Metaheuristic algorithms.

Lab 6: Multiobjective optimization

Multiobjective optimization means to simultaneously optimize several objective functions (criteria). The function to be optimized is vectorial $F:R^n \to R^r$, 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 of this 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 a evolutionary one. The main difference between multiobjective EAs and single objective EAs is related with 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 $F:[0,4]\to RxR$, $F(x)=((x-1)^2,(x-2)^2)$. 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, that described 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 http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/

Hint: exMOEA.sci