ple we examined the effect of parameter changes on the performance of BCB. Three rules of thumb were developed. First, we found that it was crucial to scale the values of the decision variables. Second, we found that, in general, $\sigma_r >> \sigma_m$ is best. This is partially due to the influence of the Euclidean distance between parents on the placement of the center of the $n-1$ dimensional hypersphere in addition to σ_m. Third, we found that the performance of BCB is more sensitive to the value of penalty–2 than for penalty–1. We then reported on our computational experience with a termination criterion based upon a standard deviation threshold for the final generation. Lastly, we summarized a collection of experiments in which gradient information was included with the hope to improve the performance of BCB. Unfortunately, this was not borne out in our tests. Following the computational experiments of the parameters of BCB we explained and tested two variations of BCB. Both variations seek to identify clusters and outliers in a given population. It is clear that the extra work involved in identifying clusters and outliers is not beneficial in the serial version of BCB but provides some impetus for the development of a parallel version of BCB.

Future research efforts will attempt to validate the performance of BCB and its variations against other ESs on a common set of test problems. In addition we plan to examine the effectiveness of linking BCB with a local search engine. Here the hope is that BCB will identify peaks (or valleys) of interest in multi-modal constrained optimization problems. Then, instead of continuing with BCB, a switch to an efficient local optimization routine will be made. It would also be of interest to explore the performance of BCB if σ_r and the penalties were more adaptive. For example, as the search proceeds we may be more willing to increase the value of the penalties or if there has been no improvement in the several generations we might want to decrease the penalties. We might also consider decreasing the mean of the normal distribution governing σ_r as the search progresses or if the search appears stalled we might want to increase the mean. Lastly, we plan to report on experiments with BCB in a parallel computing environment.

6. References

1. Back, T. (1996) Evolutionary Algorithms in Theory and Practice, Oxford University Press.

2. 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.

3. Beasley, D., D.R. Bull, and R.R. Martin (1993) "A Sequential Niche Technique for Multimodal Function Optimization," *Evolutionary Computation*, Vol. 1, No. 2, pp. 101-125.

4. Dasgupta, D. and Z. Michalewicz (1997) "Evolutionary Algorithms—An Overview," in Evolutionary Algorithms in Engineering Applications, Dasgupta and Michalewicz (eds.) Springer, 3-28.

5. Glover, F. (1998) "Genetic Algorithms, Evolutionary Algorithms and Scatter Search: Changing Tides and Untapped Potentials," Informs Computer Science Technical Section Newsletter, Vol. 19, No. 1.

6. Goldberg, D.E. and J. Richardson (1987) "Genetic Algorithms with Sharing for Multimodal Function Optimization," in *Proceedings of the 2nd International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 41-49.

7. Mahfoud, S.W. (1995) "Niching Methods for Genetic Algorithms," Doctoral dissertation, U. of Illinois at Urbana-Champaign, Urbanna, IL, USA, IlliGAL Report 95001.

8. Mengshoel, O.J. and D.E. Goldberg, "Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement," IlliGAL Report No. 99004, January 1999.

9. Padula, S.L., N. Alexandrov, and L.L. Green, "MDO Test Suite at NASA Langley Research Center," in Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellvue, WA, September 1996, AIAA , 410-420. (Accessible as a website http://fmad-www.larc.nasa.gov/mdob/MDOB/index.html)

10. Reeves, C.R. (1997). "Genetic Algorithms for the Operations Researcher," Journal on Computing, Vol. 9, No. 3, 231-250 (with commentaries and rejoinder).

11. Rechenberg, I. (1973) Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Hozboog, Stuttgart.

12. Schwefel, H.-P. (1965) Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungstechnik, Diplomarbeit, Technische Universitat Berlin.

13. Schwefel, H.-P. (1977) Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie, Vol. 26, Interdisciplinary Systems Research, Birkhauser, Basel.

14. Sobieszczanski-Sobieski, J., K. Laba, and R.K. Kincaid, "Bell-Curve Based Evolutionary Optimization Algorithm," in *Proceedings of the 7th AIAA/USAF /NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, MO., September 2-4, 1998. AIAA Paper 98-4971, pp. 2083-2096.

15. Vanderplaats, G.N. "CONMIN-A Fortran Program for Constrained Function Minimization," NASA TM X-62282, August, 1973.

16. Baker, J.E. (1987) "Reducing Bias and Inefficiency in selection algorithms," in Proceedings 2nd International Conference on Genetic Algorithms, J.J. Grefenstette (ed.) Lawrence Erlbaum and Associates, Hillsdale, NJ pp. 14-21.

we found to be 0.025, the mesh size.) Solutions which are not part of some cluster are considered outliers. The outliers are then sorted according to fitness. Next, we overwrite the worst half of the outliers, and generate replacement solutions by mating random pairs of the remaining outliers. The idea here is that by intensifying the search around high quality outliers we will increase the chances of surveying the region near the best local optima. This technique has some similarity to *crowding* in that we are applying selection pressure. Table 11 reports the means of 100 replications of BCB for method 1 (**Mtd 1**) . The results given in table 11 indicate that this method of population control has mixed results at best for the discrete Levy function.

