the domain, boundary extension preserving continuity improves the performance of crossovers defined on glued space. We would like to clarify that in this paper, we intended to show the potentiality of well-designed unbiased crossovers rather than to design a method outperforming state-of-the-art methods such as [2]. Since the genetic framework we used, the same as that of [78], is old-fashioned and uses simple genetic operators except for the new crossover, we expect that if the proposed crossover is combined with good operators such as self-adaptive mutation of CMA-ES [26, 34], it will be much more improved. Moreover, since the proposed idea is to remove the inherent bias of crossover operators in a given bounded domain, we expect that it can also be applied to other complicated crossover operators in the state-of-the-art real-coded GAs, such as G3 with PCX [16], StGA (Stochastic GA) [82], Cellular GA [20], GA with orthogonal crossover [51], and so on. However, we leave such hybridization to future work, since such improvement is beyond the scope of this paper.

A Distribution of Offspring: Box Crossover

In this appendix, we will compute the distribution function of offspring produced by the box crossover to formally show that the crossover is biased toward the center of the domain. First, consider one-dimensional case, i.e., $n = 1$. We assume that parent random variables $X, Y \sim U([0, 1])$. Given offspring z, we can divide parent domain $[0, 1] \times [0, 1]$ into four sub-domains as in Figure 11. Then,

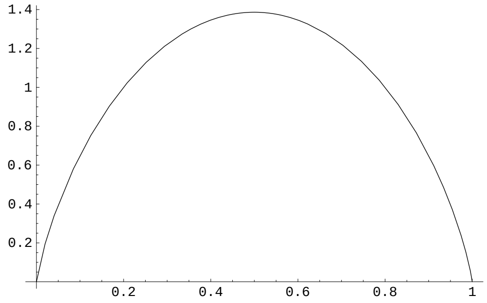


Figure 12: Probability density function $f(z)$ of offspring

the cumulative distribution function for offspring random variable Z is computed in the following.

$$\begin{aligned}
Pr(Z \le z) &= Pr(Z \le z|X, Y \in I_1) + Pr(Z \le z|X, Y \in I_2) \\
&\quad + Pr(Z \le z|X, Y \in I_3) + Pr(Z \le z|X, Y \in I_4) \\
&= \int_{I_1} 1 + \int_{I_2} \frac{z-Y}{X-Y} + \int_{I_3} \frac{z-X}{Y-X} + \int_{I_4} 0 \\
&= \int_0^z \int_0^z 1 dxdy + \int_0^z \int_z^1 \frac{z-y}{x-y} dxdy + \int_z^1 \int_0^z \frac{z-x}{y-x} dxdy + \int_z^1 \int_z^1 0 dxdy \\
&= z^2 + 2\int_0^z \int_z^1 \frac{z-y}{x-y} dxdy \\
&= z^2 + (1-z)^2 \ln(1-z) - z(z\ln z + z - 1).
\end{aligned}$$

Then, the probability density function $f(z)$ of Z becomes

$$\begin{aligned}
f(z) &= \frac{d}{dz} Pr(Z \le z) \\
&= -2(z\ln z + (1-z)\ln(1-z)).
\end{aligned}$$

Figure 12 shows the probability density function $f(z)$ of Z. This is the same as empirically observed in Figure 1(b). It is obvious that box crossover is biased toward the center of the domain. For the general n-dimensional case, we obtain the probability density function $f(z)$ as follows:

$$f(z) = (-2)^n \prod_{i=1}^{n} (z_i \ln z_i + (1-z_i)\ln(1-z_i)).$$

References

[1] B. Akay and D. Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 2010. (in press).

[2] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1769–1776, 2005.

[3] T. Bäck, D. B. Fogel, and T. Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators.* Institute of Physics Publishing, 2000.

[4] P. J. Ballester and J. N. Carter. Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 706–717, 2003.

[5] P. J. Ballester and J. N. Carter. An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 901–913, 2004.

[6] P. J. Ballester and J. N. Carter. An effective real-parameter genetic algorithms for multimodal optimization. In *Proceedings of the Adaptive Computing in Design and Manufacture VI*, pages 359–364, 2004.

[7] H.-G. Beyer. *Theory of Evolution Strategies*. Springer, 2001.

[8] H.-G. Beyer and K. Deb. On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 5(3):250–270, 2001.

[9] B. Chandra and P. P. Varghese. Moving towards efficient decision tree construction. *Information Sciences*, 179(8):1059–1069, 2009.

[10] Y.-P. Chen, W.-C. Peng, and M.-C. Jian. Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(6):1460–1470, 2007.

[11] P. C. Chu and J. E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.

[12] P. Cortez and A. Morais. A data mining approach to predict forest fires using meteorological data. In *Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence*, pages 512–523, 2007.

[13] S. Das and P. N. Suganthan. Differential evolution - a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[14] S. Dasgupta, S. Das, A. Biswas, and A. Abraham. On stability and convergence of the population-dynamics in differential evolution. *AI Commun.*, 22(1):1–20, 2009.

