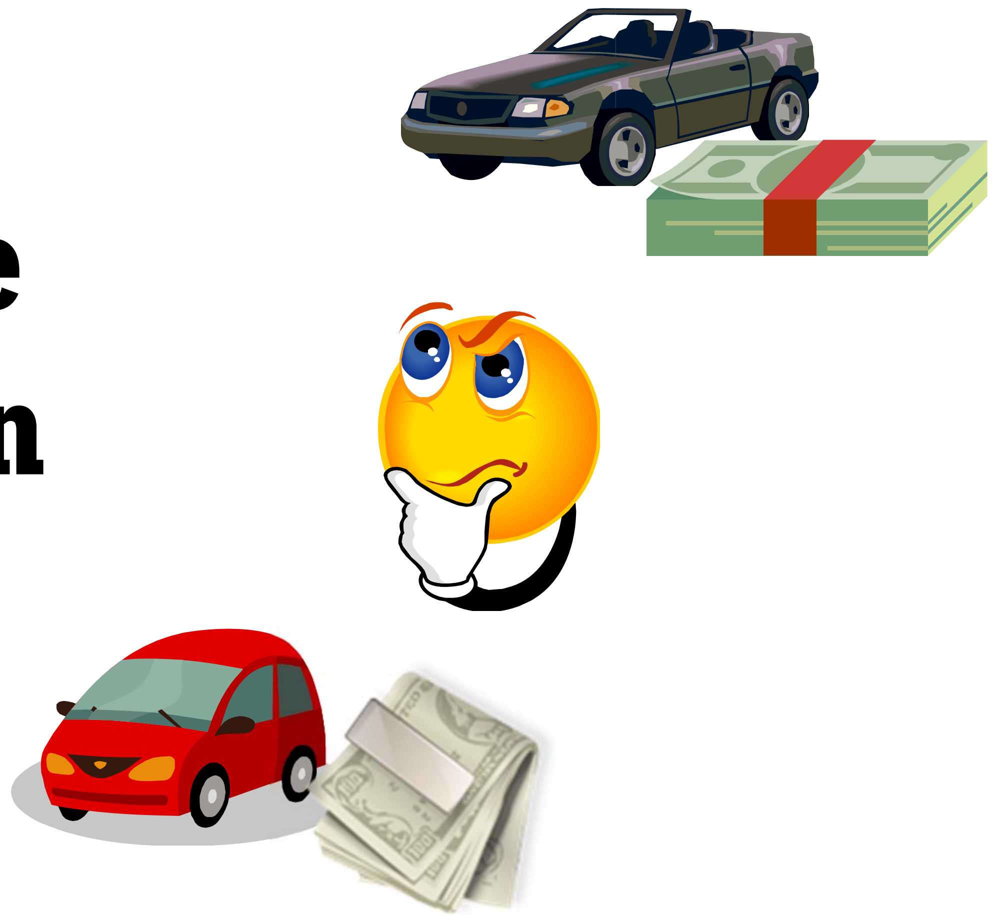
Evolutionary Algorithms:
Multiobjective Optimization, Genetic Programming

May 21, 2013

Prof. Chang Wook Ahn

Sungkyunkwan Evolutionary Algorithms Lab.
School of Info. & Comm. Eng.
Sungkyunkwan University

Evolutionary
Multiobjective
Optimization



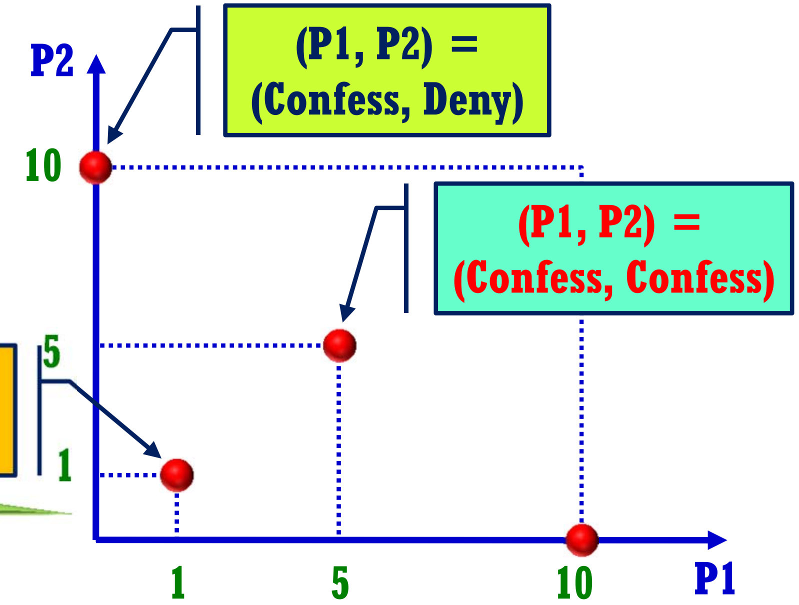
Multiobjective Optimization (1)

❖ Multibojective Optimization - Concept

➢ **Example: Prisoners Dilemma (PD) Problem**

- ✓ 2 prisoners are arrested for a crime.
- ✓ They can make a decision about two actions: Confess, Deny
- ✓ They have to make their own decision simultaneously
- ✓ They want to minimize their prison years under the conditions:

	P2: Confess	P2: Deny
P1: Confess	5, 5	0, 10
P1: Deny	10, 0	1, 1



Prison Years of the Prisoners

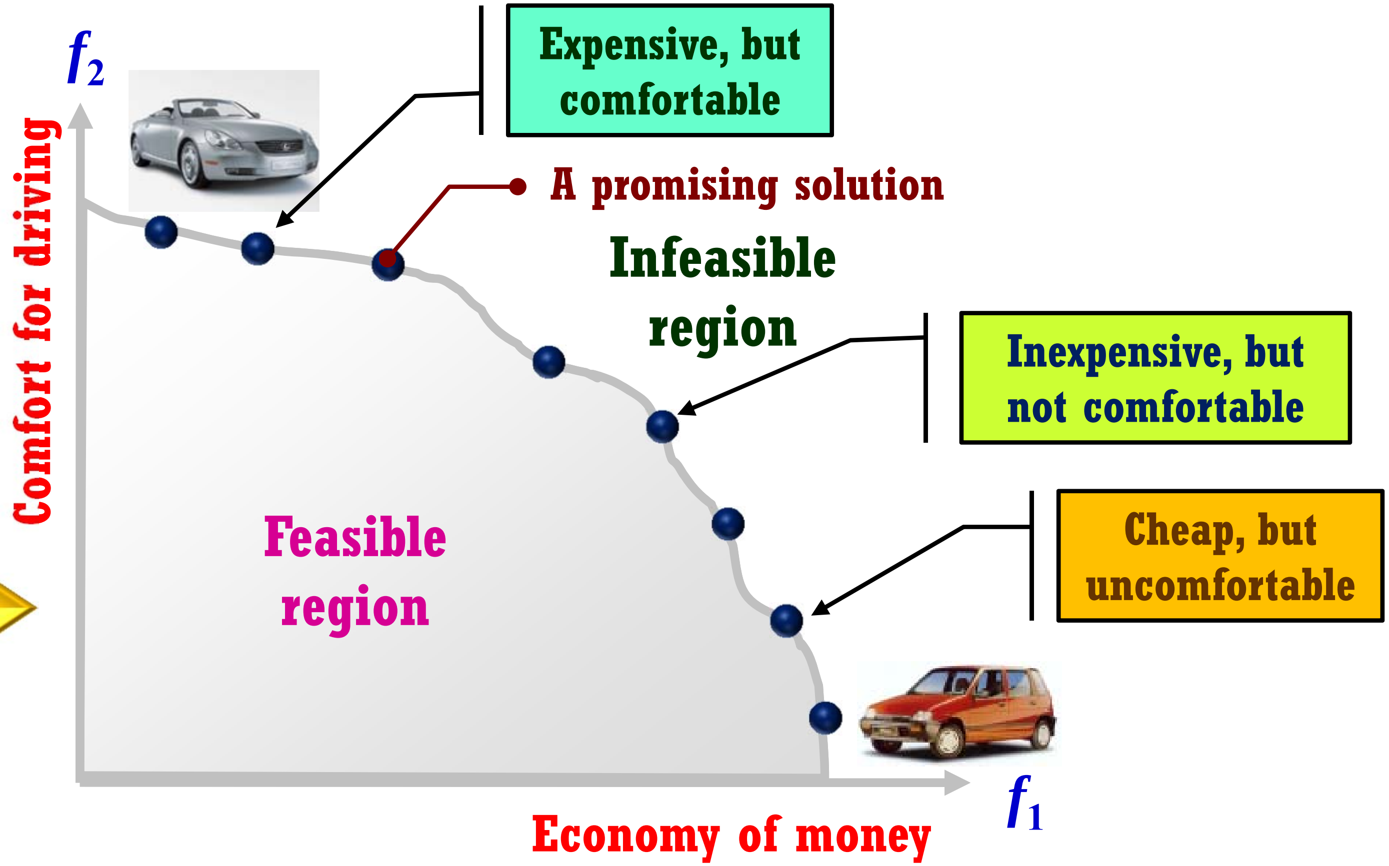
The prisoners will make the decision of (P1, P2)=(Confess, Confess) as they don't trust each other!

