

AIAA 2002-1675

Bell-Curve Genetic Algorithm for Mixed Continuous and Discrete Optimization Problems

Rex K. Kincaid

*College of William and Mary,
Williamsburg, VA 23187-8795*

Michelle Griffith

*College of William and Mary,
Williamsburg, VA 23187-8795*

Ruth Sykes

*College of William and Mary,
Williamsburg, VA 23187-8795*

Jaroslav Sobieszczanski-Sobieski

*NASA Langley Research Center, M/S 240,
Hampton, VA 23681, AIAA member*

**43rd AIAA Structures,
Structural Dynamics and Materials
April 22–25, 2002/Denver, CO**

Bell-Curve Genetic Algorithm for Mixed Continuous and Discrete Optimization Problems

Rex K. Kincaid*

College of William and Mary, Williamsburg, VA 23187-8795

Michelle Griffith†

College of William and Mary, Williamsburg, VA 23187-8795

Ruth Sykes‡

College of William and Mary, Williamsburg, VA 23187-8795

Jaroslav Sobieszczanski-Sobieski§

NASA Langley Research Center, M/S 240, Hampton, VA 23681, AIAA member

1. Introduction.

This paper is the next installment in a series (Sobieszczanski-Sobieski et al. 1998, Kincaid et al. 2000, 2001a,b and Plassman and Sobieszczanski-Sobieski 2000) that has introduced a variant of the Genetic Algorithm in which the reproduction mechanism was modified to base it on the Gaussian probability distribution, the bell curve. The bell-curve based (BCB) heuristic procedure, first presented in Sobieszczanski-Sobieski, Laba, and Kincaid (1998), is similar in spirit to Evolutionary Search strategies (ESs) and Evolutionary Programming methods (EPs) but has fewer parameters to adjust. In Sobieszczanski-Sobieski et al. (1998) BCB was tested on a structural design optimization problem. The quality of solutions generated were verified by comparing BCB solutions to ones generated by a standard nonlinear programming technique. No attempt was made to analyze the sensitivity of the BCB parameters. Kincaid, Weber and Sobieszczanski-Sobieski (2000) provide a preliminary investigation into BCB parameter selection as well as document improvements in the performance of BCB. Computational results for continuous, discrete and mixed continuous and discrete design optimization problems with constraints is reported. Further experiments with BCB for purely discrete op-

timization problems is provided in Kincaid, Weber and Sobieszczanski-Sobieski (2001a). Plassman and Sobieszczanski-Sobieski (2000) implement and test a parallel version of BCB for continuous optimization problems.

The new contribution to be presented in this installment is an extension to applications that encompass a mix of continuous and quasi-discrete, as well as truly-discrete applications. The extension combines a definition of the distance between the parent designs in the space that comprises discrete and continuous variables with application to optimization of statically indeterminate structures in which the order of design variables, e.g. the type of the cross-section, is essential. As expected, the algorithm that accommodates the order information produces better results. In addition, we provide a comparison allowing sampling from the tails of a discrete normal distribution versus a standard mutation scheme. Adding sampling from the tails brings the continuous and discrete versions of BCB into agreement. Moreover, given the same computing resources, we show that sampling from the tails of the discrete normal leads to higher quality solutions than the standard genetic algorithm mutation approach.

2. Background Information.

To illustrate the connection between BCB, ESs and EPs we define our mutation, recombination, and selection mechanisms in ES and EP terms. A new generation in our approach is selected exactly the same as a $(\mu + \mu)$ -ES. The recombination mechanism is similar to the extension of the intermediate recombination if the weights are required to sum to unity. Consider the line through two n -dimensional parent vectors \vec{P}_1 and \vec{P}_2 selected for mating as in Figure 1. Figure 1 depicts a design space simplified to three

*Department of Mathematics, The first and second authors gratefully acknowledge the support of of NASA-Langley Research Center-NAG-1-2077

†Department of Computer Science

‡Department of Computer Science

§Senior Research Scientist

Copyright © 2002 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental Purposes. All other rights are reserved by the copyright owner.

