SAND2007-4596C

Double Trouble: How global structure impacts exploration

Evolution Strategies

ABSTRACT

It is commonly believed that population-based methods are better at optimizing multimodal functions because they tend to explore more of the fitness landscape before their population converges to a compact, globally competitive region. We show that the effectiveness of this strategy is highly dependent on a function's *global structure*. When the local optima are not positioned in a "big valley", too much exploration can cause search to fail. Surprisingly, limiting the degree to which an algorithm explores can result in a better *global search* strategy.

1. INTRODUCTION

A common *global search* strategy used in evolutionary parameter optimization is to initially *explore* the search space and then *exploit* the promising regions (or region) identified during the exploration process. Intuitively, this makes sense: if an algorithm does not sample the entire search space first, its overall effectiveness may be largely determined by where it is initialized. The assumption here is that exploration, using a limited number of samples, can distinguish between different "global regions" of the search space based on average effectiveness. This simple model—explore then exploit—has proven to be an effective strategy on the standard set of artificial test functions that are frequently used to evaluate performance. Our results show that as dimensionality increases, the exploration process may be more biased to the relative size of the global region and not its quality.

Many artificial test functions have a "big valley" topology [2]. That is, a decrease in fitness implies that, on average, search is getting closer to the global optima. Although the search space is highly multi-modal, the local optima are structured such that there exists a global trend toward the best solution. The underlying *global structure* of this type of problem is roughly unimodal. Problems that exhibit this characteristic are sometimes referred to as *single-funnel* landscapes.

GECCO'07, July 7–11, 2007, London, England.

While single-funnel landscapes may be common, there also exist several real-world applications that do not have a unimodal underlying global structure. Wales [10] suggests that many optimization problems in computational biology are difficult because local optima often form distinct, spatially separate clusters within the search space. Problems of this type have multiple funnels, resulting in a landscape that has a less predictable underlying global structure.

The way that global structure impacts search is not well understood. This is partly because many of the test functions used for evaluation have single-funnel landscapes. Although there are also a few test functions that have multiple-funnels, the number of funnels increases with dimensionality. This creates a gap in our ability to evaluate global structure. That is, either we have a single funnel or we have a complex surface that often contains $\mathcal{O}(2^N)$ funnels, where N is the problem dimension. This makes it difficult—if not impossible—to understand search behavior in high dimensions. Without a scalable multi-funnel test problem, our understanding of how global structure impacts search is limited.

We have two main objectives in this paper. First, we describe several new test functions where the number of funnels remains constant as dimensionality increases. The difficulty in solving any single funnel is similar to that of the well-known Rastrigin function. Then, we show that two evolutionary algorithms, CHC [4] and an Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [7], both utilizing the "explore then exploit" philosophy, are increasingly less effective on some multi-funnel test functions. We conjecture that in high dimensions, the exploration process is more biased toward the relative size of the funnels and not their overall quality. Further empirical results on a simple multi-sphere function with low modality strongly support this conjecture.

Our results indicate that too much exploration hinders search in high dimensions. We reduced the amount of exploration in CMA-ES by using a small initial step-size and show that this results in a performance gain on our most difficult multi-funnel functions. One surprising result of this research is that limiting the degree to which an algorithm explores the search space can actually improve its global search performance.

1.1 Motivation and Background

There are several characteristics that make real-world optimization problems difficult. The degree to which an algorithm will perform well on an application partly depends on how well the algorithm can deal with the features that make the problem difficult. Since artificial test functions are used to evaluate search performance, the relative merit in any empirical study is limited by how well we understand the characteristics that make realistic parameter optimization problems difficult and by our ability to embed these features into benchmark test functions.

Researchers within the computational chemistry community have started to pay attention to how global structure affects problem difficulty [8]. Much of their attention has been devoted to studying Lennard-Jones clusters, which are a class of configuration optimization problems where the goal is to find the spatial positions for a set of atoms that has the smallest potential energy.

The energy surface of the Lennard-Jones potential is highly multimodal and the number of local optima increase with problem size. This means that multiple restarts of *local search* will become a less effective global optimization strategy as the number of the atoms in the cluster increases. From this perspective, finding the best solution for larger problems should be more difficult. However, there are several Lennard-Jones instances where finding the optimal configuration of a small cluster requires significantly more effort than finding the global solution for clusters with a greater number of atoms. For example, the optimal configuration of the $N = 38$ atom Lennard-Jones problem is more difficult to find than the global optima of other instances as high as $N = 60$ atoms, despite the fact that the $N = 38$ atom problem has considerably fewer local optima. One explanation for this discrepancy in difficulty is that the $N = 38$ atom instance has a double-funnel global structure. Assuming that a search algorithm can move between local optima, the underlying global structure of a problem may have a greater impact on problem difficulty than the number of local optima [9].

2. CREATING GLOBAL STRUCTURE

We have mentioned that there exist some test functions that have a multi-funnel global structure. For example, the Schwefel, Rana, and Whitley [12] functions all have $\mathcal{O}(2^N)$ funnels, where N is the problem dimension. Aside from their complexity in higher dimensions, these test functions also have the optimal solutions on, or near, the boundary of the search space. This means that an algorithm's performance may largely depend on how the boundary conditions are handled. We propose a test function that does not place the best solutions on the boundary of the search space and that has a constant number of funnels, regardless of the problem dimension.