Is this optimal?

What do you think about (Deny, Deny)?

Multiobjective Optimization (2)

- **What is the Aim of Multiobjective Optimization?**
  - **Example: Car Buying Problem**
    - ✓ We want to buy a car in view of "Comfort for driving" and "Economy of money"
    - ✓ But the two objectives are conflicted; thus, all possible solutions must be found!



In the Car Buying example, all the circles represent the best solutions when simultaneously considering "comfort for driving" and "economy of money"!

Multiobjective Optimization (3)

❖ What are Multiobjective Optimization Problems (MOPs)?

- **A Class of Optimization Problems that have several Conflicting Objectives**
  - ✓ **The aim is to discover all the possible solutions that are the optimal/best in view of *Multiple Conflicting Objectives***

Single Objective Optimization

For a search space Ω

There is a function $f : \Omega \mapsto \Re$

The task is to find $x^* = \arg\max_{x \in \Omega} f(x)$

subject to $g_i(x) \le 0, \; i = 1, \cdots, m$

$h_i(x) = 0, \; j = 1, \cdots, p$

Multiobjective Optimization

For a search space Ω

There are functions $f_i : \Omega \mapsto \Re$

To find $x^* = \arg\max_{x \in \Omega} (f_1(x), \cdots, f_n(x))$

subject to $g_i(x) \le 0, \; i = 1, \cdots, m$

$h_i(x) = 0, \; j = 1, \cdots, p$

Inequality Constraints

Equality Constraints

Multiobjective Optimization (4)

Multiobjective Optimization Problems

- MOPs have several **conflicting objectives** to be **maximized** simultaneously

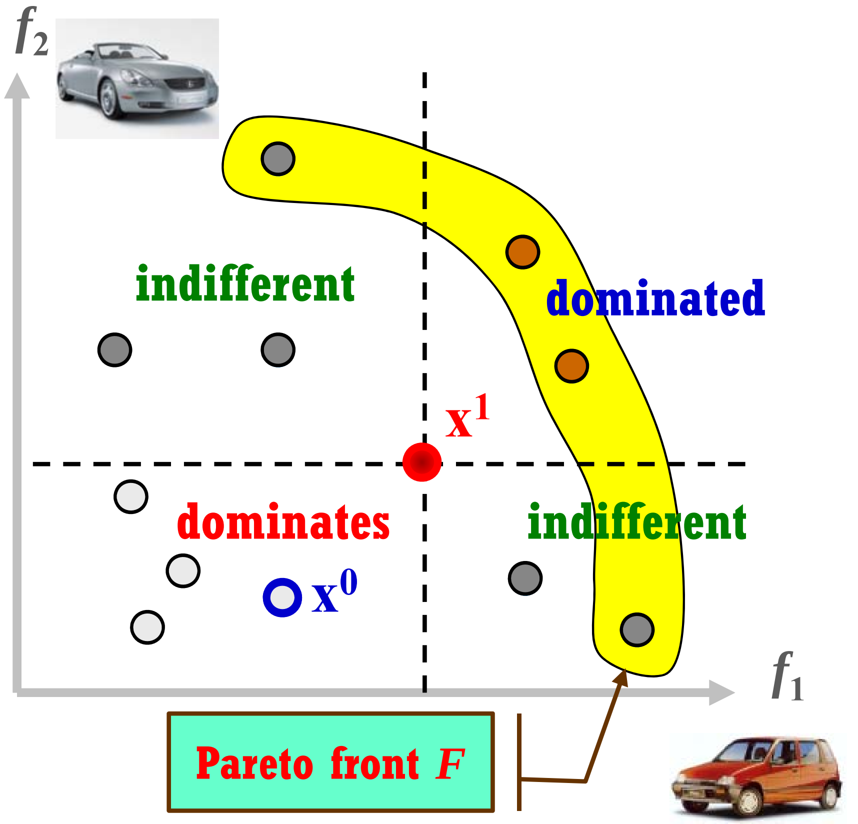
- **Due to the interdependence** of the objectives
  - ✓ MOPs normally have a set of alternative solutions

- **The solutions, known as *Pareto-optimal set*, are optimal in the sense that**
  - ✓ no solution is superior to them overall as no objective can be improved without degrading the others;

$$Q = \{\mathbf{x}^0 \in A \mid \neg\exists \mathbf{x}^1 \in \Omega_f : \mathbf{x}^1 \succ \mathbf{x}^0\}$$

where $\mathbf{x}^1 \succ \mathbf{x}^0$ indicates that x^1 (Pareto) dominates x^0.

$$\forall i : f_i(\mathbf{x}^0) \le f_i(\mathbf{x}^1) \wedge \exists j : f_j(\mathbf{x}^0) < f_j(\mathbf{x}^1)$$



