Domain Approximation and Deterministic Progression in Genetic Crossover

Kurt S. Anderson — Assistant Professor
YuHung Hsu — Graduate Student

Department of Mechanical Engineering, Aeronautical Engineering, and Mechanics
Rensselaer Polytechnic Institute, Troy, NY 12180-3590

Abstract

The fitness difference between two strings is usually neglected in most crossover operators. The neglect of this useful information often results in the less efficient propagation of search trials into fitness improved regions, and a general increase in the number of function evaluations required to obtained a converged solution. An alternative crossover strategy is presented which brings this information into consideration to assist locating search trials. The concept of the strategy involves components of solution domain approximation, progressing direction establishment, and the deterministic locating of future search trials. The implementation of this crossover scheme has empirically demonstrated its effectiveness in significantly reducing the required number of function evaluations, and improving solution quality compared with some conventional crossover strategies.

1 Introduction

Genetic-algorithm-based search methods have become increasingly popular in various applications of design optimization. This popularity arises from their flexible, yet reliable, working principles which can be effectively applied on practical design problems where the design space may be nonconvex, or disjoint [9]. The discretized search mode also makes GAs easy to be adapted to problems with discrete or integer types design variables [11].

Inspired by the biological evolution, *Genetic Algorithms* (GAs) emulate this natural process to evolve a set of candidate designs on a generation basis [10]. In a manner similar to their natural counterpart, the new generations constitute design alternatives

which ideally contain members which are better adapted to the evolutionary environment. Such an environment, represented by a fitness landscape, can be defined in terms of the objective function value and/or the level of constraint satisfaction. GAs evolve a population of designs into a direction of improving fitness on the fixed-length character strings in a domain-independent way. Selection operator serves as a simple transformation function by choosing more fit members of current population and allowing their genes to be propagated into subsequent generations with higher probability. Crossover and mutation operators are introduced to alter genetic structures so to facilitate GAs on exploring other promising designs. It is intuitive from these operators that good designs in present generation can be combined and modified to form possibly better designs and ultimately converged the population to a design which better represents the globally optimal solution.

One primary operator distinguishing GAs from other stochastic search methods is the crossover operator. Crossover allows selected members of the population to exchange characteristics of the design among themselves, and partially preserves distinctive aspects of each original design. The strength of crossover is largely attributed to the fact that it does not limit new designs to be at the vicinity of current design points; instead, the approach uses the information from two regions to advance the characteristics of new designs. Such a process permits the new design points to be escaped from local optima. Crossover is therefore regarded as the most significant factor to contribute to the search aspect of GAs [14][16]. With the recognition of the importance of crossover, a variety of investigations have been presented in the literature in which efforts were devoted to either exploiting the understanding of its mechanism (e.g., [7][18][19]), or proposing modified versions to gain better search performance (e.g., [1][2][5]).

In most of the GAs' applications appearing in the literature, crossovers are performed based on the roles of determining crossover points probabilistically. This class of crossover are not restricted to a specific type of string representations, and are independently performed from the evolutionary environment. These strategies are robust in nature, but the fitness differences between crossover pairs is a major neglected factor in these procedures. Since the fitness explicitly defines the solution landscape, differences in its values implicitly indicates potential search directions. The practice of not placing any emphasis on fitness difference neglects potentially useful information and effectively decreases the propagating speed of search trials. This results in an increasing of the number of function evaluations

necessary to evolve an entire set of designs. This situation indeed becomes a costly and cumbersome process if GAs were the sole method used for performing global search in the procedure of optimization.

This paper describes an alternative crossover strategy in an attempt to facilitate GAs on design space search process through the intelligent use of fitness value difference information. The approach arises from an observation that designs with higher levels of fitness and/or constraint satisfaction dominate the future search trials in the subsequent generations. By comparison, if one explicitly considers fitness variations between individual population members, each new trial may be propagated along towards regions which are anticipated to be favored in the *current* generation. The present idea draws upon this concept, using the fitness variations between crossover pairs, and forming a solution domain approximation to aid in the crossover process. The remainder of this paper will describe the basic of probabilistic-base crossover and the concept of the proposed crossover strategy. Numerical examples of several algebraic functions, a slider-crank mechanism design problem, and a planer five degrees-of-freedom suspension system are used to illustrate the solution efficiency of the new method.

2 Crossover Implementation in GAs

2.1 Probabilistic-base Crossover

The basic approach of crossover involves three major steps—(1) select members as parents from the population, (2) randomly determine crossover positions, either single or multiple, for each parent pair, and (3) exchange genetic characteristics to form new members. The selection is done by pairing members from the current population; however, whether any particular pair should perform crossover is governed by the user specified crossover probability P_c ($0 \leq P_c \leq 1$). P_c is in general set to a high value to promote sufficient exchanging of information among population members and maintaining certain level of genetic diversity. The conventional crossover implemented in GAs is based on the probabilistic determination of crossover sites. For a string containing L digits in length, there are $L-1$ possible positions to perform crossover where the probability of the crossover occurred at any two adjacent positions is an equal and unbiased value of $1/(L-1)$. Figure 1 illustrates several common used crossover strategies in GAs—one-point, two-point,

n-point, and uniform which can be categorized as a class of *position-uniform* crossover operator.