Rastrigin's function is a well-known multimodal test problem often used to evaluate an algorithm's global search performance. It is created by adding local optima to a simple underlying surface. Specifically, Rastrigin's function is comprised of a cosine term and a simple quadratic sphere function. Because the sphere function is unimodal, this combination produces a single-funnel function (see Figure 1).

We wanted a multi-funnel test problem with properties similar to Rastrigin's function because it would isolate global structure as the main difference impacting problem difficulty on a problem that is well-understood.

2.1 The Multi-funnel Rastrigin

Our multi-funnel version of Rastrigin's function is constructed by first creating an underlying global structure containing several quadratic sphere functions, and then adding the cosine term used in the original Rastrigin test problem to this surface. Instead of using a single quadratic to create a unimodal underlying structure, we used multiple sphere functions to create multiple funnels. The underlying global structure of our test function is the minimum of each sphere function, where each quadratic sphere creates a single funnel in the search space.

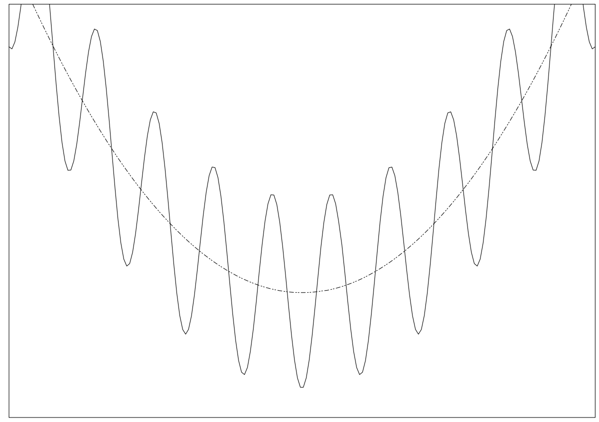

Figure 1: The original Rastrigin test function. The dashed line is the quadratic sphere function that defines its (single-funnel) global structure.

The quantity and placement of each funnel is critical for several reasons. First, if any two funnels are close, the barrier that divides them may be inconsequential. We also want the relative barrier between each funnel to scale with dimensionality. This means that the distances between the center of each funnel should also increase as dimensionality increases. Second, we would like the number of local optima in each funnel to be constructed such that the difficulty of any single funnel is consistent with that of the original Rastrigin problem.

The Rastrigin test function is often bound between $(-5, 5)$. Given this bound constraint, we propose two configurations that meet the above criteria; a two- and four-funnel instance. In both cases, the optimal funnel is located at 2.5 in each dimension. The second funnel is centered at -2.5, which is also constant across dimensions. The distance between each funnel increases proportionally with dimensionality. This construction also creates an underlying surface that is non-separable, since each funnel is located on the positive diagonal of the search space.

The four-funnel problem adds two additional funnels along the negative diagonal of the search space. The third funnel is placed at alternating values of $(2.5, -2.5)$ and the fourth funnel is located adjacent from the third, centered at $(-2.5, 2.5)$. When the problem dimensionality is odd, the final parameter for each funnel's center is zero. This keeps the third and fourth funnels equidistant from the optimal funnel. Because the centers of these two additional funnels are closer to the optimal funnel, the barriers that separate them from the global optima is also the smaller.

Multi-funnel problems are difficult when the basin of attraction to the optimal funnel is small and and when other sub-optimal funnels are nearly as deep as the optima funnel. For example, the optimal funnel for the 38 atom Lennard-

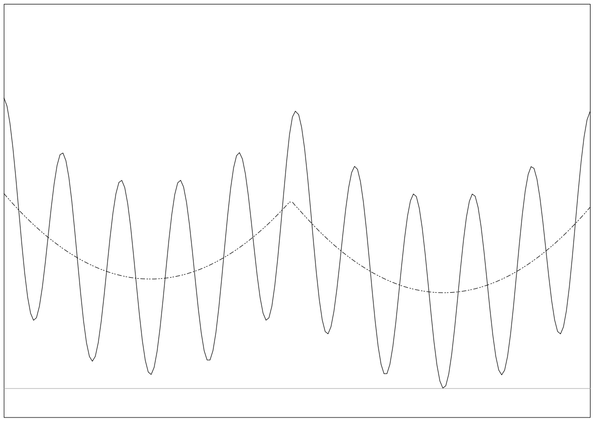

Figure 2: A double-funnel Rastrigin function.

Jones problem is believed to be about 10% of the size of the sub-optimal funnel and the best solution in the sub-optimal funnel differ from the global optima by only a small degree. [3].

We added a small weight to each sub-optimal funnel in order to decrease their overall depth. This implies that the primary funnel is the optimal funnel and its optimum remains zero at the funnel center.

We translate the cosine term used in Rastrigin's function by 2.5 so that the minimum of the local optima component is centered at the bottom of each funnel. A one-dimensional diagonal slice of the double-funnel Rastrigin function is displayed in Figure 2.

In order to make this test function more realistic and, more importantly, to understand how funnel size affects search, we scaled the sub-optimal funnels in each problem by a constant factor, denoted s_j. Multiplying a funnel by a number greater than one will create a more narrow (and thus steeper) funnel. The opposite is true when s_j is less than one; this creates a wider funnel.