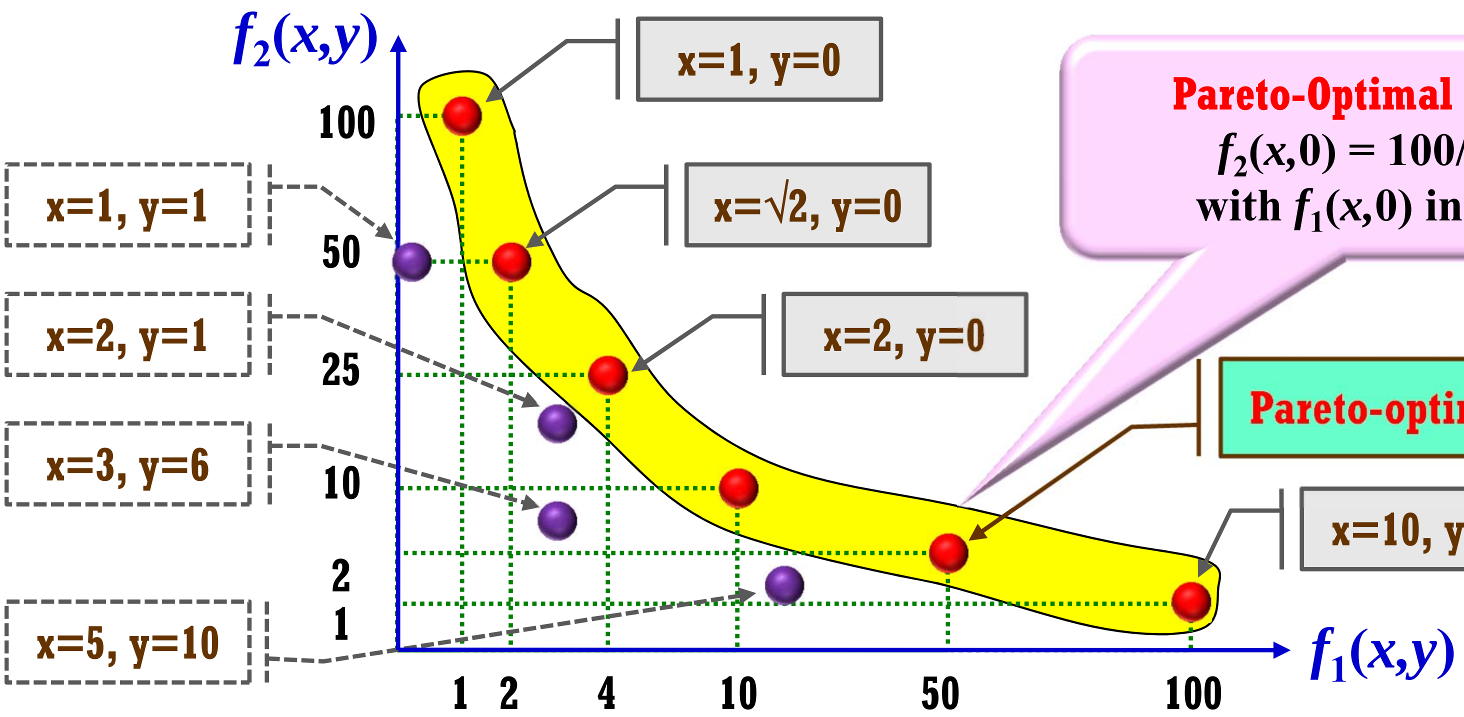
  - ✓ The image of the Pareto-optimal set is defined as the *Pareto (optimal) front*

$$F = \{ f_1(\mathbf{x}^0), f_2(\mathbf{x}^0), \cdots, f_n(\mathbf{x}^0) \mid \mathbf{x}^0 \in Q\}$$

❖ Example

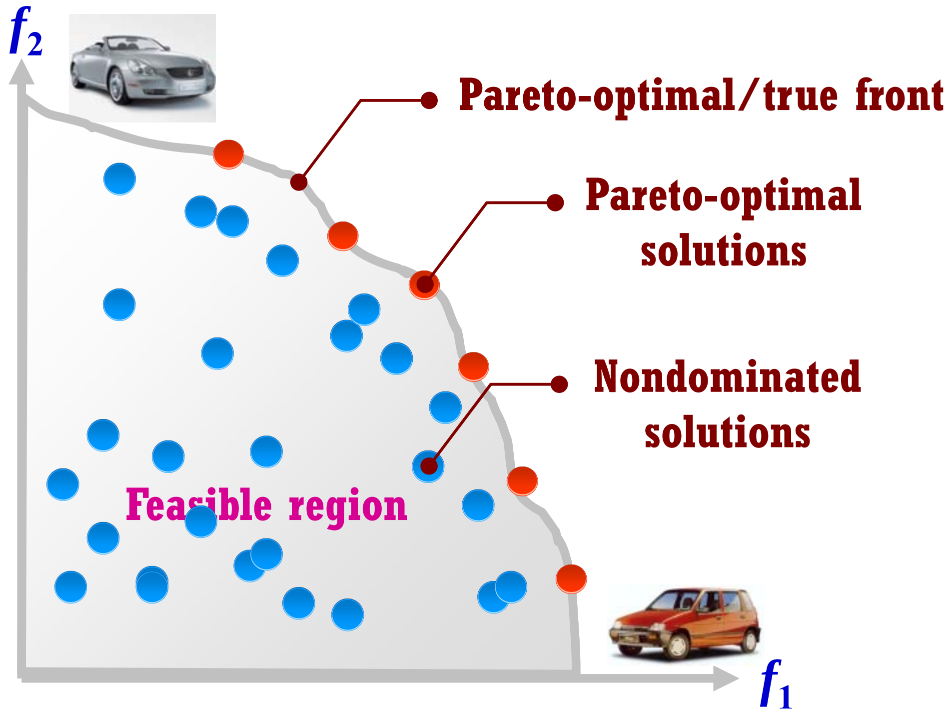
➢ **Find the Optimal Solutions w.r.t the Simultaneous Maximization of**

$$f_1(x,y) = x^2 - y\,,\;\; f_2(x,y) = \frac{100}{x^2 + y} \quad \text{where } 1 \le x \le 10,\ 0 \le y \le x^2$$



Multiobjective Optimization (6)

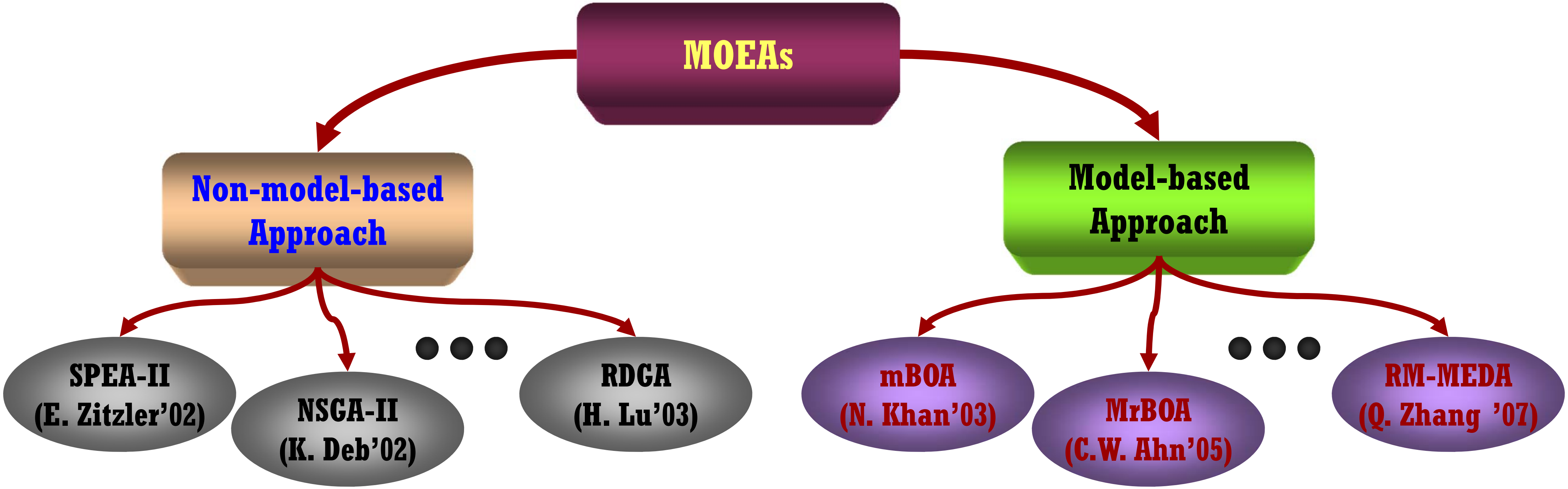
- **Goal of Multiobjective Optimization**
  - ➢ The aim is to find the *global* Pareto optimal set Q^*;
    - ✓ i.e., Place the nondominated set on the *true Pareto front* F^*
  - ☞ But, achieving the goal is not practical since there can be infinite solutions



- **Actual Goals**
  - ➢ **Higher Proximity**
    - Discover nondominated solutions that are closer to the true Pareto front
  - ➢ **Better Diversity**
    - Discover nondominated solutions that are uniformly distributed over the true front