dimensions (three design variables). First, determine the weighted mean \vec{M} of these two vectors where the weights are given by the fitness of each parent. Next, sample from a normal distribution $N(0, \sigma_m)$. The resulting point $\vec{B} = \vec{M} + |\vec{P}_2 - \vec{P}_1| * N(0, \sigma_m)$ is the child, prior to mutation. Note that in the extended version of the intermediate recombination method when the weights sum to one the unmutated child can lie anywhere on the line segment between the two parents. In our approach, rather than picking weights arbitrarily, the selection is governed by a normal distribution and the fitness weighted average of the parents. Moreover, \vec{B} is *not* restricted to lie on the line segment $\vec{P}_1\vec{P}_2$. Mutation ensues by first generating a radius r for an $n - 1$ dimensional hypersphere. The radius is a realization from a $N(0, \sigma_r)$. Typically ($\sigma_r \gg \sigma_m$). Finally the mutated child \vec{C} is selected by sampling uniformly on the surface of the $n - 1$ dimensional hypersphere. Since the child can lie anywhere on the surface the effect is similar to the rotated angle portion of an ES or EP (in $n - 1$ rather than n dimensions). However, we do not allow the hypersphere to be stretched (or shrunk) along any of its axes as is the case for an ES and EP with non-identical σ_i . We call our procedure a Bell-Curve Based evolutionary optimization algorithm (BCB). Figure 1 is a 3-dimensional view of BCB.

In the case of constraints (or simple bounds) we make two adjustments to BCB. If the point \vec{B} violates a constraint then we select another candidate for \vec{B} until we obtain a feasible one. That is, we sample from $N(0, \sigma_m)$ until $\vec{B} = \vec{M} + |\vec{P}_2 - \vec{P}_1| * N(0, \sigma_m)$ is feasible. If the child \vec{C} that is produced is infeasible, we translate the components of \vec{C} to the closest constraint. For example, if all variables are to lie within $[1, 10]$ and if $\vec{C} = (0.8, 2.0, 11.9)$, the child would become $(1.0, 2.0, 10.0)$.

In addition to the general description of BCB above there are several specific details of note. Instead of a roulette wheel selection scheme for selecting parents we implement Baker's (1987) stochastic universal sampling. In doing so we eliminate the natural bias, noted by Baker (1987), in roulette wheel selection. Although BCB's performance may be improved if knowledge of the underlying feasible region's topology is included in the selection of the initial population, here we generate the initial population randomly (uniformly). There are four parameters that must be chosen to optimize a continuous unconstrained function: population size, number of generations, σ_m , and σ_r . Kincaid et al. (2001b) provide an investigation into BCB parameter selection. We make use of their observations in this manuscript.

First, scaling with respect to the design variables is critical. BCB skews the search space towards design variables of small dimension during recombination. Hence, the search space should be scaled so that all

design variables are of similar magnitude. Two normal distributions drive BCB— $N(0, \sigma_m)$ and $N(0, \sigma_r)$. As we seek to sample along the line through the two parents, \vec{P}_1 and \vec{P}_2 , we form $\vec{B} = \vec{M} + |\vec{P}_2 - \vec{P}_1| * N(0, \sigma_m)$. Computational experience in Kincaid et al. (2001b) showed that σ_m should have a value near 1.0. Intuitively this makes sense since $N(0, \sigma_m)$ serves to inflate or deflate the value of the distance between the parents. Next, to mutate \vec{B} we sample uniformly on the surface of an $n - 1$ dimensional hypersphere with radius sampled from our second normal distribution, $N(0, \sigma_r)$. Here, the value of σ_r is expected to be larger than 1.0 to allow for the possibility of hyperspheres of larger radii. Unfortunately there is no problem specific data from which we can pick a baseline value for the radius of this hypersphere. $N(0, \sigma_r)$ must represent both the unknown radius baseline as well as its normal deviation. As expected, Kincaid et al. (2001b) found that σ_r should be much larger than σ_m . Below is an outline of the BCB procedure for continuous optimization problems.

