Recombining Angles in Differential Evolution

Thomas G. Kristensen

Abstract—In this paper we wish to investigate how optimization problems involving angles can best be handled when using Differential Evolution (DE) as the optimization technique. Specifically we state the hypothesis that angles should not be recombined naively. To investigate this hypothesis we define two simple optimization problems involving angles and investigate our hypothesis on these by creating two angle recombination strategies for the DE algorithm. Our hope is that real world problems containing angles can benefit from this study, and we therefore test our hypothesis on a problem from the field of computational chemistry.

I. INTRODUCTION

Many real world optimization problems involve finding angles which optimize some objective function. Optimizing problems with angles may pose a problem when we try to optimize them with a general purpose heuristic that is not designed to handle these. In this paper we investigate if care should be taken when recombining angles in Differential Evolution, and what practitioners can do to handle these.

The paper is organized as follows. First we describe the DE algorithm which is the subject of our study. Next we describe more formally what our concerns are when asked to optimize a problem involving angles and how we will investigate if our concerns are justified. We then present our results along with a discussion and conclude with a short summary of our findings along with some ideas for further research in the area.

II. DIFFERENTIAL EVOLUTION

Storn and Price first proposed the Differential Evolution (DE) optimization algorithm in [5]. The algorithm maintains a population of $NP$ individuals $\Theta_1, \ldots, \Theta_{NP}$. In each iteration (generation) of the algorithm, each of the members of the population is selected as parent of a new offspring. This offspring is evaluated and compared to its parent and whichever is fittest survives for the next generation. The strategy is summarized in the pseudocode below.

DifferentialEvolution($NP$, $CR$, $F$)
1 Initialize $NP$ random individuals $\Theta^1, \ldots, \Theta^{NP}$
2 while termination criterion is not met
3 do for each individual $\Theta$
4 $\Theta \leftarrow$ CreateOffspring($\Theta^i$)
5 if $\Theta$ is better than $\Theta^i$, replace $\Theta^i$ with $\Theta$

In CreateOffspring we have chosen to use the DE/rand/1/bin [4] strategy as our starting point. There are several other strategies [4] but we have chosen one that we know is used in other studies [2]. We do not believe that our choice alters the results in our study as our observations should generalize to the other recombination strategies. The below code illustrates the DE/rand/1/bin strategy with a crossover rate $CR$ and a scaling factor $F$. The procedure RANDOM() returns a random number between 0 and 1.

CreateOffspring($\Theta^{(0)}$)
1 Randomly select parents $\Theta^{(1)}, \Theta^{(2)}$ and $\Theta^{(3)}$
2 Initialize empty offspring $\Theta$
3 Let $j$ be a random number between 1 and $n$.
4 for $k \leftarrow 1, \ldots, n$
5 do if RANDOM($k$) $\leq CR$ or $k = j$
6 then $\theta_k \leftarrow \theta_k^{(1)} + F \cdot (\theta_k^{(2)} - \theta_k^{(3)})$
7 else $\theta_k \leftarrow \theta_k^{(0)}$
8 return $\Theta$

III. ANGLES

In the following we will be studying two angles $\theta$ and $\theta'$. The angle difference between these is the length of the shortest arc between them on the unit circle and the numerical difference is the number $\theta - \theta'$. In this paper, a positive numerical difference is illustrated by a counterclockwise oriented arc as illustrated on Figure 1.

Our concern with CreateOffspring is that it adds the numerical difference between the parent $\theta^{(2)}$ and $\theta^{(3)}$ to the parent $\theta^{(1)}$. This procedure does not take into account the fact, that angles with a large numerical difference can actually have a small angle difference as can be seen on Figure 1.

Fig. 1. Two examples of angles with a small angle difference but with a large numerical difference.

The problem lies in the term $(\theta^{(2)}_k - \theta^{(3)}_k)$ from CreateOffspring. We therefore propose two alternatives for this term, which take the fact that we are manipulating angles into account. The first approach utilizes the fact that if two angles have an absolute numerical difference $|\theta - \theta'|$ that is larger than $\pi$ as seen on Figure 1, we can define two numbers with an absolute value less than $\pi$ by $\pm(2\pi - |\theta - \theta'|)$ as illustrated in Figure 2. Our first strategy RANDOMSIGNDIFFERENCE
picks one of these two numbers with equal probability.