The shape of the funnel affects problem difficulty. We created variance in funnel size by scaling the sub-optimal funnels rather than scaling the optimal funnel. This way, the optimal funnel retains it shape regardless of the problem, and therefore, has a more consistent level of difficulty.

The overall form of our multi-funnel Rastrigin function is the sum of the global structure, $g(\vec{x})$, and the additional local optima, $h(\vec{x})$.

$$f(\vec{x}) \quad = \quad g(\vec{x}) + h(\vec{x})$$

The global structure is simply the minimum of each quadratic sphere. We also use this multi-sphere function, $g(\vec{x})$, without any additional local optima as a test function later in the paper.

$$g(\vec{x}) \quad = \quad \min_j \left(w_j + s_j \cdot \sum_{i=1}^{N} (x_i - \mu_{i,j})^2 \right)$$

The double-funnel function has $j = 2$ funnels and the four-funnel function is comprised of $j = 4$ funnels. Each funnel has a weight w_j that controls its depth, and a scale s_j that controls its width. The optimal funnel is never scaled ($s_1 = 1.0$) and always has a weight of $w_1 = 0$. The mean values $\mu_{i,j}$ refer to i^{th} dimension of the j^{th} funnel center, and correspond to the mean values described previously with one additional change; we shifted the centers for the sub-optimal funnels, μ_2, μ_3, and μ_4, such that the size of each basin after the weights have been applied, but prior to any scaling, were proportional to the number of funnels in the problem. In other words, we altered the centers of each sub-optimal funnel such that they all occupied the same volume in the search space before they are scaled by s_j. This was done to control the relative size of each funnel more effectively. These new center values can be replicated by applying the following substitutions.

$$\begin{aligned} \mu_2: &\quad -2.5 \quad \Rightarrow \quad -2.398179 \\ \mu_3: &\quad -2.5 \quad \Rightarrow \quad -2.236467 \\ \mu_4: &\quad -2.5 \quad \Rightarrow \quad -2.236467 \end{aligned}$$

Finally, the local optima are added.

$$h(\vec{x}) \quad = \quad -A \cdot (\cos 2\pi(x - \mu 1) + 1)$$

where the value A controls the amplitude of the cosine component and has a value of $A = 7$ in this paper. This is smaller than Rastrigin's original, which is $A = 10$, and makes our version of the single-funnel problem slightly easier.

There is a close relationship between the width of a funnel and the size of its basin of attraction. If two funnels have the same width, meaning they have not been scaled ($s_j = 1$), then the size of each funnel's basin of attraction is the same. However, applying different scaling factors to each quadratic funnel changes their relative widths. This also causes the basin of attraction size to change. A two-dimensional surface contour for the double-funnel instance is displayed in Figure 3. The optimal funnel is shaded in each case. The surface on the left has a sub-optimal funnel scaled by $s_2 = 0.87$, creating a wider optimal funnel and a larger basin of attraction. The sub-optimal funnel on the right has been scaled by $s_2 = 1.15$, which creates a more narrow optimal funnel and a smaller basin of attraction.

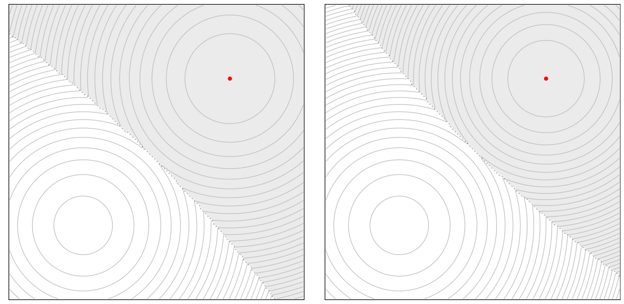

Figure 3: Funnel bias. When both quadratic functions are the same size, there is a linear basin of attraction boundary between the two functions (not shown). If the the optimal quadratic is larger (left) or smaller (right), there is a quadratic boundary defining the basin of attraction.

2.2 Measuring Bias

Small biases in low dimensions can have a large impact as dimensionality increases. It is unclear how the discrepancies in funnel size that exists in low dimensions (e.g. Figure 3) will impact the global structure in higher dimensions. In order to understand this bias, we uniformly selected 10,000

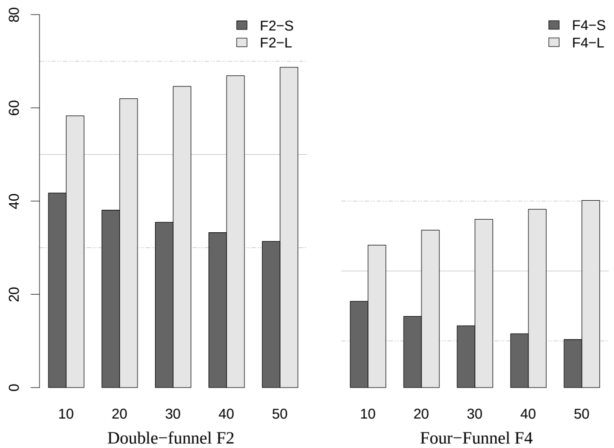

Figure 4: Understanding funnel bias. As dimensionality increases, the size of the optimal funnel changes. This gives a relatively good indication of how large each funnel is as the dimensionality increases.