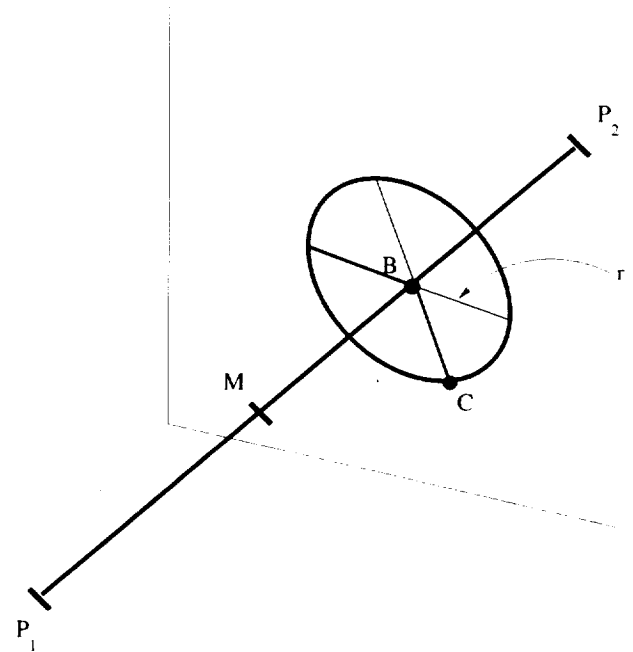


Fig. 1. BCB Geometrical Construct in 3D Space

Initialize BCB by choosing a population size, μ , a maximum number of generations, *num_gens*, σ_r and σ_m . Scale the search space so that all design variables are of similar magnitude. Randomly generate μ individuals.

Repeat *num_gens* times

Repeat μ times

sample among all the objects at a prescribed distance ϵ between the two objects. Kincaid et al. (2000) experimented with discrete problems in which the order of the attributes does NOT matter. In Section 4 we will provide computational experience for problems in which the attribute order does matter.

The tacit assumption in the previous examples is that the number of attribute positions in each object are the same. Although this need not be the case in general, it will be true for the problems we study here. Given the aforementioned distance definition we can now formally describe a discrete version of BCB. We describe the case when the positions of the attributes do matter since that is easier to explain. The symbols P_1 , P_2 , and C (two parents and the resulting child) correspond to α , β , and γ respectively in the previous discussion. Let k denote the number of attribute positions in P_1 and P_2 , let n denote the number of possible attribute values ($n > k$), and let r denote the number of attributes that are dissimilar between P_1 and P_2 .

- Step 1. Place P_1 and P_2 on a numerical axis ranging from 0 to the number of dissimilar components between P_1 and P_2 . Without loss of generality, let P_1 serve as the reference point at 0 on this numerical axis.
- Step 2. Create a truncated discrete normal distribution on the axis described in step 1. The distribution is centered on a point M whose location may be at the midpoint between P_1 and P_2 or it may be shifted toward the fittest parent. The distribution is truncated at P_1 and P_2 . The parameter σ_m must be chosen by the user.
- Step 3. Let ϵ denote the sample value chosen from the distribution in step 2. Place the child B at a distance ϵ from P_1 on the numerical axis between P_1 and P_2 .
- Step 4. Reorder the attributes of P_1 and P_2 so that the $k - r$ attributes in common appear in the last $k - r$ attribute positions. Assign the remaining r attributes of P_1 and P_2 randomly to the first r attribute positions respectively within P_1 and P_2 . Label these reordered parents \bar{P}_1 and \bar{P}_2 . Construct a child B of P_1 and P_2 as follows. First, set $B = \bar{P}_2$. Then re-assign the attributes contained in the first $r - \epsilon$ positions of P_1 to the first $r - \epsilon$ attribute positions of B .
- Step 5. Construct the final child C from B by randomly mutating B . That is, with a small probability of occurrence, p , randomly (uniformly) replace any attribute of B with any of the allowed attribute values.

In addition to experiments in which the attribute order does matter we will also explore the utility of

the truncated normal. In order to make the discrete version of BCB closer in spirit to the continuous one it would be nice to replace Step 5 above with sampling in the tails of the discrete normal distribution.

Lastly, we describe how these two algorithms interact when we solve mixed continuous and discrete optimization problems. we treat the discrete and continuous variables as a single collection of decision variables. Suppose that our design structure has b beams, and that there are t possible beam types. Further suppose that the cross-section of each beam type can be described by n continuous variables. Then our solution is a vector \vec{x} that is a concatenation of b vectors of the form

$$(x_{11}, \dots, x_{1n}; x_{21}, \dots, x_{2n}; \dots; x_{t1}, \dots, x_{tn})$$

with one such vector for each beam. Here, x_{ij} represents the j th cross-section decision variable of the i th beam type. Thus, a solution will describe not just b beams, but $b \times t$ beams. A second vector (y_1, y_2, \dots, y_b) indicates the chosen set of beam types, where y_i is in the integral range of $[1, t]$. Hence, the value of y_i indicates which beam type we have chosen for beam i . Given two parents of the form described above, continuous BCB recombines the \vec{x} 's and discrete BCB recombines the \vec{y} 's. For further information on other possible ways to solve mixed continuous and discrete optimization problems the interested reader is referred to Kincaid et al. (2001a).