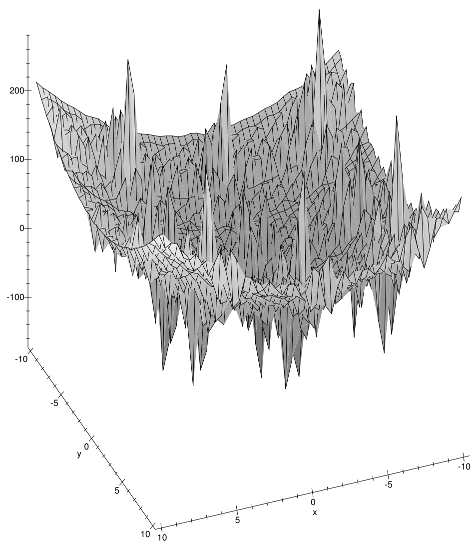

Fig 6. Levy Function

# Gens	Std Alg	Mtd 1	Mtd 2
50	-108	-100	-110
100	-134	-129	-136
200	-138	-140	-142

Table 11. Mean Best Solutions–Levy Function

The second method (**Mtd 2**) is similar except that rather than overwrite solutions we adjust the roulette wheel used to select parents. That is, when parents are to be chosen, we slightly bias the roulette wheel towards fit outliers. This is done by increasing the clustered solutions' weight by a factor of about 10%. Given an outlier and a clustered solution that are equally fit, the outlier is more likely to be chosen as a parent. This second scheme is closer in spirit to *sharing* but rather than deflate fitness values in clusters we inflate the outlier's fitness. The results in Table 11 show that this method is an improvement over method 1 and slightly improves the performance of the standard BCB. Since method 2 gave the best results for the discrete Levy function, we applied this method to the hub problem.

The results in Table 12 are for 15 replications of BCB for the 6 member 3 load continuous hub problem. The value of the fraction that controls the ratio of the distance between any pair in the cluster and the total population radius is 0.100 for the hub problem rather than 0.025 in the Levy problem. The clustered solutions' weight is increased by a factor of 5% for the hub problem. The reported results show very little difference between the standard BCB and method 2. (Method 2 tends to have a slightly lower mean and a slightly higher standard deviation.) Consequently, neither method 1 nor method 2 would be recommended in a serial setting as means to improve the performance of the standard BCB approach. However, one of our current efforts is the development of a parallel version of BCB. In the parallel environment processors are assigned portions of the population similar in spirit to the idea of clustering developed here. Hence, it is encouraging to note that performance is not degraded by identifying and performing operations on clusters.

# Gens	Std Alg Mean	S.D.	Mtd 2 Mean	S.D.
100	600.0	47.3	600.0	47.3
200	497.6	25.3	496.8	25.5
300	476.7	18.5	475.4	18.9
400	467.6	10.1	467.3	10.2
500	464.3	6.0	464.6	6.2

Table 12. 6 member hub: cluster/outlier versus standard

In a set of related experiments we identified outliers and then, instead of biasing the fitness values of the population, we initiate a *colony* of size N around each outlier (the seed). Each member of each colony is generated by perturbing each design variable of the seed (outlier) by a normal random deviate. Experiments were performed in which the selection and pairing of parents is carried out independently for each colony. Keeping the number of solutions examined constant the standard BCB procedure outperformed the non-interacting colonization scheme. We also tested a colonization approach in which colonies are allowed to interact with other colonies as well as the original population. The results of these experiments showed that slight improvements (2%) over the standard BCB performance are possible.

5. Conclusions and Discussion

Our goal was to build upon the work presented in Sobieszczanski-Sobieski et al.[14]. We began by showing the connections between BCB and traditional ESs. Next we presented an example illustrating the ability of BCB to escape local optima. Following the exam-

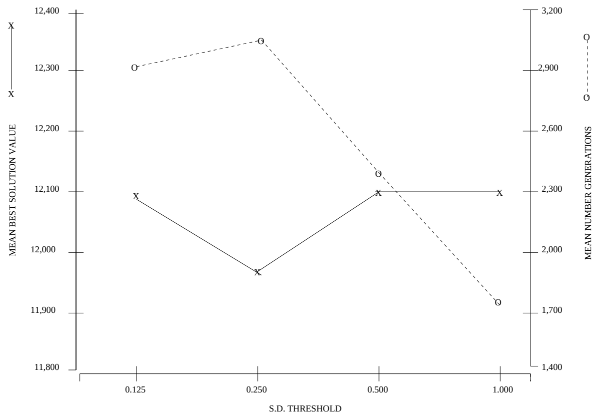

Fig 5. 20 member hub tradeoffs: generation # and solution quality vs. S.D.