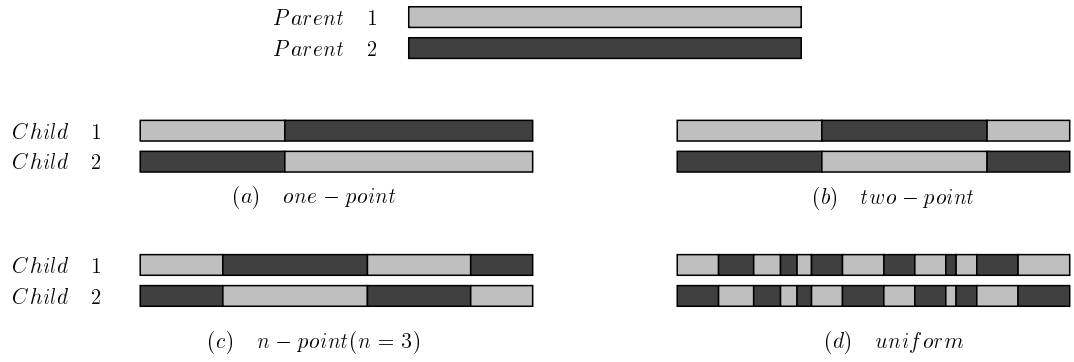

Figure 1: A common class of probabilistic-base crossover

Theoretical analysis of crossover in GAs is centered on the key theorem of schema [7]. Schemata describe the similar templates between two segments of strings from (i) bit values at some specific fixed locations and (ii) their relative distances. In GAs' terminology, the former is defined as *order* and the latter is termed *defining length*. Different schemata with various orders and defining length can be created by crossover operation allowing GAs to examine their averaged performance and gradually form strings with makeup of high order schemata. That is, the role of crossover in GAs is to construct high order schemata from highly fit, short-defining-length schemata (building blocks) [10]. However, high order schemata are likely to be destroyed by crossover which leads to the consideration of disruption in crossover—a basis for theoretical work in the majority of crossover studies.

Holland provided the initial and intuitive analysis of the disruption of 1-point crossover (a particular schema H_k will reside on children as long as the crossover does not occur within the defining boundaries of H_k [10]). This analysis leads to the conclusion that 1-point crossover is disruptive to schemata whose defining positions happen to be far apart. Syswarda proposed the uniform crossover where crossover is taken place following the indication of crossover mask [17]. Although schemata of a particular order are equally likely to be disrupted by uniform crossover irrespective of their defining length, Syswerda argues that total number of schemata disruption is lower due to the fact that longer defining length schemata are comparatively less likely to be disrupted than short defining length schemata. However, Spears and De Jong [16] are critical of multi-point

and uniform crossover based on their theoretical analysis of a general class of n-point crossover. They noted that 1- and 2-point crossover are optimal in the sense of minimizing the disruption high order schemata in which 2-point offers minimum disruption. In spite of these arguments being made from the points of disruption of good schemata, an extensive comparison performed earlier by Eshelman *et al.* [4] on different crossover operators, including 1-point, 2-point, n-point, and uniform crossover, indicated that no overall winner emerged. In fact, Eshelman has shown that the speed difference among these techniques on several test problems is not more than 20%. Therefore, these common used crossover techniques may not provide significant difference on search efficiency to distinguish their performance on practical optimization problems.

2.2 Alternative crossover techniques

Many other crossover techniques have been devised and suggested to improve search performance of GAs. One of the classes involves the memory of crossover positions (hot spots) so some positions are more probable than others. GAs adaptively learn which sites are favored for crossover and record this information in a *punctuation string*. Both original strings and these appended punctuation strings are co-evolved in the subsequent generations that allow biasing crossover to generate the individual with the best outcome. These techniques can be categorized as *position-un-uniform* crossover. Several similar techniques have been described in the literatures [3][12][14].

Another class includes problem specific knowledge to design crossover operators. Davidor [3] designed an analogous crossover for the task in robotic trajectory generation which uses the values of just a few genes to exclude some undesired crossover sites. Grefenstette *et al.* [8] developed a "greedy", heuristic crossover operator for the traveling salesman problems (TSP). The general principle of this crossover depends heavily on knowledge of city distance and uses such an information to construct offspring by choosing the better of two parental edges. Goldberg [7] also described a partially matched crossover for the TSP and several possible ways to use domain knowledge in crossover and mutation. Hajela and Yoo [9] developed a crossover technique using biological expression strategies for the handling of constraint equations in GAs. Each infeasible design is combined with the best feasible design through the use of the expression operation on a bit-by-bit basis. This gives the result that offspring has partial characteristics of feasible design so to in-

crease the level of constraint satisfaction. The underlying goal of these many suggested crossover operators is to facilitate search efficiency and to make GAs a more viable tool to be adapted on practical optimum seeking tasks.

3 Proposed Crossover Strategy

To simplify and clarify the analysis, it is assumed that a design with n design variables is represented by fixed-length binary strings of length L, and the decoded real value of this complete set of design variable is denoted by a vector of X. For many engineering optimization problems, the objective and constraint equations can be explicitly described by algebraic functions of $obj(X)$ and $g(X)$, respectively. Since GAs uses no more than fitness to direct their search, the scalar fitness for a commonly encountered problem of constrained optimization has to be formulated as a composite of the objective and constraint functions. This transforms a constrained into an unconstrained problem with augmented penalty terms in the fitness function [15]. Without loss of generality, the discussion which follows is applied to the consideration of unconstrained maximizing problems with continuous design parameters. A general expression of these problems can be written as

$$\begin{aligned} max \quad & f(X) \\ s.t. \quad & X_i^L \leq X_i \leq X_i^U \end{aligned} \tag{1}$$

where X_i^L and X_i^U are the lower- and upper-bound of parameter X_i, respectively. The function of $f(X)$ is the scalar fitness which may contain values of an objective function only or the combination of raw objective plus the measurement of constraint violation.

As mentioned in the previous section, most of the known crossover operators do not take advantage of fitness difference between a crossover pair to direct the locations of offspring. The significance of involving fitness difference to aid performing crossover can be reasoned by the fact that this information usually indicate potential search directions. That is, offspring may expect to improve their fitness or string characteristics by propagating from one parent with lower fitness to the other parent with higher fitness. However, without prior knowledge of the design or performance hyperplane, the potential search direction is usually not clear, and hence is difficult to accurately determine. The proposed crossover strategy therefore introduces the concept of domain approximation to accommodate this situation.

3.1 Approximation Basis

The simplest form to estimate potential successive search directions is based on a first-order approximation. For a pair of crossover strings with fitness of f_{high} and f_{low}, the normalized potential search direction can be mathematically expressed as

$$\vec{d}_e = \frac{f_{high} - f_{low}}{f_{high}}. \tag{2}$$

Since fitness is the only information used in GAs to conduct their search, Eq. (2) is a legitimate estimation. By using Eq. (2), progeny after crossover can then be located at the positions of

$$X^{t+1} = X^t_{low} + \alpha(X^t_{high} - X^t_{low})\vec{d}_e. \tag{3}$$

Where X^{t+1} represents the location (parameter vector) of progeny at the generation of $t+1$, $X^t_{high} - X^t_{low}$ is the position difference of parents at generation t, and $\alpha \in [0, 1]$ is a random distance coefficient to determine the location of progeny. This crossover performs by joining a line between the two candidates and choosing two points at random along this line to generate progeny. The idea behind such crossover is that the linear approximation guides the progeny towards better string structures, which generally represent improved solutions. Additionally, it is also believed that the constraint satisfaction level of progeny generated in this particular way is higher than randomly placed counterparts, if two parent strings all represent feasible designs [13][20].