\text{RandomSignDifference}(\theta', \theta'')
\begin{align*}
1 & \quad d \leftarrow |\theta - \theta'| \\
2 & \quad \text{if } d > \pi \\
3 & \quad \text{then } d \leftarrow 2\pi - d \\
4 & \quad \text{if } U_H(0, 1) < 0.5 \\
5 & \quad \text{then return } d \\
6 & \quad \text{else return } -d
\end{align*}

We can rewrite line 6 of \text{CreateOffspring} to

\[ \theta_k \leftarrow \theta_k^{(1)} + F \cdot \text{RandomSignDifference}(\theta_k^{(2)}, \theta_k^{(3)}) \]

In the \text{RandomSignDifference} approach, the sign of the difference between the two sets of angles (\Theta^{(2)} and \Theta^{(3)}) is calculated in a random matter across the set of angles. An interesting question is if this is a good strategy, or if we should stick to one strategy across all the pairs of angles (\theta_k^{(2)}, \theta_k^{(3)}) in the two sets.

Let us define the \text{ConsistentSignDifference} between two angles \theta and \theta' as the numerical shortest distance between \theta and \theta', from \theta' to \theta. As an example, examine the figures below.

\[ \theta - \theta' \quad \text{ConsistentSignDifference}(\theta, \theta') \]

To the left is the naïve difference between the two angles, which goes from \theta' to \theta in a long arc. When \theta - \theta' is numerically less than \pi, this is the \text{ConsistentSignDifference}, but in our example, this is not the case, and we find that the distance we are really interested in is \theta - \theta' + 2\pi, depicted to the right.

We can construct a procedure for calculating \text{ConsistentSignDifference} by considering the four cases sketched in Figure 3. The resulting procedure can be described by the following piece of pseudocode:

\text{ConsistentSignDifference}(\theta, \theta')
\begin{align*}
1 & \quad d \leftarrow \theta - \theta' \\
2 & \quad \text{if } \theta' < \theta \\
3 & \quad \text{then if } d \leq \pi \\
4 & \quad \quad \text{then return } d \\
5 & \quad \quad \text{else return } d - 2\pi \\
6 & \quad \text{else if } d \geq -\pi \\
7 & \quad \quad \text{then return } d \\
8 & \quad \quad \text{else return } d + 2\pi
\end{align*}

It is worth noticing that we assume that the angles \theta and \theta' are in the interval [0, 2\pi) which can easily be guaranteed by our DE implementation, by adding 2\pi to angles below zero, and by subtracting 2\pi from angles larger than 2\pi.

IV. THE FOLDING RULE PROBLEM

We have constructed two simple optimization problems to evaluate our strategies. The arguments for constructing simple evaluation problems instead of evaluating on real world problems are that the problems are easy to understand and that the optimal values are easily calculated and therefore can be compared against.

Our aim has been to construct evaluation problems that are well defined for any number of angles and which are not separable. The resulting problem is best described by an everyday object, namely a folding rule. If we unfold a rule completely, the result will be a line along the x-axis.

Each joint of the folding rule correspond to an angle. We can adjust these angles to create a conformation of the folding rule.

\[ \begin{array}{c}
\cdot \\
\cdot \\
\cdot
\end{array} \]

If we remove the wood from the folding rule but retain the metal joints, we are left with a set of points

\[ \cdot \quad \cdot \quad \cdot \]

The Folding Rule Problem is, given such a list of points, to identify a conformation of a folding rule that best covers these.