3. Refinements to Discrete BCB

In this section we test certain features of discrete BCB. First we examine the two choices for M described in Step 2 of the algorithm. That is, should M be the the midpoint between P_1 and P_2 or should it be shifted toward the fittest parent. Second, we compare the performance of discrete BCB as described above to a new version of discrete BCB in which Step 5 is replaced by allowing sampling to occur in the tails of the discrete normal distribution defined in Step 2. That is, the discrete normal distribution defined in Step 2 is no longer truncated. In addition we will limit our attention to problems in which the attribute order does NOT matter.

The test cases for these experiments is a constrained mixed continuous and discrete weighted shape selection optimization problem. We wish to select 5 shapes from a list of three (circle, triangle, and square) or more geometrical shapes. The five positions are numbered in order 1 through 5. The goal is to choose 5 shapes (possibly with repeats) and the dimensions of those five chosen shapes to minimize the total weighted sum of the perimeters. Each chosen shape's perimeter is weighted by its position in the list: the first shape in the list is weighted by 5, the second by 4, the third

by 3, and so on. The continuous variable is the radius (r) for the circle, the length of a side (s) for the square and the length of the base (b) for a right isosceles triangle. The constraints are as follows. The total area must be greater than 100 units and each continuous variable has a lower bound of 1.0 and an upper bound of 10.0. The optimal set of shapes is 4 triangles and 1 circle with an objective value of 82.9. The circle has the smallest perimeter to area ratio ($2/r$) and is chosen for position 5 with r just large enough so that 100 - (sum of the 4 triangles area) is satisfied. Small triangles are chosen for the remaining 4 shapes since its perimeter to area ratio is largest. In an attempt to understand how BCB scales with respect to the number of discrete variables we solve the position dependent shape problem with the addition of regular polygons with 5, 6, 8, and 10 sides.

Table 1. Using fitness: 15 reps, 20,000 solutions, 0.04 mutation

# Shapes	Mean Best Obj	Min Best Obj	Max Best Obj	Freq Opt Shapes
3	85.9	82.9	89.1	5/15
4	87.4	82.9	94.3	2/15
5	89.7	85.3	95.3	0/15
6	91.2	85.1	97.1	0/15
7	91.0	85.2	97.6	0/15

Table 2. Using midpoint: 15 reps, 20,000 solutions, 0.04 mutation

# Shapes	Mean Best Obj	Min Best Obj	Max Best Obj	Freq Opt Shapes
3	85.7	82.9	89.4	6/15
4	86.1	82.9	90.4	5/15
5	89.1	83.7	93.2	1/15
6	93.0	85.7	98.1	0/15
7	90.8	83.9	96.7	0/15

In the above experiment the fitness weighted mean in Step 2 of the algorithm does not perform quite as well as simply using the midpoint of P_1 and P_2 . The midpoint approach has the min and mean best objective values in 4 of the 5 cases and identifies more optimal shapes (12) than the fitness approach (7). Even so the appeal of making use of parent fitness has led us to continue with its use. One reason for doing so is to make the continuous and discrete implementations of BCB as similar as possible.

The next set of experiments seeks to compare the truncated discrete normal with mutation versus a discrete normal distribution in Step 2 of the algorithm. The table below summarizes these experiments for the discrete normal distribution which can be compared against the results in Tables 1 and 2. Here we see that the use of tails for the discrete normal is beneficial. 26 optimal shapes are found with the tails and only 12 with the mutation approach. In addition, the mean best objective value for the tails approach wins in 4 of

the 5 cases.

