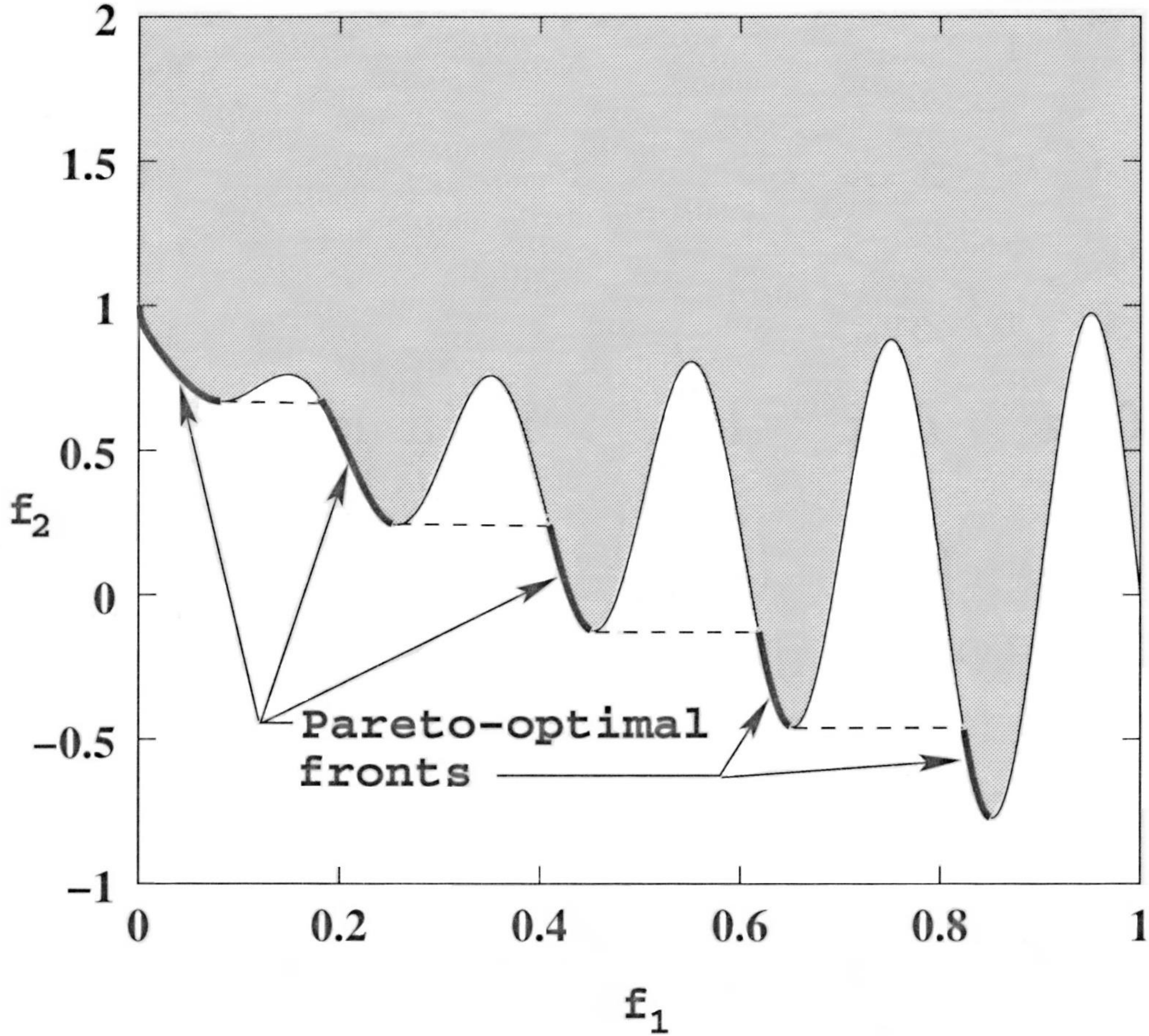
# Test functions

2. *n*=30 variable problem having a number of disconnected Pareto fronts (ZDT3)

$$
ZDT3:\begin{cases}
f_1(\boldsymbol{X}) = x_1 \\
f_2(\boldsymbol{X}) = g\cdot[1-\sqrt{f_1/g}-(f_1/g)\sin(10\pi f_1)] \\
g(\boldsymbol{X}) = 1+\dfrac{9}{n-1}\sum_{i=2}^{n} x_i
\end{cases}
$$

$$
0 \le x_i \le 1,\quad i = 1, 2, \cdots, n
$$



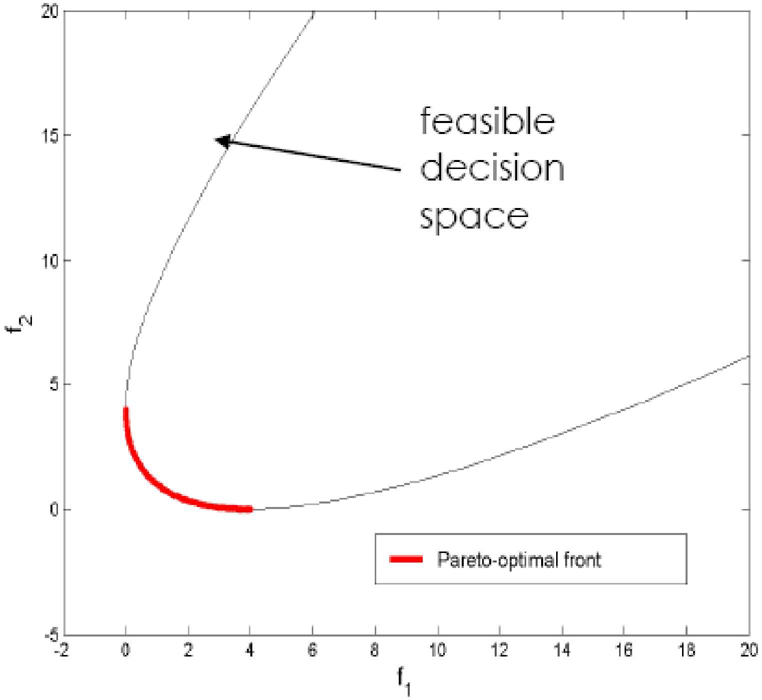
2
1.5
1
0.5
0
−0.5
−1
0
0.2
0.4
0.6
0.8
1
$f_2$
$f_1$
Pareto-optimal
fronts

