Lesson 3:

Multi-Objective Evolutionary Algorithms

António Gaspar-Cunha

Institute for Polymers and Composites/I3N, Dept. of Polymer Engineering,

University of Minho, Guimarães, Portugal

http://www.dep.uminho.pt/agc/

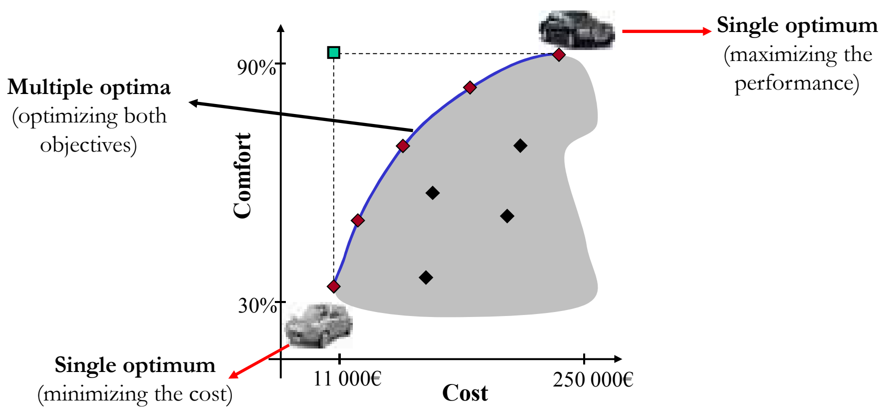
Universidad Carlos III de Madrid
January/February 2012

OUTLINE

- Concepts and Definitions
- Classical Methods
- EMO Methods
- State-of-the-art EMO
- NSGA-II
- RPSGA
- Application Example
- Niching and Speciation

Most real optimization problems are multi-objective

Example: Simultaneous minimization of the cost and maximization of the performance (comfort) when buying a car



Optimization Problem

A **partial order** is a binary relation "≤" over a set *P* which is reflexive, antisymmetric, and transitive, i.e., for all *a*, *b*, and *c* in *P*, we have that:

- $a \leq a$ (reflexivity);
- if $a \leq b$ and $b \leq a$ then $a = b$ (antisymmetry);
- if $a \leq b$ and $b \leq c$ then $a \leq c$ (transitivity).

In other words, a partial order is an antisymmetric preorder.

A partial order under which every pair of elements is comparable is called a **total order**.

CONCEPTS AND DEFINITIONS

Optimization Problem

Optimisation problem is an abstract problem where each instance is a pre-order $\preceq$ on the space of problem of solutions S.

An element of S is a solution of a problem instance if and only if it is a minimal element of S under the corresponding pre-order (alternatively, maximal).

For example, $x_1 \preceq x_2$ means that x_1 is at least as good as x_2

Assuming minimisation, the pre-order $\preceq$ can be defined as:

$$x_1 \preceq x_2 \Leftrightarrow f(x_1) \leq f(x_2), \forall x_1, x_2 \in S$$

CONCEPTS AND DEFINITIONS

Optimization Problem

Indifference of solutions - two solutions are indifferent if:

$$x_1 \preceq x_2 \wedge x_2 \preceq x_1$$

For *k* objective functions (as is the case of multi-objective optimization):

$$x_1 \preceq x_2 \Leftrightarrow f_i(x_1) \leq f_i(x_2), \forall i = 1,\ldots,k, \forall x_1, x_2 \in S$$

Incomparable solutions - if the following condition holds the solutions are incomparable: $x_1 \not\preceq x_2 \wedge x_2 \not\preceq x_1$

Thus, it is possible to say that the relation $\preceq$ is a partial pre-order.

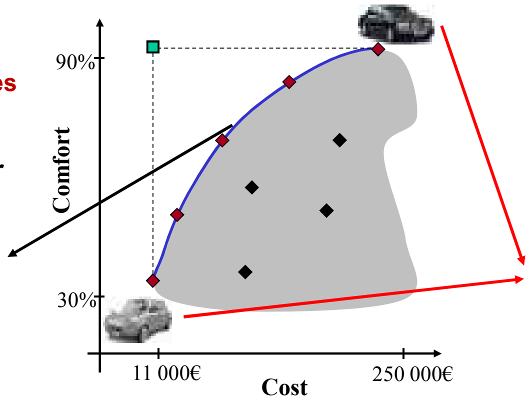
CONCEPTS AND DEFINITIONS

Single *versus* Multi-Objective

Multiple Objectives

Partial Pre-Order

Multiple Optima



Single Objective

Total Order

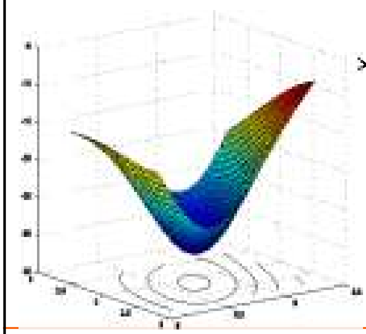
Single Optimum

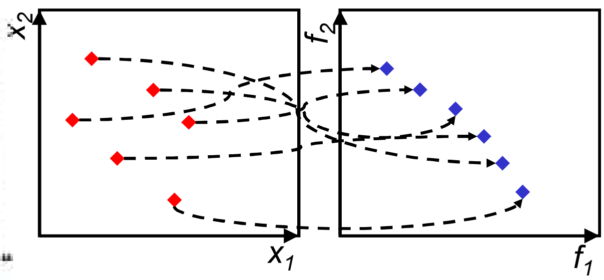
Single objective optimization is a particular case of multi-objective optimization (and not the opposite).

CONCEPTS AND DEFINITIONS

Single *versus* Multi-Objective

- Single optimum versus multiple optima
- Multi-objective requires both, search and decision-making
- There are two spaces of interest, instead of one



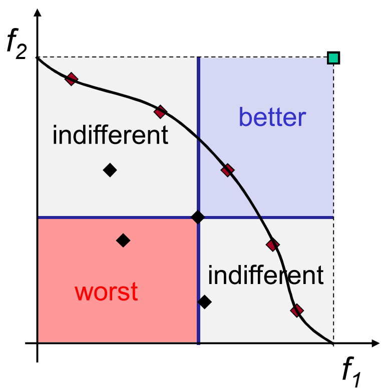


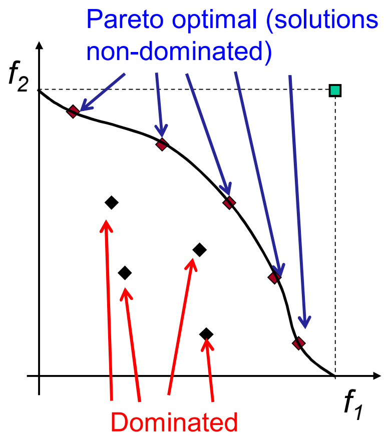
Dept. Polymer Engineering
University of Minho