We have tested a number of schemes for locating $\vec{M}$ (see Fig. 1) in addition to the fitness weighted average. We tested a scheme in which a scalar parameter a is used to bias $\vec{M}$ more towards the most fit parent. We also tested a scheme in which $\vec{M}$ is placed at the minimum of a cubic function defined along the line between parents. The coefficients of the cubic are generated from the function values and function slopes of each parent. Lastly, we tested a scheme in which the gradient vector (when available) at each parent influences the placement of $\vec{M}$. None of these methods led to better results than simply locating $\vec{M}$ at the fitness weighted average of the parents. We also devised a method to use gradient information at the parents to influence the location of $\vec{C}$, the mutated child on the $n-1$ dimensional hypersphere. Instead of the usual distribution the negative sum of the gradient vectors, projected into the orthogonal subspace, identifies the region of the hypersphere where the child will most likely be placed. This approach was unsuccessful as well. While we believe that it is unwise to not make use of gradient information when it is available, we have not been able to find a way to include gradient information in BCB so that it leads to an improved search procedure. We have found it difficult to infer from the gradient information at the parent solutions how the search space behaves between the parents. Our experiments so far have shown that these approaches do not lead to better solutions, and increase the computational requirements by an order of magnitude.

Std. Dev. Threshold	Mean Best Weight	Mean # Gens.
0.125	11,972	4173
0.250	11,898	3238
0.500	12,020	2500
1.000	12,111	1853

Table 9. S.D. Stopping Condition: 20 Member 2 Load Hub (10 reps)

Std. Dev. Threshold	Mean Best Weight	Mean # Gens.	No improve 500 gens.
0.125	12,098	2909	7/10
0.250	11,974	3051	4/10
0.500	12,100	2394	3/10
1.000	12,111	1853	0/10

Table 10. S.D. and 500 Generations without Improvement Stopping Conditions

4. Extensions to the Standard Algorithm.

The next set of results report on a set of experiments regarding the identification of clusters of solutions as well as outliers within a given generation of the BCB algorithm. The underlying objective is to prevent premature convergence to a local optima by injecting a diversification mechanism within a particular generation of BCB. This topic has been of interest to the genetic algorithm community for many years with references to niche identification. In addition, cluster and outlier identification provide a natural way to partition a population among processors for parallel computing. One general approach to prevent premature convergence is to apply selection pressure. Crowding[7,8] and sharing[3,6] are two examples of selection pressure. In crowding individuals replace existing population members according to their similarity. An individual is compared to a random subpopulation. The individual in the crowd most similar to the new individual is replaced by the new individual. Sharing derates the fitness level of individuals from the same niche proportional to the number of individuals in the current population *sharing* the same niche.

The first test function is Levy No. 5 (see Figure 6).

$$\begin{aligned} Levy(x_1, x_2) \quad = \quad & \sum_{i=1}^{5} i(cos(i-1)x_1 + i) \\ & + \sum_{j=1}^{5} j(cos(j+1)x_2 + j) \\ & + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2. \end{aligned}$$

It has hundreds of local optima and nine distinct peaks with nearly the same objective function value. However, there is one global minimum with a value of -176.138. In our tests we solved a discrete version of this problem. The search space was discretized into lattice points spaced at intervals of 0.025, with the decision variables being allowed to take on any of these values on the range $[-10.0, 10.0]$.

We tested two diversification methods. In the first method (**Mtd 1**) we periodically (every 4 generations) partition the population into clusters and outliers. A heuristic identifies clusters of solutions in which the distance between any pair in the cluster is less than some fraction of the total population radius. (This fraction is a parameter we control and whose best value straint violation is considered can we come to a decision about an appropriate value for the penalty factors. Hence, a value of about 10,000 would be acceptable based on an allowable violation of about 1.0 -E02. However, in tables 6 and 7 we see that the performance of BCB is more sensitive to the value of penalty-2 than to the value of penalty-1. In table 6 with penalty-1 fixed at 20,000 we conclude (as in table 5) that a value near 10,000 is acceptable for penalty-2. However, in table 7 with penalty-2 fixed at 20,000 we see only a small change in the mean best weight (all near 14,400) as penalty-1 is varied between 1,000 and 100,000. We do see a significant effect on the maximum constraint violation from 7 E-03 to 7 E-04 to -2.5 E-04. We note that all of the mean values are less than our agreed upon tolerance of +0.01.

Penalty Factor	Mean Best Weight	S.D.	Mean Max Const Violation
1000	6320	335	2.18e+00
2000	7540	249	1.41e+00
10,000	12,900	430	3.36e-02
20,000	14,400	733	1.81e-04
100,000	15,000	617	-1.55e-04

Table 5. Equal penalties: 20 member 2 load problem (15 reps)

Penalty Factor	Mean Best Weight	S.D.	Mean Max Const Violation
1000	6370	355	2.18e+00
2000	7560	205	1.35e+00
10,000	13,200	316	4.37e-02
20,000	14,400	733	1.81e-04
100,000	15,000	614	-4.73e-04

Table 6. Penalty 2 experiments (penalty 1 fixed at 20,000)

Penalty Factor	Mean Best Weight	S.D.	Mean Max Const Violation
1000	14,600	211	7.31e-03
2000	14,300	617	1.26e-03
10,000	14,300	604	7.68e-04
20,000	14,400	733	1.81e-04
100,000	14,700	448	-2.50e-04

Table 7. Penalty 1 experiments (penalty 2 fixed at 20,000)

