

# A Multiple Population Differential Evolution Sampler for Trade Space Visualization

Dan Carlsen<sup>\*</sup>  
Pennsylvania State University  
University Park, PA 16802  
dec204@psu.edu

Christopher Congdon<sup>†</sup>  
Pennsylvania State University  
University Park, PA 16802  
cxc242@psu.edu

## ABSTRACT

In this paper the evolutionary strategy (ES) of differential evolution (DE) is expanded upon through the use of modified speciation and crowding operators to thoroughly explore a trade space. Typically, an evolutionary strategy such as differential evolution will either converge to a single point or to a set of Pareto optimal points (in the case of multi-objective optimization). However, with the proposed modifications to DE, not only will it allow for multiple optima to be located (converged to), an entire trade space can be explored; species will move towards user-specified attractors (a way to bias the exploration) in the trade space, yet individuals will still be created elsewhere. This gives the decision maker the necessary information to determine their preferences *a posteriori*.

## Categories and Subject Descriptors

G.1 [Numerical Analysis]: Optimization; E.2 [Data Storage Representations]; F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems

## General Terms

Algorithms

## Keywords

Evolutionary Computation, Differential Evolution, Visualization, Speciation, Trade Space Exploration

## 1. INTRODUCTION

Evolutionary Strategies have been traditionally used to locate optimal solutions in an objective space. However, simply attempting to optimize a problem is not always the best

<sup>\*</sup>Graduate Student, Department of Mechanical Engineering

<sup>†</sup>Graduate Student, Department of Industrial and Manufacturing Engineering

course of action. The underlying reason being that oftentimes no optimal solution to a problem can exist that will satisfy everyone when a decision is being reached by a large number of decision makers [2].

Traditional multiobjective optimization techniques require the decision-maker (DM) to quantify their preferences *a priori*. These preferences drive the algorithm to define a Pareto Frontier of non-dominated solutions that the DM can evaluate. However, sometimes the DMs do not fully understand their preferences *a priori* [1]. In order to address this problem, the ARL Trade Space Visualizer (ATSV) has been developed to allow for thorough exploration of a trade space [9]. This research describes a tool DMs will be able to utilize in ATSV to steer the exploration of that trade space.

The overall goal is to be able to populate a multiobjective trade space using differential evolution based upon an attractor (a way to bias the exploration) on a reduced dimensional subset of the objective space. By using a combination of modified forms of speciation and crowding, the algorithm can effectively explore the unconstrained dimensions about this user-specified attractor, which could be located in either the decision or objective space, instead of merely converging to a single solution. This allows the DM to properly evaluate the complex interactions between decision variables in order to determine their preferences *a posteriori*. This exploration provides additional information regarding other areas of the trade space which may affect the DMs preference. This is the key distinction between the Design by Shopping paradigm and traditional multiobjective optimization.

In general, the objective function is:

$$\begin{aligned} \text{Minimize} \quad & |\vec{f}(\mathbf{X}) - \vec{\Phi}| \\ \text{s.t.} \quad & \\ & a_i \leq \mathbf{X}_i \leq b_i \end{aligned}$$

where  $\mathbf{X} \in \mathbb{R}^d$  is a  $d$ -dimensional decision with a support  $[a_i, b_i]$  which maps to  $\vec{f}(\mathbf{X}) \in \mathbb{R}^m$ , an  $m$ -dimensional objective space.  $\vec{\Phi} \in \mathbb{R}^k$  is a  $k$ -dimensional attractor location where  $k \leq m$ . Note that if  $k = m$  then the problem is a multiobjective optimization problem with a goal of converging to a specified point in the trade space.

Let us consider, for example, an  $n$ -dimensional input vector  $\mathbf{X} \in \mathbb{R}^n$  which maps to a 2-dimensional objective space

$\{f(\mathbf{X}), g(\mathbf{X})\}$ . If an attractor  $\Phi_L$  is placed on  $f(\mathbf{X})$  such that the objective function becomes:

$$\text{Minimize}|f(\mathbf{X}) - f_{\Phi_L}|, \quad (1)$$

where  $f_{\Phi_L}$  is the location of the attractor. Then the goal is to populate the line  $f(\mathbf{X}) = f_{\Phi_L}$  while allowing  $g(\mathbf{X})$  to freely vary. This attractor  $f_{\Phi_L}$  is then referred to as a line attractor.

With a 3-dimensional objective space  $\{f(\mathbf{X}), g(\mathbf{X}), h(\mathbf{X})\}$ , if an attractor  $\Phi_P$  is placed on  $\{f(\mathbf{X}), g(\mathbf{X})\}$ , the objective function becomes:

$$\text{Minimize}|\{f(\mathbf{X}), g(\mathbf{X})\} - f_{\Phi_P}| \quad (2)$$

where  $\{f(\mathbf{X}), g(\mathbf{X})\}$  is the point location of the attractor. Then the goal is to populate the line  $\{f(\mathbf{X}), g(\mathbf{X})\} = \{f, g\}$  while allowing  $h(\mathbf{X})$  to freely vary. This attractor  $\Phi_P$  is then referred to as a point attractor because of its 2-D representation, but in reality  $\Phi_P$  is just the 3-D representation of the line attractor  $\Phi_L$  described in the first example. This process effectively ‘‘paints’’ the trade space around the attractor, allowing preference to be specified only in a limited subset of the decision space while allowing the unspecified objectives to be explored.

The next section gives some background information for the ATSV, differential evolution, and multimodal optimization techniques such as speciation and crowding, which will be adapted for multiobjective use. Section 3 discusses the modifications to DE required to achieve the aforementioned goals. Section 4 investigates the effects of parameter selection and provides two test cases to demonstrate the implementation of this algorithm; yielding insight into the obtained results. Finally, Section 5 summarizes the key findings of this work.