Unfortunately, this linear approximation does not capture the non-linear nature of the search space encountered in most design problems. The estimated search direction may be error prone in which progeny are always biased in the direction of $\vec{d}_e$ without further adjustment. Such bias may cause slow propagation if, for instance, fitness variation of an entire population is small and string distribution does not cover the vicinity of the global optimum. Considering these deficiencies, a crossover operator, which draws upon the similar concept but employs high order domain approximation and deterministic propagation, is presented.

A second-order polynomial function, which is expressed as $aX^2 + bX + c = f$ with undetermined coefficients of a, b, and c, is introduced to describe an approximate relationship between a parameter vector X and its corresponding fitness f. The intuition behind this second-order domain approximation is to describe the nonlinearity of general

search domain which is missing from the first-order approach. In addition, the second-order polynomial equation is also a good representation of local regions containing optimal solutions because most engineering problems perform similar to quadratic responses near these optimal regions. However, there are three coefficients a, b, and c required to be determined in this approach. This is accommodated by selecting *three* strings as parents in each crossover operation where two strings are select randomly as normal and the third string is restricted to the string with highest fitness in the present population. It is important to select the *best* string as the third parent in this crossover strategy because it gives the best to date indication of where should the progeny propagate (i.e., the most promising $\vec{d}_e$).

3.2 Crossover Operation

The proposed crossover varies greatly from a more traditional crossover. The major difference is that crossover in this study employs derivatives and projections to generate new designs (progeny), and each design variable is treated independently during crossover. If the variable vectors of three parent strings are denoted as X_1, X_2, and X_{best} with fitness of f_1, f_2, and f_{best}, respectively, they have to satisfy the quadratic equation as follows:

$$\begin{cases} aX_1^2 + bX_1 + c = f_1 \\ aX_2^2 + bX_2 + c = f_2 \\ aX_{best}^2 + bX_{best} + c = f_{best}\,. \end{cases} \tag{4}$$

Although Eq. (4) describes quadratic relation between variable vector X_i and fitness f_i, values of coefficients as well as progeny vectors unfortunately can not be determined explicitly. This is due to the fact that X_i is an n-element parameter vector and directly applying Eq. (4) causes the situation of only three equations (associated with parents) with n unknowns. This problem can be solved by simply projecting the quadratic function onto the dimension corresponding to each design variable. Such an approach of treating each variable independently produces n sets of quadratic equations in which the entire variable vector can be determined explicitly by solving these sets of equations. In this case, the values of undetermined coefficients in Eq. (4) can be easily obtained but are valid only for the dimension one considered. Considering variable k of parent strings, Eq. (4) can be

rewritten as

$$\begin{cases} a_k X_{1,k}^2 & + & b_k X_{1,k} & + & c & = & f_1 \\ a_k X_{2,k}^2 & + & b_k X_{2,k} & + & c & = & f_2 \\ a_k X_{best,k}^2 & + & b_k X_{best,k} & + & c & = & f_{best}\,, \end{cases} \tag{5}$$

or expressed in the matrix form of

$$\begin{bmatrix} X_{1,k}^2 & X_{1,k} & 1 \\ X_{2,k}^2 & X_{2,k} & 1 \\ X_{best,k}^2 & X_{best,k} & 1 \end{bmatrix} \cdot \begin{bmatrix} a_k \\ b_k \\ c_k \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_{best} \end{bmatrix}. \tag{6}$$

Values of a_k, b_k, and c_k can be quickly and easily obtained from Eq. (6) provided the coefficient matrix $[X]$ is nonsingular. Based on the result of Eq. (6) and the natural property of a quadratic curve, the value of design variable X_k in the progeny strings which is expected to contribute the greatest gain on fitness improvement occurs at the location of

$$X_{progeny,k} = -\frac{b_{parent,k}}{2a_{parent,k}}. \tag{7}$$

Above calculations iterate through n-element of design variable vector to explicitly determine the new design points. Equation (7) does not rely on any random sequence to place the new design points; instead, the propagating direction and the magnitude of propagation of a new design is established explicitly. The procedures of using domain approximation of Eq. (5) and deterministic propagation of Eq. (7) distinguish this crossover strategy from conventional probabilistic-base crossover and the first-order approach. It should also be noted that there exists only one solution in Eq. (7). Thus, the new design obtained from above procedures is only assigned to one of the progeny to avoid any duplication of variable vector. For the other progeny, the value of its k-th element $X_{progeny,k}$ of variable vector X is evaluated by using first-order approach of Eq. (3). For successful operation of this crossover strategy, each variable must not attempt to grow outside the search space so $X_{progeny,k}$ is limited on the side constraint of $[X_k^L, X_k^U]$ at any instance. Figure 2 gives an illustration for locating new search points in a two-dimensional case. From a geometric point of view, the new design point is sought by first moving along $\vec{d}_1$ direction with distance $-b_{parent,1}/2a_{parent,1} = x^*$ from parent P_1 in dimension x. Similarly, these same procedures are carried out in direction $\vec{d}_2$ to complete the process. It is

this ability of fine-tuning propagating directions (i.e., $\vec{d}_e = \vec{d}_1 + \vec{d}_2$) in subsequent iteration that enables new procedures moving search points to a more correct direction. This superior tendency is clearly shown in Figure 2 comparing with other crossover strategies where the offspring are randomly distributed and are limited to the shaded area.