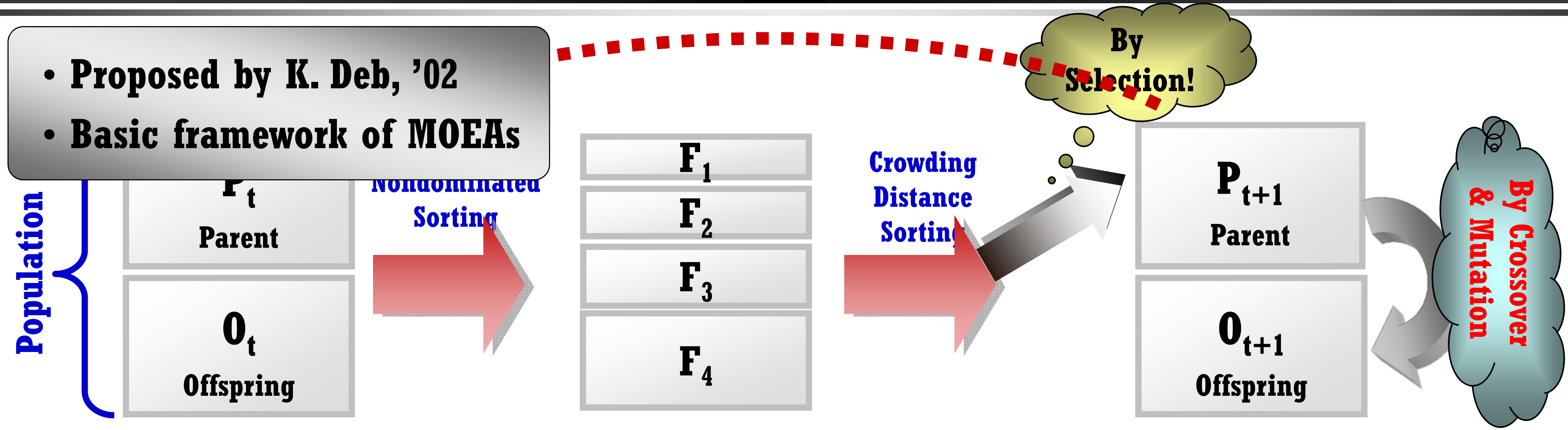
Multiobjective Evolutionary Algo.

- **Classification of Multi-Objective Evolutionary Algorithms (MOEAs)**
  - **Non-model-based approach:** NSGA (II), MOGA, SPEA (II), RDGA
    - Domination information, Sharing strategy, Elitism harmonization
    - It may be inefficient for some complicated problems
  - **Model-based approach:** m(h)BOA, MIDEA, BMOA, MrBOA, RM-MEDA
    - All the benefits of non-model-based approach + EDAs' capability
    - It generally outperforms the non-model-based approach

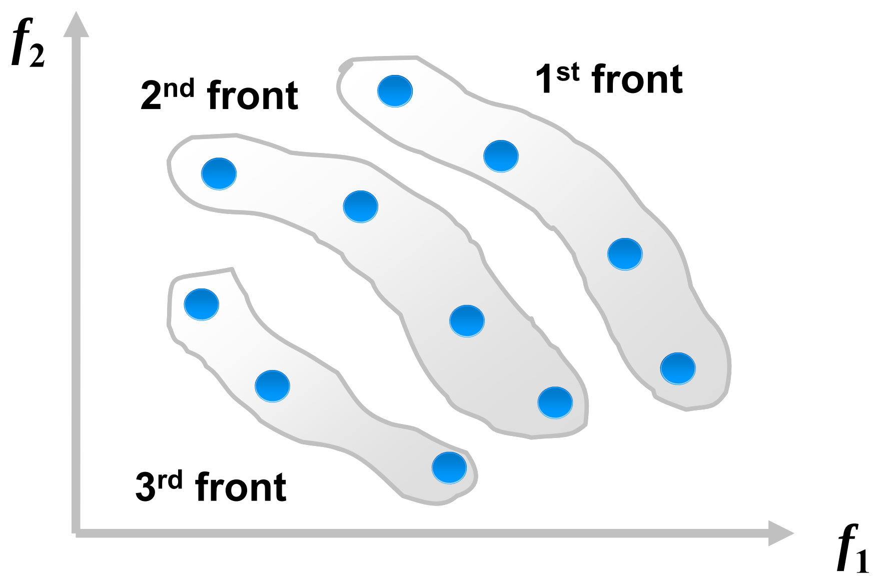


NSGA-II: Nondominated Sorting GA II

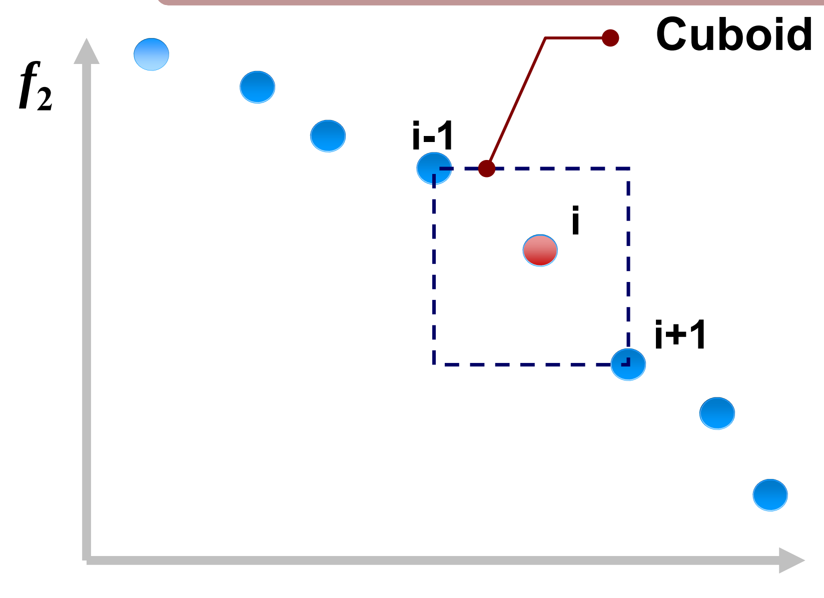
- Proposed by K. Deb, '02
- Basic framework of MOEAs



Nondominated Sorting



Crowding Distance Sorting



Robot Soccer System

❖ Path Planner for Robot Shooting Behavior (using NSGA-II)

➢ **Key objectives of path planning in the robot shooting**

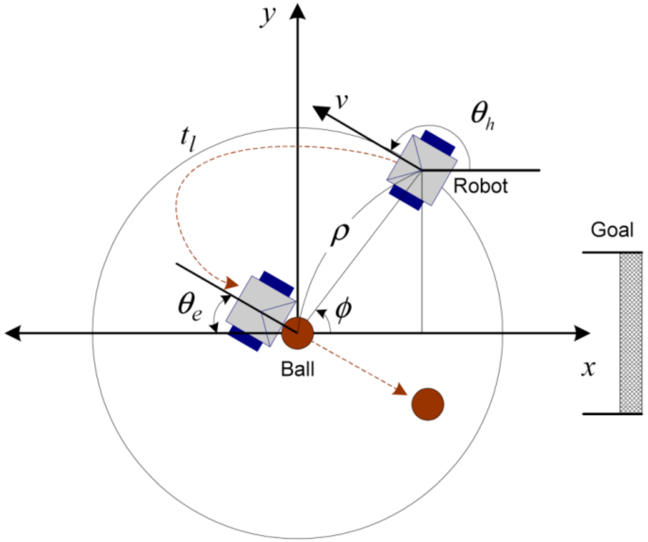
- ✓ **Robot should approach to the ball as soon as possible**
- ✓ **Robot should kick the ball to the goal accurately**