Next we describe experiments with stopping conditions for BCB. The hope is that after a sufficient number of generations BCB will have converged to one or more local optima of roughly the same quality. A natural stopping condition then is to monitor the distribution of objective function values in each generation. In particular, we check the standard deviation value for this distribution. When this value is sufficiently small we terminate BCB. Monitoring the objective function value distribution is preferable to monitoring the distance between solutions. In the latter it precludes the possibility that BCB may converge to similar valued local optima that are not physically close together in the design space. We note, however, that monitoring the objective function value distribution does *not* completely solve all difficulties. For example, it could easily be the case that the two best local optima do not have objective function values that are *close* in value. Hence, forcing the standard deviation (S.D.) of the objective function value distribution to fall below some threshold may force BCB to perform many extra generations so that all the members of the population lie near the global optima.

Tables 8, 9 and 10 and figure 5 catalog computational results for experiments with such a stopping condition. For table 8 a 6 member 3 load hub design problem was tested while for tables 9 and 10 a 20 member 2 load hub problem was examined. In table 8, 15 replications were performed keeping $(\sigma_m, \sigma_r) = (1.0, 4.0)$ with penalty values of 2000/4000 and a 20 member population. For the 20 member 2 load case in tables 9 and 10, 10 replications were performed keeping $(\sigma_m, \sigma_r) = (1.0, 4.0)$ with penalty values of 10,000/10,000 and a 40 member population. For the 6 member 3 load case S.D. thresholds of 0.125 and 0.500 had acceptable performances. The 0.125 threshold found the best overall value (460.0) but had a larger variance in solution quality than did the threshold of 0.500. For the 20 member 2 load case S.D. threshold experiments we added an additional stopping condition to avoid extremely long runs (table 10). If there was no improvement in the best observed solution value for 500 generations then the search was terminated regardless of whether or not the S.D. threshold was met. As can be seen in figure 5 and table 10 this dramatically reduced the mean number of generations for the 0.125 threshold (about 30%) and also increased the mean best weight by about 120 units (about 1%). The reduction in the mean number of generations was only about 10 % for the 0.25 threshold but resulted in the same increase in the mean best weight (120 units) as for the 0.125 threshold. Hence, the trends plotted in figure 5 for mean best weight hold even if we don't include the second stopping condition (500 generations without improvement).

S.D. Threshold	Mean Best Soln	Mean # Gens
0.125	465.9	730.8
0.250	471.2	382.1
0.500	465.8	288.4
1.000	480.3	214.9

Table 8. S.D. Stopping Condition: 6 Member 3 Load Hub (15 reps)

trend is downward, even after generating 50,000 solutions (per replication) the best observed weight is only about 650 and the average weight (20 replications) is about 740. The problem is that BCB skews the search space towards design variables of small dimension during recombination. The results in Figure 4 are for a version of BCB in which the search space is scaled so that all design variables are of similar magnitude. The resulting change in performance is dramatic. After 5,000 examined solutions the results are already superior to the 50,000 row in Figure 3. After 25,000 solutions nearly every replication is resulting in the optimal design weight of approximately 460.

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 $\vec{B} = \vec{M} + |\vec{P}_2 - \vec{P}_1| * N(0, \sigma_m)$ where $\vec{M}$ is the fitness weighted mean of the two parents. Our intuition suggests that σ_m should have a value near 1.0 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, we expect the value of σ_r to be larger than 1.0 to allow for the possibility of hyperspheres of larger radii. Unfortunately we don't have any problem 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. Hence σ_r is likely to be larger than σ_m. Table 3 summarizes a series of computational experiments for the 6 member 3 load hub design problem. We replicated fifteen times BCB (scaled version) with a population size of 20 for 500 generations. The distributional data in table 3 is for the best values seen in each of the fifteen replications.

(σ_m, σ_r)	**Mean**	**S.D.**	**Min**	**Max**
(1.0,2.0)	509.2	64.8	460.5	632.9
(1.0,3.0)	500.6	53.0	460.1	620.6
(1.0,3.5)	468.5	20.8	459.7	543.3
(1.0,4.0)	462.2	2.9	459.7	469.7
(1.0,4.5)	464.9	5.7	459.8	477.6
(1.0,5.0)	463.6	3.9	459.9	474.8
(1.0,6.0)	472.2	26.0	459.6	567.6
(2.0,4.0)	464.2	6.1	459.7	481.1
(2.5,4.0)	468.3	12.7	460.0	512.0
(3.0,4.0)	463.5	3.0	460.5	470.7
(3.5,4.0)	472.0	17.4	460.5	529.6
(4.0,4.0)	476.9	19.0	461.2	536.1

Table 3. 6 member 3 loads (15 reps, 500 gens, popsize = 20)

So, which combination of (σ_m, σ_r) is best? Rather than pin all of our conclusions on this one test case we choose three of the best combinations in Table 3—(1.0,4.0), (3.0,4.0) and (1.0,5.0)—on which to run further computational tests with a 20 member 2 load hub design problem. As a result of the 20 member tests given in table 4 we easily eliminate (3.0,4.0) from further consideration. However, the performance of (1.0,4.0) and (1.0,5.0) are relatively close. (1.0,4.0) has a lower mean weight (but higher standard deviation) and had some best solutions that were feasible (min constraint value is negative). However, since we were willing to accept as much as 1.0 E-02 as a constraint violation, both (1.0,4.0) and (1.0,5.0) have acceptable constraint violations. Although the choice is not completely clear, we selected $\sigma_m = 1.0$ and $\sigma_r = 4.0$ as our standard values. Of course there is no guarantee that other problem venues will enjoy the same performance characteristics. But our intuition is confirmed that $\sigma_m << \sigma_r$ and that a value of 1.0 works well for σ_m.

