Nature-Inspired Computing

Handling Multiple Objectives

Dr. Şima Uyar
September 2006

Multi-objective Optimization

- MOP: Multi-objective Optimization Problem
- optimization problems with multiple, often conflicting objectives
- two part problem:
  - finding set of good solutions
  - choosing best solution for particular application

Multi-objective Optimization

- MOPs present uncountable set of solutions
- solutions produce vectors representing trade-offs in objective space
- decision maker chooses among these set of possible solutions

Multi-objective Optimization Problem

- a.k.a. multi-criteria optimization, multi-performance or vector optimization problem
- problem of finding a vector of decision variables which satisfies some constraints and optimizes a vector function whose elements represent the (usually conflicting) objective functions
  - acceptable solutions to decision maker

Multi-objective Optimization Problem

Find the vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ satisfying

m inequality constraints: $g_i(\vec{x}) \le 0 \quad i = 1,2,\ldots,m$

p equality constraints: $h_i(\vec{x}) \le 0 \quad i = 1,2,\ldots,p$

optimizing the vector function:

$$\vec{f}(\vec{x}) = [\, f_1(\vec{x}),\ f_2(\vec{x}), \ldots,\ f_k(\vec{x})]^T$$

Solution Classification

- a priori preference articulation:
  - (decide → search)
  - decision maker combines different objectives into a scalar cost function.
    - makes MOP into single objective prior to optimization
  - to find other solutions, to re-optimise with different w_i.

$$f'(x) = \sum_{i=1}^{n} w_i f_i(x)$$

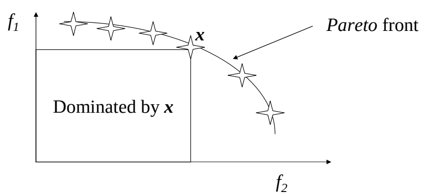
Solution Classification

- progressive preference articulation:
  - (decide ↔ search)
  - decision making and optimization interwined
  - partial preference information provides updated set of solutions for decision maker

Solution Classification

- a posteriori preference articulation:
  - (search → decide)
  - decision maker presented with set of Pareto optimal candidate solutions
  - decision maker chooses from set

Pareto Optimality



x dominates *y* if it is at least as good on all criteria and ***better*** on at least one

Pareto Optimality

A vector of decision variables $\vec{x}^* \in F$ is

Pareto optimal if there does not exist another

$\vec{x} \in F$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ *for all* $i = 1,..,k$

and $f_j(\vec{x}) < f_j(\vec{x}^*)$ *for at least one* j.

Pareto Optimality

- definition usually gives a set of solutions called the *Pareto optimal set*
- solutions in the Pareto optimal set are called *nondominated* solutions
- plot of objective functions with nondominated vectors in Pareto optimal set is called *Pareto front*

Advantages of NIC Aproaches

- population-based nature of search
  - can *simultaneously* find several Pareto optimal solutions in one run
  - traditional mathematical programming techniques can find Pareto optimum per run
- don't have to make guesses about which combinations of weights might be useful
- makes no assumptions about shape of Pareto front - can be convex / discontinuous etc
  - causes problems for most traditional mathematical programming techniques

Requirements of NIC Aproach

- way of assigning fitness,
  - usually based on dominance
- preservation of diverse set of points
  - similarities to multi-modal problems
- remembering all the nondominated points encountered
  - usually using elitism or an archive