# Schaffer's problem

# Problem Statement

- Schaffer's problem is a single variable problem with two objectives to be minimized

$$\min f_1(x) = x^2$$
$$\min f_2(x) = (x-2)^2$$
$$-10 < x < 10$$

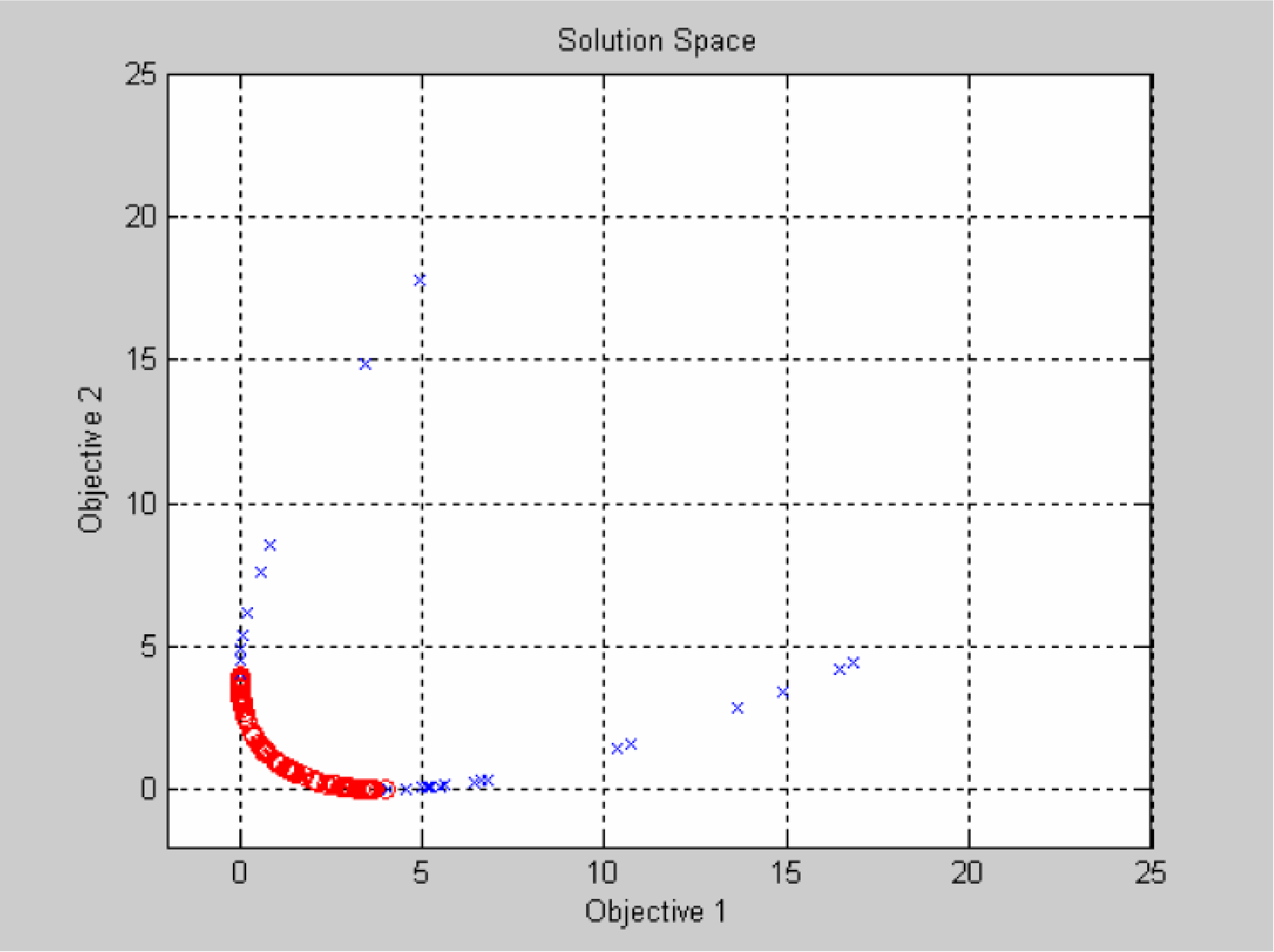



- The Schaffer's problem has Pareto optimal solutions $x \in [0, 2]$

# Fitness Function

- Two objectives have positive values and they are to be minimized
- The fitness function values are defined to be the negative of the objective function values