Criteria	**Stat**	**(1.0,4.0)**	**(3.0,4.0)**	**(1.0,5.0)**
Weight				
	Mean	12,415	13,704	12,596
	S.D.	345	515	286
	Min	11,716	12,356	12,028
	Max	13,106	14,491	12,981
Constraint				
	Mean	5.97E-03	1.91E-02	4.00E-03
	S.D.	9.45E-03	1.86E-02	5.09E-03
	Min	-1.37E-04	1.82E-03	5.52E-04
	Max	3.79E-02	7.33E-02	2.19E-02

Table 4. 20 member 2 load (15 reps, 1000 gens, popsize=20)

Next we analyze the effect of the two penalty factors on the performance of BCB. Penalty factor 1 is used if the best objective function values in the previous generations have been feasible. If the best objective function values in the previous generations have been infeasible then we apply penalty factor 2. The idea is that if the search has remained feasible for some time we may want to allow the search to move outside the feasible region (by way of a smaller penalty–1). However, if the search has been infeasible then we may want to strengthen the penalty (penalty–2) to move the search back inside the feasible region.

Tables 5, 6 and 7 record the results of a set of experiments with the 20 member 2 load hub problem. The additional BCB parameter values are a population size of 20, a maximum number of generations of 1000, and $(\sigma_m, \sigma_r) = (1.0, 4.0)$. Finally we must decide what level of constraint violation (if any) is allowed. If we require that all constraints must be met, then only maximum constraint violations less than zero are allowed. For our experiments we have decided that as long as the maximum constraint violation is on the order of 1.0 E-02 or smaller we will say that the solution is acceptable.

Table 5 forces both penalty-1 and penalty-2 to vary together. The trend in mean best weight values is dramatic and only when the maximum allowable conthis number is small it was possible to enumerate all solutions and determine an optimal solution and its corresponding optimal value. The optimal value has a volume of 613.9. Below are the BCB test results for a 2 member discrete hub design problem. BCB was replicated 15 times for each parameter setting.

Pop. Size	Mean	S.D.	Min	Max	Range
4	622.7	13.7	613.9	667.3	53.4
10	614.7	2.8	613.9	625.3	11.4
20	613.9	0.0	613.9	613.9	0.0

Table 2a. Discrete 2 member hub results: 50 generations held constant

Gens.	Mean	S.D.	Min	Max	Range
50	622.7	13.7	613.9	667.3	53.4
100	613.9	0.0	613.9	613.9	0.0
200	613.9	0.0	613.9	613.9	0.0

Table 2b. Discrete 2 member hub results: Population size 4 held constant

The number of solutions examined by BCB in row 1 of table 2a and row 1 of table 2b are identical (200) and the computational results are identical as well. However, BCB in row 2 of table 2a examines 500 solutions while BCB in row 2 of table 2b examines only 400 solutions but the row 2 of table 2b exhibits superior performance. Thus performance improves faster with increasing generations than it does for increasing population size.

3. Standard Algorithm Parameter Experiments

This section is organized as follows. We first examine the effects of decision variable scaling on the performance of BCB. Next we seek robust values for the two main parameters—σ_m and σ_r. Although we did not discuss the penalty values for constrained optimization models in our overview of BCB it is clear that these values directly affect the algorithms' performance. Let $f(\vec{x})$ denote the objective function and $g_i(\vec{x})$ denote the ith constraint, each of the form $g_i(\vec{x}) \leq \vec{0}$. The selection fitness criterion penalizes constraint violations by adding a penalty for the maximum violated constraint. That is, $fitness = f(\vec{x}) + penalty * max_i(g_i(\vec{x}))$. We use two values to penalize the maximum constraint violation. If the best values in previous generations have been feasible then we apply penalty–1. If the best values in previous generations have been infeasible then we apply penalty–2. The idea is that if the search has remained feasible for some time we may want to allow the search to move outside the feasible region (by way of a smaller penalty–1). However, if the search has been infeasible then we may want to strengthen the penalty (penalty–2) to move the search back inside the feasible region. We examine the effect of the penalty term values on performance. Next, we determine the effects on performance of a stopping condition based upon the spread of the solution values in the final population. Lastly, we comment on the placement of $\vec{M}$.

