7
On Analogy with Some Other Algorithms

Nothing tempts a person as much as a cure for all problems. Nothing delights the scientist as much as a universal algorithm capable of efficiently solving any problem. We all well know that there is no such "remedy", but we are in continuous expectation of its appearance. Differential evolution seems to be a gleam of hope. In this chapter we shall establish an analogy between differential evolution and some other popular algorithms. It is obvious that DE can be easily compared with genetic algorithms and evolution strategies; I leave this task to the mercy of the reader. But here, we shall compare DE with nonlinear simplex, a very old and famous algorithm, and with two recent and efficient metaheuristics, particle swarm optimization and free search. From the outside, drawing an analogy will help us to disclose advantages and disadvantages of differential evolution. *Ex altera parte*, on the inside, I shall make an attempt to interpret the other algorithms through Differential Evolution.

Direct search methods, the methods we are speaking about in this book, firstly were proposed in the 1950s. The state of the art at that time was presented by Swann in 1972 [Swa72]. These methods, as you know, are used in one or more of the following cases.

1. Calculation of the objective function is time-consuming.
2. The gradient of the objective function does not exist or it cannot be calculated exactly.
3. Numerical approximation of the gradient is slow.
4. Values of the objective function are "noisy".

In order to disengage oneself from constraint-handling techniques and to devote one's attention to the algorithms themselves we shall consider only boundary constraints (2.15).

7.1 Nonlinear Simplex

The class of simplex direct search methods was introduced in 1962 [SHH62]. The most famous direct search method was suggested by Nelder and Mead in 1965 [NM65]. I shall briefly describe this method [Wri95].

Four operations, characterized by scalar parameters, are defined: *reflection* (ρ), *expansion* (χ), *contraction* (γ), and *shrinkage* (σ). In the original version of the algorithm these parameters should satisfy

$$\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1 \quad \text{and} \quad 0 < \sigma < 1\,. \tag{7.1}$$

The standard version of the algorithm assumes that

$$\rho = 1, \qquad \chi = 2, \qquad \gamma = 1/2 \quad \text{and} \quad \sigma = 1/2\,. \tag{7.2}$$

The Nelder–Mead algorithm is:

1. **Order.** Order the $D+1$ vertices in $\mathrm{I\!R}^D$ space to satisfy
$$f(x_1) \le f(x_2) \le \cdots \le f(x_{D+1})\,. \tag{7.3}$$
2. **Reflect.** Find the *reflection point* x_r from
$$x_r = \bar{x} + \rho \cdot (\bar{x} - x_{D+1})\,, \tag{7.4}$$
where $\bar{x} = \sum_{i=1}^{D} x_i/D$. Evaluate $f_r = f(x_r)$.
If $f_1 \le f_r < f_D$ then accept the reflected point x_r and terminate the iteration.
3. **Expand.** If $f_r < f_1$ then calculate the *expansion point* x_e,
$$x_e = \bar{x} + \chi \cdot (x_r - \bar{x})\,, \tag{7.5}$$
and evaluate $f_e = f(x_e)$. If $f_e < f_r$ then accept x_e and terminate the iteration, otherwise accept x_r and terminate the iteration.
4. **Contract.** If $f_r \ge f_D$ then perform a *contraction* between the better of x_{D+1} and x_r.
   (a) **Outside.** If $f_D \le f_r < f_{D+1}$ then *outside contraction*
$$x_c = \bar{x} + \gamma \cdot (x_r - \bar{x})\,, \tag{7.6}$$
   and evaluate $f_c = f(x_c)$. If $f_c \le f_r$ then accept x_c and terminate the iteration, otherwise perform a *shrink.*
   (b) **Inside.** If $f_r \ge f_{D+1}$ then perform an *inside contraction*
$$x_{cc} = \bar{x} - \gamma \cdot (\bar{x} - x_{D+1})\,, \tag{7.7}$$
   and evaluate $f_{cc} = f(x_{cc})$. If $f_{cc} < f_{D+1}$ then accept x_{cc} and terminate the iteration, otherwise perform a *shrink.*

5. **Shrink.** Evaluate f at the D points for the next iteration from

$$v_i = x_1 + \sigma \cdot (x_i - x_1), \qquad i = 2, \dots, D+1\,. \tag{7.8}$$

In spite of world-wide popularity this algorithm suffers from the following drawbacks [Tor89, Wri96].

