Metaheuristic algorithms.

Lab 6: Multiobjective optimization. Multimodal optimization

---

1. Multiobjective optimization

Multiobjective optimization means to simultaneously optimize several objective functions (criteria). The function to be optimized is vectorial, $F:R^n->R^r$, and its components can be denoted as follows $F=(f_1,f_2,\ldots,f_r)$.
The optimization criteria are usually conflicting, therefore the problem does not have a unique solution. In such a case we are looking for some trade-off solutions (called Pareto optimal) characterized by the fact that they cannot be improved with respect to all their components (any improvement with respect to one criterion leads to a decrease of quality with respect to other criteria).
There are different approaches to solve such a problem. The main approaches are:

- *Aggregation methods:* the multiobjective problem is transformed in a one-objective optimization problem by combining all optimization criteria in a single one. Thus the new objective function becomes: $f(x)=w_1f_1(x)+w_2f_2(x)+\ldots+w_rf_r(x)$ where $w_1,w_2,\ldots,w_r$ are weights associated to objective functions. For each set of weights one can obtain a different solution.
- *Direct approximation of the Pareto optimal set:* it uses a population of elements which will approximate the Pareto optimal set. The approximation process can be an evolutionary one. The main difference between multiobjective EAs and single objective EAs is mainly related to the selection process. In the MOEAs the selection process is based on the dominance relationship between the elements (see Lecture 10).

Examples of test functions used to evaluate the performance of multiobjective algorithms are available at: http://en.wikipedia.org/wiki/Test_functions_for_optimization or at http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/

Application 1. Let us consider the function $F:[0,4]->RxR$, $F(x)=((x-1)^2,(x-2)^2)$. Estimate the optimal Pareto set and the corresponding Pareto front.

Variant 1. By using the aggregation technique

a) Construct the aggregated objective function:

```
function y=fw(x)
    w=0.1;
    y1=(x-1)*(x-1);
    y2=(x-2)*(x-2);
    y=w*y1+(1-w)*y2;
endfunction
```

b) Apply an evolution strategy (for instance as implemented in SE.sci) or Particle Swarm Optimization, or Differential Evolution (see lab 5) to optimize the aggregated objective for the following values of w: (0.1,0.2,0.3,…,0.9). The corresponding results should be collected in a list.

c) Plot the points having as coordinates the values of the objective functions computed at the previous step (the plotted set of points will be illustrate an approximation of the Pareto front):

```
function pareto(x)
    f1=(x-1).^2;
    f2=(x-2).^2;
    plot(f1,f2,'*');
endfunction
```

The function pareto should be called for the list of solutions cosntructed at step (b).

Variant 2. Use the NSGA-II and MOGA algorithms implemented in SciLab (functions optim_nsga2 and optim_moga) to solve the same problem and plot the true and the approximated Pareto fronts.

Exercise. Compare the behavior of NSGA-II and MOGA for the test functions ZDT1 and ZDT3 described at
https://www.researchgate.net/profile/Kalyan_Deb/publication/3949503_Scalable_multi-objective_optimization_test_problems/links/02e7e51938c309279e000000/Scalable-multi-objective-optimization-test-problems.pdf?origin=publication_detail

Hint: exMOEA.sci

2. Multimodal optimization

Depending on their goal, the optimization problems can belong to one of the following categories:

- *Global optimization:* the aim is to find a configuration which minimizes or maximizes the value of the objective function (among all possible configurations in the search space).
- *Local optimization:* the aim is to find the best configuration in the neighborhood of a given (initial) element. A local optimum is better than the elements in its neighborhood but it could be worse than the global optimum.
- *Multimodal optimization:* the aim is to find all optima (both local and global); it is useful when there are several global optima and/or the local optima correspond to interesting configurations. The interest in identifying all optima might appear in engineering design (e.g. find all resonance points of a mechanical or electrical system).

The identification of all optima can be done in at least two ways:

- By iterating a local optimization algorithm (starting from different initial configurations)
- By using a single run of a population-based algorithm which enforce the population to discover several niches in the search space, each niche being related to an optimum.

The global optimization metaheuristics favor the populations which converge in a neighborhood of the optimum, such that at the end of the iterative process the population is characterized by a small diversity. In the case of multimodal optimization the strategy should be different, as the population should re-organize itself in species associated to the regions of all optima. Such a speciation could be achieved through explicit division of the population in sub-populations (such a strategy is not effective if the number of optima is unknown) or through implicit speciation (the

elements group themselves in species as an effect of some specific mechanisms). The most popular speciation mechanisms are:

- *Sharing*: the fitness of each element is penalized (by using a sharing function) if it belongs to a crowded region.
- *Crowding*: the main idea is that during the selection process an element competes only with elements which are in its neighborhood (e.g. with the closest element)

Example: Crowding DE (proposed by R. Thomsen „Multimodal optimization using crowding-based differential evolution", CEC 2004) is different from a standard DE (see lab 5) only with respect to the selection step. In the standard DE, each trial element (z_i) is compared only with the corresponding current element (x_i). In crowding DE the trial element replaces the element from the current population which is the closest and it replaces it if it is better.

Fig 1 illustrates the effect of this change in the DE selection.

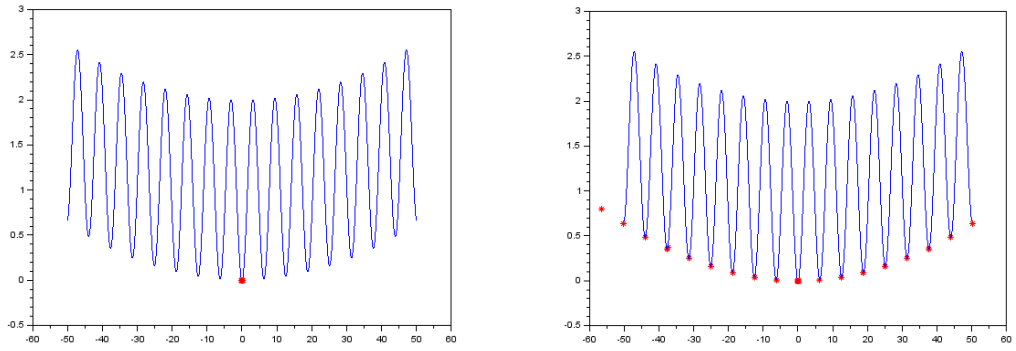

Fig. 1. Distribution of a population with 30 elements after 500 generations. Standard DE (left) and crowding DE (right)

Application 2. Modify the implementation of the DE algorithm (lab 5) in order to incorporate the crowding-based selection and test for one-dimensional and 2-dimensional multimodal test functions. Hint: see crowdingDE.sci