# Eval.	Mean	S.D.	Min	Max	Range
5,000	899.4	89.4	776.9	1134.4	357.5
10,000	792.8	63.5	666.5	900.0	233.5
25,000	761.1	54.9	654.2	864.0	209.8
50,000	743.5	54.8	642.9	843.3	200.4

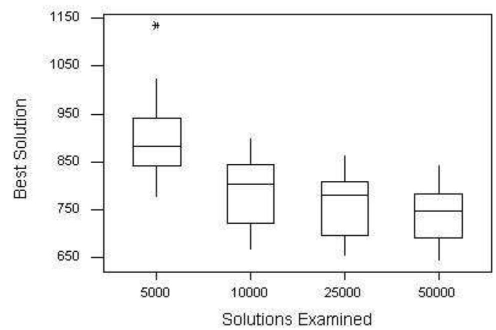

Fig 3. 6-Member Hub Problem: Unscaled

# Eval.	Mean	S.D.	Min	Max	Range
5,000	488.5	12.2	469.5	511.6	42.1
10,000	465.2	4.0	460.6	476.9	16.3
25,000	460.5	0.6	459.7	462.0	2.3
50,000	460.2	0.4	459.6	461.0	1.4

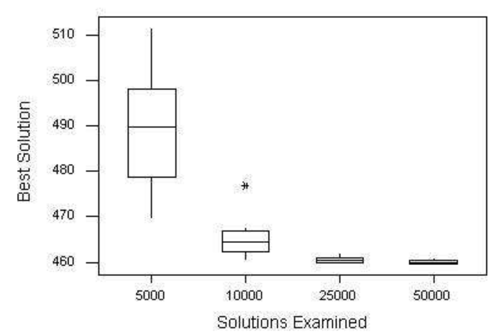

Fig 4. 6-Member Hub Problem: Scaled

The first set of test results indicate that scaling with respect to the decision variables is critical. Figure 3 records the results of 20 replications of BCB with an increasing number of solutions examined. The design problem is minimum weight 6 I-beam members (36 design variables) with 3 loads hub problem. From Sobieszczanski-Sobieski et al.[14] we know that the best solutions have a weight of about 460 units. The distributional data in Figure 3 shows that, although the

Sobieszczanski-Sobieski et al. (1998) to analyze the sensitivity of the BCB parameters. We rectify that deficiency here as well as demonstrate how to implement BCB for discrete and mixed continuous and discrete constrained optimization problems.

2. BCB and Local Optima

To explore how BCB behaves in the presence of non-global local optima a 2 symmetric-member 3 load continuous hub case was examined. We also report on a discrete version of this design optimization problem when symmetry is ignored. There are seven design variables–two for the top and bottom of the I-beam, two for central portion of the I-beam web, and one for the angle (A) between the two I-beam members. See figure 2 for details. We tested three versions, all under the same concentrated force load seen in figure 2. In the first version BCB was allowed to run as usual. We observed that BCB converged to a V-shape which is the global optimum. The seven best solutions in the final population are given in Table 1a. Next, we restricted the algorithm to explore a sub-region (A < 0.0) of the search space that did not include the global optimum. BCB converged to the known local optimum (an inverted V) in that sub-region. Table 1b catalogs the seven best solutions in the final population. Lastly we forced the I-beams to be horizontal (A=0) and, as expected the design variables tend to go to their upper bounds (see Table 1c).

We conjecture that BCB is able to avoid entrapment in local optima due to two factors. First, the search space is explored sufficiently uniformly during the early generations so that children are placed near local optima. Second, once those children are near local optima, those with the best fitnesses are the most likely to be chosen as parents for the next generation. In this manner, sub-populations near non-global optima die out because the solutions are not chosen as parents. Similarly, sub-populations near global optima grow larger because the solutions are chosen as parents.

We formulated a 2 member hub problem as a purely discrete optimization problem by limiting the choices for I-Beams to ones given in standard engineering tables. In these tables, each row represents a particular I-Beam shape with six columns designating the dimensions of the I-Beam. (There are six rather than seven columns since B_1 and B_2 are forced to be the same in the tables we used.) The 2 member hub design problem then is to select 2 rows of this table (or select one row twice) to form a hub with minimum volume that satisfies the physical constraints. For our test case we have 90 rows from which to select hub members leading to a total of 1800 feasible designs. Since