random points from the region $(-5, 5)$, which are the bounds of the problem. We repeated this 1000 times and asked the question: what percentage of points are in the optimal funnel? A point is in the optimal funnel when:

$$s_1 \cdot \sum_{i=1}^{N} (x_i - \mu_1)^2 \leq g(\vec{x})$$

We created four multi-funnel functions based on the bias they exhibit. As mentioned, in every case, we scaled the optimal funnel by $s_j = 1.0$ and added nothing to it (e.g. $w_1 = 0$). The second funnel in every problem (e.g. $j = 2$) has a weight of $w_2 = 0.5 \cdot N$, where N is the problem dimension. The weights for the third and fourth funnels on the four-funnel problem were set to $w_3 = w_4 = 0.625 \cdot N$. These values create a sub-optimal solution that reflect a 8%-10% difference from the global optima when considering the range of all the local optima in the search space. On the double-funnel instances, we scale the sub-optimal funnel by $s_1 = 1.15$, to create a more narrow optimal funnel, and by $s_1 = 0.87$, which renders the optimal funnel wider. The four-funnel problems use $s_1 = 1.08$ and $s_1 = 0.9$ to change the funnel width. Again, these values were experimentally chosen based how they change the relative size of the optimal funnel.

Table 1 summarizes and labels each function based on these parameter settings. For the remainder of this paper, F2-S and F4-S refer to the two- and four-funnel problems where the optimal funnel is more narrow than the other funnels, which creates a smaller basin of attraction. F2-L and F4-L correspond to the two- and four-funnel problems where the optimal funnel is wider and the basin of attraction is therefore larger.

Figure 4 shows the relative bias for each multi-funnel instance as dimensionality increases. On both F2-L and F4-L, the optimal funnel increases as dimensionality increases. The size of the optimal funnel decreases on problems F2-S and F4-S. These statistics give a relatively good indication of how much volume each funnel occupies in the search space. For example, in $N = 50$ dimensions, F2-S occupies about 30% of the search space and F2-L occupies about 70% of the search space. We show that this distinction greatly impacts the effectiveness of search.

Name	Type	Funnel (j)	s_j	w_j
F2-S	double-funnel	1	1.0	0
		2	0.87	$0.5 \cdot N$
F2-L	double-funnel	1	1.0	0
		2	1.15	$0.5 \cdot N$
F4-S	four-funnel	1	1.0	0
		2	0.9	$0.5 \cdot N$
		3	0.9	$0.625 \cdot N$
		4	0.9	$0.625 \cdot N$
F4-L	four-funnel	1	1.0	0
		2	1.08	$0.5 \cdot N$
		3	1.08	$0.625 \cdot N$
		4	1.08	$0.625 \cdot N$

Table 1: The parameter settings for the six multi-funnel test functions used in this paper. The values s_j and w_j refer to the scale and weight applied to the j^{th} funnel of the problem.

3. ALGORITHMS

It is commonly believed that population-based methods are better at optimizing multimodal functions because they tend to explore more of the fitness landscape before their population converges to a compact and globally competitive region search space. This approach is appealing because, by comparing the values of each candidate solution, population-based algorithms have a better perspective of the entire search space than a purely *local search* method. This strategy has been reinforced with empirical studies.

In this paper, we describe and evaluate two evolutionary algorithms that utilize this approach: CHC, a non-traditional genetic algorithm, and CMA-ES, an evolution strategy with covariance matrix adaptation.

3.1 CHC

CHC [4] diverges from the traditional genetic algorithm in that it selects two parents for recombination in a uniform random way – there is no bias toward selecting better individuals. Instead, selective pressure is created using *cross-generational* truncation selection: newly created offspring must compete with the parent population for survival.

CHC also uses a modified version of uniform crossover, where exactly half of the non-matching bits are exchanged. The children under this scheme are always the same maximal Hamming distance from both parents. Further steps are taken to ensure that parents are not allowed to cross unless they are sufficiently different. Eshelman refers to this as *incest prevention* [4]. Initially, two strings must be different by at least $L/4$ bits, where L is the length of an individuals string. This *difference threshold* decreases by 1 each time crossover fails to produce an improving individual. This means that over time, crossover may occur on strings that are more similar. Eshelman states that incest prevention coupled with a difference threshold enables CHC to perform a coarse grain search initially, which preserves diversity, and eventually resort to a more fine grain exploitive search.

No mutation is used to alter one generation to the next. Instead, when the difference threshold decreases to zero, CHC initiates a restart mechanism called *cataclysmic mutation*, that re-initializes the entire population by randomly flipping 35% of the bits of the best individual.

3.2 CMA-ES

The canonical *evolution strategy* is an iterative process where a population of μ distinct parents produce λ offspring based on mutation distributions around the parents. Intermediate recombination creates a single parent based on the average position of the current population. A (μ, λ) selection strategy considers only the best μ of the current population for recombination.

An evolution strategy with *Covariance Matrix Adaptation*, or CMA-ES, uses a covariance matrix to explicitly rotate and scale the mutation distribution [7]. The orientation and shape of the distribution are directly calculated based on the *evolution path* and local information extracted from large populations. Hansen and Ostermeier define the reproduction phase from generation g to generation $g+1$ as:

$$x_k^{(g+1)} = \langle x \rangle_\mu^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{D}^{(g)} z_k^{(g+1)}$$

where $z_k^{(g+1)}$ are randomly generated from an $N(0, I)$ distribution. This creates a set of base points that are rotated and scaled by the eigenvectors ($\mathbf{B}^{(g)}$) and the square root of the eigenvalues ($\mathbf{D}^{(g)}$) of the covariance matrix C. The single global step size, $\sigma^{(g)}$, is calculated based on the length of the evolution path. Finally, the points are translated to center around $\langle x \rangle_\mu^{(g)}$, the mean of the μ best parents of the population.