More formally, given a set of \( n + 1 \) target points \( T = \{t_0, \ldots, t_n\} \), we wish to identify a list of \( n \) angles \( \Theta = \{\theta_1, \ldots, \theta_n\} \) which minimizes

\[ \sum_{t \in T} \min_{p \in \Theta} (|t - p|) \]
where $P_0$ is the set of points obtained by folding up a virtual folding rule with unit length segments as illustrated on Figure 4 (a). This is done by fixing $p_0$ in $(0, 0)$ and choosing $p_i$ as

$$p_i = p_{i-1} + (\cos(\sum_{j=1}^{i} \theta_j), \sin(\sum_{j=1}^{i} \theta_j)).$$

In a related problem, we demand the point $t_i$ to be matched with $p_i$ and we therefore seek to minimize the quantity

$$\sum_{i=0}^{n} |t_i - p_i|$$

as illustrated in Figure 4 (b). This related problem we name the Restricted Folding Rule Problem.

Our constructed problems have two properties: (1) if the target points are generated by a folding rule we know the optimal fitness (zero) and (2) changing an angle $\theta_i$ changes the points $p_1, \ldots, p_n$. The first property makes it easier to set goals for our tests and the second property ensures that the variables in the problems are not separable.

V. EXPERIMENTAL SETUP

We have compared the original unmodified recombination strategy (ORIGINAL) with the two proposed alternatives, RANDOMSIGNDIFFERENCE and CONSISTENTSIGNDIFFERENCE on our two folding rule problems.

We have performed tests for a varying number of angles $n$; for each $n$ we have performed 1,000 runs where the target points are generated by a rule and with random populations of 50 members. The three strategies are used in our DE implementation with a scaling factor $F$ of 0.5 and a crossover rate of 0.75. Both the Folding Rule Problem and the Restricted Folding Rule Problem are used as benchmarks in the tests.

An optimization is terminated when the best individual reaches a fitness below $10^{-3}$ (success) or when the number of fitness evaluations exceeds 10,000 (failure). We record the AES (Average Evaluations pr. Success) and the percent of successful runs.

We have furthermore tested our three recombination strategies on a simple energy minimization problem from computational chemistry, namely identifying the angles that minimize the energy of the alanine dipeptide [1]. We have used the same parameters as the other experiments on the structure which contains six rotatable bonds. The energy of the structure was calculated using the MMFF94 force field in Open Babel 2.2.0 trough the Pybel ([3]) binding library and the goal value was set to the smallest value of 100,000 Monte Carlo samples. Each strategy was tested 2,000 times and each test was given 10,000 evaluations.

VI. RESULTS

For clarity we present our results as shaded matrices where the shading of an entry illustrates how many successful runs were obtained or how many evaluations where required for success.

In Figure 5 we have illustrated the percentage of successful runs on our two folding rule problems; an omitted box means that no successful runs where observed. CONSISTENTSIGNDIFFERENCE outperforms the two other strategies in all but two runs where it is beaten by one of the other strategies by exactly one point. The most impressive difference is where $n$ is 6 in the Restricted Folding Rule Problem where the CONSISTENTSIGNDIFFERENCE strategy has a success rate of 53% as opposed to the ORIGINAL strategy with only 22%.

When we look at the AES (Figure 6) we see that the CONSISTENTSIGNDIFFERENCE has a lower number of fitness evaluations than the other two methods. The improvement is, however, not as impressive as one would hope. Tables with the exact numbers can be obtained at

http://cs.au.dk/~tgk/.
The alanine dipeptide experiments are summarized in Table I and indicate that the altered recombination strategies have less impact on problems from computational chemistry. As can be seen from the table we actually perform more evaluations before reaching the desired goal when using the two altered strategies and this experiment is therefore contradictory to the previous, indicating that the original DE strategy is more robust across problems with more complex fitness landscapes.

### VII. Conclusion

To sum up, our experiments indicate that the choice of recombination strategy has a large effect on the percentage of successful runs in simple cases, but that it has little effect on the number of fitness evaluations in these. Our experiments furthermore illustrate that the choice of optimization problem can have a large impact on the observed difference between the recombination strategies and in some cases it can lead to contradictory conclusions. We can not confirm nor refute our hypothesis that angles should not be handled naively.

Our studies pose two interesting questions: (1) is testing new ideas on sandbox problems a valid evaluation method and (2) is over engineering functions to specific optimization scenarios a problem in the field. It is our hope that others in the field will investigate these two interesting questions.

### REFERENCES