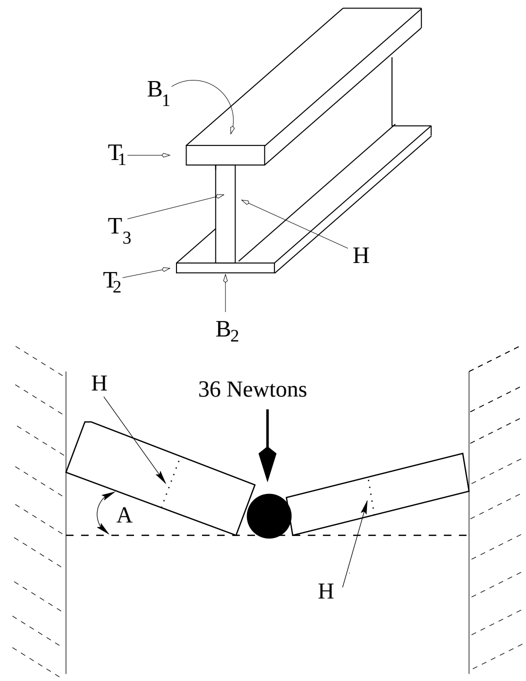

Fig 2. Two member hub diagram

DV	# 1	# 2	# 3	# 4	# 5	# 6	# 7
B1	4.82	5.21	4.88	2.00	5.84	2.01	5.85
B2	2.70	2.00	2.01	2.00	2.00	2.01	2.06
T1	0.10	0.10	0.12	0.21	0.10	0.17	0.10
T2	0.10	0.10	0.10	0.13	0.10	0.15	0.10
T3	0.10	0.10	0.13	0.14	0.10	0.10	0.10
H	3.00	3.00	3.02	3.00	3.02	3.12	3.00
A	46.3	48.0	44.2	42.4	44.9	45.0	44.3

(a) A > 0.0, V shape weight approx. 600

DV	# 1	# 2	# 3	# 4	# 5	# 6	# 7
B1	4.23	4.11	2.95	4.23	5.47	3.76	4.03
B2	6.00	5.99	6.00	5.97	6.00	5.82	5.72
T1	0.10	0.10	0.10	0.10	0.10	0.10	0.10
T2	0.16	0.13	0.10	0.16	0.19	0.13	0.14
T3	0.22	0.25	0.25	0.23	0.19	0.27	0.26
H	3.43	3.89	4.59	3.42	3.66	3.58	3.45
A	-27.5	-25.5	-32.2	-27.6	-23.4	-28.0	-30.2

(b) A < 0.0; Inverted V shape; weight approx. 1100

DV	# 1	# 2	# 3	# 4	# 5	# 6	# 7
B1	6.00	6.00	6.00	6.00	5.91	6.00	5.96
B2	6.00	6.00	6.00	6.00	5.97	5.97	6.00
T1	1.00	0.99	1.00	1.00	0.66	0.97	0.90
T2	1.00	1.00	1.00	1.00	1.00	1.00	1.00
T3	1.00	1.00	1.00	1.00	1.00	0.99	0.99
H	8.00	8.00	8.00	8.00	8.00	8.00	8.00
A	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(c) A = 0.0; Horizontal shape; weight approx. 5100

Table 1. Local Optima for 2-member Hub Problem

angle. Although these rotation angles are a natural idea they are not used in most ESs. For example Dasgupta and Michalewicz[4] never mention them. The mutation regions describe a hypersphere if all of the σ_i values are identical and a hyperellipsoid otherwise. With the σ_i and α_{ij} there are potentially $n(n+1)/2$ parameters under the control of the ES. These may be fixed or self-adapting.

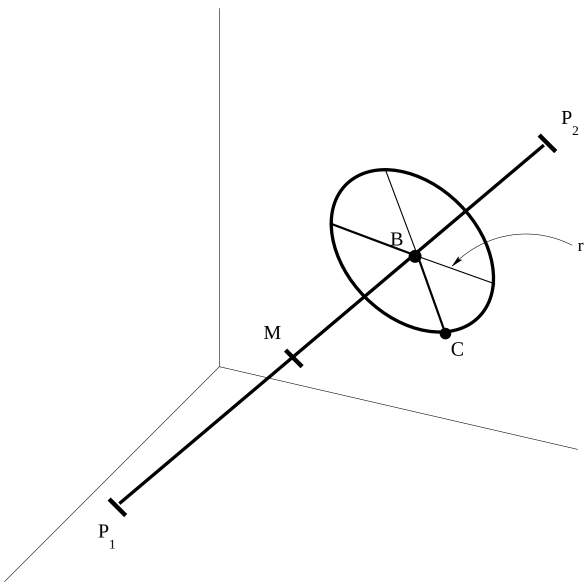

Fig 1. BCB Geometrical Construct in 3D Space

Our heuristic procedure, first presented in Sobieszczanski-Sobieski et al.[14], is similar in spirit to ES methods but has far fewer parameters. To illustrate the connection to ESs we define our mutation, recombination, and selection mechanisms in ES terms. A new generation in our approach is selected exactly the same as a $(\mu + \lambda)$–ES. The recombination mechanism is similar to the extension of the intermediate recombination if the weights are required to sum to one. Consider the line through two n-dimensional parent vectors $\vec{P_1}$ and $\vec{P_2}$ selected for mating. 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,σ_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 $\overline{P_1 P_2}$. 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 $(\sigma_r >> \sigma_m)$. 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 (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 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.

We have applied BCB to continuous, discrete, as well as mixed continuous and discrete variable optimization problems (both constrained and unconstrained). The first suite of test problems is a minimum weight (volume) design of a hub structure also found in Balling and Sobieszczanski-Sobieski[2]. Each member of the hub is an I-beam rigidly attached to the hub and to the wall. The beam cross-sectional dimensions are the design variables, and the constraint functions reflect the material allowable stress and the 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. Additional details may be found in Padula et al.[9]. 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. Design variable domains can be chosen as either continuous, discrete, or mixed continuous and discrete. For discrete domain design variables both standard tables of I-beam dimensions and a fixed range of integral millimeter units were tested. The second test problem is a standard problem taken from the global optimization literature.