To compute covariance, CMA-ES uses a time dependent portion of the evolution path. The evolution path updates after each generation using a weighted sum of the current path, $p_c^{(g)}$, and a vector that points from the mean of the μ best points in generation g to the mean of the μ best points in generation $g+1$. A principle components analysis on the evolution path is used to update the mutation distribution.

When a larger population (λ) is used, the best μ individuals may help describe the topology around the mean of the current generation [6]. This is called the rank-μ update. Hansen and Kern have empirically shown that CMA-ES performs well on multi-modal problems when large populations and rank-μ updates are employed.

The distribution used by CMA-ES is initially isotropic. As a result, the initial value of σ is critical for exploration. Hansen and Ostermeier suggest that the quality of solutions found by CMA-ES using a small initial step-size are often determined by the location of the starting point [7]. Auger and Hansen suggest an initial step-size that is half of the constrained region [1].

4. EMPIRICAL RESULTS

The goal in this section is to understand how global structure affects the algorithms described in the previous section. This is not meant to be a competition between algorithms in order to identify which strategy is the most effective. In fact, our results with respect to *relative algorithm performance* should not be interpreted too strongly because algorithm- and function-specific parameter settings can easily change their performance rank. We are also aware that the local optima in our problem are separable and symmetric, and unrealistic feature that CHC exploits. Instead, we are showing that *both* algorithms are affected by global structure. We provide empirical evidence that suggests that, as dimensionality increases, global structure alone can cause *both* of these algorithms to fail.

We evaluated CMA-ES and CHC on each test function from $N = 10$ to 50 dimensions. Our primary goal is to estimate the probability that an algorithm will find the optimal solution. We estimate this by running 100 trials of each algorithm and measuring the proportion of trials that found the center of the optimal funnel.

Most evolutionary algorithms stop when they have reached a predetermined number of evaluations. For example, although CHC undergoes cataclysmic mutation when the diversity of its population is too homogeneous, it will continue to do so until it has reached the maximum number of evaluations. CMA-ES, on the other hand, has well defined stopping criteria such that it may terminate before using the maximum allowable evaluations. To make our comparisons fair, we added random restarts to CMA-ES. Each algorithm in our comparison ran for exactly 15,000N evaluations, where N is the problem size. This is a sufficient number of trials to allow each algorithm to converge at least once per trial.

Every search algorithm has a set of control variables that determine its behavior. Typically, CHC uses a population size of 50, regardless of dimensionality. We also tested a version of CHC with a population size of 100. The bit-precision is usually set to 10 or 20 bits. A higher precision search will tend to restart less often because the difference threshold that controls the restart mechanism is based on the string length. We encoded each parameter using 20 bits.

The default parameters for CMA-ES are also robust. The value of the initial step-size is usually $\sigma_0 = (U - L)/2$, where L and U are the lower and upper bounds of the constrained search space. Since our problems are bound between $(-5, 5)$, this corresponds to a step-size of $\sigma_0 = 5$. Increasing the population size has also been shown to increase the performance of CMA on multimodal functions [5, 1]. We ran CMA-ES using three population sizes: $\lambda = 4 + \lfloor 3 \log(N) \rfloor$ (the default), $\lambda = 5N$, and $\lambda = 10N$.

4.1 Perspective

In order to assess how the multiple funnels affect these algorithms, we ran each algorithm on Rastrigin's single-funnel function. We used the same amplitude that our multi-funnel problems used, $A = 7$, and shifted the optimum to the value of $\mu_1 = 2.5$ in every dimension. No weight was added (e.g. $w_1 = 0$) and no additional scaling was used (e.g. $s_1 = 1$). The exact form of our modified version Rastrigin single-funnel function, F1, is:

$$F1 = \sum_{i=1}^{N} (x_i - 2.5)^2 - 7 \cdot (\cos 2\pi(x_i - 2.5) + 1)$$

Figure 5 shows the number of times each algorithm was successful as dimensionality increases. CMA-ES performed the best with the largest population sizes, $\lambda = 10N$ being the most effective. CHC using a population size of 50 was more effective than CHC utilizing the larger population size of 100. Both of these results are consistent with previously reported behavior [5, 11].