[15] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[16] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4):371–395, 2002.

[17] K. Deb and H.-G. Beyer. Self-adaptation in real-parameter genetic algorithms with simulated binary crossover. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 172–179, 1999.

[18] K. Deb and A. Kumar. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multi-objective problems. *Complex Systems*, 9:431–454, 1995.

[19] K. Deb, K. Sindhya, and T. Okabe. Self-adaptive simulated binary crossover for real-parameter optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1187–1194, 2007.

[20] B. Dorronsoro and E. Alba. A simple cellular genetic algorithm for continuous optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2838–2844, 2006.

[21] L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Proceedings of the Workshop on Foundations of Genetic Algorithms*, pages 265–283, 1991.

[22] L. J. Eshelman, K. E. Mathias, and J. D. Schaffer. Crossover operator biases: Exploiting the population distribution. In *Proceedings of the International Conference on Genetic Algorithms*, pages 354–361, 1997.

[23] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval-schemata. In *Proceedings of the Workshop on Foundations of Genetic Algorithms*, pages 187–202, 1993.

[24] D. E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5:139–167, 1991.

[25] G. Guanqi and Y. Shouyi. Evolutionary parallel local search for function optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(6):864–876, 2003.

[26] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159–195, 2001.

[27] F. Herrera, M. Lozano, and A. M. Sánchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, 18(3):309–338, 2003.

[28] F. Herrera, M. Lozano, and A. M. Sánchez. Hybrid crossover operators for real-coded genetic algorithms: an experimental study. *Soft Computing*, 9(4):280–298, 2005.

[29] F. Herrera, M. Lozano, and J. L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12(4):265–319, 1998.

[30] T. Higuchi, S. Tsutsui, and M. Yamamura. Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature*, pages 365–374, 2000.

[31] M. N. Howell, T. J. Gordon, and F. V. Brandao. Genetic learning automata for function optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32(6):804–815, 2002.

[32] B.-G. Hu, H.-B. Qu, Y. Wang, and S.-H. Yang. A generalized-constraint neural network model: Associating partially known relationships for nonlinear regressions. *Information Sciences*, 179(12):1929–1943, 2009.

[33] H. Huang, H. Qin, Z. Hao, and A. Lim. Example-based learning particle swarm optimization for continuous optimization. *Information Sciences*, 2010. (in press).

[34] O. Ibáñez, L. Ballerini, O. Cordón, S. Damas, and J. Santamaría. An experimental study on the applicability of evolutionary algorithms to craniofacial superimposition in forensic identification. *Information Sciences*, 179(23):3998–4028, 2009.

[35] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[36] C. Igel, T. Suttorp, and N. Hansen. A computational efficient covariance matrix update and a $(1+1)$-CMA for evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 453–460, 2006.

[37] J. Jägersküpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007.

[38] C. Z. Janikow and Z. Michalewicz. An experimental comparison of binary and floating point representations in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 31–36, 1991.

[39] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu. Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. *Information Sciences*, 2010.

[40] F. Kang, J. Li, and Z. Ma. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences*, 2010. (in press).

[41] H. Kita. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms. *Evolutionary Computation*, 9(2):223–241, 2001.

[42] H. Kita, I. Ono, and S. Kobayashi. Theoretical analysis of the unimodal normal distribution crossover for real-coded genetic algorithms. In *Proceedings of the International Conference on Evolutionary Computation*, pages 529–534, 1998.

[43] H. Kita, I. Ono, and S. Kobayashi. Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms. In *Proceedings of the Congress on Evolutionary Computation*, pages 1581–1587, 1999.

[44] O. Kramer. Premature convergence in constrained continuous search spaces. In *Proceedings of the Parallel Problem Solving from Nature*, pages 62–71, 2008.

[45] O. Kramer. *Self-Adaptive Heuristics for Evolutionary Computation.* Springer, 2008.

[46] O. Kramer, B. Gloger, and A. Goebels. An experimental analysis of evolution strategies and particle swarm optimisers using design of experiments. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 674–681, 2007.

[47] R. A. Krohling and L. S. Coelho. Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(6):1407–1416, 2006.

[48] S. Kukkonen and J. Lampinen. An extension of generalized differential evolution for multi-objective optimization with constraints. In *Proceedings of the Parallel Problem Solving from Nature*, pages 752–761, 2004.

[49] S. Kukkonen and J. Lampinen. GDE3: the third evolution step of generalized differential evolution. In *Proceedings of the Congress on Evolutionary Computation*, pages 443–450, 2005.

[50] C. l. Sun, J. Zeng, and J. Pan. An improved vector particle swarm optimization for constrained optimization problems. *Information Sciences*, 181(6):1153–1163, 2011.

[51] Y.-W. Leung and Y. Wang. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 5(1):41–53, 2001.

