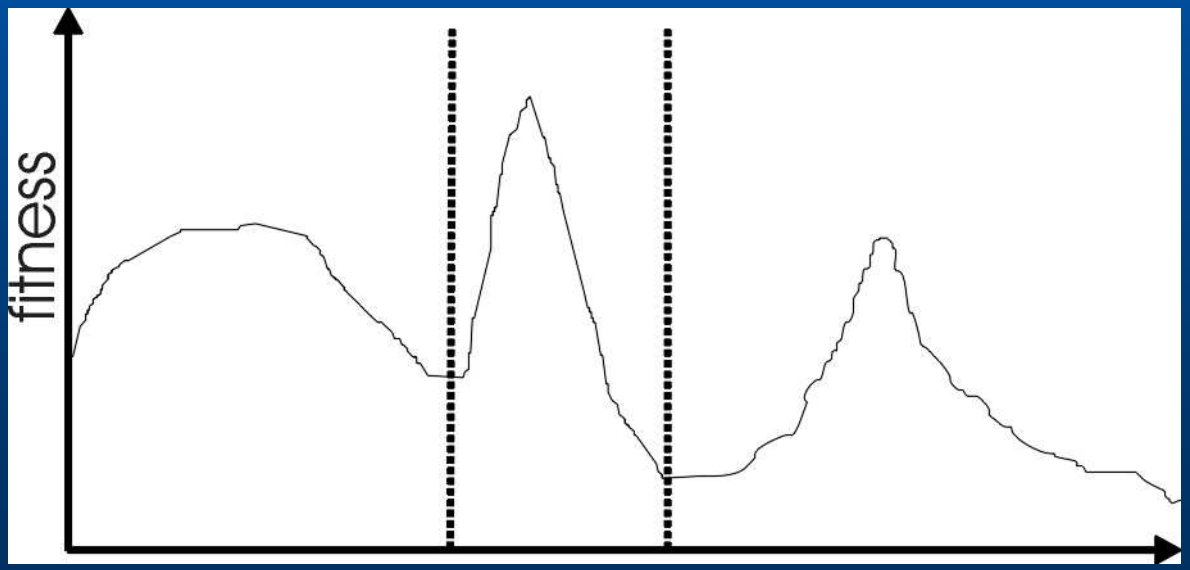
Multi-objective & Multi-modal Optimization

Multimodal Problems

Motivation 1: Multimodality

Most interesting problems have more than one loc ally optimal solution.

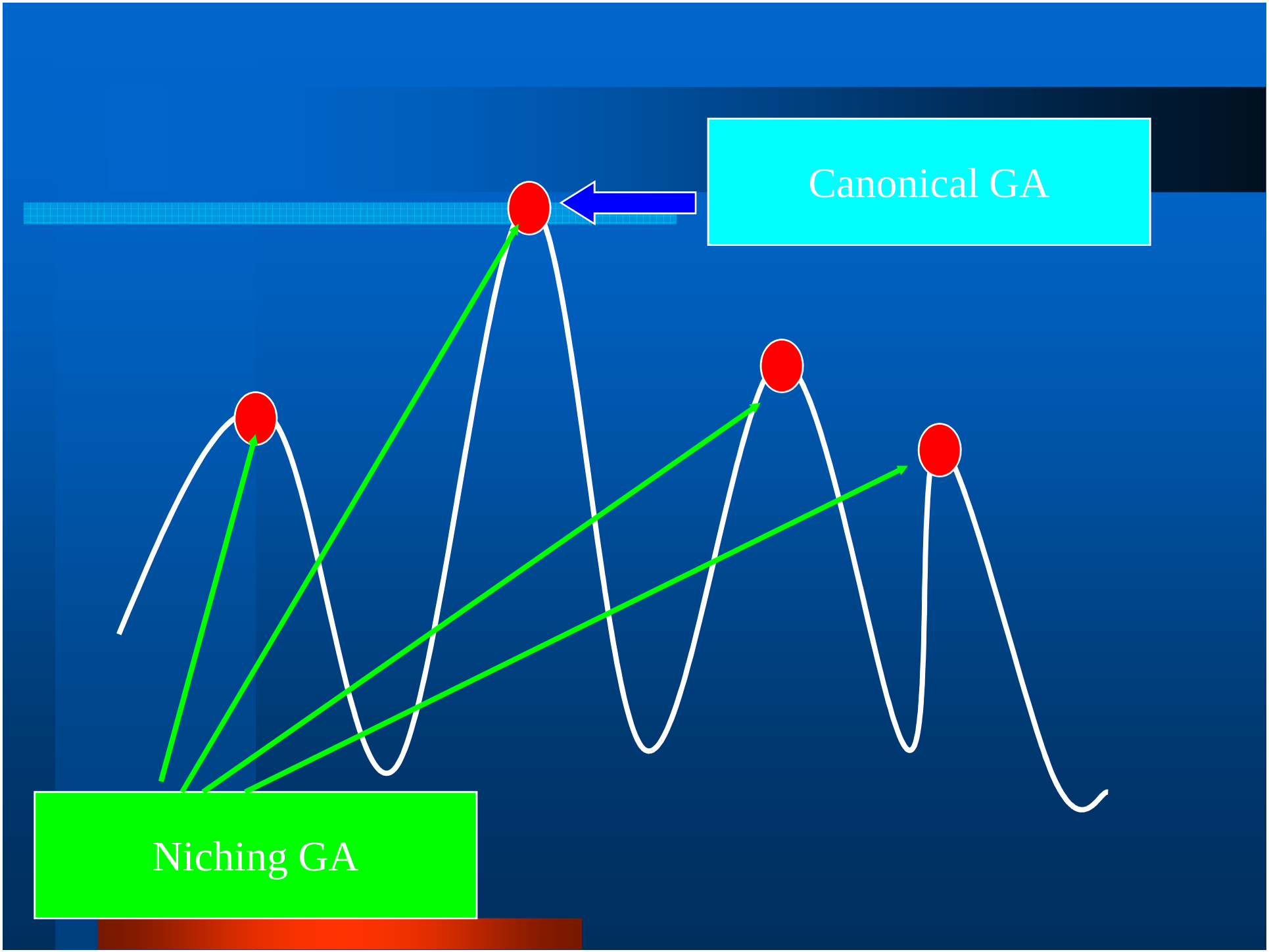


Motivation 2: Genetic Drift

- Finite population with global mixing and selection eventually converge around one optimum
- Often might want to identify several possible peaks
- This can aid global optimisation when sub-optima has the largest basin of attra ction

Niching Genetic Algorithms: Motivation

- Traditional genetic algorithms with elitist selection are suitable to locate the optimum of **unimodal functions** as they converge to a single solution of the search space.
- Real problem, however, often require the identificat ion of optima along with some local optima.
- For this purpose, **niching methods** extend the simple genetic algorithms by ***promoting the formati on of subpopulations in the neighborhood of the local optimal solutions.***


Canonical GA
Niching GA

Niching Genetic Algorithms: The Idea

- Niching methods have been developed to reduce the effect of **genetic drift** resulting from the selection operator in the simple genetic algorithms.
- They maintain population diversity and permit genetic algorithms to explore more search space so as to identify multiple peaks, whether optimal or otherwise.
- The **fitness sharing method** is probably the best known and best used among the niching techniques.