Table 3. Discrete Normal with tails: 15 reps, 20,000 solutions

# Shapes	Mean Best Obj	Min Best Obj	Max Best Obj	Freq Opt Shapes
3	84.0	82.9	88.4	13/15
4	85.6	82.9	90.6	7/15
5	89.0	82.9	99.9	5/15
6	91.3	86.0	99.9	0/15
7	91.8	82.9	99.6	1/15

4. Hub Design Problem

Our standard test problem in Sobieszczanski-Sobieski et al. (1998), Kincaid et al. (2000) and Kincaid (2001) has been a minimum weight (volume) design of a hub structure also found in Balling and Sobieszczanski-Sobieski (1994). Each member of the hub structure is an I-beam rigidly attached to the hub and to the wall, forming a two-dimensional, wheel-with-spokes pattern whose two-beam version is illustrated in Figure 3. The structure load has three components, horizontal and vertical concentrated forces, and a concentrated moment, all acting on the hub in the plane of the structure. Various loading cases (henceforth called "loads") are formed from the above load components. Three loading cases are applied to a 6-beam structure, and two loading case are used in case of a 20-beam structure to reduce the computing elapsed time. The beam cross-sectional dimensions are the design variables, and the constraint functions reflect the material allowable stress as well as overall and local buckling. The top and bottom flanges of the I-beam are not necessarily of the same dimensions. Hence, the cross-section of each I-beam requires six design variables. Figure 3 illustrates a 2-member hub problem. Additional details may be found in Padula et al.(1996). The utility of the hub structure as a test case stems from its ability to be enlarged by adding as many members as desired without increasing the dimensionality of the load-deflection equations. These remain 3 by 3 equations for a 2-dimensional hub structure regardless of the number of members. While analytically simple, the hub structure design space is complex because the stress, displacement and buckling constraints are rich in nonlinearities and couplings among the design variables. Our goal is to extend this problem so that, in addition to choosing the dimensions of the beams so as to minimize the volume of the hub frame, we would also select (among a finite list) the beam type.

As a first step in this direction we consider 3 beam cross-section types - I, circular, and triangular. In all cases the overall length of the beam is held constant amongst the cross-section types. The I-beam has 6 design variables; the circular beam has 2 design vari-

ables; and the triangular beam has 3 design variables. Our first experiment is with a 6 beam 3 load hub design problem. Each beam has the same nominal length of 120.0 and the following BCB parameters values:

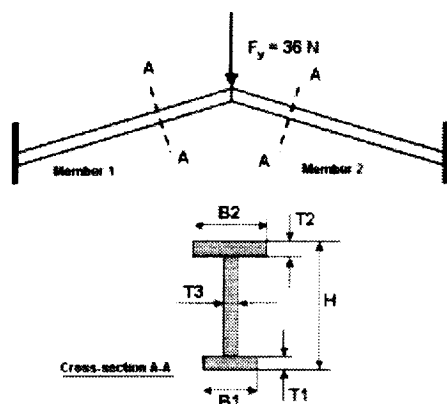


Figure 3. 2-member hub description

population size : 20
 number of generations : 1250
 replications : 50
 PENALTY1 : 2000.0
 PENALTY2 : 4000.0
 fitness-related mean for discrete
 normal number of swaps

As a benchmark we force mixed BCB to choose all I-beams and obtain the following results:

	Order NOT Important	Order Important
min vol	576.62	554.51
mean vol	687.81	681.02
s.d. vol	63.63	126.33

The case in which the order of matching elements in the parents matters leads to slightly better results. This trend has been borne out in all of computational experience although the order is important approach seems to require more generations to converge. We expect this is due to the larger neighborhood of feasible solutions when order counts.

With the penalties shown above, the maximum constraint violations occurred as follows:

	Order NOT Important	Order Important
mean max. violation	-0.00039	-0.0085
s.d. max. violation	0.0027	0.0596

The following table lists the results when mixed BCB is allowed to choose one of three possible shapes (I-beam:1, circle:2, or triangle:3) for each of the six beams when the order does not matter in computing the distance between discrete components.