## 2. BACKGROUND

Multimodal and multiobjective optimization share similar goals. The process of locating multiple local optima of a multimodal function and defining a Pareto Frontier both require extensive exploration. In both cases the algorithm must resist the natural desire to converge to a single solution. Both techniques share some commonality with the desired goals of this research (to explore a trade space). This section provides an overview of the current literature pertaining to trade space exploration, differential evolution and the evolutionary exploration techniques of speciation and crowding. The goal is to provide a motivation for the proposed work.

### 2.1 Design by Shopping

The ATSV is essentially an implementation of the Design by Shopping paradigm proposed by Balling [1]. The Design by Shopping paradigm is an idea where decision-makers are presented with a large number of designs in a trade space and then choose an optimal solution (in their eyes) from this set. The major advantage of this is that it allows the decision-makers to form their design preferences after visualizing the entire design space *a posteriori* and then choose an optimal based upon those preferences [9].

ATSV is a data visualization interface which allows users to visualize complex design spaces through the use of multi-dimensional visualization techniques such as customizable

glyph plots, parallel coordinates, linked views, brushing, and histograms. In addition the user can specify upper and lower bounds on the design space variables, assign variables to glyph axes and parallel coordinate plots, and dynamically brush variables. Also, preference shading to help in visualizing a user’s preference structure and algorithms for visualizing the Pareto frontier have been added to the interface to help in shaping a decision-maker’s preference [9].

### 2.2 Differential Evolution

Differential evolution was developed by Storn and Price [7]. Like other evolutionary algorithms it incorporates operators of mutations, crossover, and selection. It uses a population  $NP$  of  $d$ -dimensional vectors to minimize an objective function. The population size  $NP$  does not change during the run from generation to generation. The initial population is randomly generated with the hope of covering the entire decision space. Mutation is performed by adding a weighted difference between two random population members to a third random member. Crossover then occurs by mixing this mutated member’s parameters with another predetermined target population member. Selection then occurs between the target and trial member by comparing their fitness values. The member with the better fitness survives. Each population member serves as the target member during a generation so that in a generation there are  $NP$  competitions.

Differential evolution was chosen for this work because it requires few parameter settings, is a real-coded algorithm, and is separable. This separability is desirable in a sampling algorithm because it allows variables to be decoupled from each other, which fosters diversity about unconstrained dimensions of the trade space.

#### 2.2.1 Operators

This section discusses the Evolutionary Operators that are employed in DE that are relevant to this work.

**Mutation.** There are many DE mutation schemes in the literature, refer to [5] for a detailed comparison of the options. For this report, the strategy  $DE/rand/1/exp$  is utilized. This strategy works by selecting 3 random parents and recombining them in various fashions in order to generate a child. Eq. 3 demonstrates this approach, with a mutant vector  $\vec{V}_{i,g}$  [5]:

$$\vec{V}_{i,g} = \vec{x}_{r_0,g} + F \times (\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) \quad (3)$$

where  $F \in (0, 1+)$  is the user defined scale factor parameter and  $\vec{x}_{r_i,g}$  where  $i \in (1, 2)$  is the randomly selected individual from the current species.

**Mating.** After a mutated vector  $\vec{V}_{i,g}$  is generated, it is mated with a sequentially-determined ‘‘parent’’ population member  $\vec{x}_i$  through a cross-over strategy. A starting point,  $i \bmod d$ , is selected and the probability of cross-over is the user-defined parameter  $CR$ . If cross-over is successful, the  $i^{\text{th}}$  gene of the mutated vector  $\vec{x}_i$  is swapped with the  $i^{\text{th}}$  gene of parent vector  $\vec{x}_i$ . Then,  $i$  is indexed and the process is

repeated until the cross-over is unsuccessful. If cross-over is unsuccessful, the mating process stops.

**Selection.** After a child is generated by the mutation/mating operators, it competes for survival with the originating parent. Only the individual with the best fitness survives to the next generation.

## 2.3 Multiobjective/Multimodal Optimization

Both multiobjective and multimodal optimization force diversity and delay convergence of the sampling set in order to meet their stated objectives. This section will discuss two tools which can be used to achieve the desired results.

### 2.3.1 Speciation

One technique to locate multiple optima is the idea of speciation used by Li in the development of his species-based DE (SDE)[3]. Speciation is a niching method where the algorithm's population is grouped based on Euclidean distance. In DE, it keeps the randomly chosen members used for cross-over and mutation to those within the species. This effectively limits the algorithms step-size to the neighborhood of each individual species, therefore reducing selection pressure for the global population. This distance calculation is

$$|\vec{x}_i, \vec{x}_j| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (4)$$

where  $\vec{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,d}\}$  and  $\vec{x}_j = \{x_{j,1}, x_{j,2}, \dots, x_{j,d}\}$  are  $d$ -dimensional decision vectors which represent two individuals  $i$  and  $j$  from the population. A radius parameter  $r_s$ , which represents the distance from the center of a species to its boundary, must be user specified.

At the beginning of each iteration the entire population  $\mu$ :

$$\mu = \sum_{i=1}^{NS} NP_i, \quad (5)$$

where  $NS$  is the number of species and  $NP_i$  is the population of species  $i$ , is sorted based on fitness and the fittest member is selected to be a species seed. The next individual's distance to the seed is checked and if it falls within a specified radius  $r_s$  from the seed it is considered to be part of that species; otherwise it is set to be another species seed.