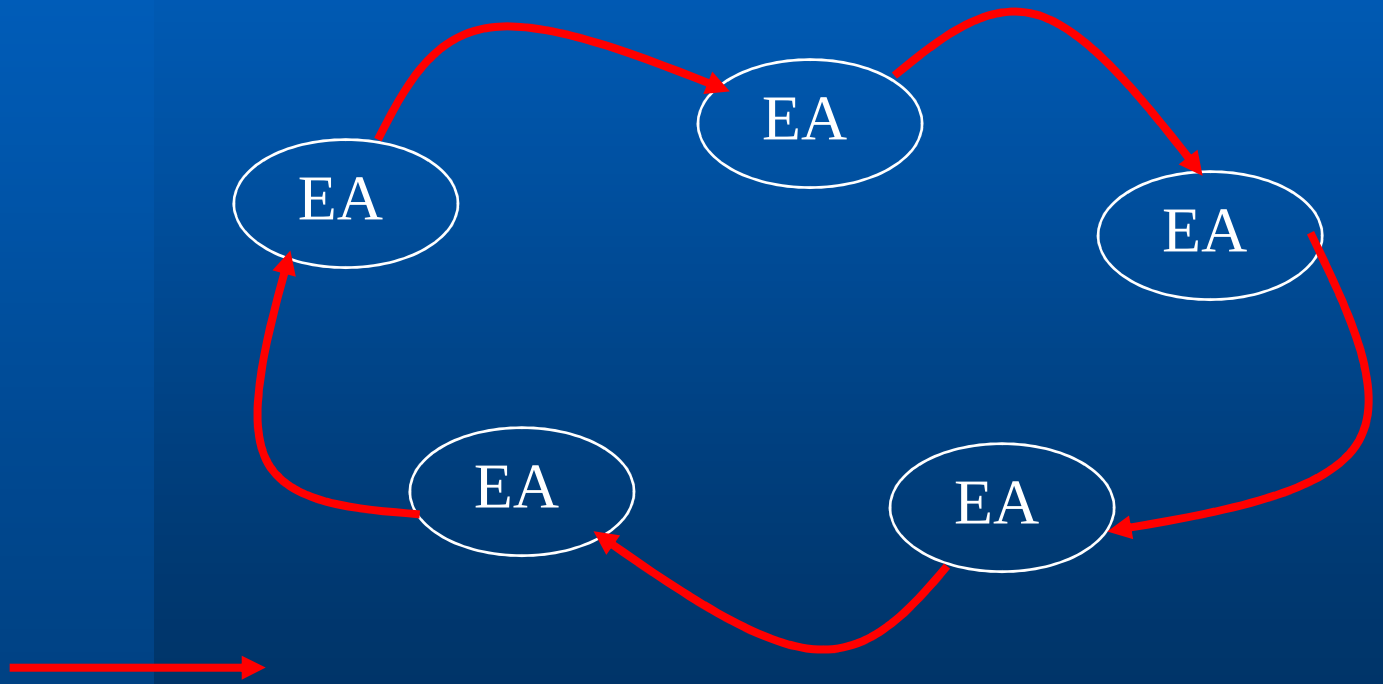
Ecological Meaning (1)

- In natural ecosystem, a niche can be viewed as an organisms task, which permits species to survive in their environment.
- Species are defined as a collection of similar organisms with similar features.
- The subdivision of environment on the basis of an organisms role reduces inter-species competition for environmental resources.
- This reduction in competition helps stable sub-populations to form around different niches in the environment.

Implications for Evolutionary Optimisation

- Two main approaches to diversity maintenance:
- Implicit approaches:
  - Impose an equivalent of geographical separation
  - Impose an equivalent of speciation
- Explicit approaches
  - Make similar individuals compete for resources (fi tness)
  - Make similar individuals compete with each other for survival

Implicit 1: "Island" Model Parallel EAs



Periodic migration of individual solutions between populations

Island Model EAs contd:

- Run multiple populations in parallel, in some kind of communication structure (usually a ring).
- After a (usually fixed) number of generations (an ***Epoch***), exchange individuals with neighbours
- Repeat until ending criteria met
- Partially inspired by parallel/clustered systems

Island Model Parameters 1

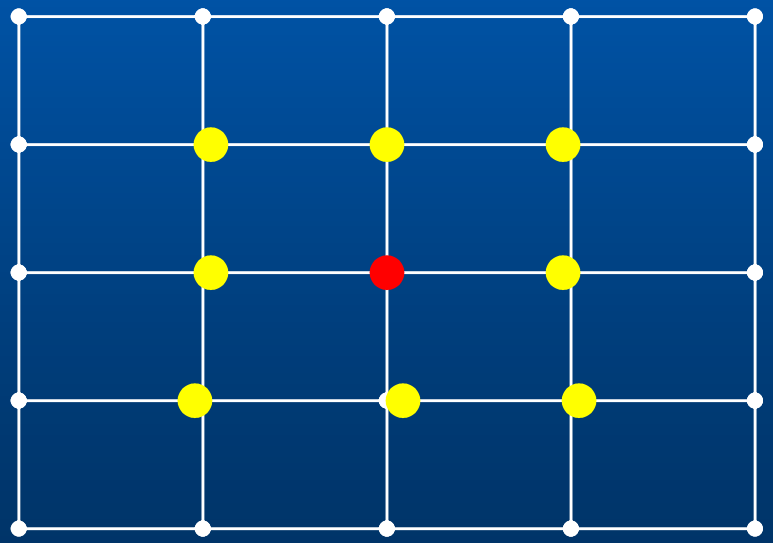
- Could use different operators in each island
- How often to exchange individuals ?
  - Too quick and all pops converge to same solution
  - Too slow and waste time
  - Most authors use range~ 25-150 gens
  - Can do it adaptively (stop each pop when no improvement for (say) 25 generations)

Island Model Parameters 2

- How many, which individuals to exchange ?
  - Usually ~2-5, but depends on population size.
  - More sub populations usually gives better results but there can be a “critical mass” i.e. minimum size of each sub population needed
  - Martin et al found that better to exchange randomly selected individuals than best
  - Can select random/worst individuals to replace

Implicit 2: Diffusion Model Parallel EAs

- Impose spatial structure (usually grid) in 1 population



Current individual

Neighbours

Diffusion Model EAs

- Consider each individual to exist on a point on a grid
- Selection (hence recombination) and replacement happen using concept of a neighbourhood a.k.a. *deme*
- Leads to different parts of grid searching different parts of space, good solutions spread across grid over a number of gens

Diffusion Model Example

- Assume rectangular grid so each individual has 8 immediate neighbours
- Equivalent of 1 generation is:
  - Pick point in population at random
  - Pick one of its neighbours using roulette wheel
  - Crossover to produce 1 child, mutate
  - Replace individual if fitter
  - Circle through population until done

Explicit 1: Fitness Sharing Method

- The sharing method essentially modifies the searc h landscape by ***reducing the payoff in densely pop ulated regions.***
- This method rewards individuals that uniquely ex ploit areas of the domain, while discouraging high ly similar individuals in a domain.
- This causes population diversity pressure, which helps maintain population members at local opti ma.

