Exploitation of Sub-populations in Evolution Strategies for Improved Numerical Optimization

Brian D. Rigling
Department of Electrical Engineering
University of Michigan
Ann Arbor, MI 48105
brigling@eecs.umich.edu

Frank W. Moore
Systems Analysis Department
Miami University, Oxford, OH 45056
moorefw@muohio.edu
(513) 529-5929

ABSTRACT
This paper describes the use of a modified Differential Evolution strategy that identifies multiple solutions to the numerical optimization of multidimensional objective functions. Traditional approaches to this class of problems, such as Newton's Method, are restricted for use on continuous, differentiable functions; in addition, the solution identified by these approaches is often dependent upon the initial guess. The ability to find the multiple solutions is therefore restricted by one's ability to choose appropriate initial conditions. The Differential Evolution strategy described in this paper is not restricted by continuity and differentiability requirements, and can therefore robustly exploit the concept of sub-populations to converge to multiple solutions in a multi-dimensional problem space.

1 Introduction
Newton's method and other classic approaches to numerical optimization (Patrick 1972) may be applied to a limited class of multidimensional objective functions. This class of functions is characterized by properties such as stability, continuity, and differentiability. Functions not included in this class necessitate the use of a more robust approach (Nam 1994; Horn and Goldberg 1994; Shang 1997). When attempting to identify multiple solutions to a given objective function (Gilbert 1994), frequently one's only recourse is to repeat the selected numerical optimization technique under a wide variety of initial conditions. This approach is computationally inefficient because multiple initial guesses will frequently converge to the same solution. Furthermore, no guarantee exists that all solutions in the problem space will be found. Practical problems from science and engineering frequently benefit from the determination of as many optimal solutions as may be found in a specified region of interest. A methodology with the ability to force simultaneous convergence to multiple solutions is thus desired (Goldberg and Richardson 1987).

Evolutionary algorithms (Fogel 1995) use the principles of fitness-proportionate selection and reproduction to explore exponentially large problem spaces for sufficiently optimized solutions. These algorithms have demonstrated the capability to converge to optimized solutions under adverse conditions of instability (Greenwood 1997), discontinuity (Harp and Samad 1992), and lack of differentiability (Nacaskul 1997). In addition, recent investigations have produced evolutionary algorithms that exhibit remarkably good scalability with increasing problem complexity (Yao and Liu 1998). Evolution strategies (Rechenberg 1994; Back and Hoffmeister 1994) are evolutionary algorithms that manipulate vectors of real numbers; as such, evolution strategies have proven to be particularly useful for the task of optimizing functions of real variables (Back & Schwefel 1995).

Differential evolution (Price and Storn 1997) is a simple evolution strategy for solving difficult nonlinear numerical optimization problems. Differential evolution provides a fast and robust alternative to such traditional nondeterministic techniques as the simplex method (Nelder and Mead 1965) and the Levenberg-Marquardt algorithm (More 1978). The research described in this paper expands differential evolution, which normally converges to only one solution, by introducing the use of sub-populations to realize simultaneous convergence to multiple solutions.

2 Solution Approach
The basic algorithm for differential evolution is given in (Price and Storn 1997, p. 78). The primary extension to the original algorithm implemented during this research consists of introducing the characteristic of sub-populations (Davidor 1991; Punch 1998). A randomly seeded population is created as usual and evolves in the manner described by Price and Storn. However, there is one significant difference: each
population member identifies with one of \( N \) sub-
sub-populations and only mates within that sub-
subpopulation (Mahfoud 1995), where \( N \) is related
to the number of global and local optima. A
penalty (Myung and Kim 1998) is applied
individuals that are too close, in a Euclidean
distance sense, to members of different sub-
sub-populations. The resulting effect is to drive
different sub-populations to converge to different
solutions. Thereby, one may calculate an
undetermined number of solutions to a numerical
problem in a single run of the program. Note that
the number of sub-populations does not need
necessarily equate to the number of global and
local optima. In practice, the number of sub-
sub-populations should be less than or equal to the
number of global and local optima, as having too
many sub-populations will cause a confluence of
sub-populations about a single solution. Future
research may consider the possibility of
dynamically selecting the most appropriate
number of sub-populations for
a given multidimensional objective function.

The implementation of the sub-populations
is not as simple as assigning numbers to
population members and applying penalties
when appropriate. Some care must be taken on a
problem by problem basis to insure efficient
convergence. Each sub-population only
regenerates within itself; therefore, the size of
each sub-population must be sufficiently large to
provide genetic diversity. Second, the size of the
penalty applied must be adjusted correctly. A
very large penalty will keep sub-populations
strictly divided, which may restrict evolution,
depending on the size of the problem space. On
the other hand, an excessively small penalty may
have little or no effect upon the evolutionary
process. A middle ground must be reached which
will allow sub-populations to evolve separately
but freely. What constitutes an appropriate
penalty is completely dependent on the dynamic
range of the problem space – how high and low
function values may be – as well as how rapidly
the function varies. Techniques for automatically
identifying optimal dynamic penalty values for a
specific problem were not considered for this
research.