This is done for all  $NP$  individuals in the population. Once the species are established, a check is done to ensure that each species has at least  $M$  individuals. In DE,  $M$  is usually set to a number greater than or equal to 3 because DE requires 3 or more individuals within each species to implement its mutation and cross-over operators. If the species does not contain enough individuals then new individuals are randomly created within the species radius until at least  $M$  individuals exist. After the DE is run for each species, the entire population is resorted based on fitness.

Adding individuals to the species could potentially create a population larger than  $\mu$ , but if this is the case, only  $\mu$  individuals total are selected to be species seeds and members of a species. Another potential problem is that some species

may converge more quickly than others. When this happens, some individuals may become redundant. To combat this problem, whenever an individual is created it is checked against the other individuals within the current species. If it is a copy, it is replaced by a new random individual [3].

### 2.3.2 Crowding

Another multimodal optimization technique of interest for this work is that of crowding [10]. The concept of crowding is used to maintain diversity in the population, which encourages exploration and increases the probability of converging to multiple optima. It achieves this by only allowing a newly generated offspring  $p_o$  to compete with the individual  $p_i$  that is most similar to it. Thomsen defined similarity by Euclidean distance in the genotypic space, but a phenotypic-based implementation can also be employed. This offspring will then replace  $p_i$  in the next generation if it has a better fitness. The pool of individuals that  $p_i$  is chosen from is a random subset (crowd) of the population with a crowd size set by the crowding factor ( $CF$ ); usually taken to be a small number such as 2 or 3.

Due to the  $CF$  usually being set to a low number, crowding has been known to experience a problem called replacement error, where the offspring replaces an individual which is not similar to it [10]. One way to combat this is to set the  $CF$  equal to the population size, guaranteeing that it replaces the most similar individual in the population. This has the drawback of a longer runtime, but this increase in runtime is usually insignificant when compared to fitness evaluation times.

## 3. METHODOLOGY

This paper presents a new methodology hereby referred to as Adaptive Sampling Differential Evolution (ASDE). The foundation of this algorithm is based upon the traditional DE operators coupled with a speciation strategy presented by Li [3] and a crowding strategy presented by Thomsen [10] but tailored for the Design by Shopping Paradigm [1]. This section presents the ASDE algorithm, describes its key operators and discusses strategies to set parameters for specific applications within the Design by Shopping Paradigm.

### 3.1 Problem Formulation

The objective of this algorithm is to bias samples of a  $d$ -dimensional decision space, which maps to an  $m$ -dimensional objective space, towards a  $k$ -dimensional attractor  $\vec{\Phi}$  while exploring the non-preferred dimensions of the trade space. Fitness is defined as the Euclidean distance to the attractor, which may be located in either the decision (genotypic) or objective (phenotypic) space.

For this study, the decision space is constrained to a subset of  $\mathbb{R}$ , requiring a modification of the mutation operators to ensure feasibility. To address scaling issues, each dimension is transformed to a 0-1 scale. The user must specify a dimensional tolerance parameter  $\epsilon$ , for each dimension, which represents how distinguishable the user requires each dimension to be. Any vector where all genes fall within  $\epsilon$  of another vector will be considered identical to that vector.

### 3.2 Operators

This section details the modifications from the traditional DE for use with ASDE.

### 3.2.1 Mutation

For ASDE, the mutation approach is similar to that from DE, but an additional feasibility check is added, which is discussed in the mating section.

### 3.2.2 Mating

After a cross-over decision is made over each dimension, the feasibility of the generated child (candidate) is required. If the child is not feasible, the mutation and mating process is repeated up to  $NP$  times, where at that time a randomly-generated candidate is created to compete against the parent individual. This candidate is guaranteed to be feasible in the decision space based upon the criteria for random number generation. This feasibility cycle is derived from Storn [6] and was modified for ASDE to incorporate the randomly-generated alternatives.

### 3.2.3 Selection

In ASDE, crowding is used to modify the selection operator. This crowding is implemented similar to Thomsen [10], the key difference being the usage of speciation. This crowding strategy is implemented without the requirement of any additional user-defined parameters, since  $CR = NP$ . This guarantees that the candidate individual competes against the closest member (within the species). Li [3] argued against Thomsen’s crowding scheme because of the computational expense of searching for the nearest neighbor of a candidate. However, by incorporating crowding into a species-based algorithm this search is limited to within the species, thus reducing its cost.

### 3.2.4 Speciation

While the general idea of speciation is appealing for this work, a few aspects of it need to be considered more carefully; the fact that even though the species converge to different locations, they still converge, and the implications of having to specify a radius. With respect to the first concern, ways to prevent the entire population from converging need to be considered since the goal is to populate the whole trade space instead of just having multiple species converge to the attractor.

A speciation strategy similar to Li [3] is implemented, with two key differences. First, the radius parameter  $r_s$  is eliminated in an effort to improve problem scalability. Second, the speciation strategy is combined with the previously mentioned crowding strategy.

The main concern regarding Li’s speciation strategy pertains to the use of  $r_s$ . Specifying the radius  $r_s$  requires problem specific knowledge which can greatly impact performance and can be difficult to interpret from an operator perspective. If the number is too small, each population member will become its own seed, generating a total of  $NP$  total species. Each species will then need to be increased to  $M$  members by to ensure that the DE strategy can perform. Li suggests that these extra members be generated randomly which can lead to a random-walk condition, as  $(M - 1)$  randomly-generated individuals compete with the species

seed (worst-case). This leads to a total of  $(M - 1) \times NP$  randomly-generated individuals in a generation. The next section discusses how ASDE addresses these issues.