Explicit 2: Crowding

- Attempts to distribute individuals evenly amongst niches
- Relies on the assumption that offspring will tend to be close to parents
- Uses a distance metric in phenotype genotype space
- Randomly shuffle and pair parents, produce 2 offspring
- 2 parent/offspring tournaments - pair so that d(p1,o1)+d(p2,o2) < d(p1,02) + d(p2,o1)

Multi-objective Problems

Multi-Objective Problems (MOPs)

- Wide range of problems can be categorised by the presence of a number of n possibly conflicting objectives:
  - Buying a car: speed vs. price vs. reliability
  - Engineering design: lightness vs strength
- Two part problem:
  - Finding set of good solutions
  - Choice of best for particular application

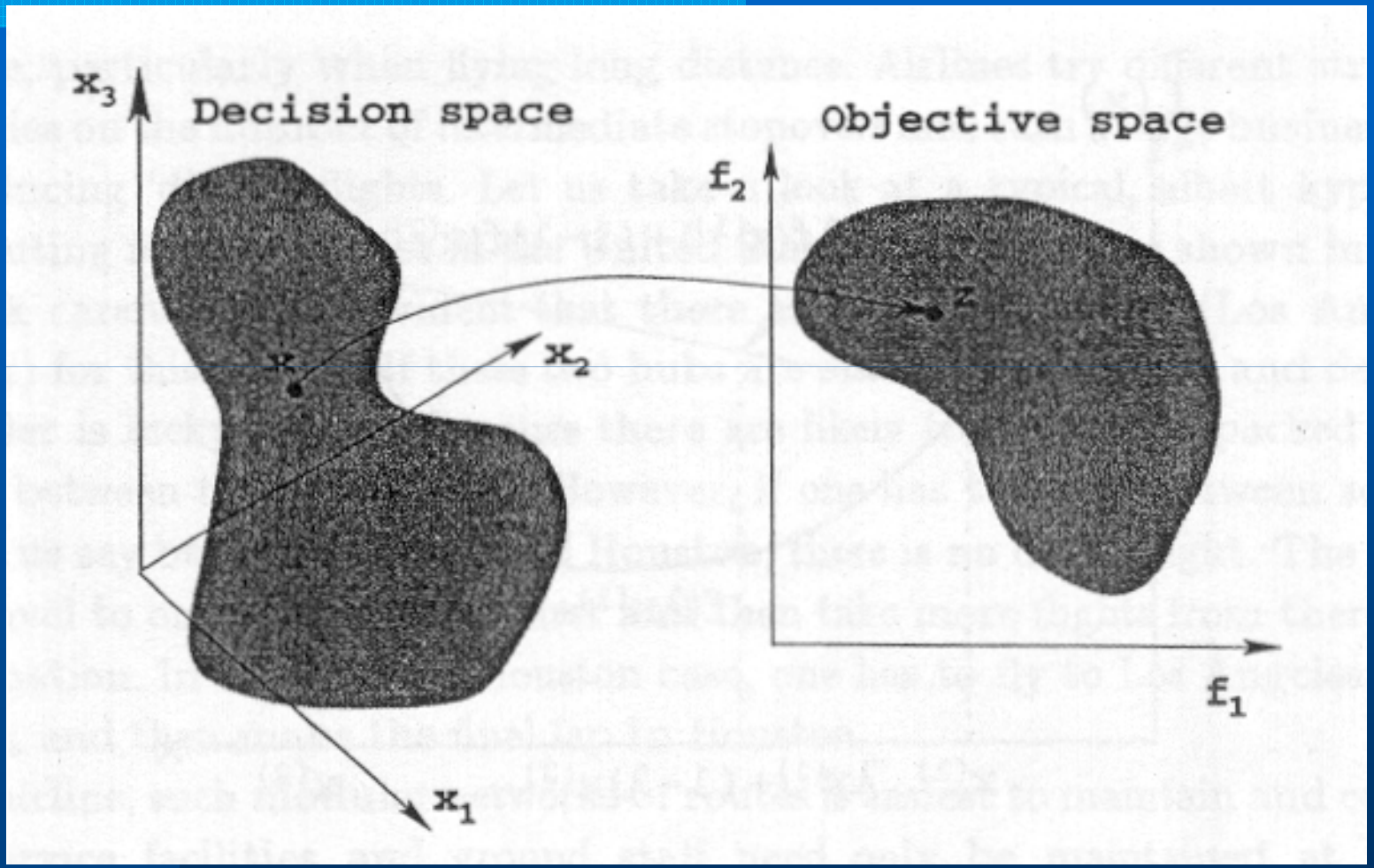
Multi-Objective Optimization

- Optimization problems with multiple, conflicting objectives.

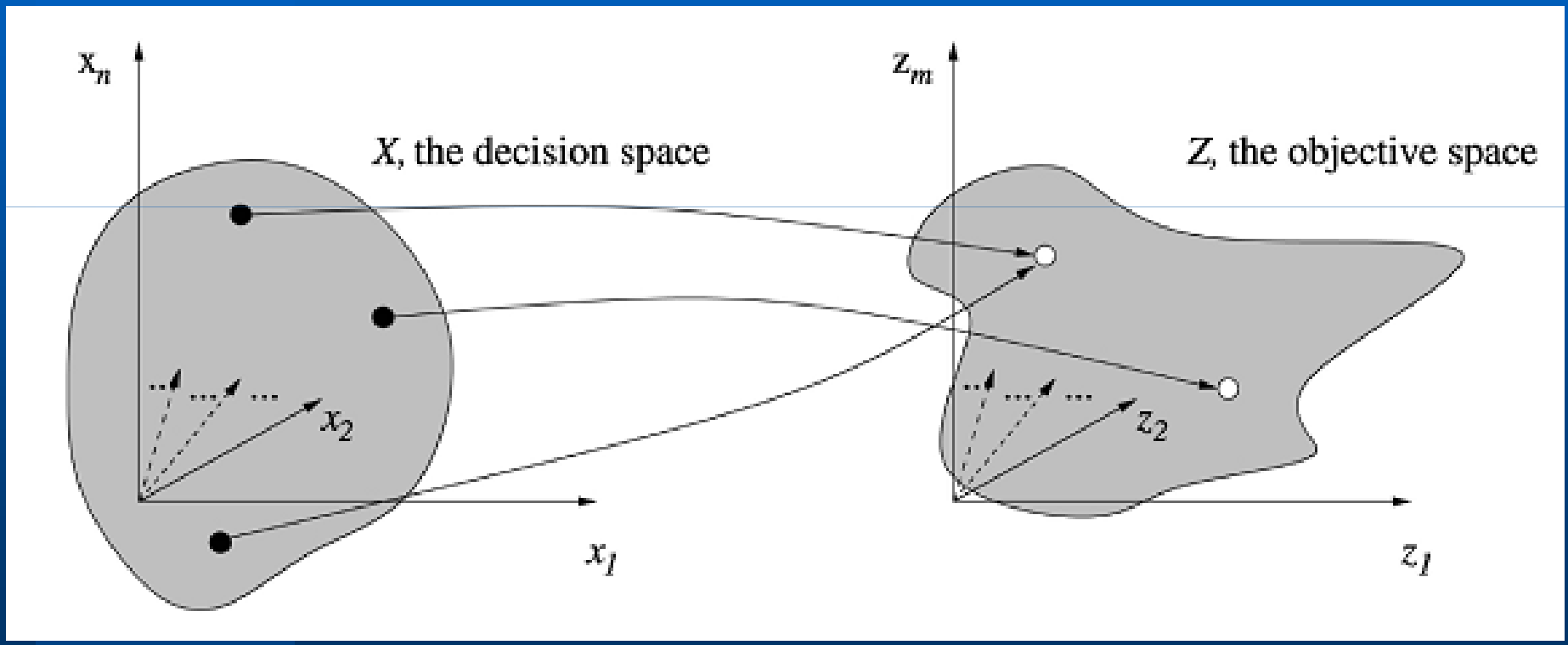
$$
\begin{aligned}
&\text{Minimize/Maximize} && f_m(x), && m = 1,2,\ldots,M;\\
&\text{subject to} && g_j(x) \geq 0, && j = 1,2,\ldots,J;\\
& && h_k(x) = 0, && k = 1,2,\ldots,K;\\
& && x_i^{(L)} \leq x_i \leq x_i^{(U)}, && i = 1,2,\ldots,n;
\end{aligned}
$$