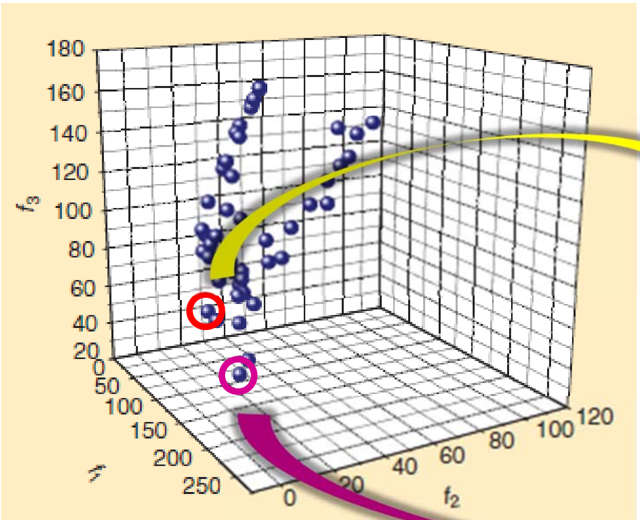
$$f_1 = K_t \cdot t_l$$ **(Elapsed time)**

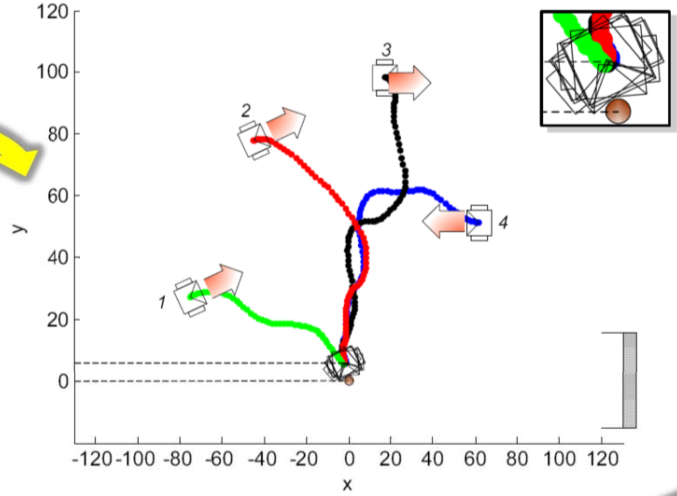
$$f_2 = K_\theta \cdot \theta_e$$ **(Heading direction error)**

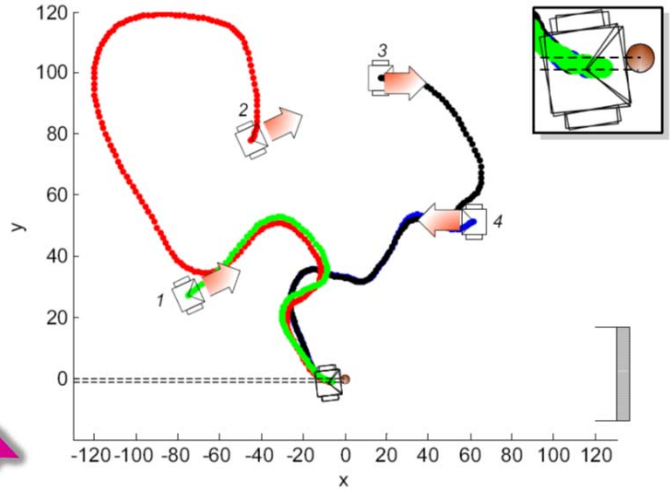
$$f_3 = K_\phi \cdot |\pi - \phi|$$ **(Posture angle error when kicking a ball)**

Dependencies exist!
Spending more time returns a more accurate position!

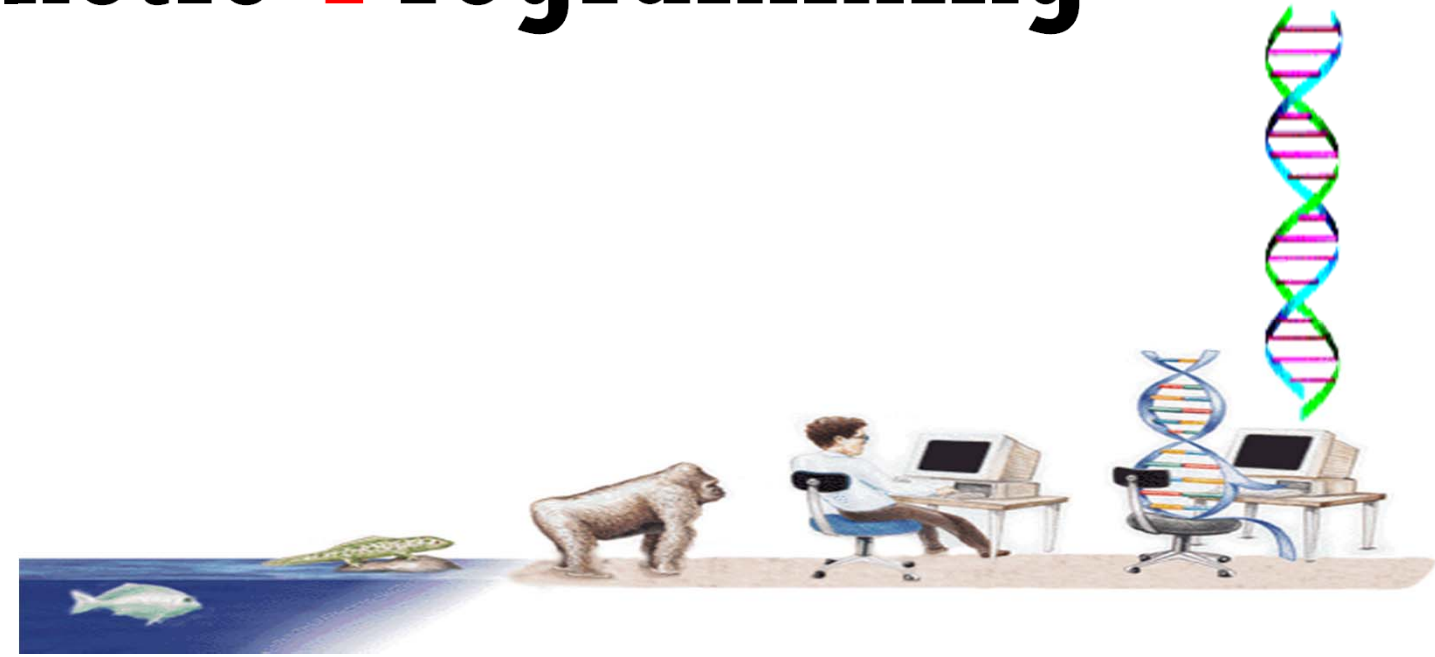








Genetic Programming



Genetic Programming (1)

❖ From “Manual” to “Automatic/Intelligent”

➢ The Challenge

“How can computers learn to solve problems without being explicitly programmed? In other words, how can computers be made to do what is needed to be done, without being told exactly how to do it?” by Arthur Samuel (1959)

➢ Criterion for Success

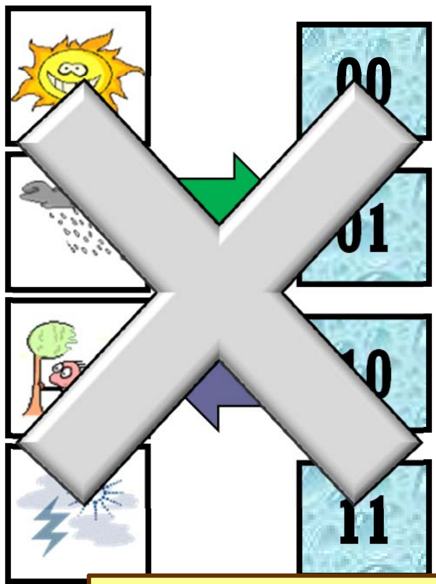
“ The aim is to get machines to exhibit behavior, which if done by humans, would be assumed to involve the use of intelligence.” by Arthur Samuel (1983)