## 3.3 Identifying Species

ASDE eliminates  $r_s$ , and replaces the parameter by a species parameter,  $NS$ . This parameter is more intuitive to the user, as it directly corresponds to the desired amount of diversity in the search space. By specifying  $NS$  and the population size  $NP$ , the user is generating a total of  $\mu$  parents in each generation, where  $\mu = NS \times NP$ . This also eliminates the need for the parameter  $M$ , as all species have  $NP$  members.

## 3.4 Generating Species

For ASDE, the defining member of each species is a species seed. By definition, the species seed will have the highest fitness of a given species, and the membership of each individual is determined solely on that individual’s location with respect to the species seed. Figure 1 describes the layout of the algorithm. This algorithm begins with a empty

```

input :  $L$ -a list of all individuals
output:  $S$ -a list of all dominating individuals identified
         as species seeds
          $\varphi$ -a list of all species  $\theta$ 

begin
   $\varphi \in \{\emptyset\}$ ;
  while not reaching the end of  $L$  do
     $L_{fit} \leftarrow \text{SORT}(L, \text{FITNESS})$ ;
     $p_{seed} \leftarrow p_0 \in L_{fit}$ ;
     $\theta \leftarrow p_{seed}$  Remove  $p_0$  from  $L_{Individuals}$ ;
     $L_{dist} \leftarrow \text{SORT}(L, \text{DISTANCE})$ ;
    for  $i = 1 : NP - 1$  do
       $\theta \leftarrow p_i$ ;
      Remove  $p_i$  from  $L$ ;
    end
    Let  $\varphi \leftarrow \theta$ 
  end
end

```

Figure 1: ASDE Algorithm for Determining Species

set of species  $\varphi$ . It takes  $L$  as an input, which is a compilation of surviving individuals from all species in the previous generation.  $L$  is sorted by fitness and placed in  $L_{fit}$ . The individual with highest fitness ( $p_0$ ), as determined by  $L_{fit}$ , is removed from  $L$  and becomes the first species seed  $p_{seed}$ . The list  $L$  is then sorted by Euclidean distance to  $p_0$  and becomes  $L_{dist}$ . At this time, the  $NP - 1$  closest individuals to  $p_0$  are inserted into species  $\theta$  that has seed  $p_{seed}$ , while being removed from  $L$ . Species  $\theta$  is added to the set of species  $\varphi$ ,  $L_{fit}$  is resorted from the remaining members of  $L$ , and the process is completed until there are  $NS$  total species in  $\varphi$ , each consisting of  $NP$  individuals.

While in SDE the species radius  $r_s$  is a user specified parameter, this strategy calculates an average species radius vector  $\vec{r}_s$ , which is the  $d$ -dimensional vector which represents the average distance of all  $NP - 1$  individuals from the species seed  $p_{seed}$ . This  $\vec{r}_s$  addresses the scaling issue of  $r_s$  by allowing each dimension to maintain its influence over the domain.

While SDE uses  $r_s$  for determining number and contents of the species, ASDE only uses  $\bar{r}_s$  for a boundary of randomly generated individuals. These randomly generated individuals occur either when the mating/mutation process generates either an infeasible point or a duplicate of a previously-generated parent. The next section will discuss the tracking of duplicate individuals through the use of a tabu-list.

### 3.5 Tabu List

Since the primary objective of ASDE is to efficiently sample points on the trade space for computationally expensive functions, there may be benefit in ensuring that a specific decision point is never sampled twice. In order to eliminate the risk of redundant calculations, each individual can be tested against a tabu list to ensure that it has not already been evaluated. When a tabu list is incorporated into an ES, the following decisions are required:

- Are duplicates deleted, or do they just inherit the matching solution’s fitness?
- What precision is required to declare a duplicate (see discussion below on  $\epsilon$ )?
- If a duplicate is deleted, how is it replaced?

Each of these questions have an effect on the balance of exploration and exploitation of the algorithm. For this paper, a tabu list in which duplicates are deleted and replaced by a randomly-generated member located within the species radius is used. These randomly-generated members help to explore the trade space.

The definition of a duplicate in relation to a continuous variable is critical, since by measure theory the probability of randomly selecting a specific point in space for a continuous variable is 0 ( $P(X = x) = 0, \forall X \in \mathbb{R}^d$ ). Of course, a computer representation of a continuous number is finite, but the probability of selecting a specific point is effectively 0. The concept of  $\epsilon$  allows the DM to specify a level of differentiation for each decision variable. It is assumed that the DM knows the critical resolution of each decision variable *a priori* and can specify an  $\epsilon$  where further differentiation is insignificant. This allows the algorithm to effectively employ fitness-sharing, so that individuals within a specified  $\epsilon$  for each dimension will not be evaluated again.

### 3.6 Convergence

Since the purpose of this strategy is to sample a decision space vs. specifically target a set of optimal solutions, the evaluation of convergence is left to the DM. From an algorithm perspective, the goal is to maintain the balance between exploration and exploitation with the goal of intelligently sampling the trade space. Mechanisms to address species pre-convergence must be addressed in future work.

One mechanism to ensure diversity is to replace a species at each generation with randomly-generated individuals to explore different regions of the trade space. In ASDE, this concept was implemented by dropping the last species (which contains the set of worst-performing individuals) and replacing the entire species with a set of random individuals,

seeded by the best overall individual from the previous generation. This strategy is promising as it injects a significant number of new candidate solutions, but these solutions do not compete with the best known solutions until the following generation because they are located in a different species (for the next generation)

A strategy to increase convergence to the attractor is in allowing each species to run for more than a single generation between resorting the species. This strategy will cause each species to “hone-in” on locally-optimal solutions prior to reintegration and re-speciation. For this study, new species are determined after each generation, similar to SDE [3].