In Sobieszczanski-Sobieski et al.[14] we tested two mating schemes for BCB with the continuous hub problem. Mating scheme 1 chose two parents from a roulette wheel in which the sector sizes were determined by a fitness value equal to the sum of the weight of the structure (objective value) and the maximum constraint violation. Mating scheme 2 chose one parent from a roulette wheel based on objective values and a second parent from a roulette wheel based on maximum constraint violation. Here we implement Baker's[16] stochastic universal sampling rather than roulette wheel selection. In doing so we eliminate the natural bias present in roulette wheel selection. The quality of solutions generated in Sobieszczanski-Sobieski et al.[14] for a continuous hub design problem were verified by comparing BCB solutions to ones generated by CONMIN[15] a standard nonlinear programming technique. No attempt was made in

Performance of a Bell-Curve Based Evolutionary Optimization Algorithm

Rex K. Kincaid*
College of William and Mary, Williamsburg, VA 23187-8795

Michael Weber†
College of William and Mary, Williamsburg, VA 23187-8795

Jaroslaw Sobieszczanski-Sobieski‡
NASA Langley Research Center, Hampton, VA 23681, AIAA member

An evolutionary search strategy utilizing two normal distributions to generate children is presented. This Bell-Curve Based (BCB) evolutionary algorithm is similar in spirit to $(\mu + \lambda)$ evolutionary strategies but with fewer parameters to adjust. Extensive tests regarding the sensitivity of BCB parameters to performance are provided. The test suite includes continuous variable constrained hub design problems, mixed discrete and continuous variable constrained hub design problems, and an unconstrained highly multi-modal discrete optimization problem.

1. Introduction.

Evolutionary methods are exceedingly popular with practitioners of many fields; more so than perhaps any optimization tool in existence. Historically Genetic Algorithms (GAs) led the way in practitioner popularity[10]. However, in the last ten years Evolutionary Strategies (ESs) have gained a significant foothold[5]. One partial explanation for this shift is the interest in using GAs to solve continuous optimization problems. The typical GA relies on a binary representation of the design variables which is cumbersome. This is not true for ESs which work with real-valued design variables. One of the earliest ES references is Schwefel[12] in which an ES with a population of size one was examined. Not until Schwefel[13] and Rechenberg[11] were populations of varying sizes considered. For detailed current references on evolutionary methods in general and ESs in specific see Back[1] and Dasgupta and Michalewicz[4]. To understand the connection between our work and ESs we describe an ES.

In this description we do not provide any explanations for initialization procedures, termination criterion, or how constraints should be handled. What remains is to define mutation, recombination, and selection. There are two standard ES approaches to selection—$(\mu + \lambda)$–ES and (μ, λ)–ES. In the first approach the best μ individuals out of μ parents plus λ children are selected for the next generation, while in the second approach μ individuals out of λ children $(\lambda >> \mu)$ are selected. Hence, in the first approach fit individuals may continue from one generation to the next while in the second approach no parents survive to the subsequent generation. Currently, the (μ, λ)–ES is considered to be the superior approach[1] (p. 68). We describe only one sexual recombination procedure, and an extension, but a much wider variety exist in the ES literature. Consider two selected parents $\vec{P}_1$ and $\vec{P}_2$ each described as a real-valued vector. A child is produced by taking the mean of each of the parent's components–$(\vec{P}_1 + \vec{P}_2)/2$. Back[1] refers to this as intermediate recombination. An extension of this recombination procedure is to allow arbitrary weights in the interval $[0, 1]$ to be assigned to each parent prior to taking the mean.

Lastly, mutation is an asexual operator. Let n denote the number of components in a vector $\vec{P}$ describing an individual in the population. Several random variables are associated with $\vec{P}$. A random variable σ_i is associated with each of the $i = 1, \ldots, n$ components of $\vec{P}$. In addition there are $n(n-1)/2$ random variables α_{ij}, one for each of the pair-wise interactions among the components of $\vec{P}$. Together these define a matrix $\mathbf{C}$ with σ_i as the ith diagonal entry and α_{ij} as the ijth off diagonal (symmetric) entries. The mutation of $\vec{P}$ is completed by forming $\vec{P} + \mathrm{N}(\vec{0},\mathbf{C})$. $\mathrm{N}(\vec{0},\mathbf{C})$ denotes a realization of a random vector sampled from an n-dimensional normal distribution with expectation $\vec{0}$ and a covariance matrix $\mathbf{C}^{-1}$. The effect of each σ_i is to govern how the stretching of the ith component along the ith axis occurs. Each α_{ij} describes a rotation

*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

‡Computational AeroSciences and Multidisciplinary Research Coordinator

AIAA 2000-1388

Performance of a Bell-Curve Based Evolutionary Optimization Algorithm

Rex K. Kincaid
College of William and Mary,
Williamsburg, VA 23187-8795

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

Jaroslaw Sobieszczanski-Sobieski
NASA Langley Research Center, Hampton, VA 23681,
AIAA member

41st AIAA Structures,
Structural Dynamics and Materials
April 3–6, 2000/Atlanta, GA