Genetic Programming (1)

❖ A Computer Program in C

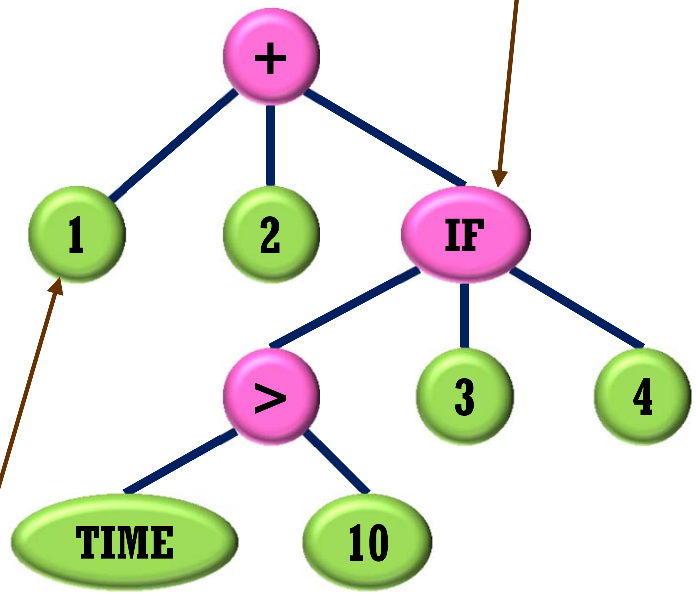
- ➢ Usually, C program codes are used for solving problem on computers
- ➢ But, numeric (e.g., binary) codes are used for doing tasks
- ➢ It is not flexible to solve problems on computers!
- ➢ Why not use the program codes directly?

```
int foo (int time)
{
   int temp1, temp2;
   if (time > 10)
       temp1 = 3;
   else
       temp1 = 4;
   temp2 = temp1 + 1 + 2;
   return (temp2);
}
```



Functional node:
{+, -, *, %, IF-ELSE}

(+ 1 2 (IF (> TIME 10) 3 4))

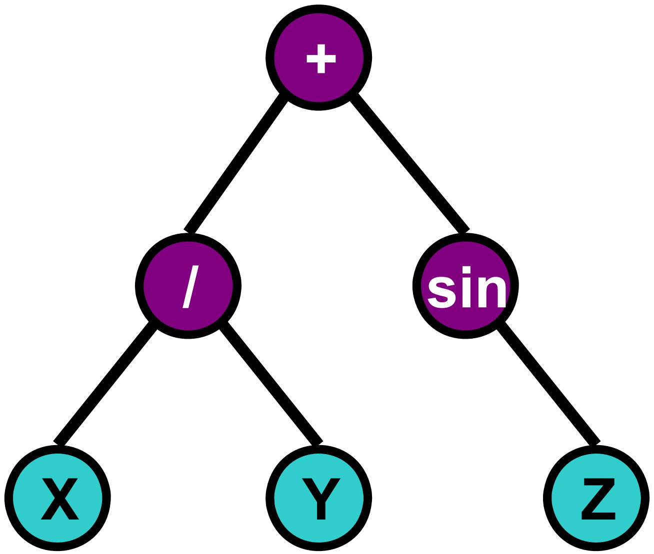


Terminal node:
{X, Y, ..., random}

Genetic Programming (2)

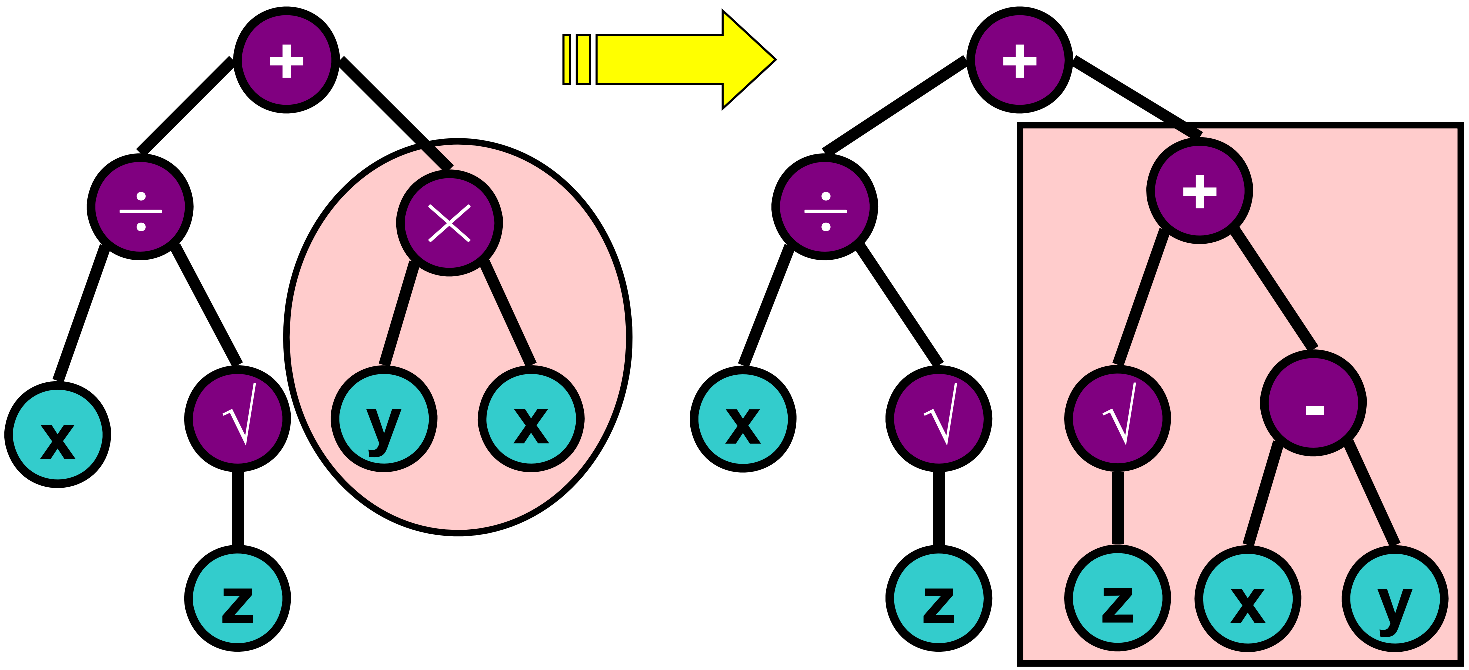
- **What is Genetic Programming (GP)?**
  - **Breeding computer programs** to resolve problems
  - Key principle is very similar to GAs; but something different
    1) Sameness: Evolutionary procedures by selection and variation
    2) **Difference**: **Representation** by tree and graph (**non-linear representation**)
       All possible computer programs can be encoded!