CONCEPTS AND DEFINITIONS

Non-Dominance Concept

$$\text{maximize} \quad f_1(x_i), f_2(x_i)$$





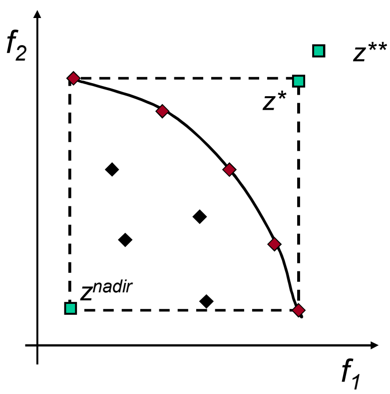
CONCEPTS AND DEFINITIONS

Non-Dominance Concept

z* - Ideal point: *does not exist, upper bound of Pareto-optimal set (maximization)*

z- Utopian point:** *also, does not exist*

z^{nad}- Nadir point: upper bound on Pareto-optimal set (maximization)



Multi-Objective Optimization Problem

$$\begin{array}{llll} \min\limits_{x_i} & f_l(x_i) & i=1,\cdots,n & l=1,\cdots,M \\ \text{subject to} & g_j(x_i)=0 & j=1,\cdots,J & \\ & h_k(x_i)\geq 0 & k=1,\cdots,K & \end{array}$$

where f_l are the M objective functions of the n parameters x_i, and g_j and h_k are the J equality ($J\geq0$) and K inequality ($K\geq0$) constraints, respectively

Non-Dominance Definition

Definition: A solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$ (or mathematically $x^{(1)} \preceq x^{(2)}$), if both conditions 1 and 2 are true:

1. **The solution $x^{(1)}$ is no worst than $x^{(2)}$ in all objectives, or $f_j(x^{(1)}) \ntriangleright f_j(x^{(2)})$ for all j = 1, 2, ..., M.**
2. **The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective, or $f_j(x^{(1)}) \triangleleft f_j(x^{(2)})$ for at least one $j \in \{1, 2, ..., M\}$.**

Definition (Non-dominated set): Among a set of solutions P, the non-dominated set of solutions P' are those that are not dominated by any member of the set P.

Minimization case

CONCEPTS AND DEFINITIONS

Non-Dominance Definition

Table : Relations on objective vectors and approximation sets. The relations $\prec$, $\prec\prec$, $\lhd$, and $\preceq$ are defined accordingly, e.g., $z^1 \prec z^2$ is equivalent to $z^2 \succ z^1$ and $A \lhd B$ is defined as $B \rhd A$.

relation	objective vectors	
strictly dominates	$z^1 \succ\succ z^2$	z^1 is better than z^2 in all objectives
dominates	$z^1 \succ z^2$	z^1 is not worse than z^2 in all objectives and better in at least one objective
better		
weakly dominates	$z^1 \succeq z^2$	z^1 is not worse than z^2 in all objectives
incomparable	$z^1 \parallel z^2$	neither z^1 weakly dominates z^2 nor z^2 weakly dominates z^1

Minimization case

CONCEPTS AND DEFINITIONS

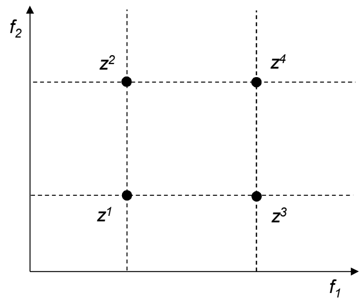
Non-Dominance Definition

relation	approximation sets	
strictly dominates	$A \succ\succ B$	every $z^2 \in B$ is strictly dominated by at least one $z^1 \in A$
dominates	$A \succ B$	every $z^2 \in B$ is dominated by at least one $z^1 \in A$
better	$A \rhd B$	every $z^2 \in B$ is weakly dominated by at least one $z^1 \in A$ and $A \neq B$
weakly dominates	$A \succeq B$	every $z^2 \in B$ is weakly dominated by at least one $z^1 \in A$
incomparable	$A \parallel B$	neither A weakly dominates B nor B weakly dominates A

Minimization case

CONCEPTS AND DEFINITIONS

Examples of dominance relations on objective vectors



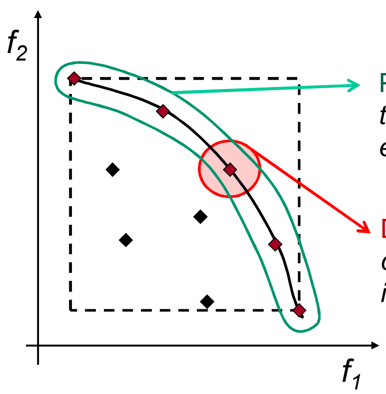
$$z^1 \succ z^2, z^1 \succ z^3, z^1 \succ z^4, z^2 \succ z^4, z^3 \succ z^4, z^1 \succ z^1,$$

$$z^1 \succ\succ z^4, z^1 \succeq z^3, z^1 \succeq z^4, z^2 \succeq z^2, z^3 \succeq z^3, z^3 \succeq z^4, z^4 \succeq z^4 \text{ and } z^2 \parallel z^3$$

Minimization case

CONCEPTS AND DEFINITIONS

Decision making



Pareto Optimality (*a set of optimal trade-offs, all objectives have equal importance*)

Decision Making (*choose the best compromise based on preference information*)

CONCEPTS AND DEFINITIONS

Decision making

- Decision making is the process of selecting a single solution among alternatives.
- Use additional preference information to “complete” the partial pre-order => choosing among incomparable solutions.
- The original problem formulation does not contain any type of preference information. Thus it is not known by the EA.
- This information must be provided by a Decision Maker:
  - **A priori**: decision making before the search (single objective);
  - **A posteriori**: decision making after the search (multiple solutions are found);
  - **Progressively**: decision making during the search;
  - Combination of the above.

CARACTERÍSTICAS

Historical Perspective (*only some examples*)

First algorithms	Schaffer (1985) – VEGA Kursawe (1990) –VOES
Classic algorithms	Fonseca and Fleming (1993) – MOGA Srinivas and Deb (1994) – NSGA Horn, Nafpliotis and Goldberg (1994) – NPGA
Elitist algorithms	Zitzler and Thiele (1999) – SPEA, (2001) – SPEA2 Deb and co-authors (2000) – NSGA-II Knowles and Corne (2000) – PAES, PESA

CARACTERÍSTICAS

Historical Perspective (*only some examples*)

Algorithms incorporating preferences concerning the solutions

- Fleisher (2003) – Simulating Annealing
- Zitzler and Künzli, (2004) – IBEA
- Emmerich et al. (2005) – SMS-EMOA
- Zitzler et al. (2008) – SPAM

CLASSICAL METHODS

Practical Resolution

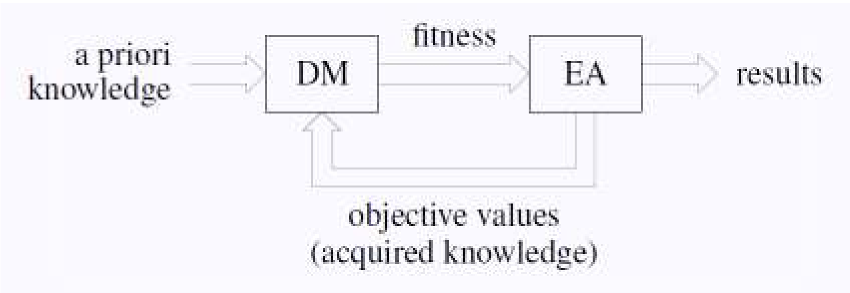
1- In a first step the EAs were used to find single solutions by:

- Using traditional aggregating function approaches
- A priori articulation of preferences

2- In a subsequent step EAs were used to get an approximation of the Pareto-optimal front as whole (MOEAs):

- All objective are optimized simultaneously
- EAs can use comparisons directly (including Pareto-dominance)
- Not considering preferences makes the problem formulation simpler
- A posteriori articulation of preferences
- Solve many problems in one optimisation run

Aggregation Methods



CLASSICAL METHODS

Aggregation Methods

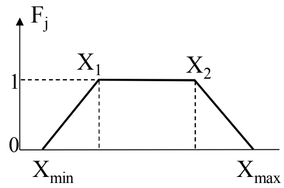
OBJECTIVE FUNCTION

$$FO_i = \sum_{j=1}^{q} w_j F_j \qquad \begin{cases} 0 \le w_j \le 1 \\ \sum w_j = 1 \\ 0 \le F_j \le 1 \\ 0 \le FO_i \le 1 \end{cases}$$

To maximize: $$F_j = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

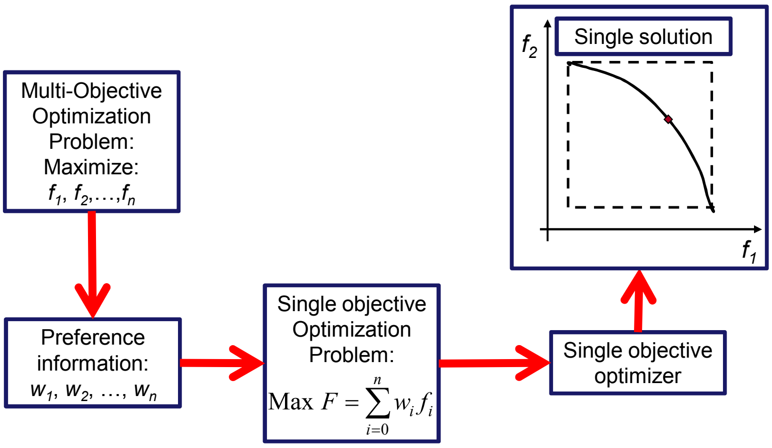
To minimize: $$F_j = 1 - \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (2)$$

Within range: $$F_j = \begin{cases} eq.1 \ \ if \ \ X_{min} \le X \le X_1 \ (X_{max} = X_1) \\ 1 \ \ if \ \ X_1 \le X \le X_2 \\ eq.2 \ \ if \ \ X_2 \le X \le X_{max} (X_{min} = X_2) \end{cases}$$



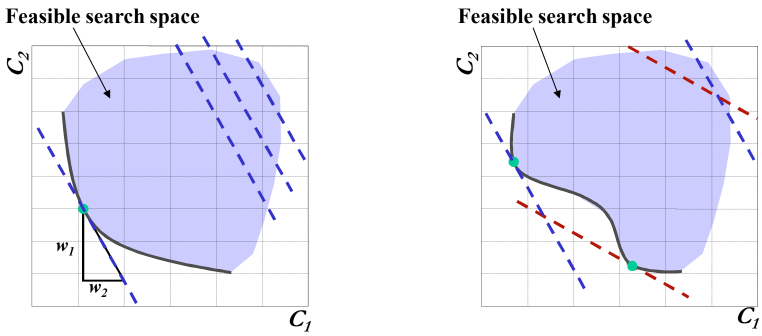
CLASSICAL METHODS

Aggregation Methods



CLASSICAL METHODS

Limitations of the Objective Function Approach



Advantages:

- Easy to implement
- Suitable for convex multiple optima surfaces

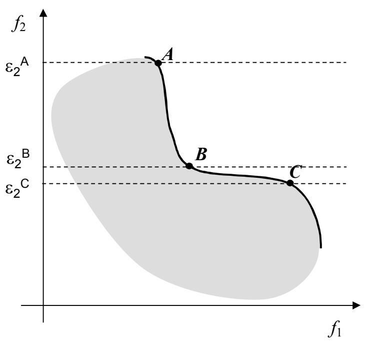
Disadvantages:

- Difficulties with mixed (simultaneous *maximization and minimization*) and non-convex problems
- Different weights can yield the same optimum
- When the relative weights change, a new optimization run needs to be carried out

CLASSICAL METHODS

ε-Constraint Method

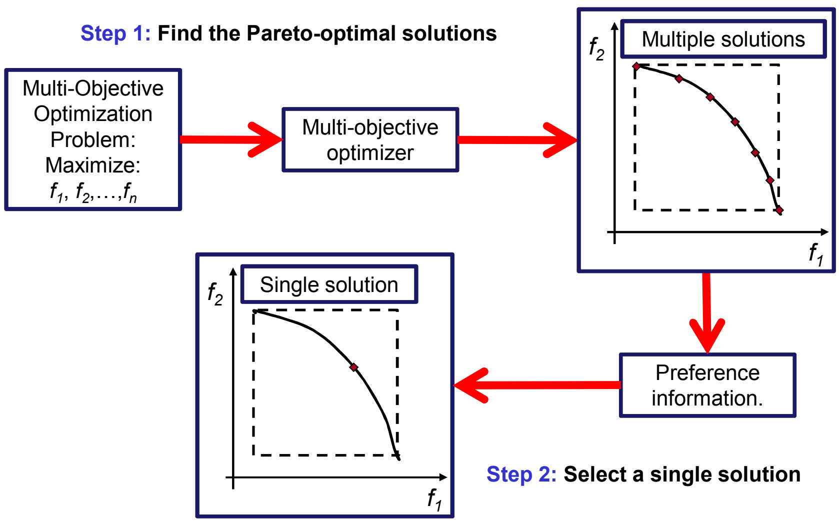
$$\text{Maximize } f_1(x),$$
$$\textit{subject to} \quad f_2(x) \geq \varepsilon$$



- Constrain all but one objective
- Need to know relevant **ε** vectors
- Non-uniformity in Pareto-optimal solutions
- However, any Pareto-optimal solution can be found with this approach

EMO METHODS

Multi-Objective Optimization



Dept. Polymer Engineering
University of Minho

Advantages and Disadvantages

Advantages of a posteriori approach:

- No interaction with the decision maker is needed
- The preference information is not required
- The decision maker may learn from the Pareto-front approximation

- Disadvantages:

- Computationally demanding
- No interaction with the decision maker is allowed
- The eventual existence of preference information is not used
- These algorithms are based on some implicit preferences
- In practice, most of the Pareto-front found may be irrelevant
- The approximation quality in the region of interest may be insufficient