# Pareto Plot (20 Generations)



# Tanaka Problem

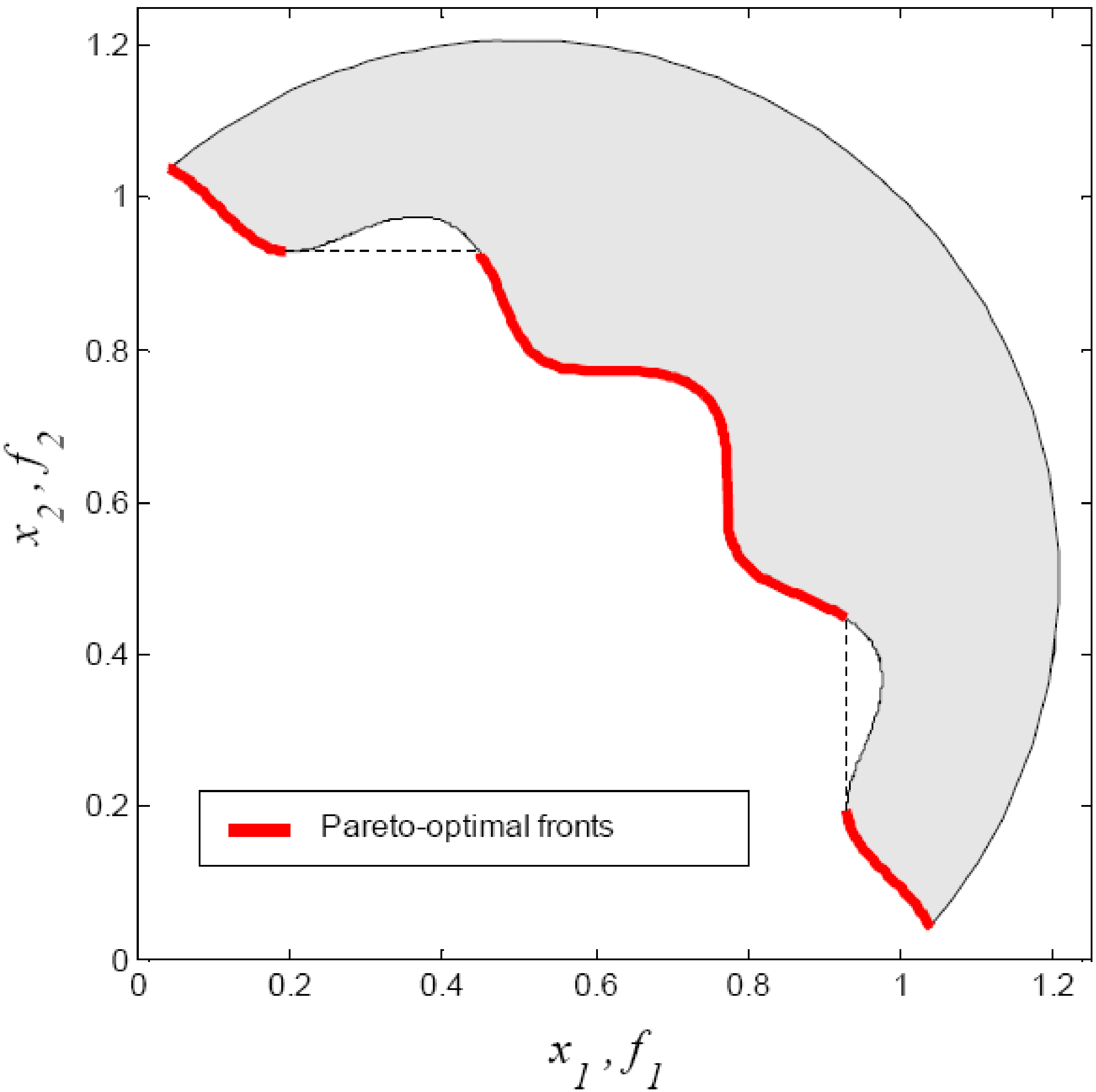
# Problem Statement

- Tanaka problem is a constrained optimization problem with two objectives to be minimized

$$\min f_1(x_1, x_2) = x_1$$
$$\min f_2(x_1, x_2) = x_2$$
$$\text{subject to } C_1(x_1, x_2) = x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\tfrac{x_1}{x_2}\right) \geq 0, \quad 0 \leq x_1 \leq \pi,$$
$$C_2(x_1, x_2) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \quad 0 \leq x_2 \leq \pi.$$