M objective functions: $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T$

Decision Space vs. Objective Space



Decision Space vs. Objective Space



Objectives in Multi-Objective Optimization

- Two goals in a MOO
  - To find a set as close as possible to the Pareto-optimal front
  - To find a set of solutions as diverse as possible
  - Ex. Airline Route: Cost vs. Time
- With respect to single objective
  - Two goals instead of one
  - Dealing with two search space
    - Objective space & Decision space

MOPs 1: Conventional approaches

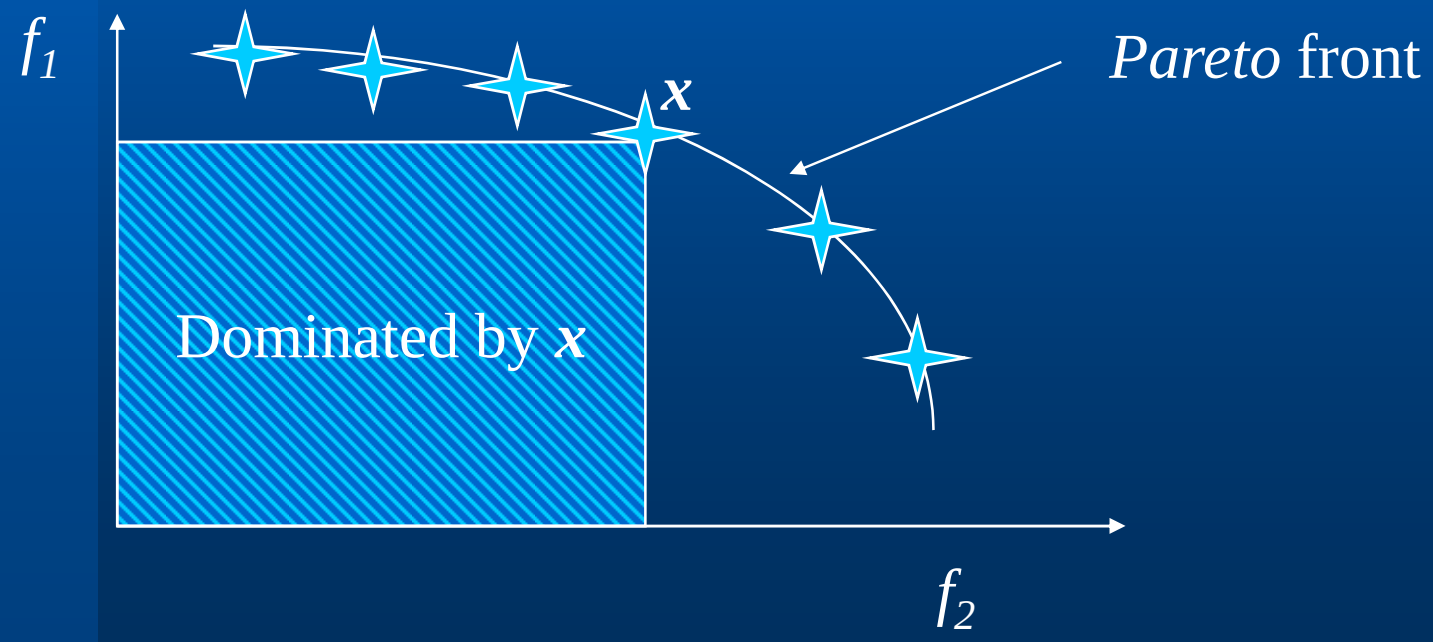
- Rely on using a weighting of objective function values to give a single scalar objective function which can then be optimised:

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

- To find other solutions have to re-optimise with different $w_{i.}$

MOPs 2: Dominance

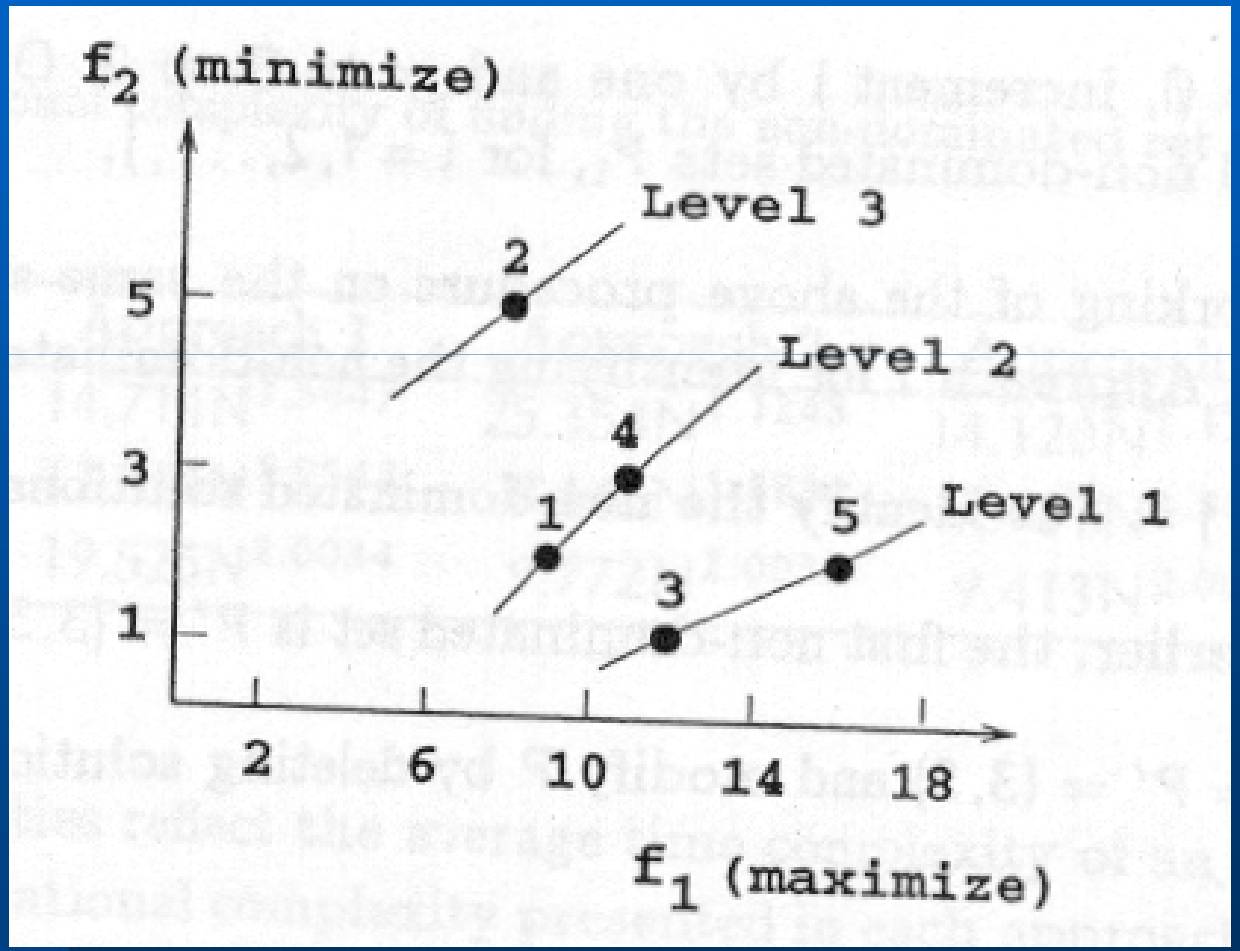
- we say x dominates y if it is at least good on all criteria and ***better*** on at least one