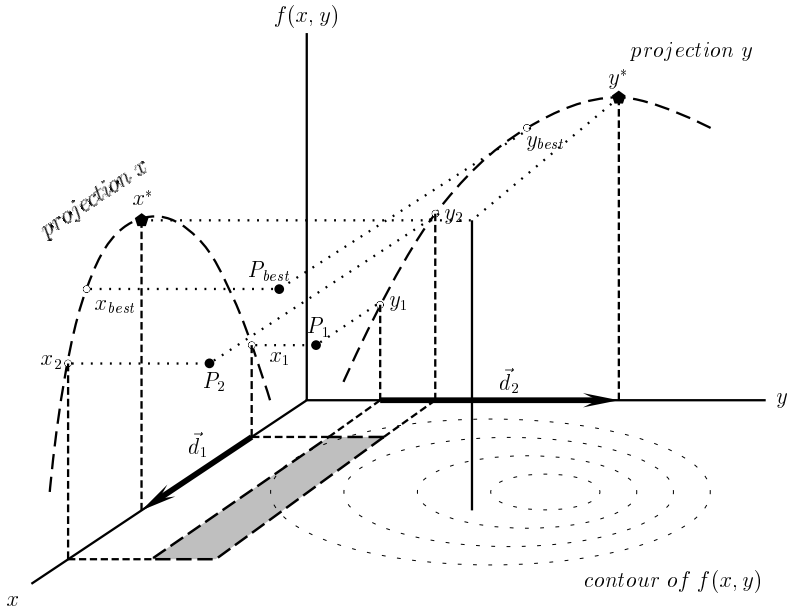

Figure 2: Location of search point produced by the new crossover strategy

3.3 Properties of New Strategy

There are some important properties associated with this second-order crossover strategy. The first is that the locations of progeny are not always limited to the bound of $X^t_{high} - X^t_{low}$ as in the case of first-order approach (recall Eq. (3)). Depending on the values of a_k and b_k, the variable value $X_{progeny,k}$ can be either within the bound (e.g., x^* in dimension x) or in its extensions (e.g., y^* in dimension y) as shown in Figure 2. This provides the flexibility of the procedure which allows focusing on a particular design variable, and uses fitting procedures to maximize its contribution on fitness improvement. Another property of this operation is the capability of exploring new points through the modification of variable vectors X of individual strings. Therefore, new schemata, which do not appear in their parents, can be created as long as all individuals do not have the same fitness. The introduction of new schemata maintains genetic diversity in the population, thus avoiding

the possibility of premature convergence to local optima.

This new strategy also performs similar to conventional probabilistic-base techniques during different search stages. Specifically, at the beginning of a run, when the difference of variable vectors and fitness are typical high, this strategy will also produce new designs with great variations. Likewise, later in the run, when the population is typical more converged and the variation of variable vector is lower, the procedure permits more local exploitation allowing evolution to continue. The quadratic domain approximation in the former case acts like a global approximation scheme while it performs similar to local interpretation in the latter case.

Additionally, the optimal solution can be quickly obtained by proposed crossover strategy once search points are already near it. This is based on the fact that a quadratic function is used to interpret a general nonlinear space. If the space is quadratic in nature, then the optimal solution X_{opt} obtained from aforementioned approximated model is theoretically the exact optimum. This characteristic greatly improves the efficiency of GAs at later search stage without the incorporation of additional costly computation of gradients. Moreover, this strategy is highly efficient on performing multi-dimensional search in parallel. The procedure emphasizes on each design variable and simultaneously maximizes the payoff associated with each variable in one crossover operation. Thus, it is expected that GA with this crossover strategy is possible to discover near-optimum solutions with less computational effort than conventional probabilistic-base schemes.

To offer better robustness for this crossover strategy on handling various types of problems, two different schemes may be possible for its implementation. One is to apply the proposed strategy on *entire* population while the other way applies this strategy on only a fraction of the population with traditional probabilistic-base crossover being performed on the remaining strings. The former offers greatest potential on exploiting an identified promising area, but may suffer information lost from other areas. The latter circumvents this shortcoming which allows other domain information to be integrated for approximation improvement and more correct search direction predictions. The latter case is particularly useful for a problem with highly noisy, poorly behaved, or irregular search space. However, the present paper does not implement the second case but mainly focus on the first approach to validate its efficiency and practical usefulness.

4 Test Functions

This section focuses on the behavior of GAs utilizing the proposed new crossover strategy when applied to three test problems. The first is De Jong's test suite [7], which is generally used to test the efficiency and capability of the algorithm and represent the common difficulties in optimization problems. This test suit involves test function with very different characteristics. These functions were chosen because of the significant difficulties these characteristics (e.g., multi-modal, high degree of coupling between variables, noise, discontinuities, etc.) pose to optimization.

The second problem, a slider-crank mechanism, is more application oriented. Paradis and Willmert (1983) described a mechanism design problem with explicitly known objective function and constraints. A four-bar slider-crank mechanism has to be designed so a desired coupler curve is generated. This four-bar mechanism, as depicted in Figure 3, involves eight design variables b_i with the space bounds defined on the respective interval of (1,30), (1,30), (-10,0), (1,30), (0,10), (0,10), (0,3), and (0,3). The coupler point P has to generate the desired curve where the path coordinates are given in Table 2. The desired positions (x_d, y_d) of P are tabulated as a function of the crank rotations $r(k)$ relative to the starting angle b_8. This optimization problem of the slider-crank mechanism is formulated as follows:

$$minimize \qquad f_2 = \sum_{i=1}^{8} \left[x_p(r(k)) - x_d(k)\right]^2 + \left[y_p(r(k)) - y_d(k)\right]^2 . \tag{8}$$

Subject to the movability constraints:

$$g_1 \;=\; -0.85b_2 + b_1 + b_3 - b_6 \leq 0 \tag{9}$$

$$g_2 \;=\; -0.85b_2 + b_1 - b_3 + b_6 \leq 0\,. \tag{10}$$

The movability constraints ensure that the linkage can operate for the complete range of crank rotations. Usually, the values of state or response variables cannot be solved in closed form from the governing equations of the multibody system. However, the coordinates $P(x_p, y_p)$ of this slider-crank mechanism can be analytically derived as a function of rotation $r(k)$.