- It fails when the simplex collapses into a subspace, or becomes extremely elongated and distorted in shape. In most of these cases, the objective function has highly elongated contours and a badly conditioned Hessian.
- It fails when its search direction becomes nearly orthogonal to the gradient.
- It is very sensitive to an increase of the problem dimension.

Now, let us examine a DE strategy from the RAND/DIR group (see Chapter 3). For the dimension D, at each iteration (generation g) we shall randomly extract $D+1$ individuals from the population $\mathbb{P}^g$. The worst individual of this subpopulation (x_{D+1}) belongs to the negative class C_-; the others (x_i, $i = 1, \dots, D$) form the positive class C_+. There is no average shift in this strategy. So, the strategy can be rewritten as

$$\omega = \bar{x} + F \cdot (\bar{x} - x_{D+1})\,, \tag{7.9}$$

where $\bar{x} = \sum_{i=1}^{D} x_i / D$.

It is obvious that such a strategy is identical to the *reflection* in the Nelder–Mead algorithm (7.4) taking into account that $F \equiv \rho$. Moreover, a wide range of differentiation constant values, $F \in (-1, 2+)$, can be easily interpreted as an *expansion*, $F \geq 2$, and an inside, $F \in (-1, -0)$, or outside, $F \in (+0, 1)$, *contraction.*

Unlike the logic of passing from one step to another in the Nelder–Mead algorithm, differential evolution immediately (or after crossover) selects the best individual among the target and the trial ones. The value of the differentiation constant is either controlled by an adaptation scheme or could be perturbed randomly. Also, the "simplex" is created randomly on the basis of the population for each individual per generation. Furthermore, there is no restriction on the number of used individuals.

I shall emphasize the following advantages/features of the DE approach in comparison with the Nelder–Mead simplex:

- *Search is performed in random subspaces.*
  1. The fact that in strategy (7.9) or, more generally (3.7), any number of individuals (usually $n < D$) can be used illustrates the creation of a simplex in subspaces of the search space.
     This makes the algorithm less sensitive to the problem dimension. On the other hand, such a strategy is more flexible in assigning a descent direction.

2. The subspace, generated from a population, better fits the optimal zones within each new generation.
   Thus, even if an inefficient simplex has appeared at the next step there is a great probability of constructing an efficient one. Often, a "bad" simplex executes the role of an explorer by testing unknown regions.
3. Introduced in (3.7), *average shift* allows us to correct a rough direction. In addition, hybridization with the best individual (RAND/BEST/DIR group) localizes, in some cases, the optimum more briskly.

- *Existence of many simultaneous optimizers.*
  DE could be perceived as a set of autocorrelated simplex optimizers. Such a distributed organization provides more thorough exploration, and improves convergence and precision of solution.

So, as you can see, DE overcomes all the drawbacks stated for the Nelder–Mead simplex. The DE strategy (7.9) of the RAND/DIR group inherits and develops the ideas underlying the simplex algorithm.

7.2 Particle Swarm Optimization

The first particle swarm optimization (PSO) method was proposed by J. Kennedy and R.C. Eberhart in 1995 [KE95]. It progressed simultaneously with DE and, at present, possesses one of the best performances among evolutionary algorithms, or even more widely, among metaheuristics. PSO issued from the metaphor of human sociality. It was born of attempts to simulate human cognition and to apply this model to a real optimization problem.

The idea is to use a set of individuals (a swarm of particles) for search space exploration. A particle represents a vector solution $x_i \in \mathbb{R}^D$, $i = 1, \dots, NP$ of an optimization task (the same notation as for DE is used here). At each iteration t the particle changes the position influenced by its velocity $v_i(t)$.

$$x_i(t) = x_i(t-1) + v_i(t)\,. \tag{7.10}$$

In order to update the velocity two rules are combined.