### 3.7 Parameters for ASDE

The new parameters/decisions of interest are:

- *NP*: Species Population Size
- *NS*: Number of Species
- *F*: Scale factor parameter for mutation
- *CR*: Cross-Over Ratio for Mating
- $\epsilon_i$ : The level of differentiation for input  $i$
- *RANDOM?*: An option that replaces the individuals from the last species of a generation with randomly generated individuals.

### 3.8 Complexity

The computational complexity of this algorithm is determined by the number of sorting operations and the total population size. For each generation, there is one sort by fitness ( $L_{\text{fit}}$ ) and *NS* sorts by distance ( $L_{\text{dist}}$ ). For each  $L_{\text{dist}}$  calculation, a total of  $\mu - 1$  distance calculations are required. This gives a computational complexity for speciation of  $\mathcal{O}(NS \times (\mu - 1))$  which is similar to the results found by Li, but with a fixed number of species which makes the run-time in the control of the DM, based upon the user specified parameters *NP* and *NS*.

For the crowding operator, each individual must be compared against *NP* individuals in order to determine its closest neighbor. This requires a total of  $NS \times NP^2$  which is identical to  $NP \times \mu$ . Since typically  $NP > NS$ , the overall computational complexity of this algorithm will be  $\mathcal{O}(NP \times (\mu - 1))$  which is smaller than Thomsen’s  $\mathcal{O}(\mu^2)$  for crowding and Li’s  $\mathcal{O}(\mu \times NS)$  for speciation.

### 3.9 Measures

Traditional optimization algorithms measure performance based upon measures such as the number of functional evaluations (FEs) and reliability. On the other hand, an appropriate set of measures for comparing a sampling algorithm is the first 4 standardized moments of the distribution of samples (mean, variance, skewness and kurtosis). In particular, the kurtosis of the sample is a key performance measure. The sample kurtosis measures the bias of the sampling towards an attractor. A kurtosis of  $-1.2$  indicates no selection pressure, or a uniform distribution across the dimension. A kurtosis of 0 indicates a normal distribution while a kurtosis approaching inf indicates a degenerate condition with all

mass existing at a point [8]. For this investigation, all four of these measures are used to judge the effectiveness of the algorithm’s parameter settings.

## 4. EXPERIMENTATION

The test cases that will be modeled in this study were chosen to demonstrate the effectiveness of the ASDE algorithm to thoroughly explore the trade space.

### 4.1 2-D Simple Example

This example involves a direct mapping of the decision variables to the objective variables in 2-dimensions.

$$\begin{aligned} f(x, y) &= x \\ g(x, y) &= y \\ \Phi_f &= 0 \end{aligned}$$

where  $\Phi_f$  is an attractor on the objective variable  $f(x, y)$ . This scheme is  $d = 2$  for the decision space,  $m = 2$  for the objective space and  $k = 1$  for the attractor. The goal of this experiment is to demonstrate the effectiveness of ASDE to explore the line  $f(x, y) = 0$  throughout the range of  $g(x, y)$ . A visualization of this trade space is shown in Figure 2.

#### 4.1.1 Parameter Investigation

An experiment was conducted in order to understand the underlying dynamics of the algorithm and sensitivity of its parameters on the 2-D Simple Test Function. The factors of interest with their tested values were:

- *NP*: Species Population Size (10, 20)
- *NS*: Number of Species (3, 5)
- *CR*: Cross-Over Ratio for Mating (0.6, 0.9) input  $i$
- *RANDOM?*: Replace Last(lowest fitness) Species with Randomly-Generated Individuals

The scale factor ( $F$ ) was set to 0.5 based upon Price’s guidelines [4] and testing by both Li and Thomsen [3, 10]. The testing ranges for *NP* and *NS* were selected based upon the requirement to limit the *FE* requirement. While  $CR = 0.9$  was selected by Price, Li and Thomsen, the effect of *CR* on exploration was tested by adding  $CR = 0.6$  as a test parameter.

A full factorial experiment was run with each combination of the above parameter settings. For each setting, a set of 10 seeds were selected for random number generation. The statistics for these 10 runs were averaged for analysis. Each combination of parameters were run for 1000 functional evaluations, based upon the functional requirements set forth by the ATSV group. The results of this parameter study are shown in Figure 3.

#### 4.1.2 Key Findings

The results of the parameter study found that for the unconstrained variable  $g(x, y)$  the settings  $NS=5$ ,  $NP=20$ ,  $CR=0.9$  and *RANDOM?=Y*. Figure 4 demonstrates the progression of sampling over 1000 FEs for this optimally selected set

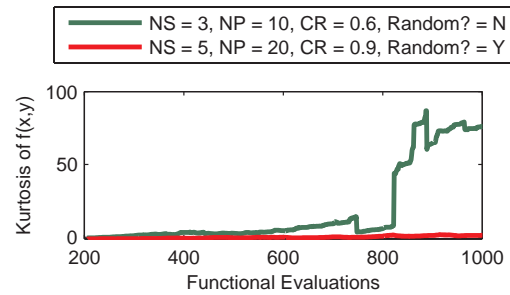


Figure 6: Moving average of kurtosis over FEs (200 FE Window) for Simple XY Example. This plot compares the change in kurtosis with respect to the number of functional evaluations between two extreme parameter settings.

of parameters. In contrast to the optimally selected parameters for exploration of  $g(x, y)$ , a time-series comparison of parameter settings which quickly target  $\Phi_f$  is shown in Figure 5. The key observation from this trial is how the algorithm quickly begins to find and “paint” the attractor in under 400 FEs. By monitoring the kurtosis of the sampling population with respect to the number of FEs, a sense of how quickly the algorithm finds the attractor can be established (Figure 6). It is proposed that this measure can be used to actively adapt ASDE to achieve the DMs sampling objective; this could be an area of future research.