$$x_p \;=\; x_2 + b_4 \cos(b_7 + \theta_2) \tag{11}$$

$$y_p \;=\; y_2 + b_4 \sin(b_7 + \theta_2)\,. \tag{12}$$

Where x_2, y_2, and θ_2 can be obtained from the following calculations.

$$x_2 = b_1 \cos(b_8 + r) + b_5 \tag{13}$$

$$y_2 = b_1 \cos(b_8 + r) + b_6 \tag{14}$$

$$\theta_2 = \arcsin\left(\frac{b_3 - y_2}{b_2}\right). \tag{15}$$

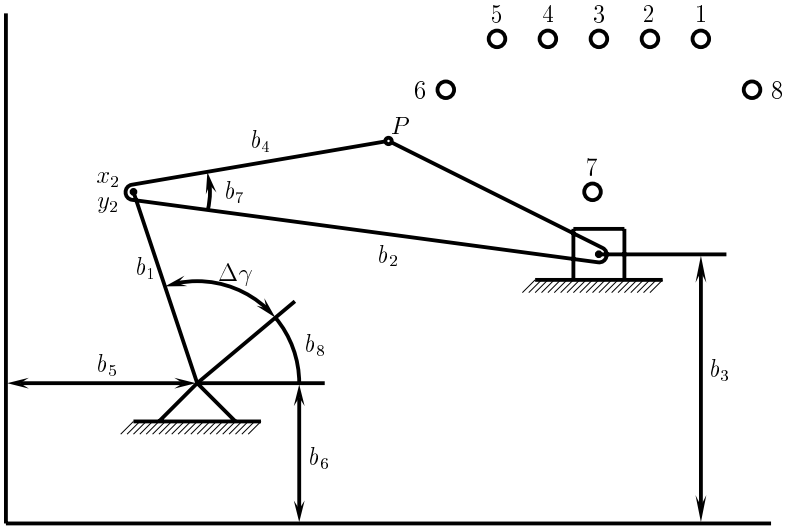

Figure 3: Slider-crank mechanism and defined coupler points

Table 1: Path coordinates and prescribed driving angles

Point k	1	2	3	4	5	6	7	8
$x_d(k)$	26	23	20	17	14	10	20	30
$y_d(k)$	16	16	16	16	16	13	7	13
$r(k)$ in degrees	0	22	44	66	88	120	221	314

The third problem is the design of a planar five degree-of-freedom vehicle model for transient dynamic response. The suspension system of this vehicle model of Figure 4 is to be designed to minimize the extreme acceleration $\ddot{q}_1$ of the driver's seat for a variety of road conditions. The objective function is thus a measurement of cumulative acceleration level for a period of driving time expressed as:

$$f_3 = min \int_{t=0}^{T} |\, \ddot{q}_1 \,|^2 \, \mathrm{d}t \,. \tag{16}$$

Further, it is desirable to constrain this model for a reasonable optimization result. These constraints involve the maximum acceleration of driver's seat g_1, relative displacements

between the chassis and the driver's seat g_2, the chassis and the front and rear wheels g_3 and g_4, and the road surface and the front and rear wheels g_5 and g_6. These constraints are expressed as follows:

$$g_1 = |\ddot{q}_1| \leq 400\, in/sec^2 \tag{17}$$

$$g_2 = |q_1 - q_2| \leq 2\, in \tag{18}$$

$$g_3 = |q_2 - q_4| \leq 5\, in \tag{19}$$

$$g_4 = |q_2 - q_5| \leq 5\, in \tag{20}$$

$$g_5 = |q_4| \leq 2\, in \tag{21}$$

$$g_6 = |q_5| \leq 2\, in\,. \tag{22}$$

Some numerical data associated with this model are kept fixed during optimization: $m_1 g = 290\,\text{lb}$, $m_2 g = 4500\,\text{lb}$, $m_4 g = m_5 g = 96.6\,\text{lb}$, $\text{I} = 41,000\,\text{lb-in-sec}^2$, $L = 120\,\text{in}$, $k_4 = k_5 = 1500\,\text{lb/in}$, and $c_4 = c_5 = 5\,\text{lb-sec/in}$. Lower and upper bounds for design variables $p = [k_1,\, k_2,\, k_3,\, c_1,\, c_2,\, c_3]$ are [50, 200, 200, 2, 5, 5] and [500, 1000, 1000, 50, 80, 80], respectively.

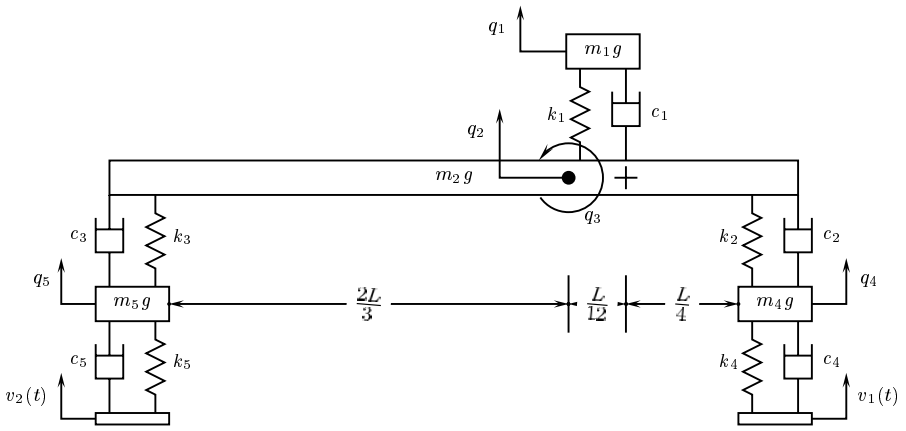

Figure 4: Planar five degrees of freedom vehicle model

A number of experiments were conducted to provide a reasonable basis for comparison the performance of a GA between the new strategy (NS), 1-point crossover (1P), 2-point crossover (2P), and uniform crossover (UC) strategies on these three test problems. The stochastic nature of GAs' search however renders such a comparison study somewhat difficult. To minimize these variations, each function was run 10 times, each with a

different random number seed but using the same j-th seed for j-th run. This is to avoid the situation where successive random number treated some crossover positions preferentially. The experiments used C++ GALib code running on a SUN workstation and the setting of GA's parameters for all three problems was as follows:

coding :	binary, 20-bit per variable
population :	30
crossover rate :	0.6
mutation rate :	0.005
reproduction :	tournament selection with size 2
fitness scaling :	sigma truncation scaling

5 Discussion of Results

5.1 Test Function 1

The performance of the NS on De Jong's test suit showed a promising result to facilitating GAs' search on four of five functions. The NS yielded a significantly greater speed on fitness improvement and discovered the optimal solution with shorter generations than other conventional strategies. The step function posses more difficulties than other functions for NS to conduct its search. It is due to the procedure of domain approximation not being able to provide a good estimation of search space contour. Majority of the offspring is therefore produced by the first-order approach resulting in a less efficient search propagation.