Another control input to the extended
differential evolution algorithm is the distance to
be maintained between sub-populations.
Selection of an appropriate value for this input is
a delicate matter that depends upon how rapidly
the function varies, as well as the estimated
proximity of one solution to another. One must
estimate how closely grouped solutions in the
problem space may be, such that a given sub-
subpopulation cannot converge to a solution and
mistakenly conceal another nearby solution
within its envelope. Automatic selection of
optimal distance values remains the subject of
future research.

3 Results

Two different functions of two independent
variables were designed to test the extended
differential evolution algorithm’s ability to
converge to multiple solutions in a timely
manner. Each of these functions is a nonlinear
numerical optimization problem that is difficult
to solve via traditional approaches. As shown in
Fig. 1, the first function consists of nine closely
grouped solutions. The surrounding contour of
this function (Eq. 1) is bi-cubic in nature,
providing rapid convergence to the general area
of the solutions. However, the close proximity
and equal weighting of these solutions makes
simultaneous convergence to all nine solutions
extremely difficult. In addition, since the
solution at \((2, 2)\) is surrounded by the other eight
solutions, it is more difficult to discover; an
initial guess converging from outside the general
solution area is more likely to converge to one of
the surrounding roots.

Applying Newton’s method to the function
described in Eq. 1, one hundred initial guesses
(shown in Fig. 2) were randomly generated and
allowed to converge to within a specified error
tolerance. As shown in Fig. 3, only eight of the
nine solutions were found, with the central
solution conspicuously absent.

Applying the extended differential evolution
algorithm described in this paper, one hundred
initial guesses were randomly generated and
randomly assigned to five equally sized sub-
sub-populations, as shown in Fig. 4. After 100
generations (Fig. 5), sub-populations have begun
to converge to many of the external solutions. By
generation 200 (Fig. 6), convergence to the
external solutions has dynamically changed the
objective function, via the protective envelope
extended by a sub-population, such that those
population members that have not yet converged
will behave as though those solutions do not
exist. Continuing the evolution through a total of
400 generations (Fig. 7) yields convergence to
all nine solutions to within several digits of
accuracy, which can be further improved through
successive generations.
\[ F(x, y) = |(x - 1) * (x - 2) * (x - 3)| + |(y - 1) * (y - 2) * (y - 3)| \quad (Eq. 1) \]

Figure 1. A multidimensional objective function with multiple global optima (Eq.1).

Figure 2. Random guesses used for Newton’s Method (Eq.1).
Figure 3. Newton’s Method fails to identify all zeroes in the solution space.

Figure 4. Random grouping of guesses into sub-populations for differential evolution (Eq.1).
Figure 5. Differential evolution with five sub-populations after 100 generations (Eq.1).

Figure 6. Differential evolution with five sub-populations after 200 generations (Eq.1).
After 400 Generations (40100 Total Evaluations)

Figure 7. Differential evolution with five sub-populations after 400 generations (Eq.1).

As shown in Fig. 8, the second objective function (Eq. 2) consists of four distantly separated global solutions and four closely spaced, centrally located local minima. The local minima serve as a distraction from the global optima. As a result, Newton's Method was incapable of identifying all of the globally optimal solutions to this problem. As with the first example, one hundred initial guesses (Fig. 9) were randomly generated and randomly assigned to five equally sized sub-populations. Figs. 10-11 trace the convergence to all four globally optimal solutions. One may also note a number of population members led astray by locally optimal (globally sub-optimal) solutions. This result leads to the conclusion that an increased number of local optima makes finding global optima a more demanding task and therefore requires additional sub-populations.

4 Discussion

This research introduced the use of sub-populations into the differential evolution algorithm for the purpose of simultaneously identifying multiple solutions to the numerical optimization of multidimensional objective functions. The examples described above demonstrate how the extended differential evolution algorithm can improve upon classical approaches such as Newton's Method. The concept of sub-populations providing multiple convergence is not restricted to application in the realm of differential evolution, evolution strategies, or genetic algorithms. Sub-populations may be useful in any application of genetic or evolutionary programming wherein the possibility of multiple solutions is of interest (Williams and Leggett 1982; Lorenz 1986; Rodl and Tovey 1987; Hel-Or, Rappaport, and Werman 1993). The major contribution of this research was to produce a modified differential evolution algorithm that provided a simple, efficient, and easily visualized solution to an important class of difficult numerical optimization problems.

Bibliography


\[ F(x, y) = \left( \alpha \sqrt{(x - 2.5)^2 + (y - 5.5)^2} \right) \times \left( \alpha \sqrt{(x - 5.5)^2 + (y - 2.5)^2} \right) \times \left( \alpha \sqrt{(x + 0.5)^2 + (y - 2.5)^2} \right) \times \left( 0.1 + \alpha \sqrt{(x - 1.5)^2 + (y - 1.5)^2} \right) \times \left( 0.1 + \alpha \sqrt{(x - 3.5)^2 + (y - 1.5)^2} \right) \times \left( 0.1 + \alpha \sqrt{(x - 1.5)^2 + (y - 3.5)^2} \right) \times \left( 0.1 + \alpha \sqrt{(x - 3.5)^2 + (y - 3.5)^2} \right) \]

(Eq. 2)

Figure 8. A multidimensional objective function with global and local optima (Eq.2).

Figure 9. Initial guess distribution for differential evolution with sub-populations (Eq.2).
Figure 10. Differential evolution with five sub-populations after 800 generations (Eq.2).

Figure 11. Differential evolution with five sub-populations after 1600 generations (Eq.2).