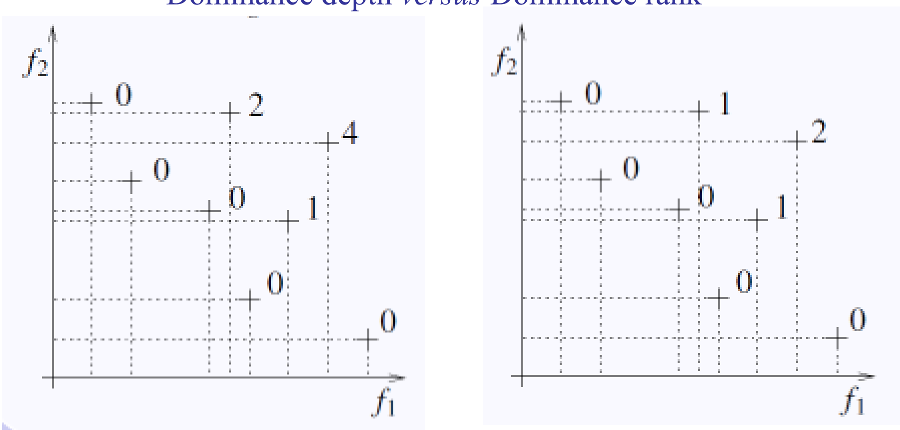
Preferences articulation

Pareto ranking:

- Non-dominated individuals are best (assign cost 0)
- Assigning cost to other individuals is more subjective
- Dominance rank vs. dominance depth (or non-dominated sorting) vs. dominance Strength

Preferences articulation

Dominance depth *versus* Dominance rank



Preferences articulation

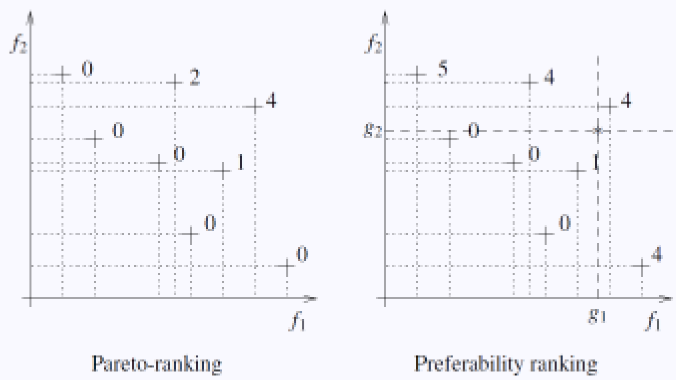
Some preference information is usually available :

- Weights
- Reference directions
- Goals
- Ideal values to be approached as much as possible
- Engineering specifications to be met or improved upon
- Priorities

EMO METHODS

Preferences articulation

Example of preferability-based ranking



EMO METHODS

Preferences articulation

1. All aggregations of the objectives:

- Weighted sum, minimax, . . .
- Goal programming variants (goal + reference direction)
- Other comparison operators
- Guided dominance (Branke et al. 2001), lexicographic order

2. Other MCDM techniques:

- ELECTRE (Benayoun, 1966), PROMETHEE (Brans and Vinke, 1985), GRIP (Figueira et al., 2009), etc.

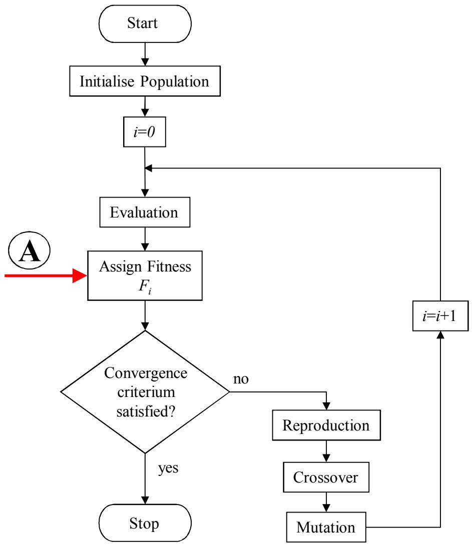
Challenge:

- Allow the DM to state its preferences easily

Dept. Polymer Engineering
University of Minho

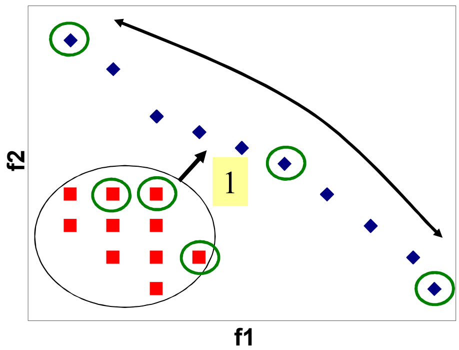
EMO METHODS

MULTIOBJECTIVE OPTIMISATION



EMO METHODS

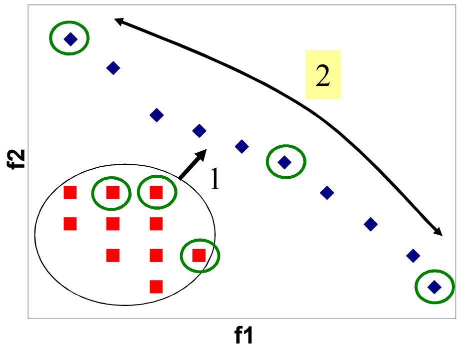
Basic functions of a MOEA:



1. Guiding the population towards the Pareto set: *Fitness assignment*

EMO METHODS

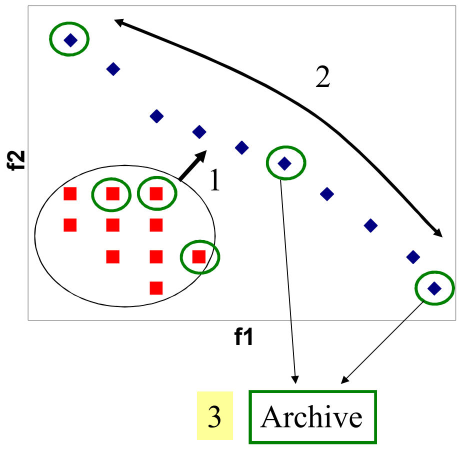
Basic functions of a MOEA:



1. Guiding the population towards the Pareto set:
   Fitness assignment
2. Maintaining a diverse nondominated set
   Density estimation

EMO METHODS

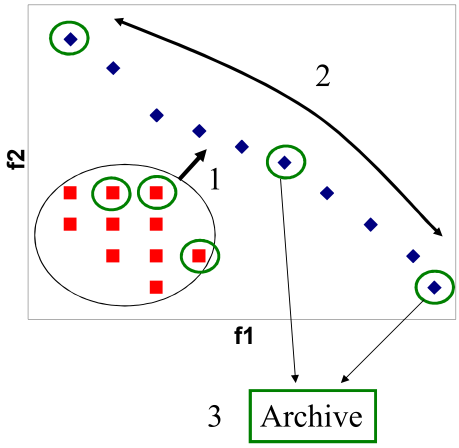
Basic functions of a MOEA:



1. Guiding the population towards the Pareto set:
   Fitness assignment
2. Maintaining a diverse nondominated set
   Density estimation
3. Preventing nondominated solutions from being lost
   Elitist population *(archiving)*

EMO METHODS

Basic functions of a MOEA:



1. Guiding the population towards the Pareto set:
   Fitness assignment

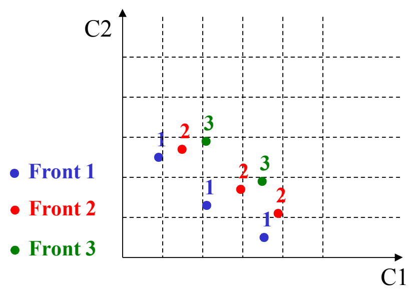
2. Maintaining a diverse nondominated set
   Density estimation

3. Preventing nondominated solutions from being lost
   Elitist population *(archiving)*

EMO METHODS

TRADITIONAL MOEAs:

- **Schaffer (1985) – VEGA**
- **Fonseca and Fleming (1993) – MOGA**
- **Horn, Nafpliotis and Goldberg (1994) – NPGA**
- **Srinivas and Deb (1995) – NSGA**



Step 1- *Ranking Function*

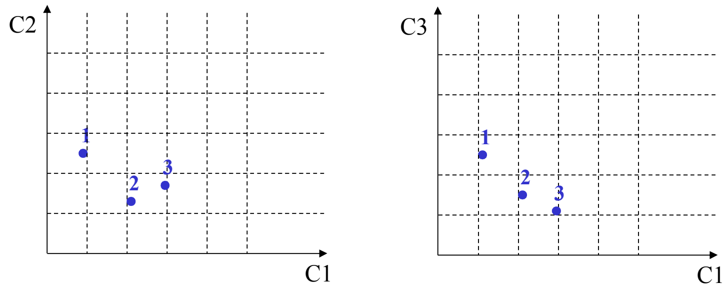
FO(Front 1)>*FO*(Front 2)>*FO*(Front 3)

Step 2- *Sharing*

$$FO_i' = \frac{FO_i}{m_i'}$$

EMO METHODS

Dificulty with the traditional MOEAs



If only criteria 1 and 2 are considered, point 3 is dominated.

If criterion 3 is also considered point 3 is non-dominated

EMO METHODS

Non-Elitist EMO Procedures

- Vector evaluated GA (VEGA) (Schaffer, 1984)
- Vector optimized EA (VOES) (Kursawe, 1990)
- Weight based GA (WBGA) (Hajela and Lin, 1993)
- Multiple objective GA (MOGA) (Fonseca and Fleming, 1993)
- Non-dominated sorting GA (NSGA) (Srinivas and Deb, 1994)
- Niched Pareto GA (NPGA) (Horn et al., 1994)
- Predator-prey ES (Laumanns et al., 1998)

Elitist EMO Methods

- Distance-based Pareto GA (DPGA) (Osyczka and Kundu, 1995)
- Thermodynamical GA (TDGA) (Kita et al., 1996)
- Strength Pareto EA (SPEA) (Zitzler and Thiele, 1998)
- Non-dominated sorting GA-II (NSGA-II) (Deb et al., 1999)
- Pareto-archived ES (PAES) (Knowles and Corne, 1999)
- Multi-objective Messy GA (MOMGA) (Veldhuizen and Lamont, 1999)
- Other methods: Pareto-converging GA, multi-objective micro-GA, elitist MOGA with co-evolutionary sharing

EMO METHODS

Nondominated Sorting Genetic Algorithm (*NSGA-II*):

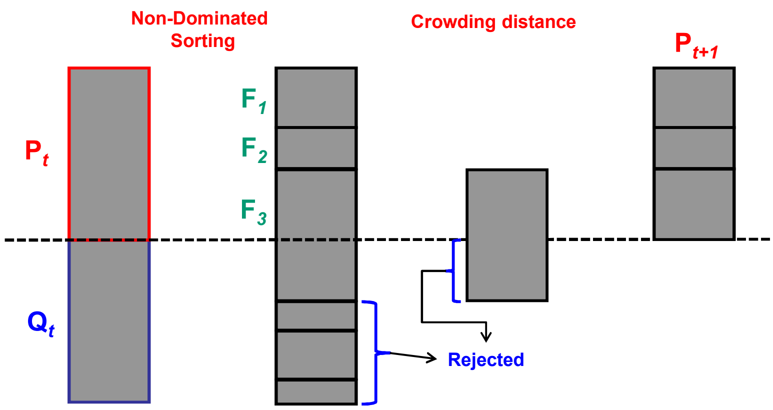
Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal and T. Meyarivan

IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 6, NO. 2, APRIL 2002

Dept. Polymer Engineering
University of Minho

NSGA - II

Nondominated Sorting Genetic Algorithm (NSGA-II)



Order of the NSGA -II: $O(N \log^{M-1} N)$

NSGA - II

Global Algorithm

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \texttt{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $\lvert P_{t+1} + \mathcal{F}_i \rvert \leq N$	until the parent population is filled
$\texttt{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding-distance in $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
$\text{Sort}(\mathcal{F}_i, \prec_n)$	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \lvert P_{t+1} \rvert)]$	choose the first $(N - \lvert P_{t+1} \rvert)$ elements of $\mathcal{F}_i$
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

NSGA - II

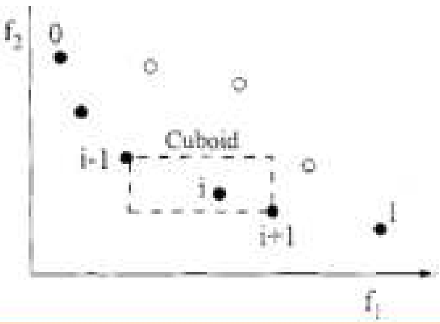
Fast non-dominated sort

fast-non-dominated-sort(P)
for each $p \in P$
  $S_p = \emptyset$
  $n_p = 0$
  for each $q \in P$
    if $(p \prec q)$ then — If p dominates q
      $S_p = S_p \cup \{q\}$ — Add q to the set of solutions dominated by p
    else if $(q \prec p)$ then
      $n_p = n_p + 1$ — Increment the domination counter of p
  if $n_p = 0$ then — p belongs to the first front
    $p_{\text{rank}} = 1$
    $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$
$i = 1$ — Initialize the front counter
while $\mathcal{F}_i \neq \emptyset$
  $Q = \emptyset$ — Used to store the members of the next front
  for each $p \in \mathcal{F}_i$
    for each $q \in S_p$
      $n_q = n_q - 1$
      if $n_q = 0$ then — q belongs to the next front
        $q_{\text{rank}} = i + 1$
        $Q = Q \cup \{q\}$
  $i = i + 1$
  $\mathcal{F}_i = Q$

NSGA - II

Crowding distance assignment

crowding-distance-assignment($\mathcal{I}$)
$l = |\mathcal{I}|$ — number of solutions in $\mathcal{I}$
for each i, set $\mathcal{I}[i]_{\text{distance}} = 0$ — initialize distance
for each objective m
  $\mathcal{I} = \text{sort}(\mathcal{I}, m)$ — sort using each objective value
  $\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$ — so that boundary points are always selected
  for $i = 2$ to $(l - 1)$ — for all other points
    $\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m)/(f_m^{\max} - f_m^{\min})$