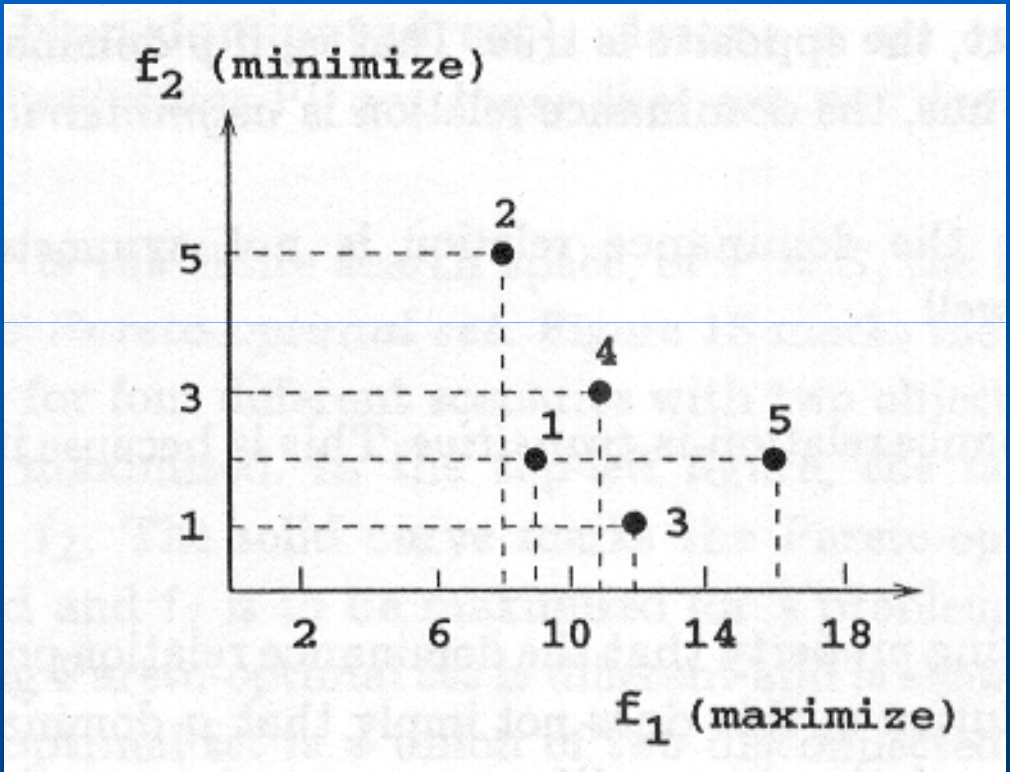
Concept of Domination

- A solution x^1 is said to dominate the other solution x^2, if both conditions 1 and 2 are true:
  1. The solution x^1 is no worse than x^2 in all objectives
  2. The solution x^1 is strictly better than x^2 in at least one objective

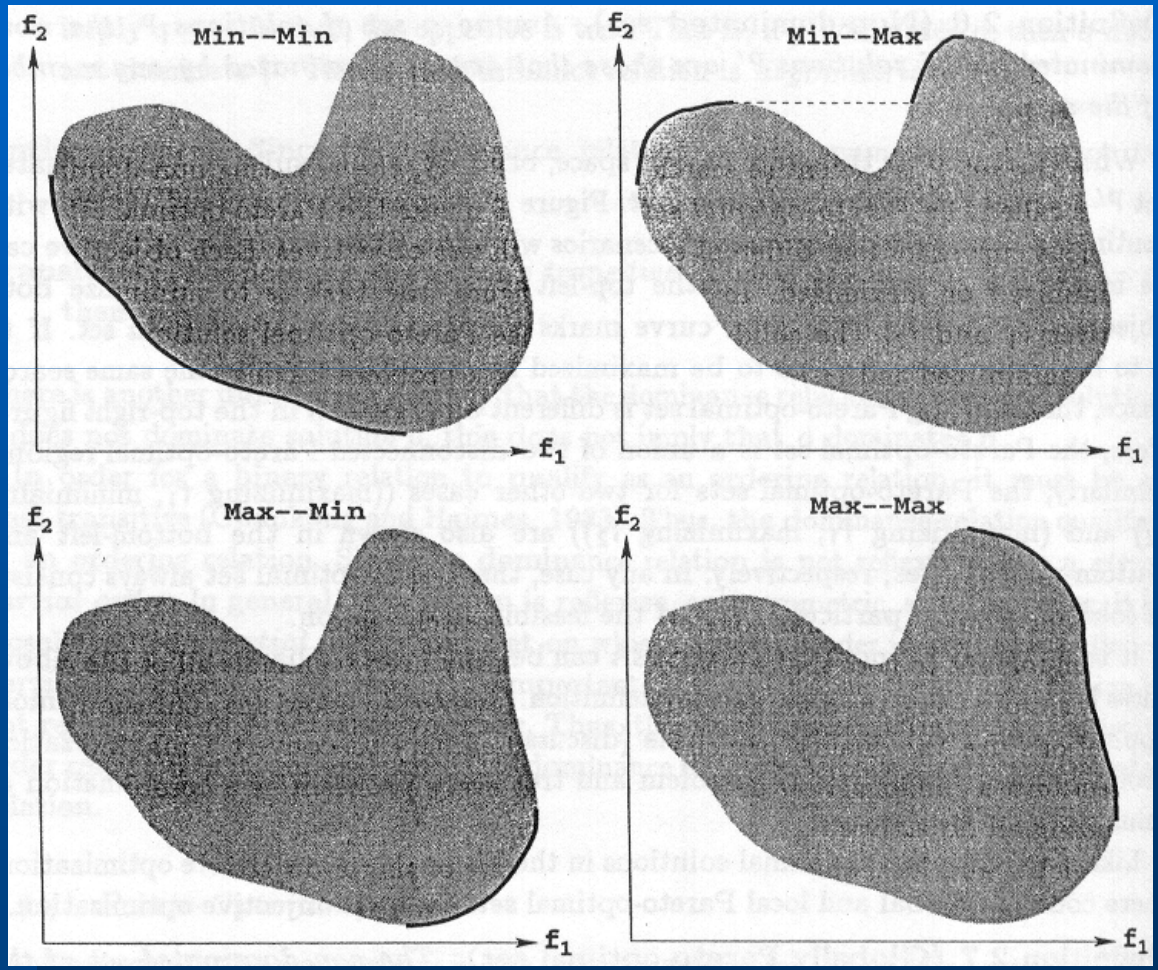
Non-dominates Sorting Example



Dominance Example



Pareto-Optimality Example



Pareto-Optimality

- 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.
- Globally Pareto-optimal set
  - The non-dominated set of the entire feasible search space S is the globally Pareto-optimal set
- Locally Pareto-optimal set

Non-dominated Sorting of a Population

1. Set all non-dominated sets P_j, (j=1,2,…) as empty sets. Set non-domination level counter j = 1.
2. Find the non-dominated set P′ of P
   1. Set solution counter i=1 and create an empty non-dominate set P′.
   2. For a solution j∈ P (but j ≠ i), check if solution j dominates solution i, If yes, go to Step 2-4.
   3. If more solutions are left in P, increment j by one and go to Step 2-2; otherwise, set $P'=P' \cup \{i\}$.
   4. Increment i by one. If i ≤N, go to Step 2-2; otherwise stop and declare P′ as the non-dominated set.
3. Update $P_j = P'$ and $P = P \setminus P'$.
4. If P ≠ Φ, increase j by one and go to Step 2. Otherwise, stop and declare all non-dominated sets Pj, for j=1,2,…,i.