- Note the variable space is also the objective space

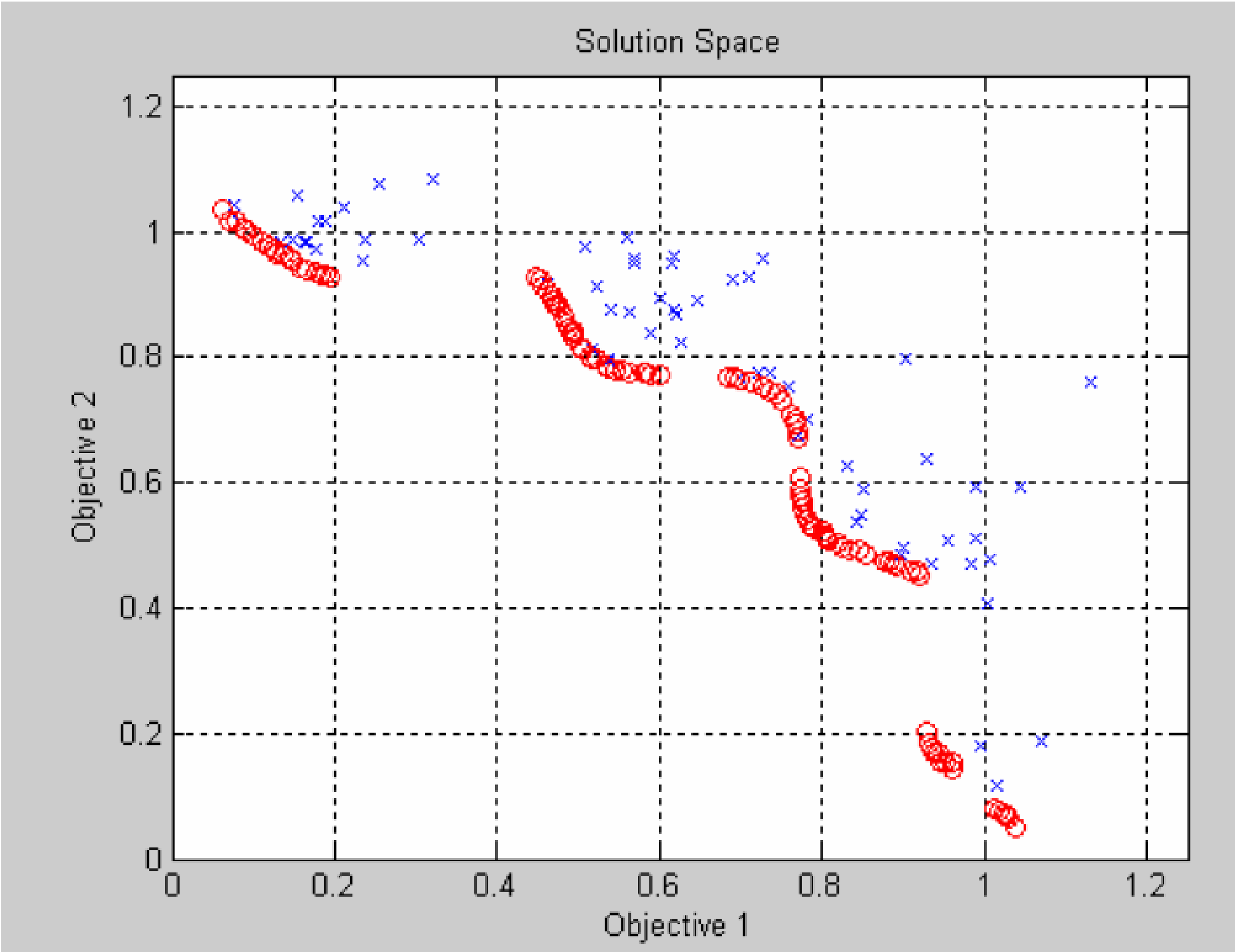
# Tanaka Problem



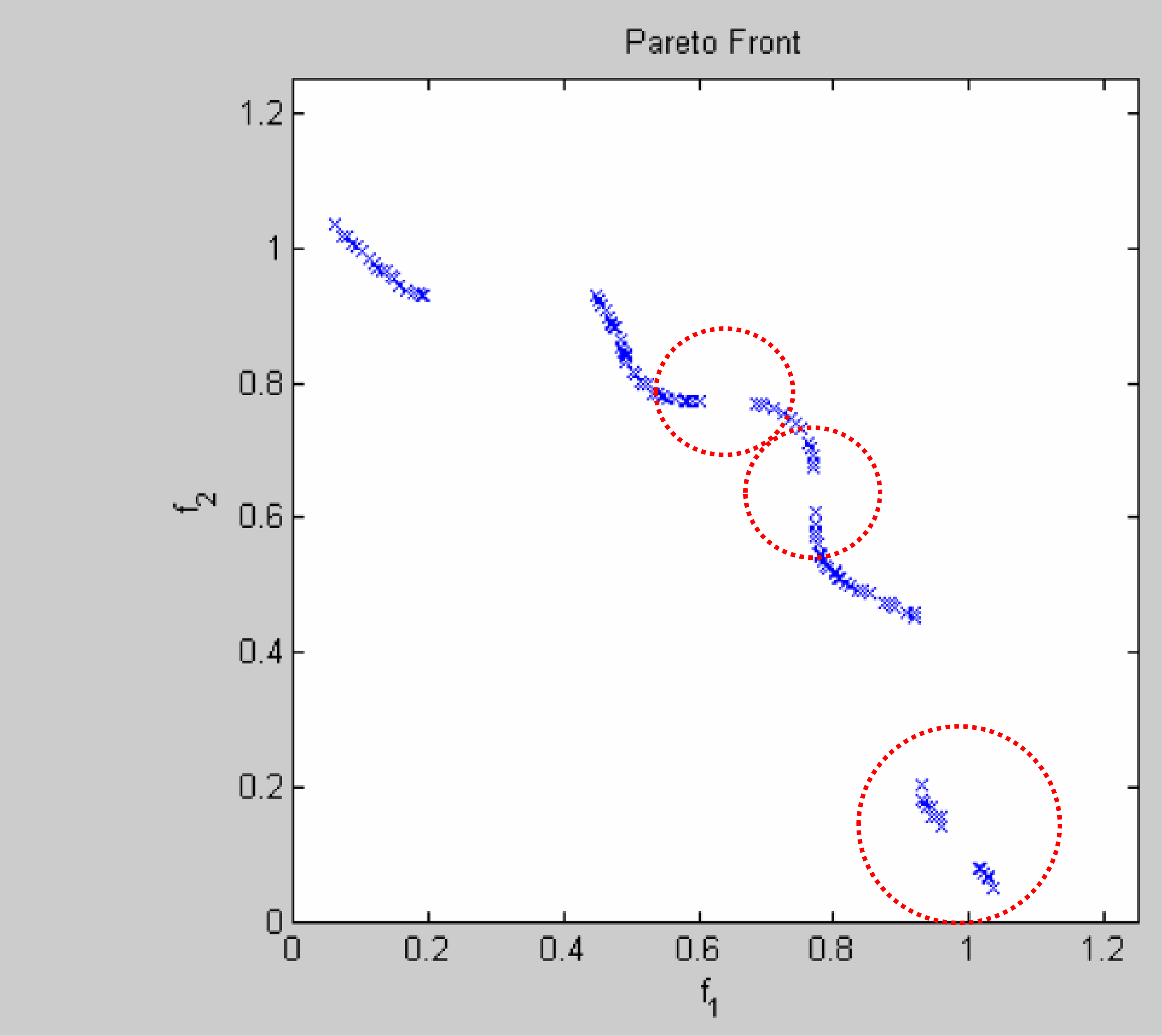
# Fitness Function

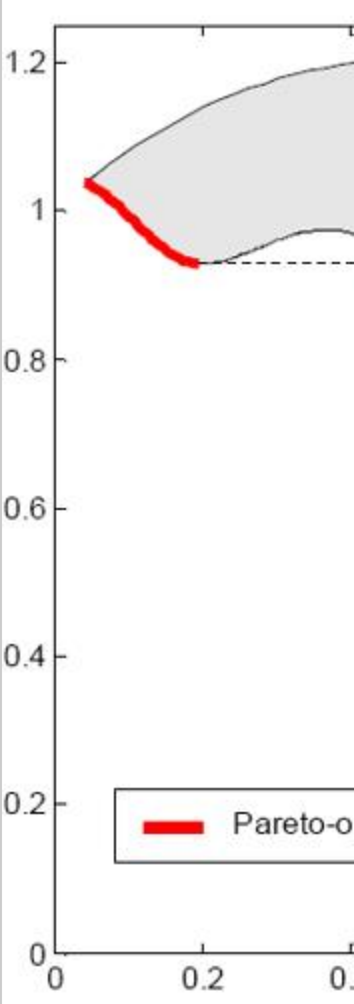
- Two objectives have positive values and they are to be minimized
- The fitness function values are defined to be the negative of the objective function values
- Infeasible solutions are assigned with -10 to reduce the chance of surviving