Table 4: Frequencies of designs chosen (order does not matter)

Design	#	Min Best Vol	Mean Best Vol	Std Dev Best Vol
1,1,1,2,2,2	1	456.90	456.90	-
1,1,2,1,2,2	1	711.10	711.10	-
1,2,1,3,2,2	1	448.84	448.83	-
1,2,2,2,1,2	1	404.51	404.51	-
1,2,2,2,2,2	2	383.29	383.29	0.001
1,3,2,2,2,2	1	429.78	429.78	-
1,3,2,2,3,2	1	464.50	464.50	-
2,1,1,2,2,2	1	416.85	416.85	-
2,1,2,2,2,2	1	462.20	462.20	-
2,1,2,2,2,3	1	592.75	592.75	-
2,1,2,3,2,2	1	447.83	447.83	-
2,1,3,2,2,1	1	477.77	477.77	-
2,2,2,1,2,2	1	417.72	417.72	-
2,2,2,1,3,2	1	464.91	464.91	-
2,2,2,2,1,1	1	461.79	461.79	-
2,2,2,2,1,2	1	383.20	383.20	-
2,2,2,2,2,1	1	471.11	471.11	-
2,2,2,2,2,2	8	361.91	365.98	10.51
2,2,2,2,2,3	1	426.61	426.61	-
2,2,2,2,3,2	2	401.99	422.61	20.62
2,2,2,3,2,1	1	533.66	533.66	-
2,2,2,3,2,2	2	416.74	447.42	30.68
2,2,3,2,2,2	2	401.51	401.51	0.002
2,3,2,2,2,2	2	424.75	426.89	2.14
2,3,2,2,2,3	1	576.26	576.26	-
2,3,3,2,2,3	1	541.52	541.52	-
2,3,3,3,2,2	1	502.52	502.52	-
3,1,2,2,2,2	1	489.31	489.31	-
3,1,3,3,2,2	1	541.69	541.69	-
3,2,2,2,1,2	1	429.18	429.18	-
3,2,2,2,2,2	7	401.49	401.49	1.65
3,3,2,2,2,2	1	527.12	527.12	-

Best Designs:

2,2,2,2,2,2 with a minimum volume of 361.91

3,2,2,2,2,2 with a minimum volume of 401.49

One of these two designs was selected
 30% of the time

The following table lists the results when mixed BCB is allowed to choose one of three possible shapes (I-beam:1, circle:2, or triangle:3) for each of the six beams when the attribute order does matter in computing the distance between discrete components.

**Table 5: Frequencies of designs chosen
(order does matter)**

Design	#	Min Best Vol	Mean Best Vol	Std Dev Best Vol
1,2,2,1,2,2	1	421.07	421.07	-
1,2,2,2,2,2	3	383.19	383.44	0.21
2,2,1,2,1,2	1	404.90	404.90	-
2,2,2,1,1,1	1	479.39	479.39	-
2,2,2,1,2,2	3	399.77	477.93	61.39
2,2,2,2,1,2	2	386.97	436.16	49.19
2,2,2,2,1,3	2	434.42	488.10	53.67
2,2,2,2,2,2	22	361.91	362.00	0.15
2,2,2,2,2,3	1	412.54	412.54	-
2,2,2,3,2,1	1	527.49	527.49	-
2,2,2,3,2,2	3	407.82	427.82	14.48
2,2,2,3,3,2	1	451.82	451.82	-
2,2,3,2,2,1	1	441.57	441.57	-
2,2,3,3,2,2	1	517.07	517.07	-
2,3,1,3,2,2	1	538.54	538.54	-
2,3,2,1,2,2	1	485.36	485.36	-
2,3,2,2,2,2	1	410.65	410.65	-
2,3,2,2,3,2	1	444.29	444.29	-
3,1,2,2,2,2	1	476.33	476.33	-
3,2,2,2,2,2	1	423.71	423.71	-
3,2,2,2,3,2	1	441.13	441.13	-

Best Designs:

2,2,2,2,2,2 with a minimum volume of 361.91

1,2,2,2,2,2 with a minimum volume of 383.19

2,2,2,1,2,2 with a minimum volume of 386.97

2,2,2,1,2,2 with a minimum volume of 399.77

One of these four designs was chosen
60% of the time.

With the same penalties as before, the maximum constraint violations occurred as follows:

	Order NOT Important	Order Important
mean max violation	-0.076	-0.0619
s.d. max violation	0.0849	0.06

The above tables indicate the trends we have seen in variety of problem instances. The "order does matter" approach results in fewer, higher quality solutions and the distribution of final solutions is tighter (smaller variance) with a higher quality mean.