1. **Simple Nostalgia**
   In the view of psychology it realizes the tendency of an organism to repeat successful behaviors from the past or, in case of failure, to return to the last success. Let p_i be the best solution attained of the ith particle up to the present iteration, thus the velocity is updated in the following way.

$$v_i(t) = v_i(t-1) + \rho_1 \cdot (p_i - x_i(t-1))\,. \tag{7.11}$$

2. **Social Influence**
   In spite of an infinite number of possibilities to represent a social behavior two methods were distinguished for defining a neighborhood:

- *gbest* — considers the entire population as a neighborhood;
- *lbest* — defines the subpopulation surrounding the particle.

The *gbest* case is more practical and generally gives better results. Let p_g be the best solution of the population, so the mathematical expression of the social influence is $\rho_2 \cdot (p_g - x_i(t-1))$.
In the view of sociology this term represents the tendency of an organism to emulate the success of others.

To sum up, each of the particles is updated per iteration in the following way.

$$\begin{aligned} v_i(t) &= v_i(t-1) + \rho_1 \cdot (p_i - x_i(t-1)) + \rho_2 \cdot (p_g - x_i(t-1)) \\ x_i(t) &= x_i(t-1) + v_i(t)\,. \end{aligned} \tag{7.12}$$

The constants ρ_1 and ρ_2 are control parameters. They define which of two behaviors is dominant.

However, this algorithm in such a form has several drawbacks, leading mainly to premature convergence. In order to improve its performance some modifications were proposed:

- Limitation of $\rho_{1,2}$ up to 2 and its relaxation $\rho_{1,2} \cdot rand(0,1]$
- Limitation of the velocity $v_i \in [-V_{\max}, +V_{\max}]$, $V_{\max} = H - L$
- Introduction of the inertia weight w applied to the previous velocity $v_i(t-1)$, which understates the influence of the preceding behaviors on the current one

All this results in the next update formula:

$$\begin{aligned} v_i(t) = w(v_i(t-1)) &+ \rho_1 \cdot rand(0,1] \cdot (p_i - x_i(t-1)) \\ &+ \rho_2 \cdot rand(0,1] \cdot (p_g - x_i(t-1)) \\ v_i \in [-V_{\max}, +V_{\max}]\,; &\quad \rho_1, \rho_1 \in (0,2]\,. \end{aligned} \tag{7.13}$$

Let us compare PSO with DE now. The PSO strategy (7.13) consists of three components:

1. Damped history of previous velocities $w(v_i(t-1))$
2. Personal behavior $\rho_1 \cdot rand(0,1] \cdot (p_i - x_i(t-1))$
3. Social behavior $\rho_2 \cdot rand(0,1] \cdot (p_g - x_i(t-1))$

The history of previous velocities in terms of psychology characterizes the memory of an organism, and the damping mechanism realizes its property — forgetting. Memory always appears together with personal qualities. If the strategy is built only on personal behavior (second component), the algorithm will not work at all. Thus, memory induces positive effects for an individual movement of the particle, and, on the other hand, it retards its social reaction.

DE does not contain the first two aspects of PSO in pure form. Most likely the transversal technique (see Chapter 6) could illustrate personal behavior.

The individual makes a random walk in the search space and then chooses its optimal (best) position. Here, in DE, the walk is defined by the state of the population, whereas in PSO the next position (of the walk) is defined by the personal optimal position p_i and by the particle's velocity v_i. In PSO terms, DE (differentiation) is more social strategy. Nevertheless, I have made an attempt to introduce into DE the memory aspect in the form of damped preceding velocities (difference vectors). Various damping mechanisms, linear and nonlinear, were tested. As a result, such a modification induced only an inertia of search and showed decrease of convergence.

The third aspect of PSO (social behavior) can be interpreted in DE by the following strategy.

$$\omega = V_b + F^* \cdot (V_b - ind)\,, \qquad F^* = F \cdot rand(0,1]\,. \tag{7.14}$$

This strategy is an example of the RAND/BEST group (see Chapter 3, Equation (3.8)). We easily catch an identity between this strategy and the social aspect of PSO ($\omega = x_i(t),\ V_b = p_g,\ F = \rho_2 - 1,\ ind = x_i(t-1)$).