# Pareto Plot



# Non Dominated Solutions



## Now, it is your turn to finish Assignment #4

# Decision Problems

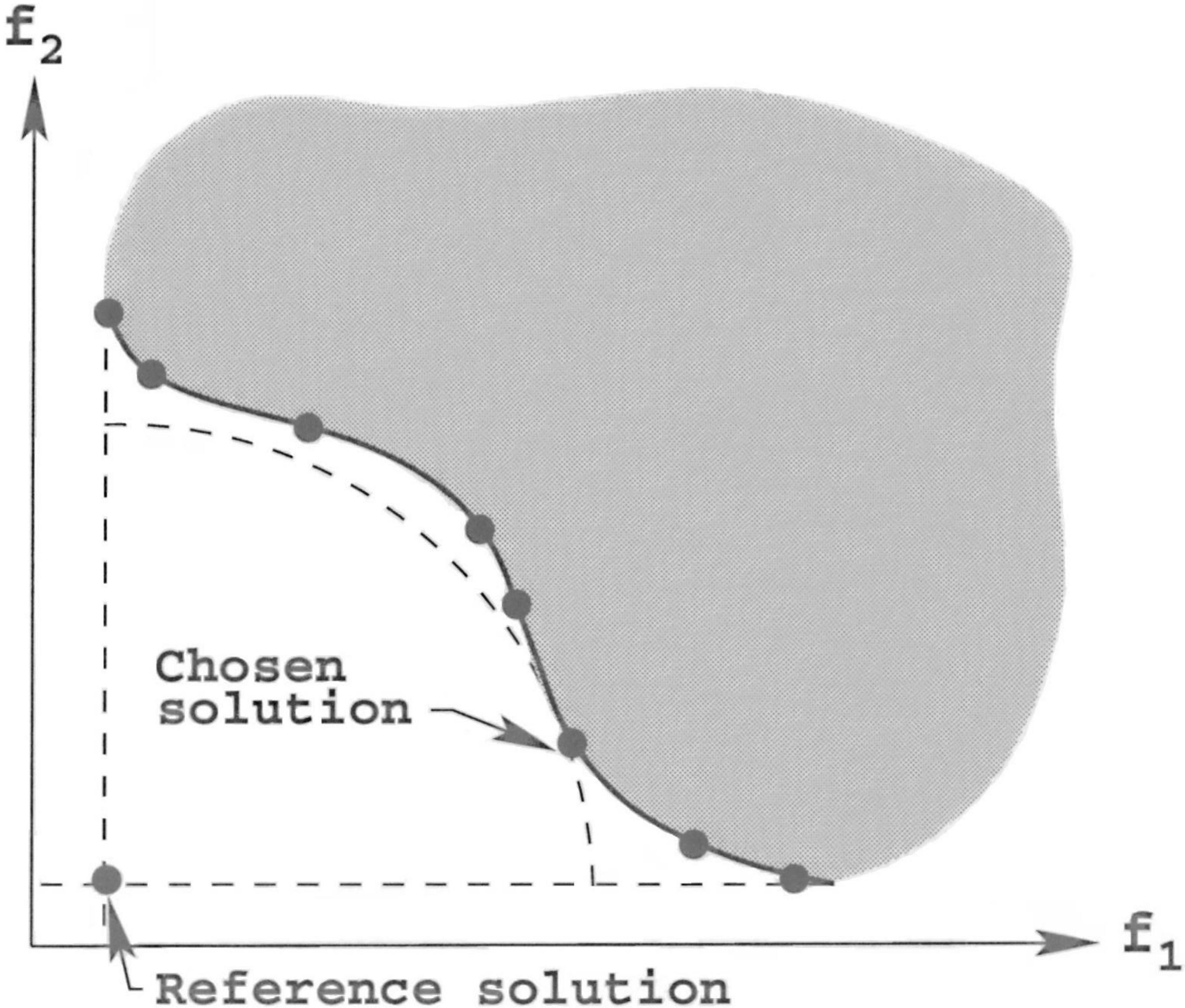
- How does one choose a particular solution from the obtained set of non-dominated solutions?
- **Optimization-level techniques**
  - the optimization technique is directed towards a preferred Pareto-optimal region during the optimization phase
- **Post-optimal techniques**
  - Compromise Programming Approach
  - Marginal Rate of Substitution Approach
  - Pseudo-Weight Vector approach

# Comprise programming

- ‘The method of global criteria’
- Pick a solution which is minimally located from a given reference point ***z*** (non-existent solution), by using a distance metric *d*

$$d(\boldsymbol{f},\boldsymbol{z}) = \left( \sum_{m=1}^{M} \left| f_m(\mathbf{x}) - z_m \right|^p \right)^{1/p}$$

$p$=2 ➔ $l_2$-metric ➔ Euclidean distance


$f_2$
Chosen solution
$f_1$
Reference solution

# Marginal Rate of Substitution Approach

- The amount of improvement in one objective function which can be obtained by sacrificing an unit decrement in any other objective function.

- The solution having the maximum marginal rate of substitution is the one chosen by the method.