Next, we consider a 20 beam 2 load hub design problem. Again all 20 beams are of the same nominal length. The same BCB parameters were used as in the previous experiment except that Penalty 1 was increased to 10,000 and Penalty 2 was increased to 20,000. With these penalties, the program had a condition added that the maximum constraint violation must be less than 0.02. Again, as a benchmark, the

following results are obtained when mixed BCB is allowed only to choose I-beams:

	Order NOT Important	Order Important
min. vol.	17089.629	17112.098
mean vol.	18984.661	17869.609
s.d. vol.	2051.516	1285.047

The use of the "order does matter" definition of discrete distance yields results that are only slightly better with respect to the mean but are significantly better with regard to the variance in the results.

The following table lists the results when mixed BCB is allowed to choose one of three possible shapes (I-beam:1, circle:2, or triangle:3) for each of the six beams.

	Order NOT Important	Order Important
min. vol.	12865.855	13394.102
shapes	2,2,1,3,2,2,1,3,2,2, 2,3,2,1,1,2,2,2,2,1	2,2,2,2,2,2,1,3,2,1, 2,2,2,3,3,2,2,2,2,2
mean vol.	16140.339	14278.798
s.d. vol.	1694.326	1070.669

Here we can see a significant improvement with the use of the "order does matter" discrete distance definition.

Next we increase the number of choices for the beam type (the only discrete variable) from 3 to 6. The shape choices for each beam are 1:I, 2:circle, 3:triangle, 4:rectangle, 5:U, or 6:C. This problem was run with all 6 beams having the same nominal length and the following BCB parameters:

population size : 20
number of generations : 1250
replications : 50
PENALTY1 : 2000.0
PENALTY2 : 4000.0
fitness-related mean for discrete
normal number of swaps

The experiment was first run for the case when "order does NOT matter" and, of the 50 replications, only the following solutions were repeated:

Design	#	Min Best Vol	Mean Best Vol	Std Dev Best Vol
1,2,2,2,2,2	2	383.22	383.34	0.12
2,2,2,2,6,2	2	456.41	467.58	11.17

The range of values for all the solutions was 361.91 (2,2,2,2,2,2) to 977.27 (5,6,4,5,2,1) with an average value of 533.11 and standard deviation of 119.91.

The same experiment was repeated for the case when order does matter and, of the 50 replications, only the following solutions were repeated:

Design	#	Min Best Vol	Mean Best Vol	Std Dev Best Vol
2,2,1,2,2,2	2	383.20	383.32	0.12
2,2,2,2,2,2	6	361.91	376.94	33.22
2,2,2,2,5,2	2	395.20	415.09	19.89
2,2,5,2,2,2	2	395.20	449.81	54.61

The range of values for all the solutions was 361.91 (2,2,2,2,2,2) to 1322.02 (5,2,6,2,2,2) with an average value of 526.35 and standard deviation of 148.02. If the single outlier value of 1322.02 is discarded, the maximum value is 701.03 (1,2,2,2,3,5) and the average value over the 50 replications is 507.85 with a standard deviation of 85.75. In this manner we have a tighter range of values then with the "order does not matter" definition of distance.

Next we consider the 20 beam 2 load problem and use the same BCB parameters as in our previous 20 beam 2 load case except that 6 shape choices (1:I, 2:circle, 3:triangle, 4:rectangle, 5:U, or 6:C). are now available. This yielded the following results over 20 replications:

	Order NOT Important	Order Important
min. vol.	14949.145	14359.097
shapes	2,4,4,2,2,6,1,2,2,2, 6,2,2,3,2,3,4,6,4,5	4,2,4,5,1,1,4,2,2,4, 2,3,3,2,5,4,3,2,2,6
mean vol.	16827.441	15803.128
max. vol.	19001.340	17734.045
shapes	1,3,4,6,4,6,5,6,2,6, 5,2,2,5,4,4,2,2,1,1	4,5,4,5,4,1,2,1,2,6, 2,2,1,2,2,1,5,2,6,1
s.d. vol.	1233.146	901.371

The use of the "order does matter" definition for discrete distance yields results that are both slightly better with respect to the mean and less spread out (smaller standard deviation) over the multiple runs.