Given the same precision, CHC will restart more frequently using a smaller population, but the restarts are not random. Cataclysmic mutation initializes the new population by altering 35% of the best individual. Although any single bit

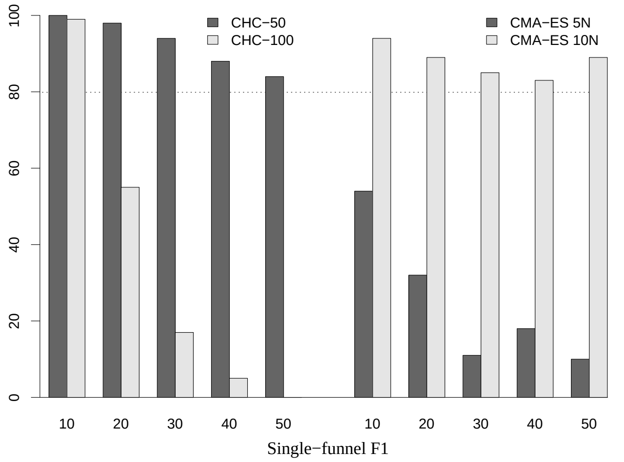

Figure 5: Success rate as dimensionality increase on the modified Rastrigin (single-funnel) function.

may result in a change that is large or small in Euclidean space, the best individuals of the new population are likely to be near the best individual. This means that CHC with a small population size is converging quickly and restarting locally.

On the other hand, larger populations are often more effective for CMA-ES on multimodal problems. Smaller population may be more efficient, in terms of evaluation calls, but also tend to get stuck in local optima more frequently.

The main goal here is to establish a reference point. We observe that both CHC and CMA-ES solve this problem more than 80% of the time, regardless of dimension.

4.2 Multi-funnel Results

What happens to the success rates of these algorithms when additional funnels are added to the search space? The success rates as a function of dimensionality for each multi-funnel problem are displayed in Figure 6. The most striking feature is the rapid decrease in each algorithm's ability to locate the global optima on problems where the optimal funnel occupies a smaller volume in the search space. In every problem where the optimal funnel's basin of attraction is smaller (F2-S and F4-S), none of the success rates are above 10%. This is especially disturbing on the double-funnel instance, F2-S, since we know that the optimal funnel occupies approximately 30% of the search space. This is hardly like searching for a needle in a haystack. When the optimal funnel has a larger basin of attraction than the other funnels (F2-L,F4-L), the affect on performance compared to each algorithm's behavior on F1 is less noticeable.

The optimal funnel's volume in the search space does not appear to be the main problem. For example, F4-L is approximately 40% of the search space, yet neither algorithm appears to struggle with this problem. This implies that the *relative* size of each funnel's basin of attraction may be more important that the *actual* size of the optimal funnel's basin of attraction. The remainder of this paper focuses on why these algorithms perform poorly—and even fail—when the optimal funnel occupies a small portion of the search space.

4.3 Understanding Exploration

There are several reasons why an algorithm may fail to

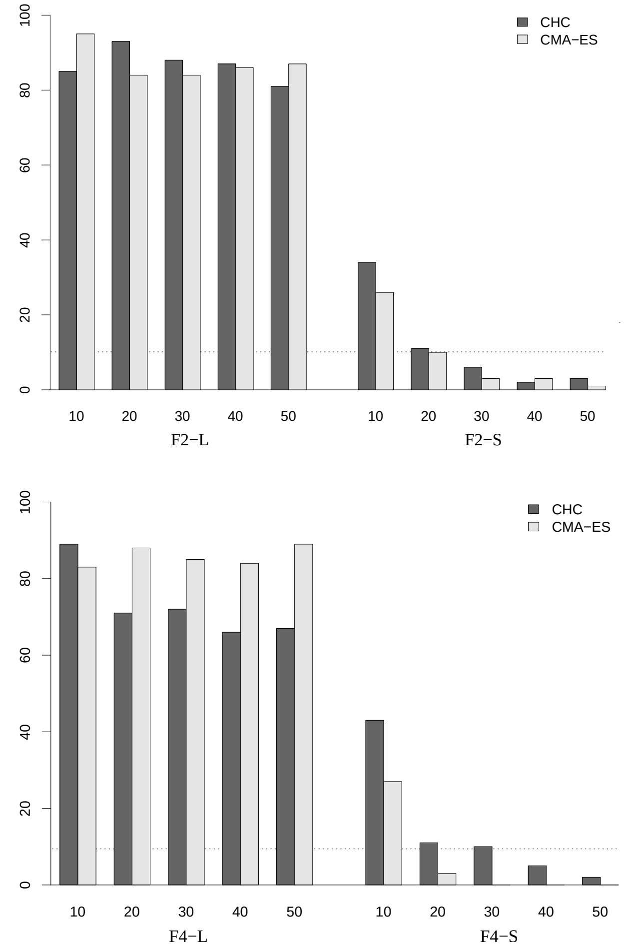

Figure 6: Success rates on the multi-funnel Rastrigin function. When the optimal funnel has a large basin of attraction (F2-L and F4-L), both CHC and CMA-ES perform well. However, when the optimal funnel is more narrow, resulting in a small basin of attraction (F2-S and F4-S), the probability that each algorithm finds the optimal solution noticeably decreases.

find the global optima, but the most common explanation is that it has converged to a local optimum. Of course, on multi-funnel landscapes, the local optimum may be in any funnel. Given the success rates for each algorithm on F1, F2-L and F4-L, we believe that the local optima in the problem are *not* the most detrimental feature.

We conjecture that when the optimal funnel is narrow, meaning it occupies a relatively small portion of the search space, exploration tends toward the funnel with the largest basin of attraction, relative to the others, and not the funnel that is the deepest.

If our conjecture is correct, we would expect that, on average, when search is failing to locate the global optima, it is getting stuck in a local optimum that is in the larger funnel. We measured the proportion of the population that was in the optimal funnel when the most effective solution was found. In other words, for each of the 100 trials, we computed the proportion of the population that was in the optimal funnel at the time when the best solutions was encountered. These proportions are very close to either 1 or 0, since the entire population is usually in a single funnel when it converges.