TREE REPRESENTATION



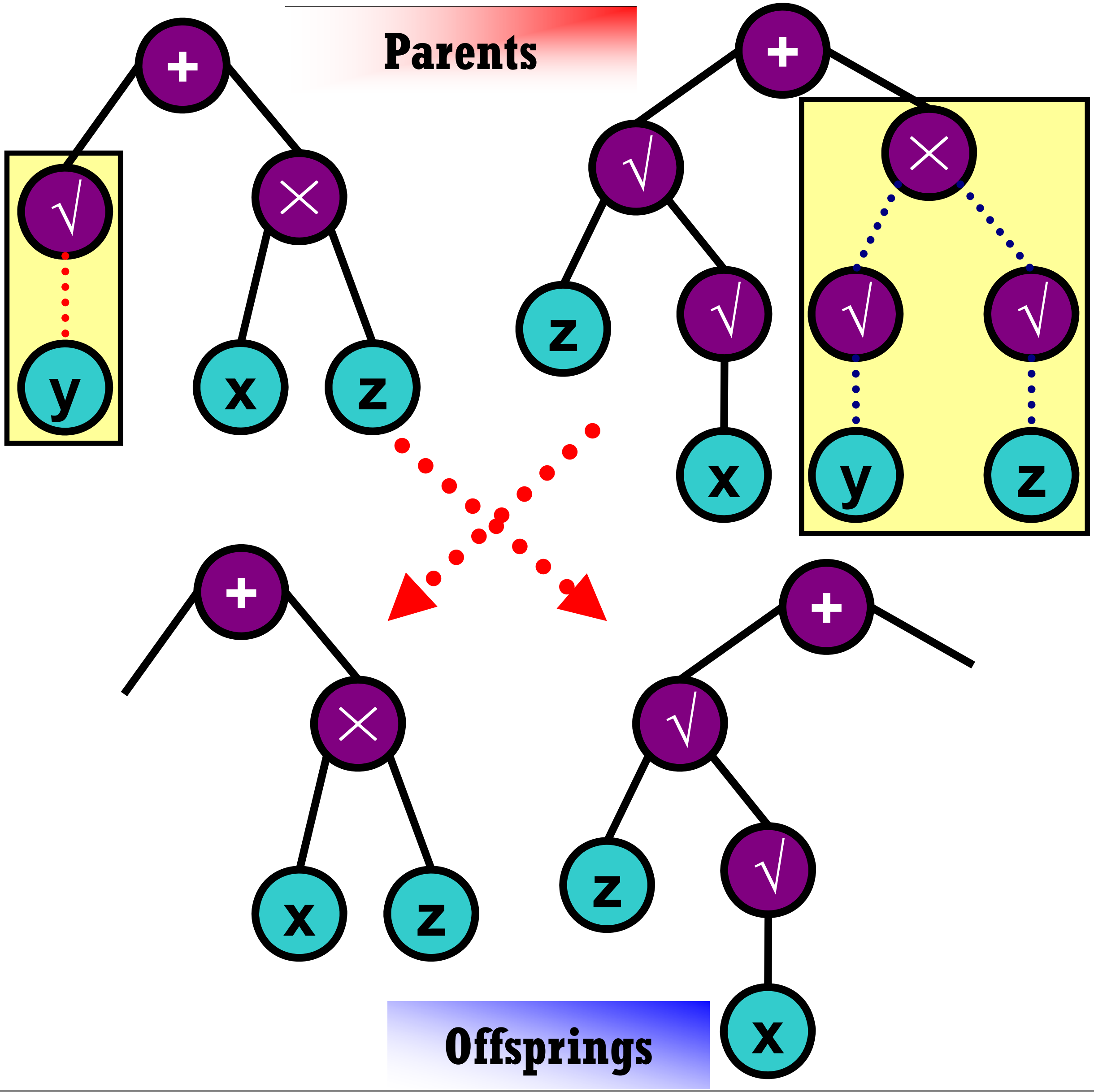
$f(x,y,z) = x/y + \sin(z)$

MUTATION

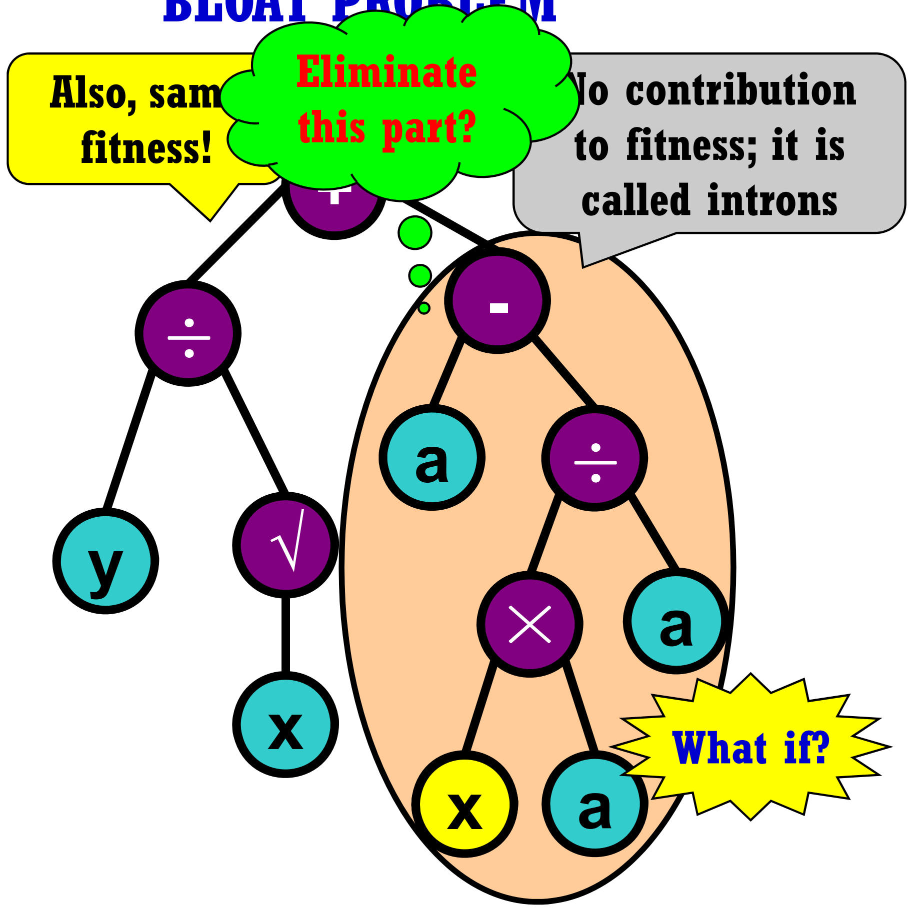


Genetic Programming (3)

CROSSOVER



BLOAT PROBLEM

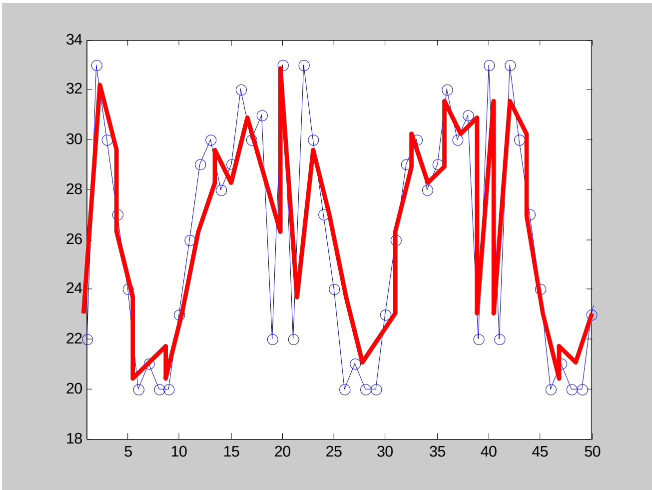


Problem: It is possible to grow rapidly in size over the course of a run while the fitness does not improve at all!

➔ Solving the problem is not easy.