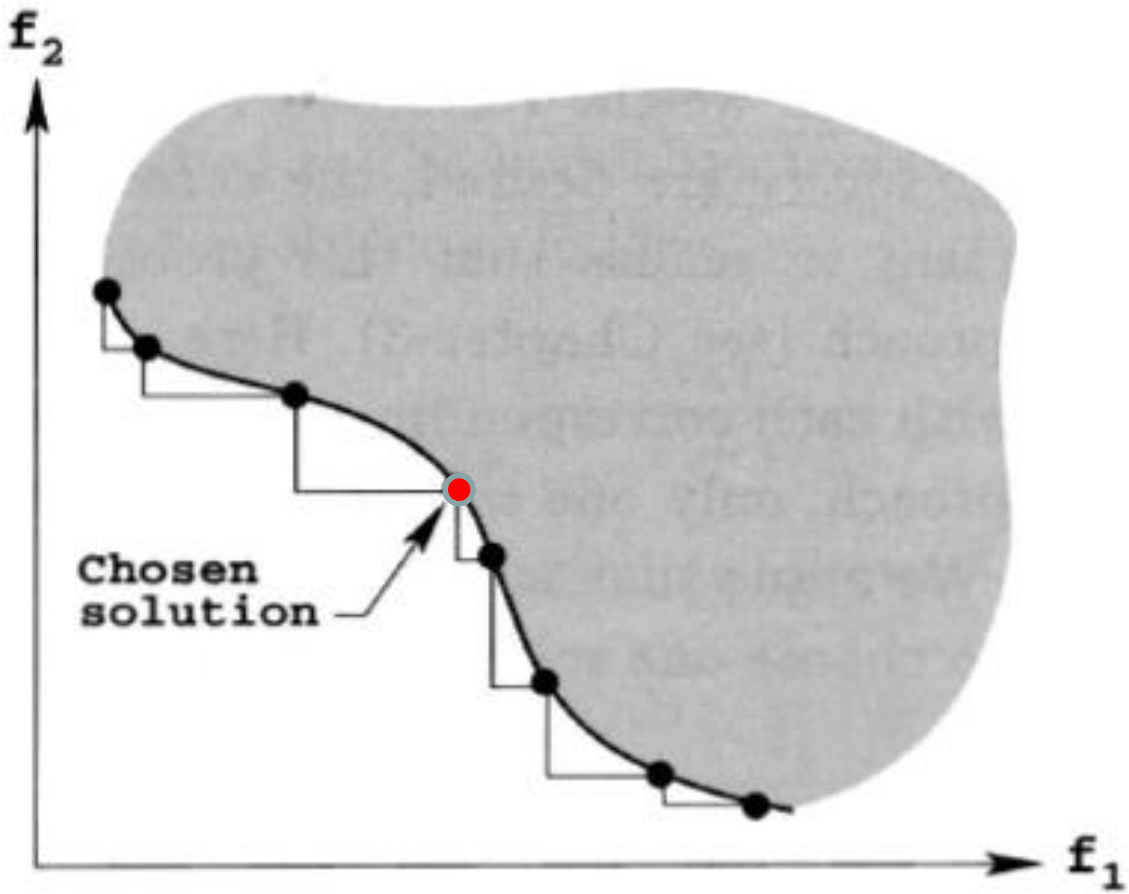
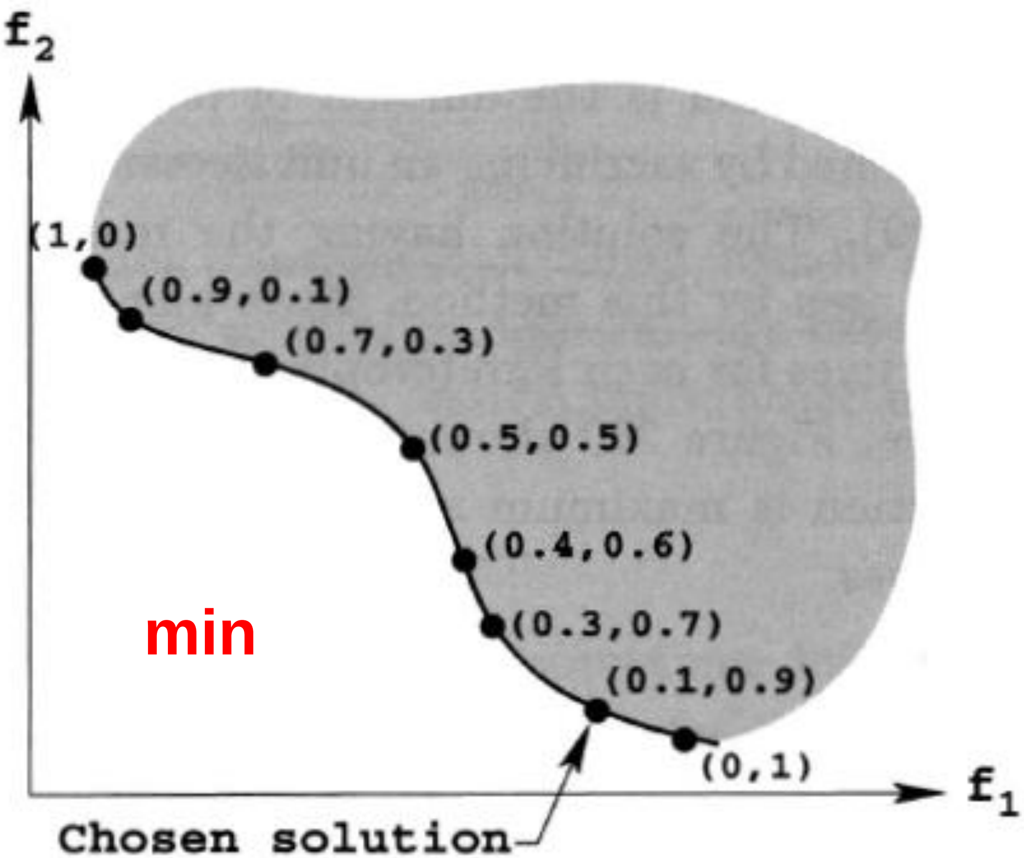



Fig. 253 on p. 377

# Pseudo-Weight Vector approach

$$w_i = \frac{\left(f_i^{max} - f_i(\mathbf{x})\right) / \left(f_i^{max} - f_i^{min}\right)}{\sum_{m=1}^{M} \left(f_m^{max} - f_m(\mathbf{x})\right) / \left(f_m^{max} - f_m^{min}\right)}$$