Finally, two complementarity features might be observed:

1. DE enlarges the social behavior by its groups of strategies.
2. PSO propagates the ideas of leadership on developing a local neighborhood.

It should be noticed that PSO uses the selection operation in an implicit form, whereas DE regards selection as a separate operation.

The success of DE may be explained by its *collective intelligence* behavior. If PSO exploits only two optimal positions (the particle's optimal position and leader's position), DE involves in evolution the positions and fitness of all the individuals of a population (population state). From a sociopsychological point of view, PSO represents the *conscious*, and DE the *unconscious* way of reasoning. It is well known that a human brain treats about only 10% of information consciously and 90% of information unconsciously (or modified states of consciousness). Perhaps, by imitating the human, the best of universal optimizers would be an alternation of PSO and DE with near to natural proportions.

7.3 Free Search

Free search (FS) is a very recent population-based optimizer. It was invented by K. Penev and G. Littlefair in 2003 [PL03]. Just as PSO emulates social cognition, FS is associated with an animal's behavior. FS partially imitates Ant Colony Optimization (ACO) adapted for continuous search [BP95]. Also, it includes: a PSO mechanism to refresh an animal's position, a DE strategy principle to create the animal's action and, also, the general structure of GA.

An animal in free search makes a journey, several exploration steps in the search space. Then, it moves at the best found position and marks it by a pheromone. The behavior of any animal is described by two aspects.

1. **Sense**
   Each of the animals has a sense to locate a pheromone. The more sensitive animal is able to find a better (promising) place for the search. The less sensitive one is forced to search around any marked position.
2. **Action**
   Each of the animals makes a decision of how to search; that is, it chooses its own neighborhood of search. So, the search journey of an animal may vary from local to global movements.

Both sense and action of an animal are random factors.

Let x_i be an animal of a population $\mathbb{P}$. The population consists of NP animals. The animals mark their positions by a pheromone $P_i \leq 1$:

$$P_i = f(x_i)/f_{\max} \,, \tag{7.15}$$

where $f_{\max} = \max_i\{f(x_i)\}$, $i = 1, \dots, NP$. Then, one endows each of the animals with the sense S_i:

$$S_i = P_{\min} + rand_i(0,1] \cdot (P_{\max} - P_{\min}) \,, \tag{7.16}$$

where $P_{\max}, P_{\min}$ are the maximum and the minimum pheromone values of a population.

At each generation the animal x_i begins its journey from any position x_k (from any animal of the population) satisfying its sense; that is,

$$x_k : \quad S_i \leq P_k, \, \forall k \in [1, \dots, NP] \,. \tag{7.17}$$

During the journey each animal performs T steps in the following way.

$$x_i^t = x_k + R \cdot (H - L) \cdot rand_t(0,1), \qquad t = 1, \dots, T \,. \tag{7.18}$$

$R \in [R_{\min}, R_{\max}] \subset \mathbb{R}^D$ is a randomly generated vector that defines a neighboring space. H, L are boundary constraints. The favors of step are considered as values of the objective function $f(x_i^t)$. The animal moves itself to the best found position

$$x_i \, : \qquad f(x_i) = \max_t\{f(x_i^t)\} \,.$$

When all animals perform their journeys, a new pheromone is distributed (7.15) and new senses are generated (7.16). Then, the population passes to the next generation.

This method has many random factors:

- Start position x_k
- Neighboring space R
- Steps of a journey x_i^t
- Generation of a sense S_i

In spite of the fact that the authors present such a randomness as a self-adaptation mechanism [Pen04, PL03], I suppose that it is exclusively a randomness. From my point of view self-adaptation is developed only in the restrictions on choosing a start position for a journey (7.17).