## 4.2 Examples

### 4.2.1 2-D Crossed Example

This example employs a rotation to the previous example,

$$\begin{aligned} f(x, y) &= x \cos(\alpha) - y \sin(\alpha) \\ g(x, y) &= x \sin(\alpha) + y \cos(\alpha) \\ \Phi_f &= 0 \end{aligned}$$

where  $\Phi_f$  is an attractor on the objective variable  $f(x, y)$  and  $\alpha = 45^\circ$  is the angle of rotation. This scheme is also  $d = 2$  for the decision space,  $m = 2$  for the objective space and  $k = 1$  for the attractor. The objective space  $\{f(x, y), g(x, y)\}$  becomes dependent on both decision variables, which provides a more complex example for ASDE to evaluate. Figure 7 visualizes the trade space of the 2-D Crossed Example. Utilizing the information learned from the 2-D Simple Example, a smaller set of parameter combinations were run for this 2-D Crossed Example. The following combinations of parameters were compared:

- *NP*: Species Population Size (20)
- *NS*: Number of Species (3, 5, 8)
- *CR*: Cross-Over Ratio for Mating (0.9) input  $i$
- *RANDOM?*: Replace Last(lowest fitness) Species with Randomly-Generated Individuals

This generated 6 combinations of parameters. Figure 8 demonstrates the progression of sampling over 1250 FEs in a 3-D Surface Histogram format. The key observation from this figure can be seen clearly in Figure 8f. Two peaks, which

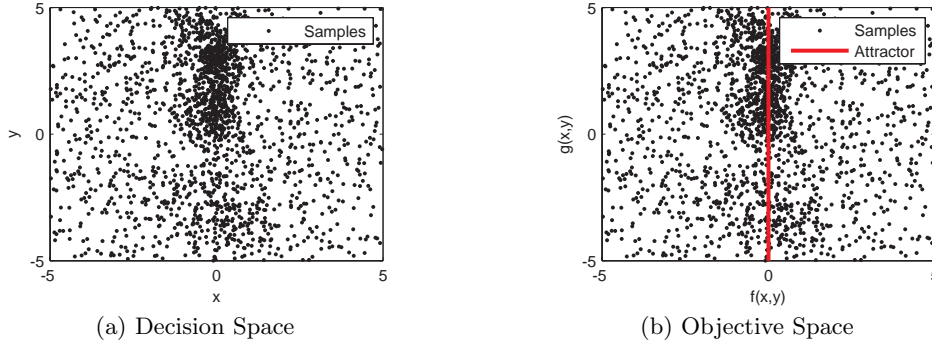


Figure 2: Trade Space of 2-D Simple Example. This example shows the results of 2000 FEs on an example with  $NS=3$ ,  $NP=20$ ,  $CR=0.6$  and  $RANDOM?=Y$

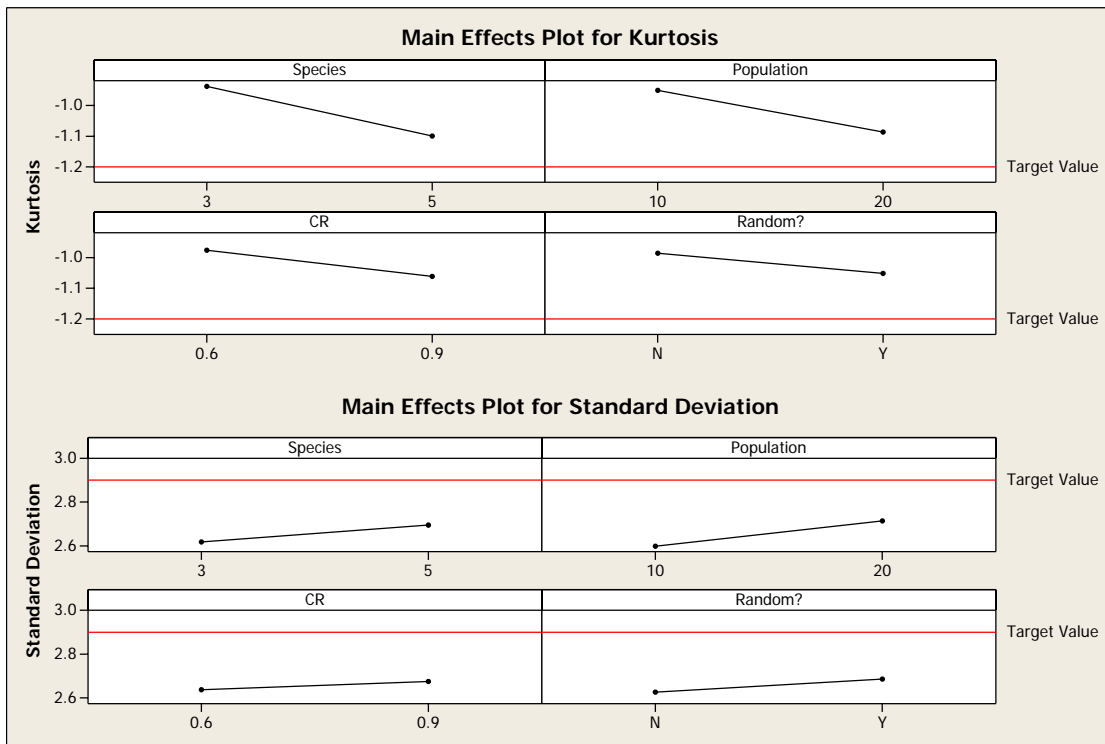


Figure 3: Main Effects Plot for the 2D Simple Example of  $g(x,y)$ . These are the results for the unconstrained objective variable,  $g(x,y)$ , averaged of 10 runs. 1000 FEs were computed for each run. The target values for the kurtosis and standard deviation of a uniform distribution with range  $[-5,5]$  are shown in red.