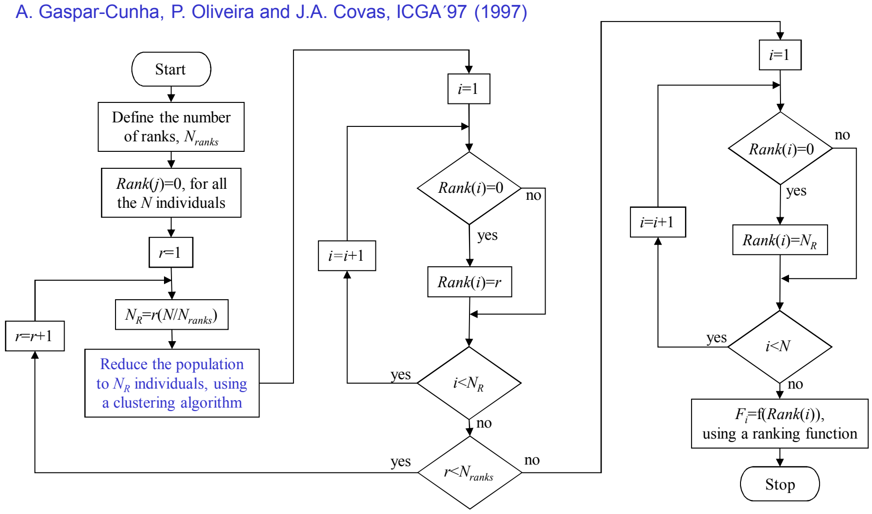


Reduced Pareto Set Genetic Algorithm (*RPSGA*):

A. Gaspar-Cunha, P. Oliveira and J.A. Covas, ICGA´97 (1997)

Reduced Pareto Set Genetic Algorithm (*RPSGA*):

A. Gaspar-Cunha, P. Oliveira and J.A. Covas, ICGA´97 (1997)

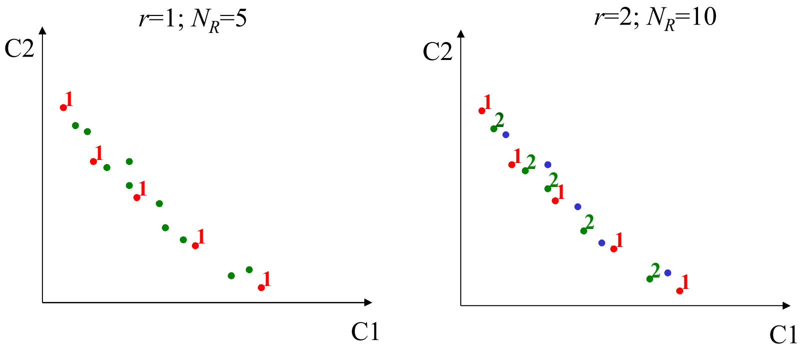


Dept. Polymer Engineering
University of Minho

RPSGA

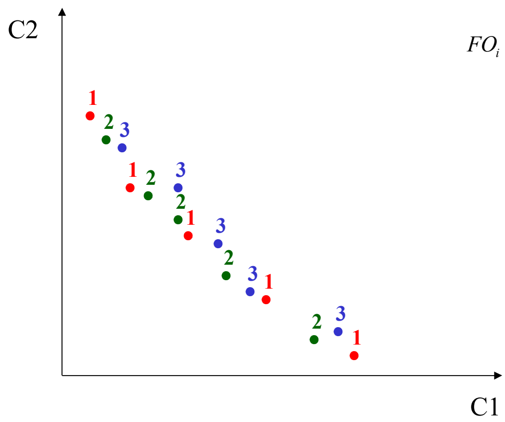
Reduced Pareto Set Genetic Algorithm (*RPSGA*):

N=15; N_{ranks}=3



RPSGA

Reduced Pareto Set Genetic Algorithm (*RPSGA*)



Ranking linear:

$$FO_i = 2 - SP + \frac{2(SP-1)(N+1-i)}{N}$$

$FO(1) = 2.00$
$FO(2) = 1.87$
$FO(3) = 1.73$

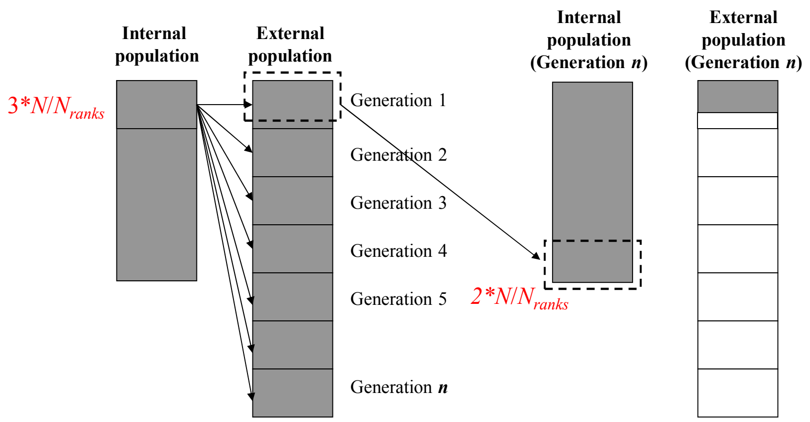
Rank functions

$$FO_i = 2 - SP + \frac{2(SP-1)(N+1-i)}{N}$$ Linear

$$FO_i = \frac{c-1}{c^N - 1} c^{N-i}$$ Exponential

where N is the number of population individuals, FOi is the objective function value for individual i, SP is the selection pressure ($1.0 < SP \leq 2.0$) and c is a constant that controls the selection pressure on the exponential ranking ($0<c<1$).

RPSGA with Elitism (*RPSGAe*)



Order of the RPSGAe: $O(N_{ranks}\, q\, N^2)$

RPSGA

How the basic functions are accomplished in the RPSGAe :

1. Guiding the population towards the Pareto set
   Fitness assignment: *ranking function based on the reduction of the Pareto Set*

2. Maintaining a diverse nondominated set
   Density estimation: *ranking function based on the reduction of the Pareto Set*

3. Preventing nondominated solutions from being lost
   Elitist population: *periodic copy of the best solutions (to the main population), selected with the method of Pareto set reduction*

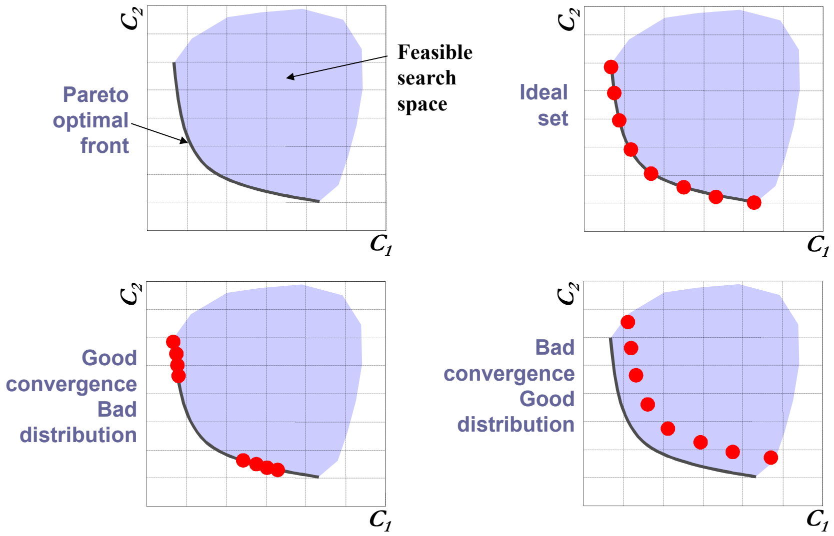
RPSGA

Algorithm Parameters:

- Number of ranks (N_{ranks}) [10;20;30,50,70]: **Strong effect**
- Density operator [Clustering, Sharing]: **Weak effect**
- Indifference limits of the clustering algorithm: [0.1,0.2;0.3;0.4;0.5]: **Weak effect**
- Type of ranking function (N) [linear; exponential]: **Weak effect**
- Size of the internal population (N) [100;200;300]: **Weak effect**
- Size of the external population (N_e) [100;200;300]: **Weak effect**

OPTIMIZAÇÃO MULTI-OBJECTIVO COM EAs – *Desempenho*

PERFORMANCE METRICS



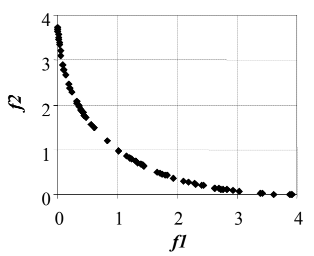
APPLICATION EXAMPLE

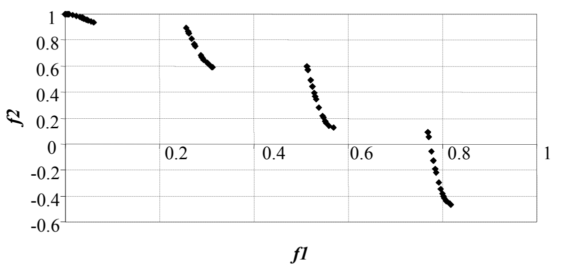
Reduced Pareto Set Genetic Algorithm (RPSGA)

Bi-objective problems to be minimized

$$f_1(x) = x^2$$
$$f_2(x) = (x-2)^2$$

$$f_1(x_1, x_2) = x_1$$
$$f_2(x_1, x_2) = (1+10x_2)\left(1-\left(\frac{x_1}{1+10x_2}\right)^2 - \frac{x_1}{1+10x_2}\sin(8\pi\, x_1)\right)$$





APPLICATION EXAMPLE

Test problems:

1- SPH-2; 2- SPH-3: $x_i \in [-10^3, 10^3]$

$$\text{minimise } f_j(x) = \sum_{i=1; i\neq j}^{m} x_i^2 + (x_j - 1)^2$$

3- ZDT6: $x_i \in [0, 1]$

$$\text{minimise } f_1(x) = 1 - \exp((-4x_1)\sin^6(6\pi\, x_1))$$

$$\text{minimise } f_2(x) = 1 + 9\left(\frac{\sum_{i=2}^{m} x_i}{m-1}\right)^{0.25}\left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$$

4- QV: $x_i \in [-5, 5]$

$$\text{minimise } f_1(x) = \left(\frac{1}{m}\sum_{i=1}^{m}\left(x_i^2 - 10\cos(2\pi\, x_i) + 10\right)\right)^{1/4}$$

$$\text{minimise } f_2(x) = \left(\frac{1}{m}\sum_{i=1}^{m}\left((x_i - 1.5)^2 - 10\cos(2\pi\,(x_i - 1.5)) + 10\right)\right)^{1/4}$$

Algorithm parameters:

- Number of variables (m): **100**
- Size of the internal population (N): **100**
- Size of the external population (N_e): **100**
- Number of ranks (N_{ranks}): **30**
- Indifference limits (*limit*): **20%**
- Number of evaluations of the objective function: **500,000**

Dept. Polymer Engineering
University of Minho

APPLICATION EXAMPLE

NEW MOEAs WITH ELITISM:

- Zitzler and Thiele (1999) – SPEA
- Deb and co-authors (2000) – NSGA-II
- Corne, Knowles and Oates (2000) – PESA
- Zitzler and Thiele (2001) – SPEA2

Dept. Polymer Engineering
University of Minho

APPLICATION EXAMPLE

Comparative study:

- Statistical comparison analysis
- 30 runs for each problem using different seed values

Results (*for each algorithm*):

1. Percentage of the Pareto frontier in which the algorithm is not beaten by the others
2. Percentage of the Pareto frontier in which the algorithm beats all the others

	PESA	NSGA-II	SPEA2	RPSGAe
SPH-2	[0; 0]	[0; 0]	[0; 0]	[100; 100]
SPH-3	[0; 0]	[0; 0]	[0; 0]	[100; 100]
ZDT6	[0; 0]	[43.5; 0.1]	[43.3; 0]	[56.6; 56.5]
QV	[35.3; 31.5]	[29.1; 7.3]	[61.2; 35.9]	[0; 0]

Zitzler, E., WEB page http://www.tik.ee.ethz.ch/~zitzler/testdata.html

APPLICATION EXAMPLE

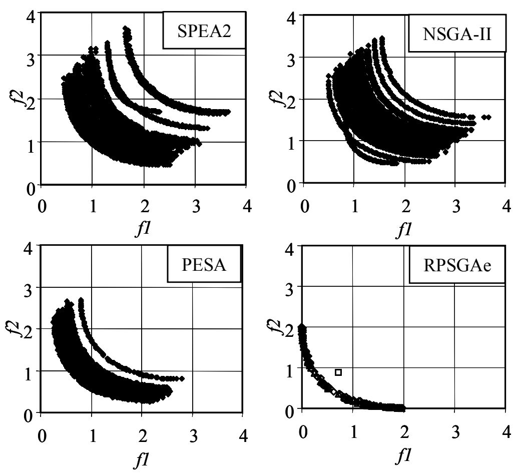
RPSGAe – Reduced Pareto Set Genetic Algorithm with Elitism

1- SPH-2

$$f_1(x) = \sum_{i=1, i\neq 1}^{n} x_i^2 + (x_1 - 1)^2$$

$$f_2(x) = \sum_{i=1, i\neq 2}^{n} x_i^2 + (x_2 - 1)^2$$

$n = 100$
$x_i \in [-1000 , 1000]$



NICHING AND SPECIATION

Niching and speciation

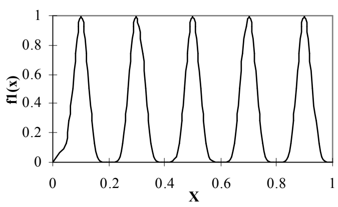
Multimodal functions

NICHING AND SPECIATION

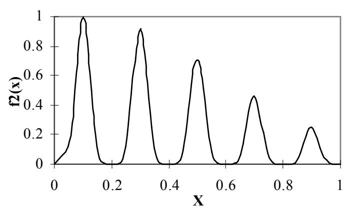
Niching and speciation
Multimodal functions

In most optimisation problems it is necessary not only to find the overall optimum, but also to identify the various local optima. Conventional GAs are not able to do this, since they converge necessarily to one point of the overall search space [GOL 89a].

$$f_1(x) = \sin^6(5\pi x)$$



$$f_2(x) = e^{-2\ln 2\left(\frac{x-0.1}{0.8}\right)^2}\sin^6(5\pi x)$$



NICHING AND SPECIATION

If GAs are applied repeatedly to determine of the maximum of the first function, they converge indifferently to any single peak. This happens because the population cannot have an infinite dimension, as assumed by the schemata theory. The problem is designated by “genetic drift” and can cause the accumulation of errors as the search proceeds [GOL 87]. However, convergence to a single peak is not desirable in functions with various similar maxima. Generally, when the search space has local maxima with different values, it will be interesting that convergence occurs to the peak with the greatest value, but also that a determined number of individuals converge for each individual peak. This is particularly important in the case of complex functions, as it provides the characterisation of their topography.