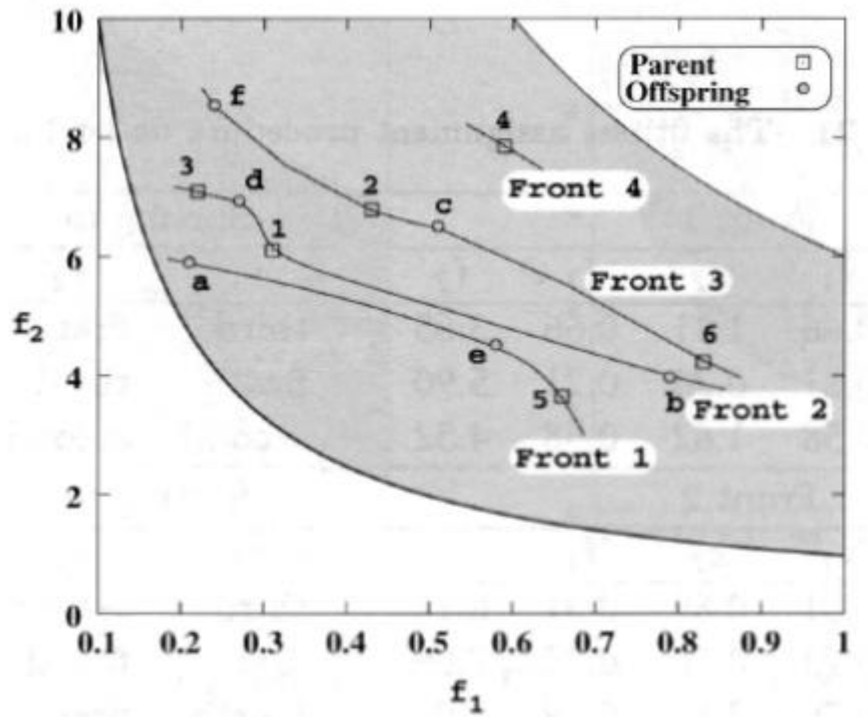



- Pseudo-weight vector $w_i$ for the $i$th objective function $f_i$
- Compute a relative trade-off value between objectives for all obtained non-dominated solutions
- Choose the solution closer to a user-preferred weight vector.
- Similar to the method used in weighted sum method.

# Hand Calculation

$$w_i = \frac{\left(f_i^{max} - f_i(\mathbf{x})\right) / \left(f_i^{max} - f_i^{min}\right)}{\sum_{m=1}^{M} \left(f_m^{max} - f_m(\mathbf{x})\right) / \left(f_m^{max} - f_m^{min}\right)}$$

| Front 2 | | | | |
|---|---|---|---|---|
| Solution | $x_1$ | $x_2$ | $f_1$ | $f_2$ |
| 1 | 0.31 | 0.89 | 0.31 | 6.10 |
| 3 | 0.22 | 0.56 | 0.22 | 7.09 |
| b | 0.79 | 2.14 | 0.79 | 3.97 |
| d | 0.27 | 0.87 | 0.27 | 6.93 |

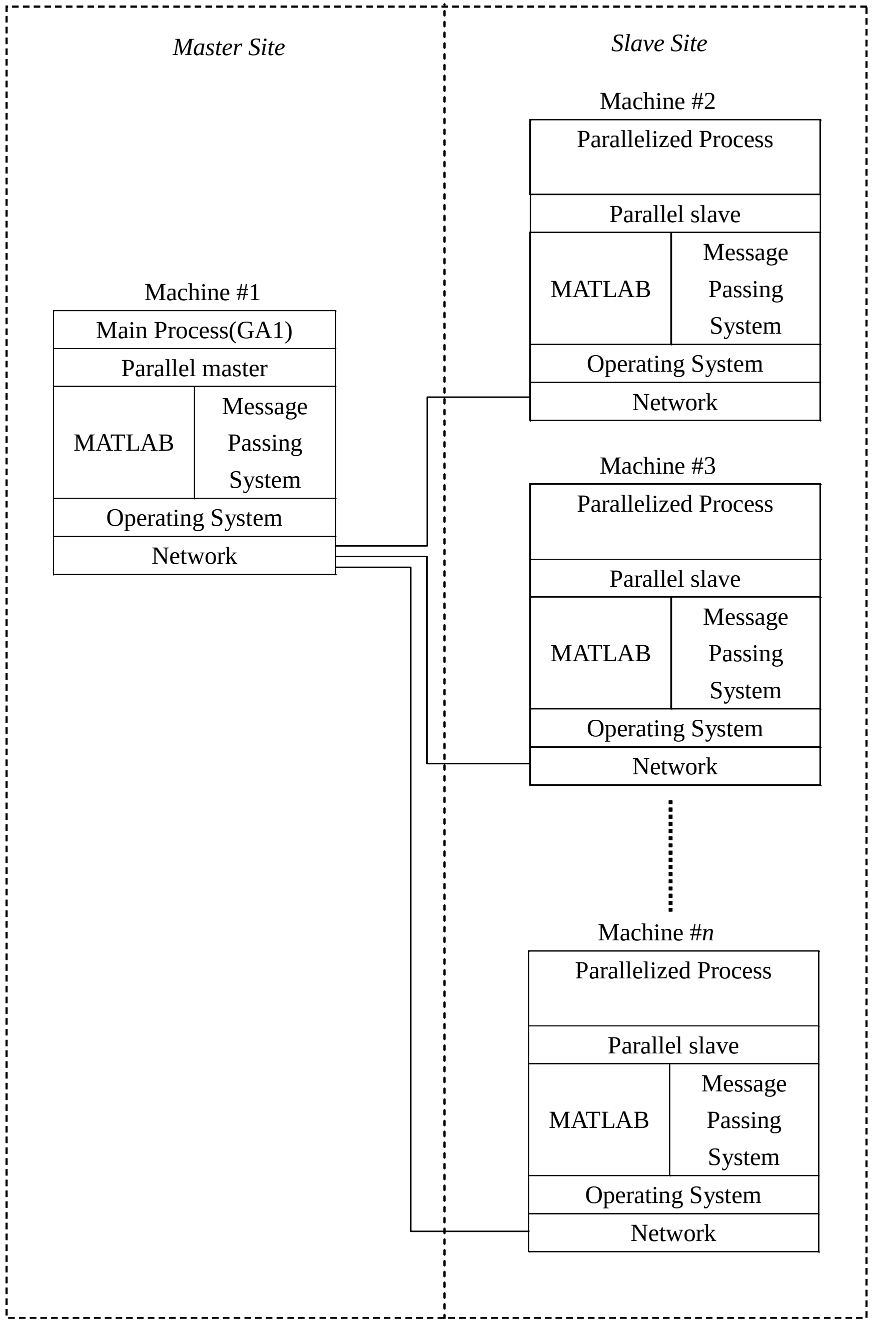


# Performance Improvements

- Exploitation v.s. Exploration
- Diversity v.s. Convergence
- Intrinsic parallel nature
  - Parallel computation
- Hybrid with other search methods (memetic algorithms)
  - GA+NM simplex search
- Hardware implementation
  - SOPC/SOC