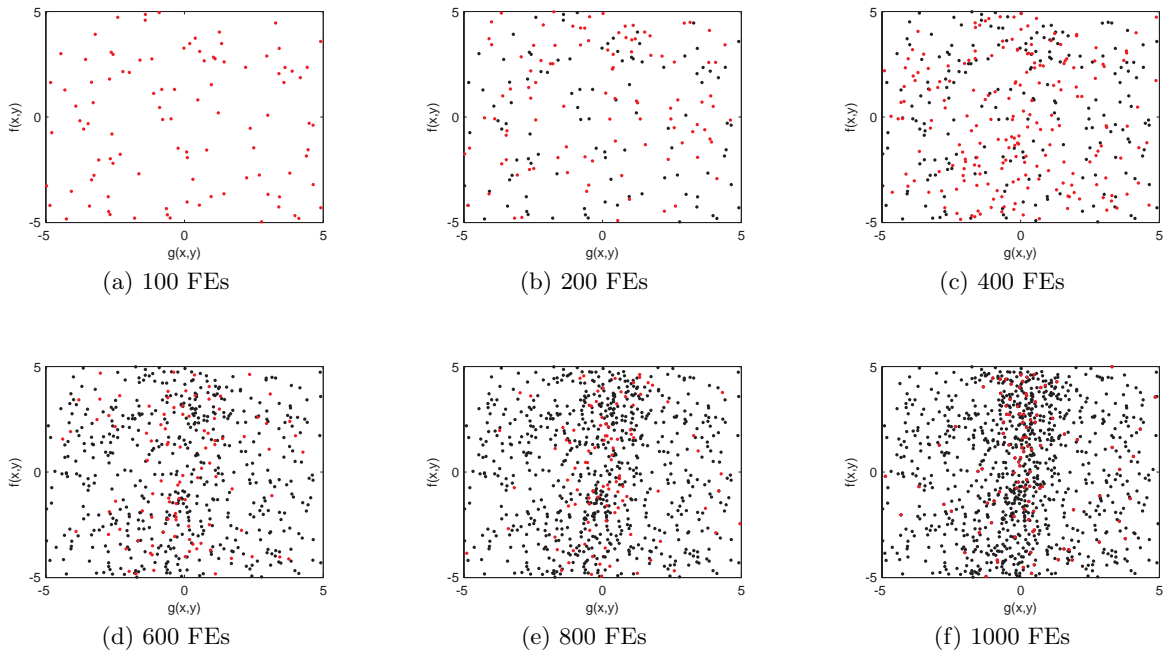


Figure 4: Time Series Plot of FEs for 2D Simple Example with  $NS=5$ ,  $NP=20$ ,  $CR=0.9$  and  $RANDOM?=Y$ . Most Recent Functional Evaluations Shown in Red.

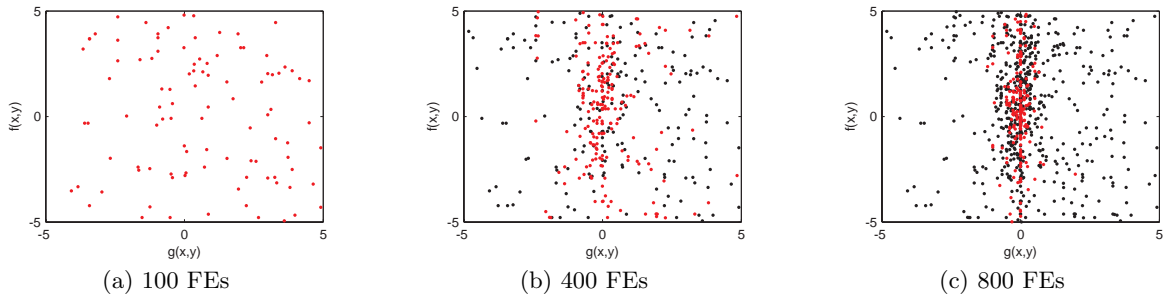


Figure 5: Time Series Plot of FEs for 2D Simple Example with  $NS=3$ ,  $NP=10$ ,  $CR=0.6$  and  $RANDOM?=N$ . Most Recent Functional Evaluations Shown in Red. This plot demonstrates the effect of a greedy selection strategy on sampling.

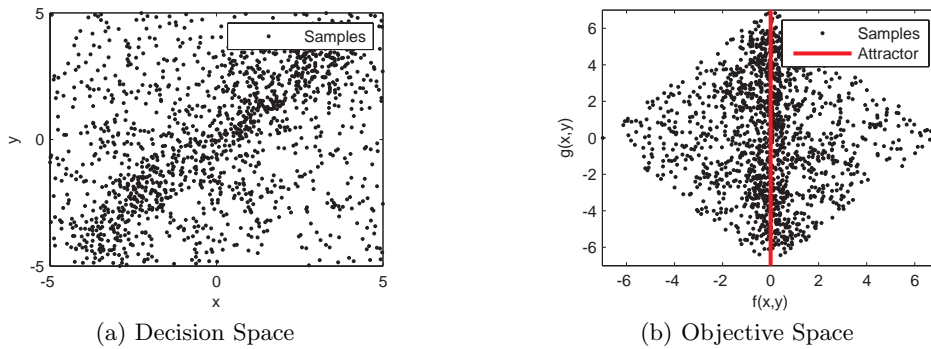


Figure 7: Trade Space of 2-D Crossed Example. This example shows the results of 2000 FEs on an example with  $NS=8$ ,  $NP=20$ ,  $CR=0.9$  and  $RANDOM?=Y$ .