NICHING AND SPECIATION

In order to deal with the above, the concepts of **niching** and **speciation** of natural evolution should be introduced in a population of chromosomes. This is based on the idea of forming stable populations of organisms by creating separated niches where they are forced to **share** the available resources [GOL 87, DEB 89].

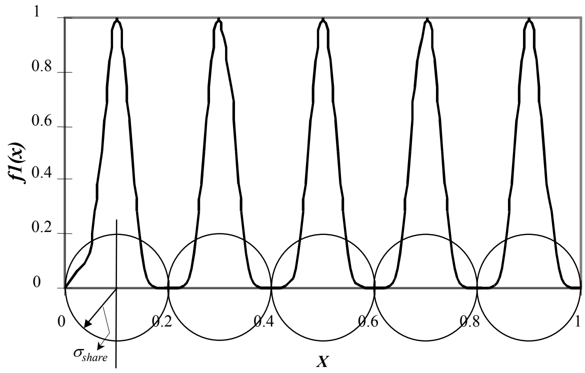
$$Sh(d_{ij}) = \begin{cases} 1-\left(\dfrac{d_{ij}}{\sigma_{share}}\right)^{\alpha} & ,if\ d_{ij} < \sigma_{share} \\ 0 & ,otherwise \end{cases}$$

sharing function, $Sh(d_{ij})$, where α is a constant and σ_{share} is the radius of a circumference defining the maximum distance between chromosomes, in order to form as many niches as the number of peaks on the search space.

NICHING AND SPECIATION

1. $0 \leq Sh(d_{ij}) \leq 1 \quad , \forall \; d_{ij}$
2. $Sh(0) = 1$
3. $\lim_{d_{ij} \to \infty} Sh(d_{ij}) = 0$

This figure illustrates this concept. The function $f_1(x)$ has 5 maxima in the search space ($0 \leq x \leq 1$). The niche size is given by the ratio between the size of the search space and the number of peaks, i.e., 1/5=0.2; consequently, σ_{share} is 0.1.



NICHING AND SPECIATION

Since the basic idea of sharing is that the objective function of an individual diminishes in the presence of its neighbours, the final objective function value (FO'_i) will result from the ratio between the initial evaluation (FO_i) and its niche count (m'_i).

$$FO'_i = \frac{FO_i}{m'_i} \qquad\qquad m'_i = \sum_{j=1}^{N} sh(d_{ij})$$

where m'_i is the sum of all the sharing functions related to this individual.

The sharing function with himself [$sh(d_{ii})$=1] will be also included.

NICHING AND SPECIATION

The distance between individuals can be determined in the real parameter space, (**phenotypic sharing**) or in the codified space (**genotype sharing**). The former will be adopted, since it has physical meaning and greater performance (according to Deb [DEB 89]). Two individuals (X_i e X_j) on a p dimensional space can be defined as:

$$X_i = \left[x_{1,i}; x_{2,i}; \ldots; x_{p,i}\right]$$

$$X_j = \left[x_{1,j}; x_{2,j}; \ldots; x_{p,j}\right]$$

$$d_{ij} = \sqrt{\sum_{k=1}^{p}\left(x_{k,i} - x_{k,j}\right)^2}$$

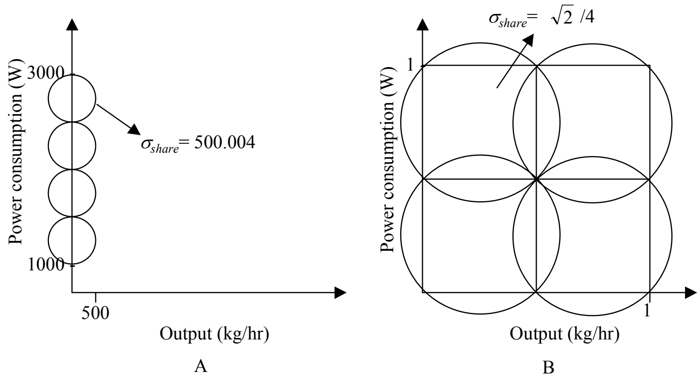
The distance between them (d_{ij}) can be defined using the norm on a p dimensional space, using the Euclidean distance.

NICHING AND SPECIATION

- Since in most optimisation problems the number of peaks is unknown, the use of the above methodology to define the value of σ_{share}, implies the use of a trial and error procedure.
- Sharing can be indistinctly applied in the space of the variables to optimise, or in the criteria space, depending on which niching is necessary. Generally it is applied in the criteria space, where the choice of the solution is carried out and where diversity along the Pareto frontier is required.

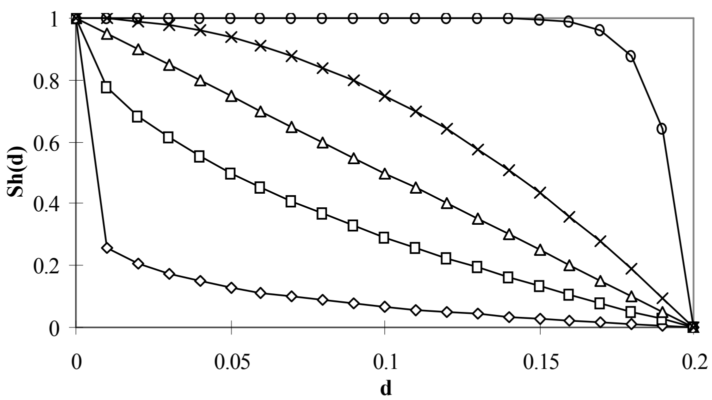
NICHING AND SPECIATION

During sharing process, scaling of all space parameters in the same interval (for example between 0 and 1) is indispensable in order to avoid the comparison between values that can be very different.



NICHING AND SPECIATION

As σ_{share}, the α parameter in the corresponding equation controls the radial size of the niche "radius", as shown in the following Figure:



NICHING AND SPECIATION

The variation of the degree of the Holder metric used also changes the shape of the niche

$$d_{ij} = \left(\sum_{k=1}^{p} \left| x_{k,i} - x_{k,j} \right|^{s} \right)^{1/s}$$