A representative case, De Jong's forth test function, is plotted in Figure 5 to illustrate the ability of NS. This particular algebraic function makes the optimal solution somewhat difficult to find because the uncorrupted function is relatively flat in the neighborhood of the optimum. The difficulties associated with this flatness is then compounded by noises which are then added to the search domain. For this high-dimension, noisy function, NS was able to quickly obtain the optimum and on average obtained better results after fewer than 20 function evaluations than the other schemes were able to achieve in > 270 evaluations. The ability of NS to perform well is due to its usage of domain information to simultaneously maximize the degree of fitness improvement associated with each variable. The accumulation of each improvement contributed to a greatly enhanced search capability and efficiency. By comparison, other crossover strategies require significant more function

evaluations to improve their solution quality. Random exchanging string characteristics in these approaches becomes an ineffective way to direct their search, especially for the increase of variable size.

5.2 Test Function 2

The constraints of this problem were handled by using external penalty method, and dynamically increased the penalty of constraint violation as generation progressed. Thus, a string, which violates the constraint at the early search stage, will not be immediately discarded from the population. Instead, the small penalties at this stage allow such strings to survive for the purpose of improving/repairing their string characteristics. This treatment of constraint violation also ensures that the genetic diversity of a population can be retained so to reduce the chance of premature convergence on the local optimum. On the other hand, a severe penalty will be applied on a string with constraint violation at the latter search stage and hence GA can focus on good quality string to exploit final solution.

As shown in Figure 6, NS performed quite well on this slider-crank mechanism design problem. It provided the fastest rate to improve solution quality and obtained the best solution. NS is able to move progeny to the vicinity of the global optimum within 2 generation and the obtained optimum design (Table 2) nearly identical to the solution obtained by Gabriele [6]. The coupler curve generalized by NS is the most accurate path as depicted in Figure 7. This example has demonstrated the usefulness of the proposed crossover for assisting GAs on optimizing practical design problems.

Table 2: Best designs of slider-crank mechanism

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	f_3	g_1	g_2
NS	9.468	30.00	-6.072	12.68	8.144	9.731	0.734	0.907	7.0732	-31.835	-0.229
1P	9.549	26.27	-0.693	17.64	4.994	4.224	0.659	1.082	10.115	-17.697	-7.864
2P	9.262	22.73	-1.449	18.34	4.989	3.759	0.667	1.167	18.568	-15.255	-4.851
UP	8.244	29.93	-5.023	19.46	3.747	2.829	0.472	1.147	13.834	-25.049	-9.345

5.3 Test Function 3

Three different vertical velocity profiles of the tire are used in this test problem to simulate various road surface disturbances. Such a profile can be generally expressed as

$$v(t) = \begin{cases} v_0 \cos[\pi f_k (t - t^{k-1})] & t^{k-1} \le t \le t^k \qquad k \text{ is odd} \\ v_0 \cos[\pi f_k (t - t^{k-1})] & t^{k-1} \le t \le t^k \qquad k \text{ is even} \end{cases} \tag{23}$$

Profile (1) is for the cases that vehicle is crossing crowned city street at an intersection. It combines two half-wavelength velocity profiles with $v_0 = 6.25$ in/sec, odd period frequency 1.25Hz for 0.8 seconds, and even period frequency 2.5Hz for 0.4 seconds. Profile (2) is a continuous sinusoidal curve with a constant frequency 2Hz and an amplitude $v_0 = 4.0$ in/sec. Profile (3) is also a continuous sinusoidal curve with a high frequency 16Hz and a large amplitude $v_0 = 32.0$ in/sec excitation. The latter two cases represent high way conditions where high amplitude/frequency inputs are generally encountered. The entire simulation lasts for $T = 20$ seconds in which the excitation durations for profiles (1), (2), and (3) are 2.4, 5, and 0.5 seconds, respectively.

The constraints were handled as in the previous case of f_2. Figure 8 shows the averaged result of the best score at each generation. Again, the NS demonstrated that it was better able to obtain a near optimal solution. For a similar solution quality, an estimated speed up factor of 3.0 between NS and the fastest algorithm among 1P, 2P, and uniform was observed. Such an enhanced efficiency is attributed to direct relating search direction with individual's fitness in crossover. A GA with NS could undergo the corrections of its search directions and modifications of variable vector through the discovery of fitness difference and fitting procedures. These two principle characteristics greatly reduced the dependency among initial population distribution, progeny locations, and random number seeds. It can thus conclude that the NS is analogous to knowledge-based search scheme but using a greatly simplified knowledge (i.e., quadratic relationships) to conduct its search instead of some complex or nested logical rules.

6 Closing Remarks

The present paper explores a new genetic crossover which integrates design fitness information to facilitate the search of GA. Key concepts include search space approximation,

estimation of search direction, and deterministic propagation of new search points. Although such an approximation is in general difficult to determine beforehand, the use of a coarse second-order approximation was shown to be a reasonable approach. Comparison of different crossover strategies showed consistent and statistically significant superior performance of the proposed crossover strategy across three test problems. This strategy also demonstrated its efficiency on obtaining better results with much less computational effort than conventional probabilistic-base strategies. Such improved performance was largely due to deterministic locating the new search points instead of random distributing them without guidance. A desirable search property was also obtained through the balance of global approximation (exploration) and local interpretation (exploitation) during various search stages. Results presented here offers evidence of its potential on solving practical engineering problems. Continuing efforts in this area are focused on examining the effect of increased problem dimensionality and complexity on this approach for multibody dynamic optimization problems.