We found that in 50 dimensions, all of the trials for CMA-ES converge to the largest funnel on F2-S and F4-S. The results for CHC are less pronounced; occasionally (about 3% of the trials) CHC found the optimal funnel. However, these number still suggest that CHC is biased toward the size of the funnel, not its depth.

The size of the population affects these results. In both CHC and CMA-ES, larger populations "pull" the distributions into the larger funnels more often. The CMA-ES results using a population size of $\lambda = 5N$ tend to get stuck in the larger funnel less than the results reported in Figure 6. Unfortunately, smaller populations also get stuck in local optima more frequently. The larger population size of 100 for CHC get stuck in the largest funnel more often than the smaller population size of 50 shown in Figure 6.

We can look at this another way. There are really only two characteristics that make this problem difficult: *global structure* and *local optima*. We can peel back different layers of difficulty to better understand which characteristic is having the greatest impact on these algorithms.

4.4 Testing a Simple Multi-sphere

When we remove the cosine term from each function, we are left with the simple underlying global structure. Recall that the global structure is:

$$g(\vec{x}) \quad = \quad \min_j \left(w_j + s_j \cdot \sum_{i=1}^{N} (x_i - \mu_{i,j})^2 \right)$$

We label these new test cases G2-L, G2-S, G4-L, and G4-S to avoid confusion with those perturbed by additional local optima. We stress that these four functions are incredibly simple; G2-L and G2-S have only two local optima in the search space; G4-L and G4-S have only four. Furthermore, each local optimum is a quadratic sphere—equally scaled, symmetric, and separable. We emphasize that a wide funnel, which created a larger basin of attraction, can now be viewed as a "wide" local optimum. The basin of attraction now refers to the local optima in the space not the funnel.

We ran each algorithm on these test functions with the hypothesis that if global structure was really the main detrimental feature, we would see results similar to those presented in Figure 6. Since we have decreased the complexity of our functions, we allowed only $5000 \cdot N$ evaluations per trial instead of $15{,}000 \cdot N$ evaluations used previously.

Figure 7 displays the success rates for these simple surfaces as a function of dimensionality. There is a close resemblance to the success rates given in Figure 6. The G2-S results are perhaps the most revealing. In 50 dimensions, the best solution has a basin of attraction that is roughly 30% of the entire search space and both algorithms only find this solution less than 5% of the time. A simple local search algorithm would, in expectation, find the best solution 30% of the time *without random restarts*.

One simple conclusion here is that, on difficult multi-funnel problems, exploration is more biased toward the size of the funnel and not the depth of the funnel. That is, too much exploration appears to "pull" search into the largest funnel, although it may not be the deepest (e.g. the funnel containing the global optimum).

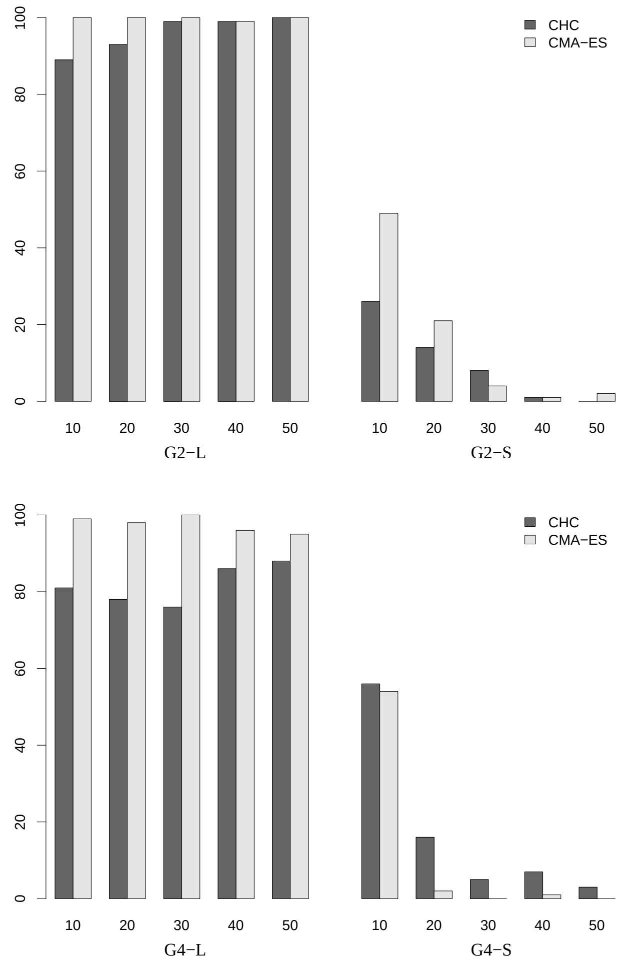

Figure 7: Success rates on the quadratic global structure. There are only two local optima on G2-L and G2-S and only four local optima on G4-L and G4-S.

4.5 Limiting exploration

The most difficult Lennard-Jones multi-funnel problems are solved by a Metropolis-type local search algorithm that operates on the local optima of the search space rather than the actual fitness function. This strategy is effective because it does not explore the entire search space, but rather exploits a single funnel at a time. By quickly comparing the best solution in each funnel, it is more adept to solving multi-funnel problems. This seems to indicate that the best global search methods for multi-funnel problems explore and exploit on a local, not global, level.

