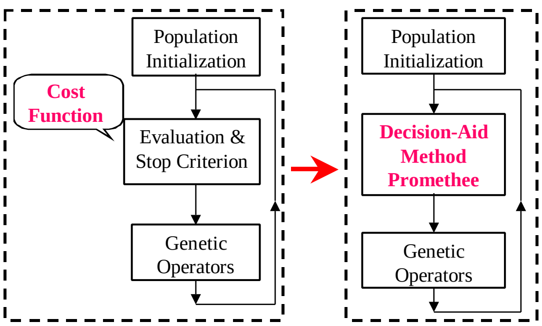
Multiple-Objectives Genetic Algorithm

Brahim Rekiek
Department of Applied Mechanics, CAD Unit, Brussels University (ULB)
Av. F. D. Roosevelt 50, CP 165/41, B-1050 Brussels, Belgium
brekiek@ulb.ac.be

Applying Genetic Algorithms (GA) to solve multiple objectives optimization problems has to deal with the twin issues of searching large and complex solution spaces and dealing with multiple, potentially conflicting objectives. Selection of a solution from a set of possible ones on the basis of several criteria is considered a difficult problem. Due to this difficulty, most of researchers reduce the problem to a mono-criterion one. Mathematical programming techniques and the popular weighted-sum approach have been developed. On the meta-heuristic side, Schaffer [4] was one of the first to recognize the possibility of exploiting Evolutionary Algorithm's to treat multiple-objectives problems.

Classical GA's use fitness-based selection, and thus require scalar fitness information. So the objectives are often artificially combined into a scalar function.



Classic and New Multi-Objectives GA Structure.

Other GA's use ranking methods to grade the population in terms of pareto dominance, without using the Decision Maker's (DM) preferences. Since the best solution may not necessarily belong to the pareto optimal set, the classical methods are a kind of local optimality search rather than a global one.

To come out of this kind of problems, we use the multi-criteria decision-aid (MCDA) method called *Promethee II* [2]. It computes a 'net flow' (*f*) associated with each solution. This flow gives us a ranking, called the Promethee II complete ranking, between the different solutions in the population.

The weights (associated with each criterion) are involved in the computation of the *f* number and represent the relative influence of each criterion. Thus the solutions are not compared according to a cost function yielding an absolute fitness of the individuals as in a classical GA, but are compared to each other thanks to flows, depending on the current population. In order to avoid a drift towards locally optimal solutions, elitism is used, i.e., the best-ever solution takes part in the evaluation of the *f* flows.

The choice of one solution over the others requires problem knowledge. It is the DM's task to adjust the weights to help the algorithm to find good solutions. Optimizing a combination of the objectives has the advantage of producing a single solution, requiring no further interaction with the DM. If this 'optimal' solution cannot be accepted, due to inappropriate settings of the weights, new runs may be required to adjust them until a suitable solution is found.

The basic steps of the classical GA are shown below.

Generate an initial population;
Evaluate fitness of individuals in the population;
repeat
  Select parents from the population;
  Recombine parents to produce children;
  Mutate children;
  Evaluate fitness of the children;
  Replace some or all of the population by children;
until *a satisfactory solution has been found;*

The new MOGA steps are the following:

Generate an initial population;
Order individuals in the population using Promethee II;
repeat
  Select parents
  Recombine best parents from the population;
  Mutate children;
  Use Promethee II to order the new population;
  Replace some or all of the population by children;
until *a satisfactory solution has been found;*

The method is integrated in the grouping genetic Algorithm (GGA) [1], and uses a group-oriented encoding. We apply it to the design of hybrid assembly lines, dealing with many objectives (cost, balance, reliability, congestion,..) [3].

References:

[1] Falkenauer E., 'Genetic Algorithms and Grouping Problems', John Wiley & Sons. First Edition, 1998.
[2] Brans J-P., Mareschal B., 'The PROMCALC & GAIA decision support system for multicriteria decision aid', *Decision Support Systems*, Vol. 12, pp. 297-310, North-Holland, 1994.
[3] Rekiek B., Pellichero F., De Lit P., Falkenauer E., and Delchambre A., 'A Resource Planner for Hybrid Assembly Lines'. Submitted to the ISPE99 Conference, Brazil, 1999.
[4] Schaffer J. D., 'Multiple objective optimization with vector evaluated genetic algorithms'. In Genetic Algorithms and their Applications, Proceedings of the First Int. Conference on Genetic Algorithms, pp. 93-100., 1985.