Acknowledgments

Partial support for this work received under National Science Foundation through award No. 9733684 is gratefully acknowledged.

References

[1] Adachi, N. and Yoshida, Y., 1995. "Accelerating genetic algorithms: protected chromosomes and parallel processing," Genetic Algorithms in Engineering Systems: Innovation and Applications, Sept. 12-14.

[2] Anderson, K.A. and Hsu, Y.H., 1998. "Crossover strategy for improved solution space exploration with genetic algorithms," ASME Design Engineering Technical Conference, Sept. 13-16, Atlanta, GA.

[3] Davidor, Y., 1991. "A genetic algorithm applied to robot trajectory generation," Davis, ed., Handbook of Genetic Algorithms, chap. 12, pp. 144-165. Van Nostrand Reinhold.

[4] Eshelman, L.J., Caruna, R., and Schaffer, J.D., 1989. "Biases in the crossover landscape," Schaffer, ed., Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, pp. 10-19.

[5] Esquivel, S.C., Levia, A., and Gallard, R.H., 1997. "Multiple crossover per couple in genetic algorithms," Proceedings of 1997 IEEE International Conference on Evolutionary Computation, pp. 103-106.

[6] Gabriele, G.A., 1993. "Optimization in mechanisms," Erdman eds., Modern Kinematics: Developments in the Last Forty Years, Chap. 11, John Wiley & Sons, New York.

[7] Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA.

[8] Grefenstette, J.J., 1992. "Genetic algorithms for changing environments," Parallel Problem Solving from Nature, 2, pp. 137-144.

[9] Hajela, P., 1990. "Genetic search—an approach to the nonconvex optimization problem," AIAA Journal, Vol. 28, No. 3, pp. 704-711.

[10] Holland, J.H., 1975. Adaptation in Natural and Artificial Systems, Univ. of Michigan, Ann Arbor, MI.

[11] Lin, C.Y. and Hajela, P., 1992. "Genetic algorithms in optimization problems with discrete and integer design variables." Journal of Engineering Optimization, Vol. 19, No. 4, pp. 309-327.

[12] Louis, S.J. and Rawlins, G.J.E., 1991. "Designer genetic algorithms: genetic algorithms in structure design." Belew and Booker, eds., Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, pp. 53-60.

[13] Rasheed, K., Hirsh, H., and Gelsey, A., 1997. "A genetic algorithm for continuous design space search," Artificial Intelligence in Engineering 11, pp. 295-305.

[14] Schaffer, J. and Eshelman, L., 1991. "On crossover as an evolutionary viable strategy," Fourth International Conference on Genetic Algorithm, pp. 61-68.

[15] Smith, A.E. and Tate, D.M., 1993. "Genetic optimization using a penalty function." Proceedings of the 5th International Conference on Genetic Algorithms, Schaffer eds., Morgan Kaufmann, San Mateo, CA. pp. 191-197.

[16] Spears W.M., 1993. "A study of crossover operators in genetic programming," International Symposium on Methodologies for Intelligent System, pp. 409-418.

[17] Syswerda, G., 1989. "Uniform crossover in genetic algorithms," 3rd International Conference on Genetic Algorithms, pp. 2-9.

[18] Whitley, L.D., 1993. Foundations of Genetic Algorithms 2, Morgan Kaufmann.

[19] Whitley, L.D. and Vose, M.D., 1995. Foundations of Genetic Algorithms 3, Morgan Kaufmann.

[20] Wright, A., 1990. "Genetic algorithm for real parameter optimization," The First Workshop on the Foundations of Genetic Algorithms and Classifier Systems, Indiana University, Bloomington, Morgan Kaufmann, pp. 205-218.