MOPs 3: Advantages of EC approach

- Population-based nature of search means you can *simultaneously* search for set of points approximating Pareto front
- 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

MOPs 4: Requirements of EC app roach

- Way of assigning fitness,
  - usually based on dominance
- Preservation of diverse set of points
  - similarities to multi-modal problems
- Remembering all the non-dominated points you've seen
  - usually using elitism or an archive

MOPs 5: Fitness Assignment

- Could use aggregating approach and change weights during evolution
  - no guarantees
- Different parts of population use different criteria
- Dominance
  - ranking or depth based
  - fitness related to whole population

MOPs 6: Diversity Maintenance

- Usually done by niching techniques such as:
  - fitness sharing
- All rely on some distance metric in genotype / phenotype space

MOPs 7: Remembering Good Points

- Could just use elitist algorithm
- Common to maintain an archive of non-dominated points
  - Some algorithms use this as second population that can be in recombination etc
  - Others divide archive into regions

Multiple Objective GA

- Use the non-dominated classification of a GA population.
- Explicitly caters to emphasize non-dominated solutions and simultaneously maintains diversity in the non-dominated solutions.

Niched-Pareto GA

- Uses a binary tournament selection scheme based on Pareto dominance.
- Solutions are selected if they dominate both the other and some small group of randomly selected solutions, but fitness sharing occurs only in the cases when both solutions are (non)dominated.

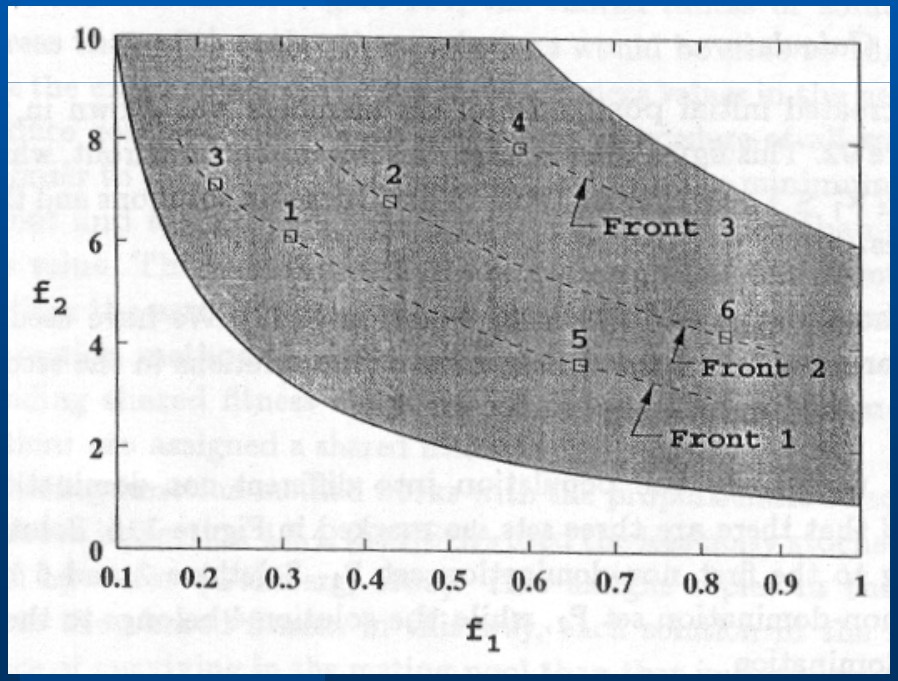
Non-Dominated Sorting GA

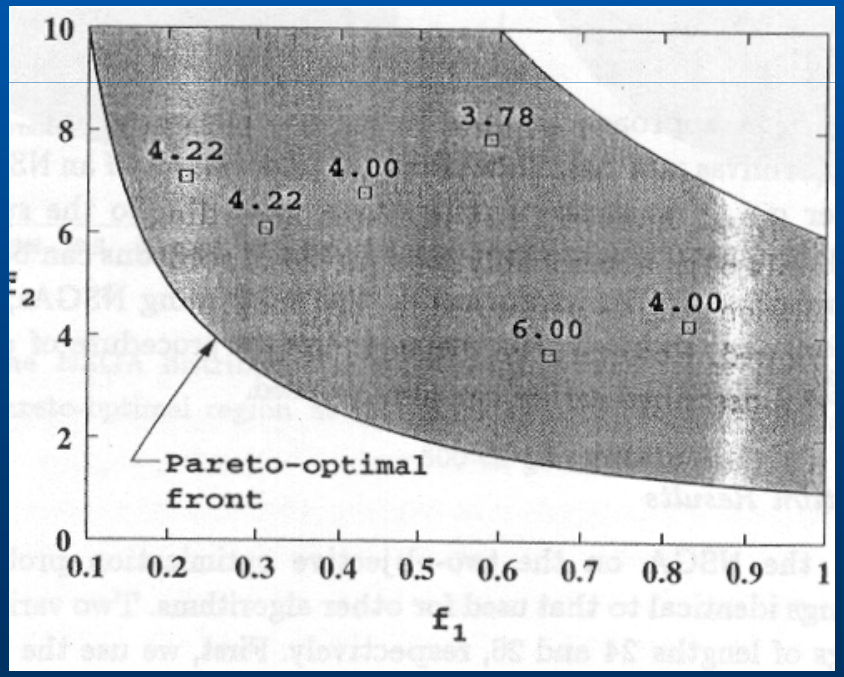
- Before selection, the population is ranked on the basis of domination (Pareto ranking)
- All nondominated individuals are classified into one category.
- To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values

Non-dominated Sorting Genetic Algorithm (NSGA)

NSGA

- Sort the population P according to non-domination
  - All solutions in the first set belong to the best non-dominated set in the population





NSGA

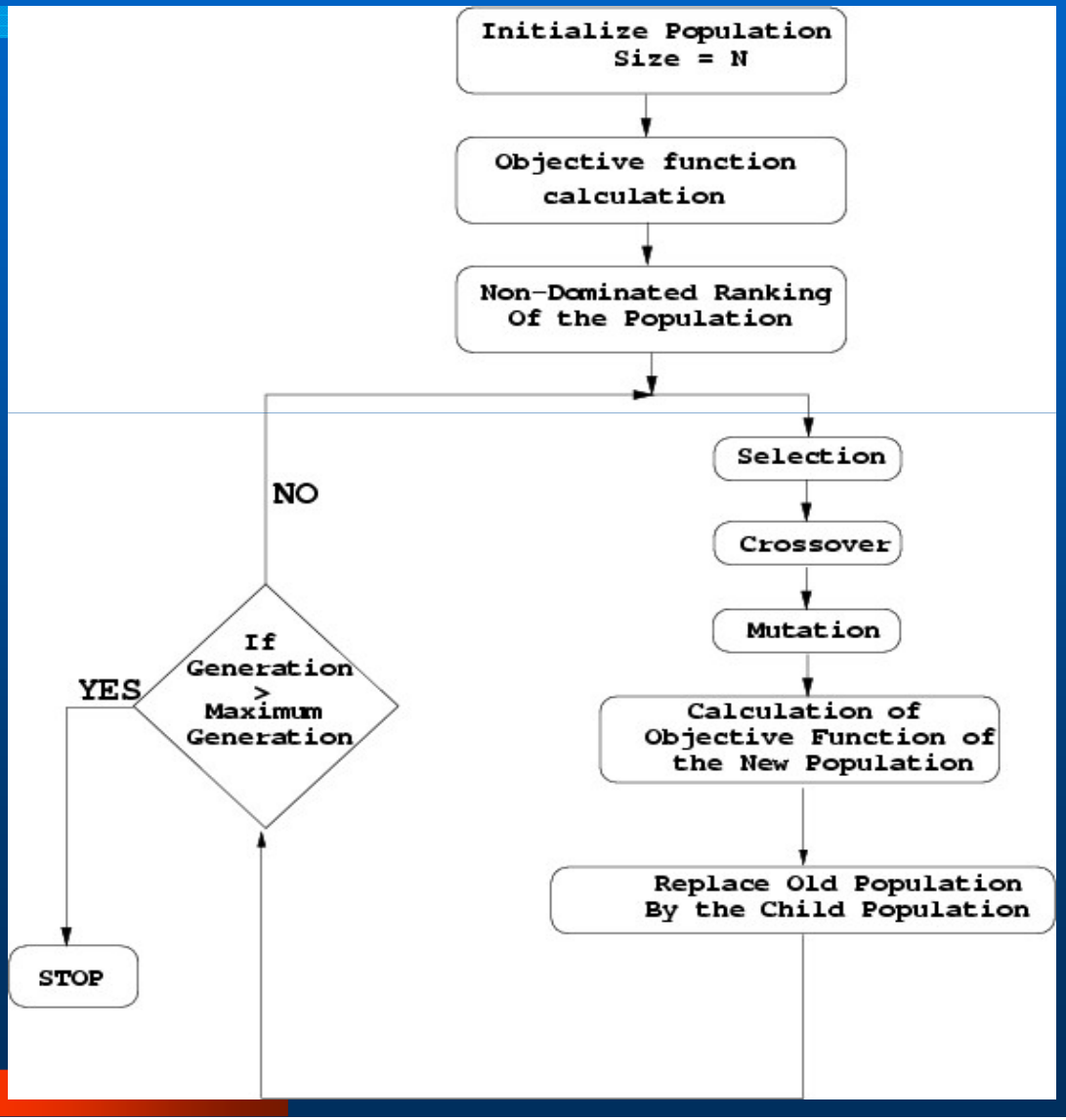
- Advantages
  - The assignment of fitness according to non-dominated sets
  - Since better non-dominated sets are emphasized systematically, an NSGA progresses to the Pareto-optimal region.
- Disadvantages
  - The sharing function approach requires the sharing parameter.
  - Performance of an NSGA is sensitive to the sharing parameter