In the spirit of this idea, we tested a rather non-intuitive initialization of CMA-ES for multimodal landscapes. Instead of starting with the default initial step-size that has been shown to be effective on many single-funnel surfaces, we initialized σ with a much smaller value. Specifically, we set $\sigma = 5\%$ of the search space domain as opposed to the standard default value of 50%. We continued using the most effective population size of $\lambda = 10N$ in order to make our results more comparable. This configuration will not explore the entire search space, because of the small step-size, yet will hopefully escape the local optima that exist within each

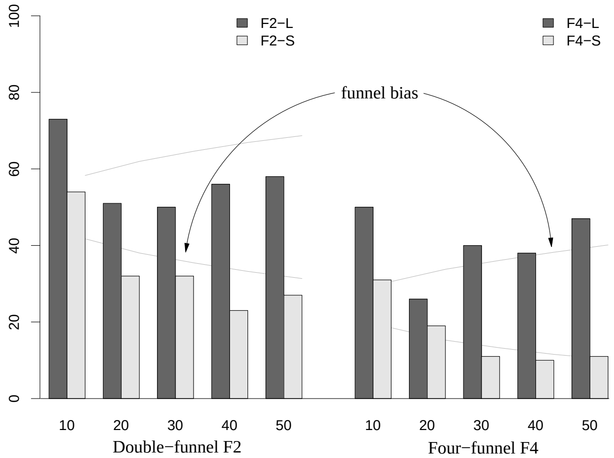

Figure 8: The results of CMA-ES initialized with a small step-size. The performance of this strategy is much more closely tied to the funnel size (gray lines).

funnel because it is using a larger population. The results are displayed in 8.

Surprisingly, limiting exploration is a much more effective strategy when the optimal basin of attraction occupies less volume in the search space. The results in Figure 8 are more tightly correlated to the size of the funnel (shown as a gray line). The success rates, compared to CMA-ES using the larger initial step-size, decrease when the optimal funnel is relatively large, but substantially improve when the optimal funnel is small. The highs are not as high, but the lows are still respectable.

This highlights one drawback to this method; the effectiveness of exploring funnels is limited to problems that contain relatively few funnels. That is, "local search methods" rely on several *restarts* in order to compare the best solutions that exist within each funnel.

5. DISCUSSION AND CONCLUSIONS

Global structure can clearly impact the performance of evolutionary optimization. When the optimal funnel is proportionally smaller than other funnels in the search space, the success rates for CHC and CMA-ES decrease dramatically. This phenomena even occurs on a simple multi-sphere function where exploration tends to pull search towards the local optima with larger basins of attraction. We believe these results will generalize to other algorithms as well. On a more limited basis, we have also tested Differential Evolution, but didn't report these results because DE failed on all our higher dimension test functions, regardless of problem structure.

Parameter settings can have a strong impact on empirical results. We intentionally created difficult multi-funnel problems to understand how global structure impacts search. Exploring how different weights (w_j) and scaling (s_j) values change our results may lead to a better understanding of how sensitive search is to bias in funnel depth and size. We found that even a small decrease in funnel size still resulted in low success rates.

Exploring the search space first, to gain a global perspective, before exploiting a particular region may be an effective strategy for "big valley", single-funnel problems. But as a general global optimization technique in high dimensions, the effectiveness of exploration may say more about our test functions than it does about the the effectiveness of this strategy. More attention is needed in this area.

This work supports an ongoing awareness that successful low-dimensional search strategies, such as exploration, are not always the best techniques in high dimensional space. Changing the way we view how exploration is affected by global structure —and implementing strategies that reflect this new philosophy—may expand the role that evolutionary algorithms play within the global optimization community.

6. REFERENCES

[1] A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *Proceedings of IEEE Congress of Evolutionary Computation*, 2005.

[2] K. Boese, A. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16, 1994.

[3] J. P. Doye. Physical perspectives on the global optimization of atomic clusters. In *Chapter in forthcoming Kluwer book Global optimization – select case studies*, 2006.

[4] L. J. Eshelman. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In R. G. J. E., editor, *Foundations of Genetic Algorithms*, pages 265–283. Morgan Kaufmann, 1991.

[5] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *PPSN*. Springer, 2004.

[6] N. Hansen, S. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[7] N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[8] R. H. Leary. Global optimization on funneling landscapes. *Journal of Global Optimization*, 18, 2000.

[9] P. M. Pardalos and F. Schoen. Recent Advances and trends in global optimization: Deterministic and stochastic methods. In *Proceedings of the Sixth International Conference on Foundations of Computer-Aided Process Design*, 2004.

[10] D. J. Wales. Energy Landscapes and properties of biomolecules. *Physical Biology*, 2:S86–S93, 2005.

[11] D. Whitley, R. Beveridge, C. Graves, and K. Mathias. Test driving three 1995 genetic algorithms: New test functions and geometric matching. *Journal of Heuristics*, 1:77–104, 1995.

[12] D. Whitley, S. B. Rana, J. Dzubera, and K. E. Mathias. Evaluating Evolutionary Algorithms. *Artificial Intelligence*, 85(1-2):245–276, 1996.