GP-Based Prediction

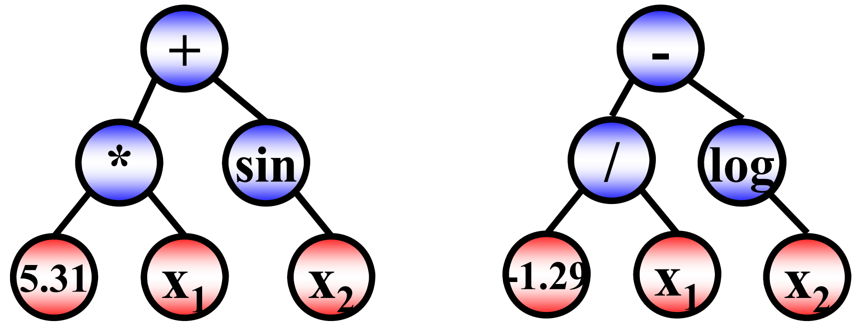
Time-Series Forecasting

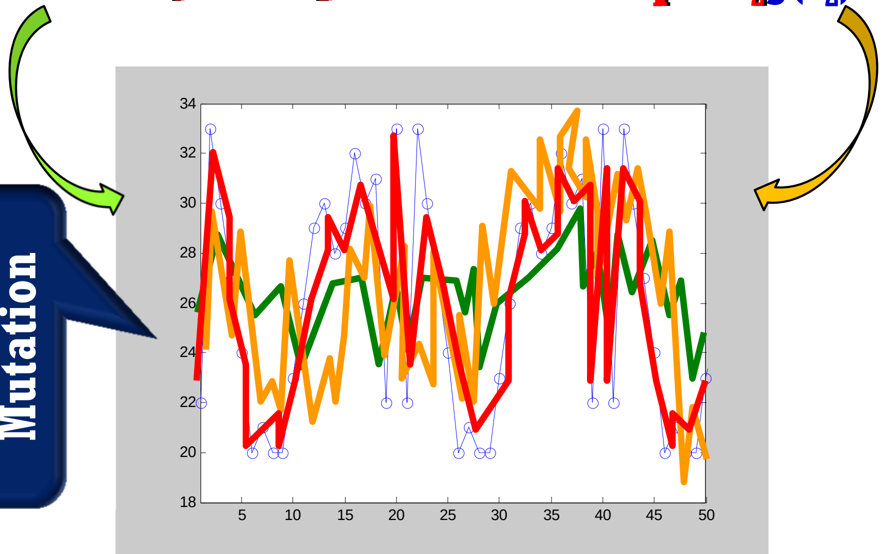


- Predicting some future outcomes from a set of historical events
- Stock prediction, Weather forecasting, etc.

GP Approach

- Using a nonlinear-type function

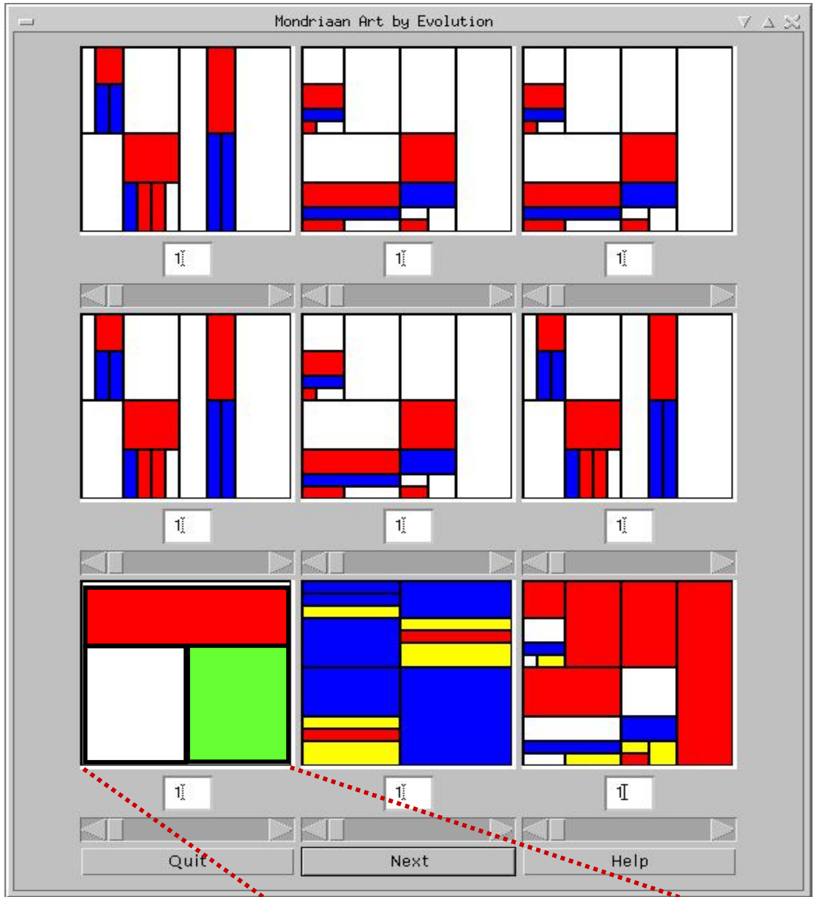


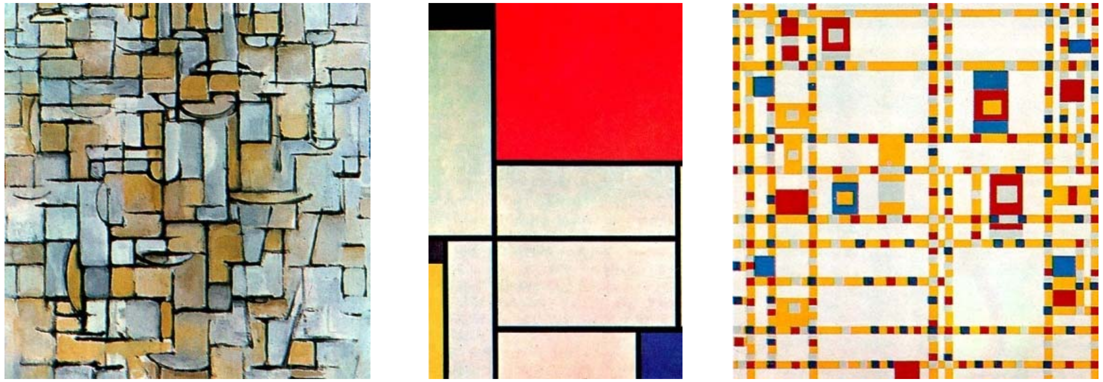


By Selection, Crossover, & Mutation

GP-Based Evolutionary Art

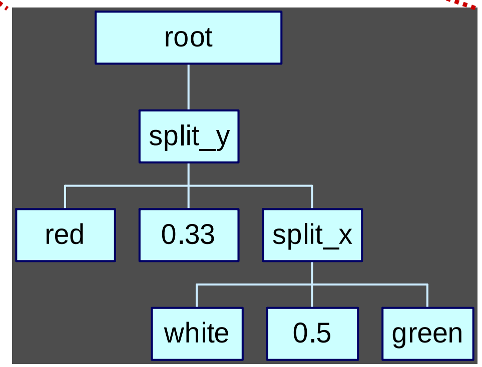
- **Mondriaan Evolver (Craenen et al.)**
  - GUI shows population of 9 pictures
  - User offers their grades
  - Computer performs one evolutionary cycle
    - ✓ Selection based on the fitness (thus, create mating pool)
    - ✓ Crossover & mutation (thus, create new population)
  - Repeat





Mondriaan's Arts

Mondriaan Representation



Summary

- **Multiobjective Evolutionary Algorithms (MOEAs)**
  - Generally, many problems consist of a set of conflicting objectives
  - Naturally, EAs deal with multiple solutions, and hence very suitable
  - MOEAs are superior to mathematical multiobjective schemes.
  - ➔ MOEAs are useful for resolving diverse problems in an attractive way.

- **Genetic Programming (GP)**
  - To develop the breeding computer programs for solving problems
  - Program codes are directly employed as data structure
  - The codes evolves in the concept of GAs.
  - ➔ GP is very flexible and powerful to deal with many real-world applications.