Non-Elitist Multi-Objective EA

- Motivations:
  - A user is usually not sure of an exact trade-off relationship among objectives.
  - Equi-spaced weight does not always result to equi-spaced trade-off solutions.
- After finding diverse set of optimal solutions, it is possible to calculate the associated weights.
  - Enables to choose from different trade-offs.

Modified NSGA-II algorithm-I

Elitism removed

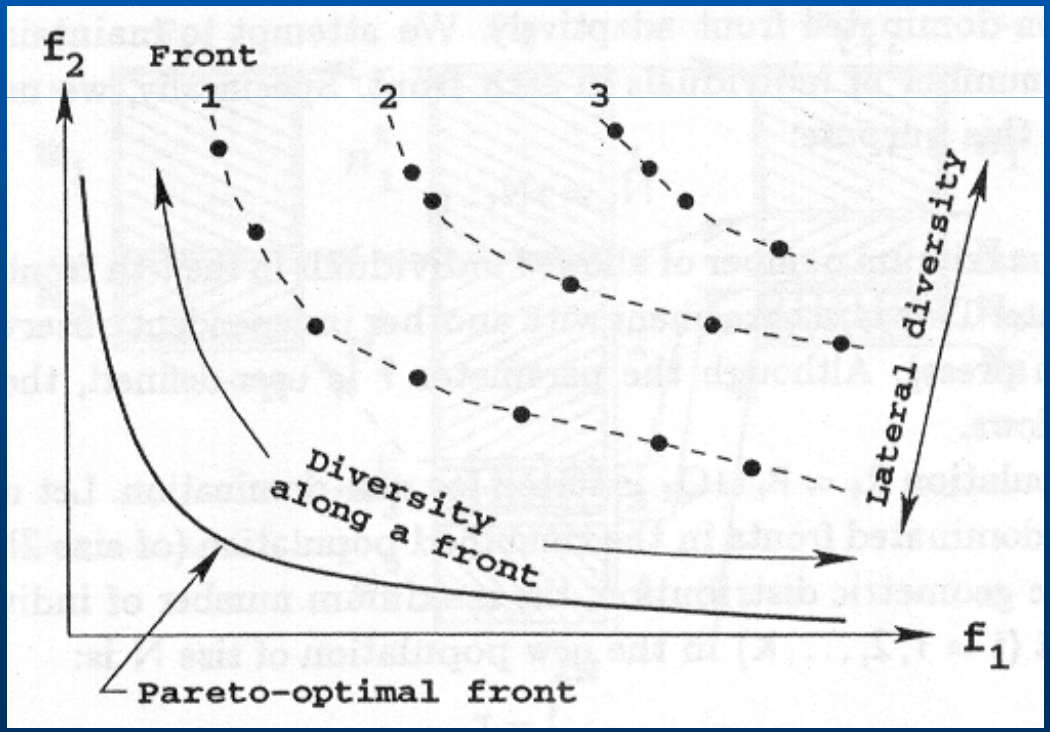


Elitist MOEA

- The presence of elites
  - GAs converge to the global optimal solution
  - Enhance the probability of creating better offspring
- Which solutions are elites in the context of multi-objective optimization?
  - A solution can be evaluated based on non-domination rank in the population

Controlling Elitism

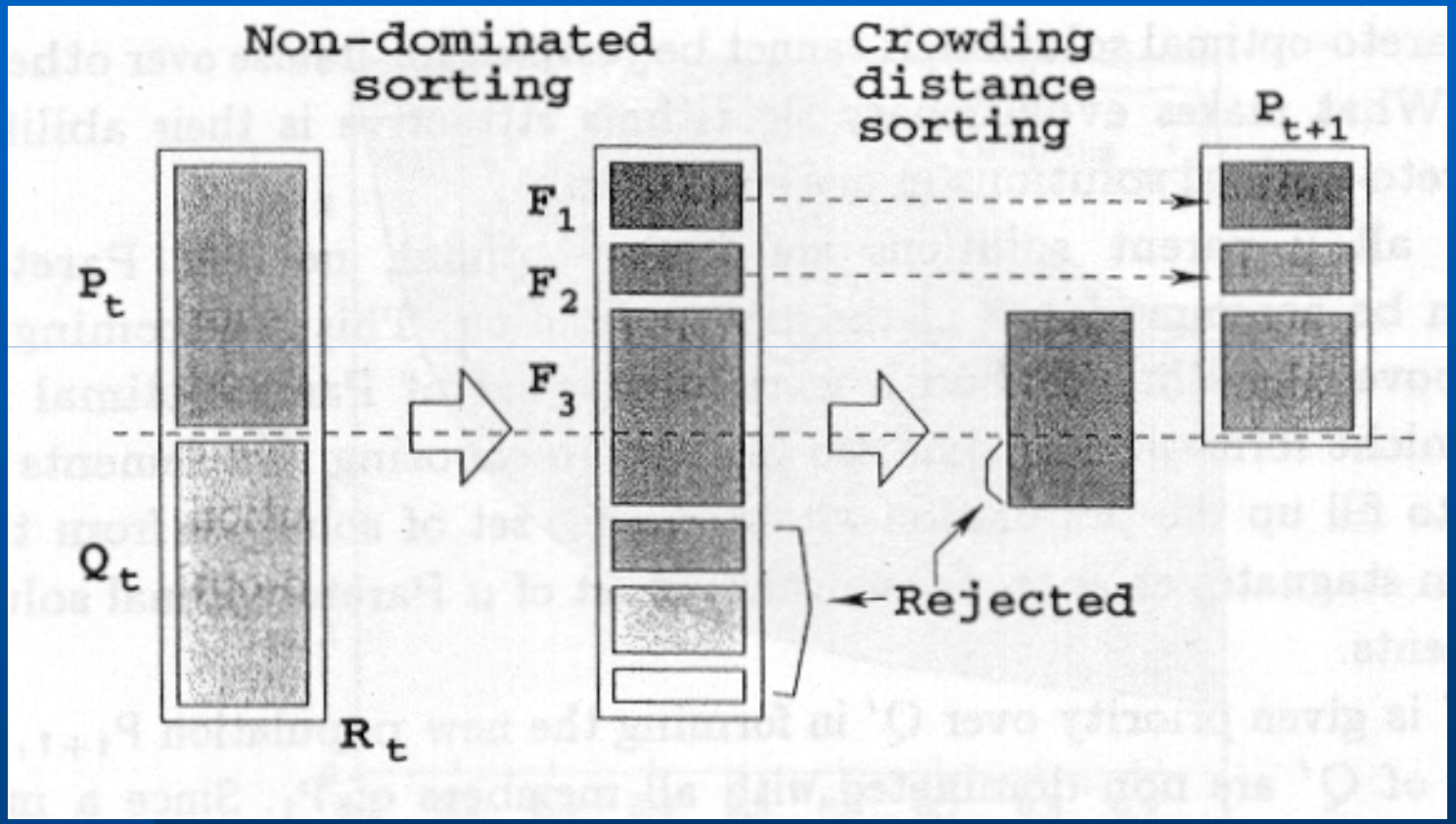
- To ensure better convergence, a search algorithm may need diversity in both aspects – along the Pareto-optimal front and lateral to the Pareto-optimal front.