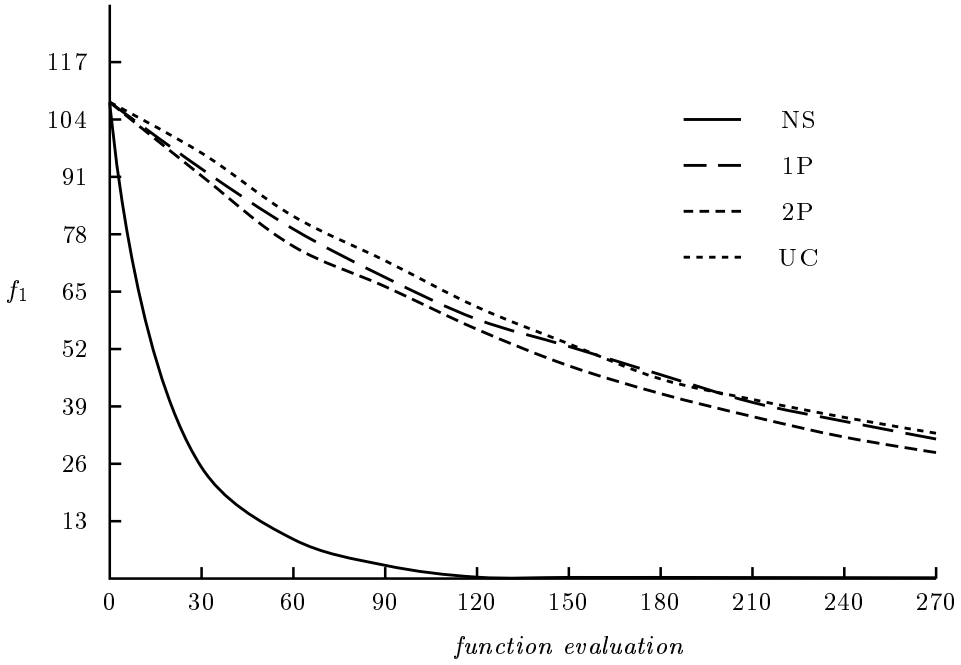

Figure 5: Convergent history for f_1

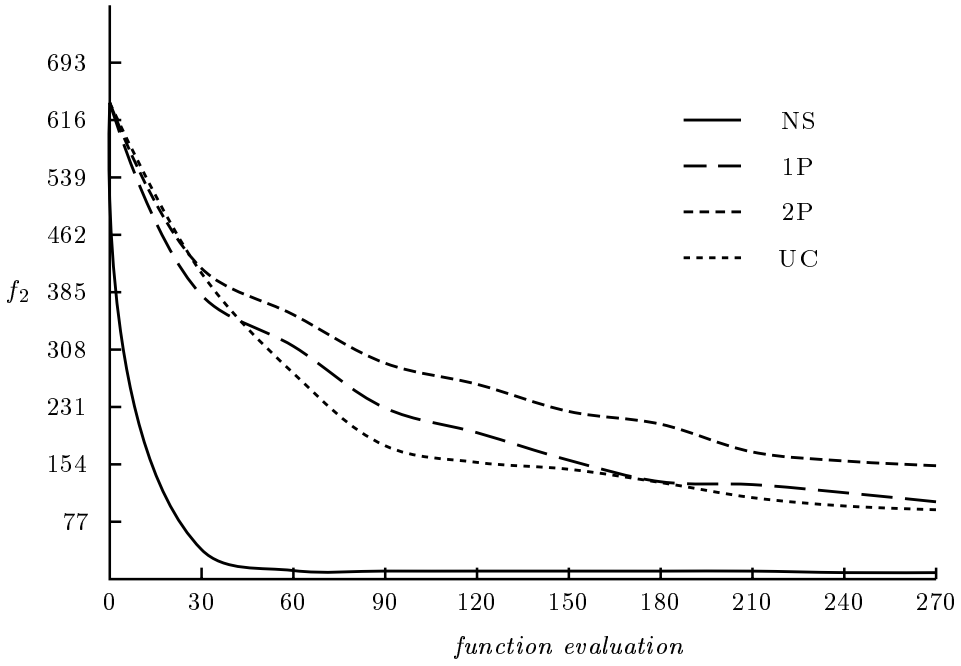

Figure 6: Convergent history for f_2

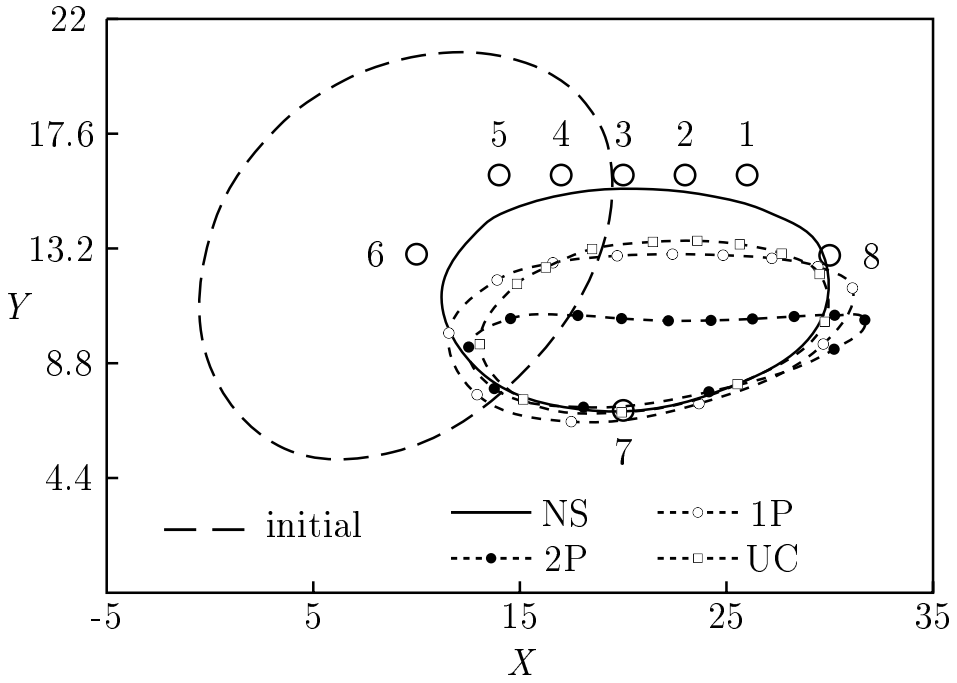

Figure 7: Obtained best coupler curves

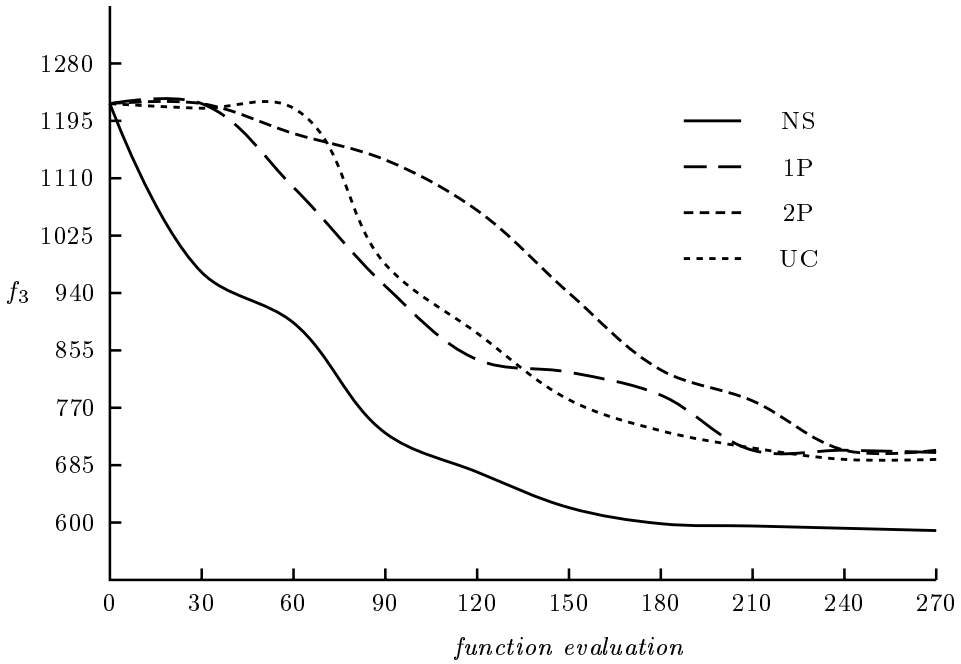

Figure 8: Convergent history for f_4