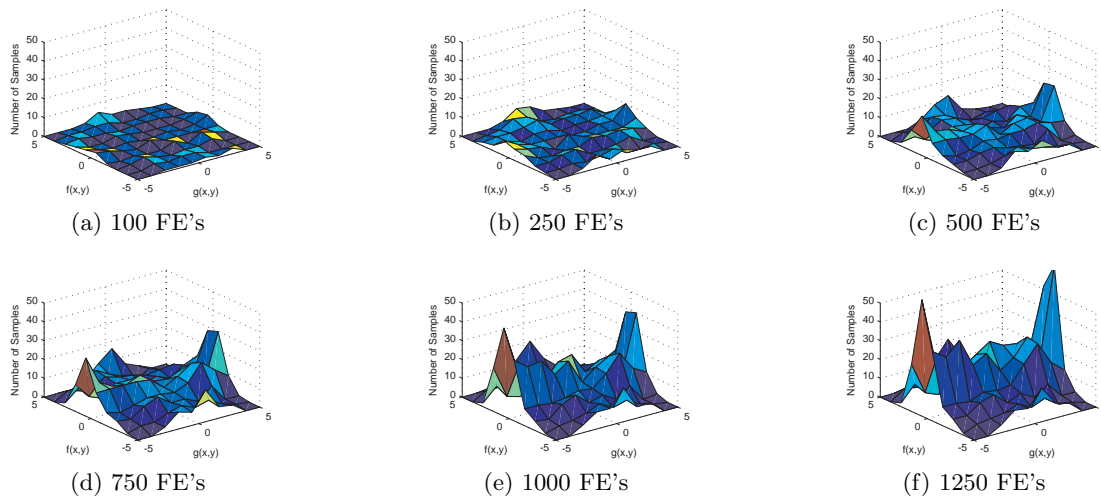


Figure 8: Time Series Plot for 2D Crossed Example. This example shows the results of 2000 FEs on an example with  $NS=8$ ,  $NP=20$ ,  $CR=0.9$  and  $RANDOM?=Y$ . The peaks that develop as the FEs increase correspond to stable species with decreasing radii. These peaks lower the kurtosis, making the distribution appear more uniform.

begin to develop  $\sim 750$  FEs begin to dominate  $\sim 1250$  FEs. This indicates that these two species have converged and no longer contribute to the exploration of the algorithm. The location and symmetry of these peaks lower the kurtosis which makes the overall distribution about  $g(x, y)$  to appear more uniform. However, by examining the standard deviation in tandem with the kurtosis it is clear that the distribution has a significantly higher spread than would be expected for a uniform distribution. Therefore, a DM must monitor each of the distribution measures discussed earlier in the paper to ensure sampling is as planned.

## 5. CONCLUSIONS

The ASDE algorithm proposed in this paper extends DE through modified speciation and crowding to thoroughly explore a trade space. This is in contrast to a traditional ES which seeks to converge to a single optimal solution (single objective) or Pareto set (multiobjective). In doing so, it provides information to the DM, allowing preferences to be formed *a posteriori* as apposed to forcing the DM to forming them *a priori*.

Two basic examples were demonstrated which provide insight into the potential of this method as an effective guided sampler. Additional research and refinement is needed to gauge ASDEs ability to transverse and map more complex multi-dimensional trade spaces.

For future work, a more thorough examination of parameter sensitivity is needed. In addition, an investigation into the real-time monitoring and control of the trade space distributions of interest via the distribution measures (i.e. kurtosis) is warranted. Finally, a strategy to monitor and address species pre-convergence is needed to limit the appearance of sharp peaks, such as were seen in Figure 8f.

## 6. ACKNOWLEDGMENTS

We would like to thank Dr. Patrick Reed for his guidance on Evolutionary Strategies, and Dr. Mike Yukish and Gary

Stump for their assistance on ATSV and the Design by Shopping Paradigm.

## 7. REFERENCES

- [1] R. Balling. Design by shopping: A new paradigm. In *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO-3)*, volume 1, pages 295–297, Buffalo, NY, 1999.
- [2] G. A. Hazelrigg. *Systems Engineering: an Approach to Information-Based Design*. Prentice Hall, Upper Saddle River, NJ, 1996.
- [3] X. Li. Efficient differential evolution using speciation for multimodal functional optimization. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Washington, DC, 2005.
- [4] K. Price. An introduction to differential evolution. In *New Ideas in Optimization*, pages 79–108. 1999.
- [5] K. Price, R. Storn, and J. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag Berlin Heidelberg, New York, 2005.
- [6] R. Storn. System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation*, 3(1):22–35, 1999.
- [7] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI, 1995.
- [8] A. Stuart and J. Ord. *Kendall's Advanced Theory of Statistics*, volume 1. Distribution Theory. Hodder Arnold, New York, 6 edition, 1994.
- [9] G. Stump, M. Yukish, T. Simpson, and E. Harris. Design space visualization and its application to a design by shopping paradigm. In *DETC 2003*, Chicago, Ill, 2003.
- [10] R. Thomsen. Multimodal optimization using crowding-based differential evolution. In *Proceedings of the 2004 Congress on Evolutionary Computation*, volume 2, pages 1382–1389, 2004.