Free search can be confronted with transversal DE (Chapter 6). The introduction of a pheromone, in FS, has purely an ideological meaning. Without loss of generality the sense generation could be calculated directly from the minimal and the maximal values of an objective function. The next DE strategy will better coincide with the ideas underlying FS:

$$\omega = x + F^* \cdot (H - L), \qquad F^* = F \cdot rand(0,1)\,. \tag{7.19}$$

$F^* \subset \mathbb{R}^D$ is a relaxed vector of differentiation. x is a randomly extracted individual. This case presents a constant difference vector $\delta = H - L$. And, contrary to the usual DE, where extracted individuals form an exploration walk, here, only F^* moves the individual through the search space.

It is clear that there are two main disadvantages.

1. The information about the state of the population is not used; that is, the algorithm loses the perfect property of self-adaptation. In the common DE case, the difference vector is created on the basis of randomly extracted individuals that partially present the population state.
2. A random choice of the vector of differentiation produces many useless steps in the global neighborhood, and a local search is needed at the end of optimization.

However, free search introduces a new individual's feature (sense) that permits controlling the sensibility of an individual to the search space. Sense suppresses exploration of the search space, but at the same time, increases convergence by exploiting promising zones. It would be very attractive to join together the intelligence of a DE strategy and the sensitivity of a FS animal.

Problems

7.1. Formulate a definition of direct search methods. Enumerate the cases where these methods should be applied. Mention at least five direct search methods that you have already met.

7.2. What four operations of nonlinear simplex do you know? Explain and sketch in each of these operations.

7.3. Enumerate the drawbacks of the nonlinear simplex method.

7.4. Which of the strategies of differential evolution nearly completely interprets nonlinear simplex? Draw an analogy between these two algorithms.

7.5. Enumerate the advantages of differential evolution as against nonlinear simplex.

7.6. What the main idea does underlie particle swarm optimization?

7.7. Which of two appearances from psychology and sociology does particle swarm optimization reflect?

7.8. What role does the inertia weight w play in the PSO algorithm?

7.9. How is the effect of memory implemented in PSO and how is its property of forgetting implemented?

7.10. Is the memory is a positive or negative aspect of the algorithm?

7.11. Does differential evolution contain the elements of memory and personal behavior, likewise PSO? If yes, explain the difference of their implementation.

7.12. Which of the DE strategies in the best way interprets the social behaviour of PSO? Implement this strategy in your DE algorithm.

7.13. Add to problem (7.12) the implementation of the memory mechanism and the effect of personal behavior peculiar to PSO. Test the new algorithm and analyze the obtained results.

7.14. Write the algorithm that will alternate the strategy of PSO (7.13) from Chapter 7 with one of the DE strategies from Chapter 3. Experiment with alternation of one strategy with another. For the following test function,

$$
\begin{aligned}
&f(X) = \frac{1}{2} + \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 1/2}{(1 + 0.001(x_1^2 + x_2^2))^2}\,, \quad -100 \le x_1, x_2 \le 100 \\
&f(X^*) = 0\,, \quad X^* = 0\,,
\end{aligned}
$$

plot the graphs of convergence depending on the percentage of one or another strategy in the algorithm. Justify the results. Compare this algorithm with the classic one from Chapter 1.

7.15. Show with an explaining sketch two main aspects of the free search algorithm.

7.16. How does one calculate the pheromone of an individual?

7.17. How does one calculate the sense of an individual?

7.18. Give a flow-graph of the free search algorithm.

7.19. What random factors does Free Search have? In your opinion are these factors advantages or drawbacks?

7.20. Which of the DE strategies coincides in the best way with the idea underlying free search? What drawbacks of this strategy do you see?

7.21. Add to your algorithm the aspect of "sense" inherent in free search. Estimate the new algorithm using, at least, the following test function

$$\begin{array}{l} f(X) = (x_1^2 + x_2^2)^{0.25} \left(\sin^2\left(50(x_1^2 + x_2^2)^{0.1}\right) + 1\right) \\ -100 \le x_1, x_2 \le 100 \;, \quad f(X^*) = 0 \;, \quad X^* = 0 \;. \end{array}$$

7.22. Find (approximate) the probability density function for the difference vector of DE. Compare this function with other familiar probability density functions. Use the obtained function to automatically generate the difference vector independently of other individuals of the population. Compare the new algorithm with the classical one.