[52] M. Li and Z. Wang. A hybrid coevolutionary algorithm for designing fuzzy classifiers. *Information Sciences*, 179(12):1970–1983, 2009.

[53] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12(3):273–302, 2004.

[54] S. Meyer-Nieberg and H.-G. Beyer. Self-adaptation in evolutionary algorithms. In *Proceedings of the Parameter Setting in Evolutionary Algorithms*, pages 47–75, 2007.

[55] E. Mezura-Montes, M. E. Miranda-Varela, and R. d. C. Gómez-Ramón. Differential evolution in constrained numerical optimization: An empirical study. *Information Sciences*, 180(22):4223–4262, 2010.

[56] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, 1996.

[57] H. Mühlenbein. The breeder genetic algorithm - a provable optimal search algorithm and its application. In *IEE Colloquium on Applications of Genetic Algorithms*, pages 5/1–5/3, 1994.

[58] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm I: Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.

[59] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998. http://archive.ics.uci.edu/ml/.

[60] N. Noman and H. Iba. Enhancing differential evolution performance with local search for high dimensional function optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 25–29, 2005.

[61] I. Ono, H. Kita, and S. Kobayashi. A robust real-coded genetic algorithm using unimodal normal distribution crossover augmented by uniform crossover: Effects of self-adaptation of crossover probabilities. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 496–503, 1999.

[62] I. Ono and S. Kobayashi. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 246–253, 1997.

[63] L. Peng, B. Yang, Y. Chen, and A. Abraham. Data gravitation based classification. *Information Sciences*, 179(6):809–819, 2009.

[64] V. Petridis, S. Kazarlis, and A. Bakirtzis. Varying fitness functions in genetic algorithm constrained optimization: the cutting stock and unit commitment problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(5):629–640, 1998.

[65] A. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part I: Basic properties of selection and mutation. *IEEE Transactions on Neural Networks*, 5(1):102–119, 1994.

[66] A. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part II: Analysis of the diversification role of crossover. *IEEE Transactions on Neural Networks*, 5(1):120–129, 1994.

[67] G. R. Raidl. Weight-codings in a genetic algorithm for the multiconstraint knapsack problem. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 596–603, 1999.

[68] J. Reed, R. Toombs, and N. A. Barricelli. Simulation of biological evolution and machine learning. *Journal of Theoretical Biology*, 17:319–342, 1967.

[69] K. S. N. Ripon, S. Kwong, and K. F. Man. A real-coding jumping gene genetic algorithm (RJGGA) for multiobjective optimization. *Information Sciences*, 177(2):632–654, 2007.

[70] J. Rönkkönen, S. Kukkonen, and K. Price. Real-parameter optimization with differential evolution. In *Proceedings of the Congress on Evolutionary Computation*, pages 506–513, 2005.

[71] K. Socha and M. Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3):1155–1173, 2008.

[72] H. Someya and M. Yamamura. A robust real-coded evolutionary algorithm with toroidal search space conversion. *Soft Computing*, 9(4):254–269, 2005.

[73] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[74] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-p. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report NCL-TR-2005001, Natural Computing Laboratory (NCLab), Department of Computer Science, National Chiao Tung University, May 2005.

[75] P. D. Surry and N. Radcliffe. Real representations. In *Proceedings of the Workshop on Foundations of Genetic Algorithms*, pages 343–363, 1996.

[76] M. Takahashi and H. Kita. A crossover operator using independent component analysis for real-coded genetic algorithm. In *Proceedings of the Congress on Evolutionary Computation*, pages 643–649, 2001.

[77] S. Tsutsui. Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring. In *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, pages 428–437, 1998.

[78] S. Tsutsui and D. E. Goldberg. Search space boundary extension method in real-coded genetic algorithms. *Information Sciences*, 133(3-4):229–247, 2001.

[79] S. Tsutsui and D. E. Goldberg. Simplex crossover and linkage identification: Single-stage evolution vs. multi-stage evolution. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 974–979, 2002.

[80] S. Tsutsui, D. E. Goldberg, and K. Sastry. Linkage learning in real-coded GAs with simplex crossover. In *Proceedings of the Fifth International Conference on Artificial Evolution*, pages 51–58, 2001.

[81] S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 657–664, 1999.

[82] Z. Tu and Y. Lu. A robust stochastic genetic algorithm (StGA) for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 8(5):456–470, 2004.

[83] Liwei Wei, Zhenyu Chen, and Jianping Li. Evolution strategies based adaptive L_p LS-SVM. *Information Sciences*, 181(14):3000–3016, 2011.

[84] A. H. Wright. Genetic algorithms for real parameter optimization. In *Proceedings of the Workshop on Foundations of Genetic Algorithms*, pages 205–218, 1991.

[85] S. Yue, J.-S. Wang, T. Wu, and H. Wang. A new separation measure for improving the effectiveness of validity indices. *Information Sciences*, 180(5):748–764, 2010.

[86] M. Zhang, W. Luo, and X. Wang. Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, 178(15):3043–3074, 2008.