5. Concluding Remarks

In this manuscript we have examined an extension of BCB that encompasses a mix of continuous and quasi-discrete, as well as truly-discrete applications. We began by testing two refinements to the discrete version of BCB. The testing of midpoint versus fitness (Tables 1 and 2) proved inconclusive. The testing of discrete normal tails versus standard mutation showed was conclusive and demonstrated that the discrete normal tails are better. Next, we implemented these refinements in a combined continuous and discrete BCB and compared the performance of two discrete distance on the hub problem. Here we found when "order does matter" it pays to take it into account.

The computational experiments we have performed lead naturally to additional questions. Will the performance of mixed BCB be improved if BCB is linked

with pattern search? Pattern search is a derivative free local search procedure that has been shown to perform well on a variety of nonlinear optimization problems (see Lewis et al. 2000 for details). Previous work (Kincaid et al. 2001b) has shown that BCB coupled with local search procedures works well for the continuous version of BCB. The reasoning is that BCB does a good job of identifying valleys of interest but takes too long getting to the bottom. Will the same be true in the mixed continuous and discrete case?

Our work here focused on a small scale structural optimization problem for testing the performance of discrete BCB. It is natural to wonder if discrete BCB will stand up against other competitors—tabu search, simulated annealing, and traditional genetic algorithms. A potential scalable problem venue to test this conjecture is the the NK-landscape model. The NK model is due to Kauffman (1989) and is a random energy model similar to a spin glass. It is designed to be tuneable. That is, as K increases the landscape becomes more rugged. When $K = 0$ a highly correlated landscape with a single peak results. At the other extreme, when $K = N - 1$ (where N is the number of decision variables) a completely uncorrelated landscape results with very many peaks.

6. References

- Baker, J.E. (1987). Balancing Diversity and Convergence in Genetic Search. Dissertation, Computer Science Dept. Vanderbilt University, Nashville, TN.
- Balling, R.J. and J. Sobieszczanski-Sobieski, "An Algorithm for Solving the System-Level Problem in Multilevel Optimization," ICASE Report No. 94-96 and NASA Contractor Report 195015, December 1994.
- Kauffman, S.A. (1989) "Adaptation on Rugged Fitness Landscapes," in D.L. Stein (ed.) *Lectures in the Sciences of Complexity*, Addison Wesley, Vol. 1, pp. 527-618.
- Kelly, J.P., M. Laguna, and F. Glover (1994) "A Study on Diversification Strategies for the Quadratic Assignment Problem," *Computers and Operations Research*, Vol. 22, No.8, pp. 885-893.
- Kincaid, R.K., Weber, M. and Sobieszczanski-Sobieski, J. (2000) "Performance of a Bell-Curve Based Evolutionary Optimization Algorithm," *Proceedings of the 41st AIAA Structures, Structural Dynamics, and Materials Conference*, Atlanta, GA, AIAA 2000-1388.
- Kincaid, R.K., Weber, M. and Sobieszczanski-Sobieski, J. (2001a) "A Bell-Curve Based Algorithm for Mixed Continuous and Discrete Structural Optimization," *Proceedings of the 42nd AIAA Structures, Structural Dynamics, and Materials Conference*, Seat-

tle, WA, AIAA 2001-1550.

Kincaid, R.K., Weber, M. and Sobieszczanski-Sobieski, J., "An Atypical ($\mu + \mu$) Evolutionary Algorithm," submitted for publication in 2001b.

Lewis, R.M., Torczon, V. and Trosset, M. (2000). Direct Search Methods: Then and Now. To appear: *J. of Computational and Applied Mathematics*.

Padula, S.L., Alexandrov, N. and Green, L.L. (1996). MDO Test Suite at NASA Langley Research Center. In: *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. (pp. 410-420). Bellvue, WA. (See <http://fmad-www.larc.nasa.gov/mdob/>)

Plassman, J. and Sobieszczanski-Sobieski, J. (2000). Parallel Scalability of Bell-Curve Based Optimization. Working Paper. NASA-Langley Research Center. Hampton, VA 23681.

Sobieszczanski-Sobieski, J., Laba, K. and Kincaid, R.K. (1998) Bell-Curve Based Evolutionary Optimization Algorithm. In: *Proceedings of the 7th AIAA/USAF /NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. (pp. 2083-2096) St. Louis, MO., AIAA Paper 98-4971.