Elitist NSGA: NSGA-II

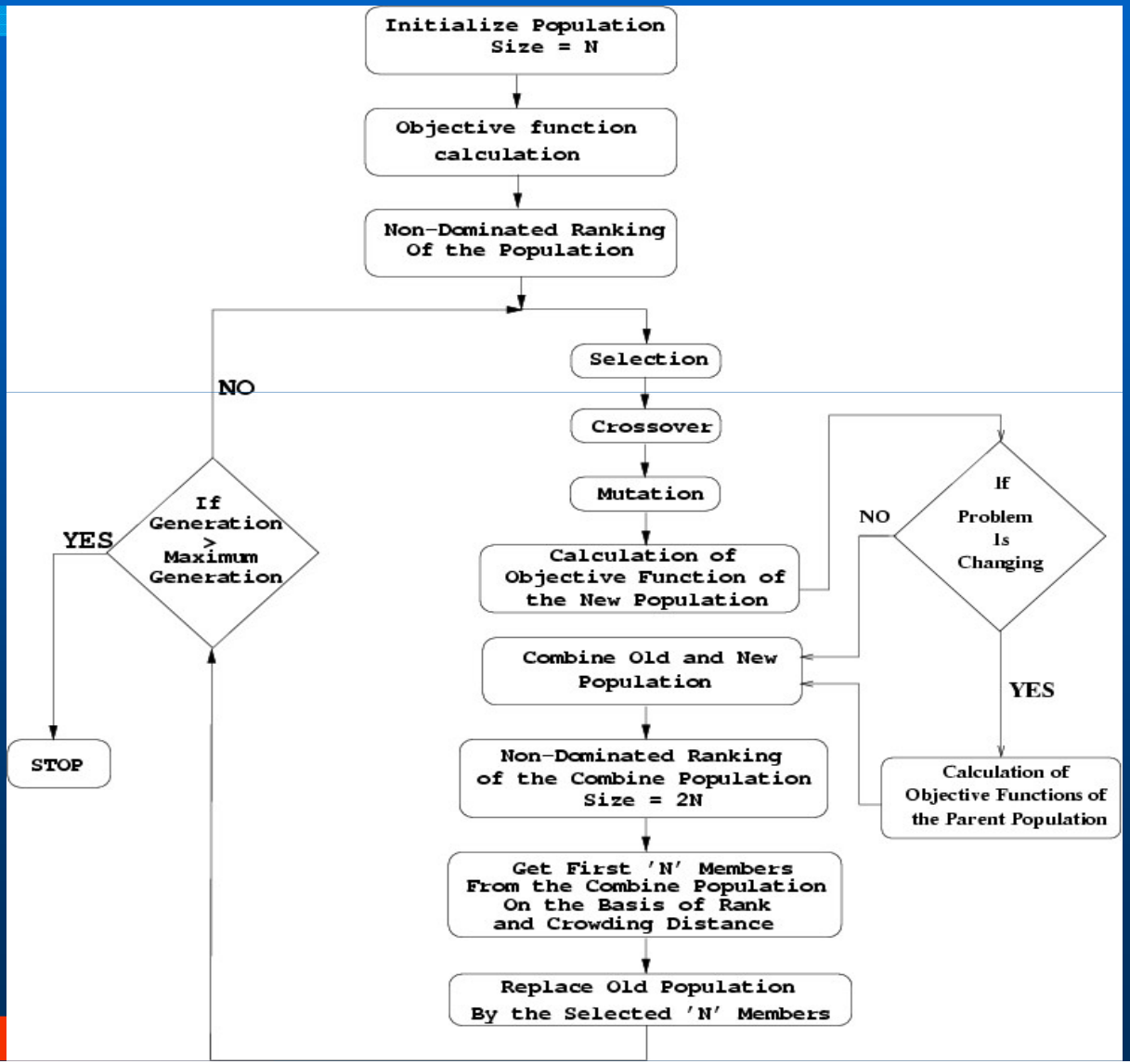
- Uses an explicit diversity-preserving mechanism.

NSGA-II: Elitist NSGA



Modified NSGA-II algorithm-II

Elitism introduced interactively



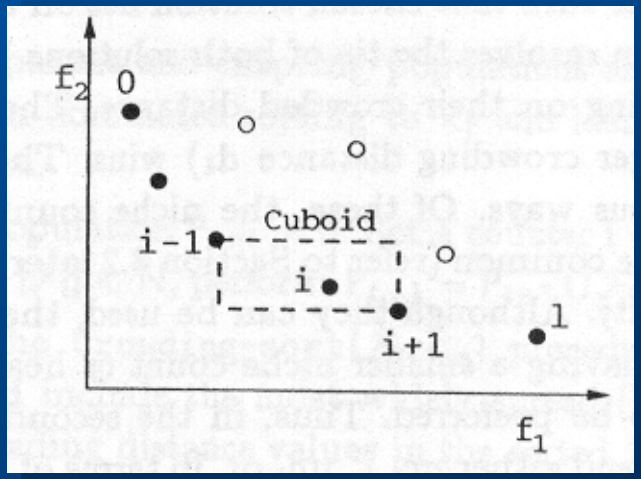
NSGA-II Procedure: Crowded Tournament Selection Operator

- A solution *i* wins a tournament with another solution *j* if any of the following conditions are true :
  - If solution *i* has a better rank, that is, $r_i < r_j$
  - If they have the same rank but solution *i* has a better crowding distance than solution *j*, that is, $r_i = r_j$ and $d_i > d_j$

NSGA-II Procedure: Crowded Tournament Selection Operator

1. Call the number of solutions in F as $l = |F|$. For each i in the set, first assign $d_i=0$.
2. For each objective function sort the set in worse order, or find the sorted indices vector: $I^m = \text{sort}(f_m, >)$.
3. Assign a large distance to the boundary solutions, and for all other solutions j=2 to (l-1) :

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{\max} - f_m^{\min}}$$



NSGA-II

- Advantages
  - No extra niching parameter is required
  - Crowding distance can be implemented in the parameter space
- Disadvantages
  - If population size is small, NSGA-II shows the poor exploration power.