# Parallel Computation of GA

- Computational scheme of parallel applications using multiple MATLAB processes
- Master-slave scheme
- TCP/IP protocol
- message passing system
- Matlab 5.X/6.X/7.X
- parmatlab / tcpudpip



*Master Site*
*Slave Site*
Machine #1
Main Process(GA1)
Parallel master
MATLAB
Message Passing System
Operating System
Network
Machine #2
Parallelized Process
Parallel slave
MATLAB
Message Passing System
Operating System
Network
Machine #3
Parallelized Process
Parallel slave
MATLAB
Message Passing System
Operating System
Network
Machine #*n*
Parallelized Process
Parallel slave
MATLAB
Message Passing System
Operating System
Network

# Part II: Applications

| Domain | Application Types |
|---|---|
| **Control** | gas pipeline, pole balancing, missile evasion, pursuit |
| **Design** | semiconductor layout, aircraft design, keyboard configuration, communication networks |
| **Scheduling** | manufacturing, facility scheduling, resource allocation |
| **Robotics** | trajectory planning |
| **Machine Learning** | designing neural networks, improving classification algorithms, classifier systems |
| **Signal Processing** | filter design |
| **Game Playing** | poker, checkers, prisoner's dilemma |
| **Combinatorial Optimization** | set covering, travelling salesman, routing, bin packing, graph colouring and partitioning |

# Constraints Handling

# Part II: Control Applications

- **Model Reduction of Discrete Interval Systems**
- **Tolerance Design of Robust Controllers for Uncertain Interval Systems**
- **Multi-objective Evolutionary Approach to the Design of Optimal Controllers for Interval Plants**
- **Robust control of interval plants**

- I-Hsum Li, Wei-Yen Wang, Chung-Ying Li, Jia-Zwei Kao, and Chen-Chien Hsu*, "Cloud-Based Improved Monte Carlo Localization Algorithm with Robust Orientation Estimation for Mobile Robots," Engineering Computations, Vol. 36, no. 1, pp. 178-203, Feb., 2019.
- Chen-Chien Hsu*, Wei-Yen Wang, Yi-Hsing Chien, and Ru-Yu Hou, "FPGA Implementation of Improved Ant Colony Optimization Algorithm Based on Pheromone Diffusion Mechanism for Path Planning," Journal of Marine Science and Technology, Vol. 26, no. 2, pp. 170-179, 2018.
- Chiang-Heng Chien, Wei-Yen Wang, and Chen-Chien Hsu*, "Multi-Objective Evolutionary Approach to Prevent Premature Convergence in Monte Carlo Localization," Applied Soft Computing, Vol. 50, pp. 260-279, Jan. 2017.
- Chiang-Heng Chien, Wei-Yen Wang, Jun Jo, and Chen-Chien Hsu*, "Enhanced Monte Carlo Localization Incorporating a Mechanism for Preventing Premature Convergence," Robotica, Vol. 35, no. 7, pp. 1504-1522, July, 2017.
- Nai-Jen Li, Wen-June Wang, and Chen-Chien Hsu, "Hybrid Particle Swarm Optimization Incorporating Fuzzy Reasoning and Weighted Particle," Neurocomputing, Vol. 167, pp. 488-501, April, 2015.
- Nai-Jen Li, Wen-June Wang*, Chen-Chien Hsu, Wei Chang, Hao-Gong Chou, and Jun-Wei Chang, "Enhanced Particle Swarm Optimizer Incorporating a Weighted Particle," Neurocomputing, Vol. 124, pp. 218-227, Jan., 2014.

- Shih-An Li, Chen-Chien Hsu*, Wen-Ling Lin, and Jui-Pin Wang, “Object Tracking Based on Hardware/Software Co-design of Particle Filter,” Advanced Science Letters, Vol. 13, no. 1, pp. 442-446, June, 2012.
- Chen-Chien Hsu*, Shu-Han Chu, and Yi-Hsing Chien, “Intelligent Digital Redesign of T-S Fuzzy Controller via Particle Swarm Optimization,” International Journal of Computational Intelligence in Control, Vol.3 no.2, pp. 49-61, 2011.
- Chen-Chien Hsu* and Tsung-Chi Lu, “Suitability of Redesigned Digital Control Systems Having an Interval Plant via an Evolutionary Approach,” ASME Journal of Dynamic Systems, Measurement and Control, Vol. 133, no. 4, pp. 041007, July, 2011.
- Chen-Chien Hsu*, Ching-Chang Wong, Hung-Chih Teng, and Cheng-Yao Ho, “Localization of Mobile Robots via an Enhanced Particle Filter Incorporating Tournament Selection and Nelder-Mead Simplex Search,” International Journal of Innovative Computing, Information and Control, Vol. 7, no. 7A, pp. 3725-3737, July, 2011.
- Shih-An Li, Chen-Chien Hsu*, Ching-Chang Wong, and Chia-Jun Yu, “Hardware/Software Co-design for Particle Swarm Optimization Algorithm,” Information Sciences, Vol. 181, no. 20, pp. 4582–4596, Oct., 2011.
- Chen-Chien Hsu*, Wern-Yarng Shieh, and Chun-Hwei Kao, “Digital redesign of uncertain interval systems based on extremal gain/phase margins via a hybrid particle swarm optimizer,” Applied Soft Computing, Vol. 10, no. 2, pp. 602-612, March, 2010.

# Interval Plant

- Interval plant:

$$G(s,\boldsymbol{a},\boldsymbol{b}) = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1} s^{n-1}}{a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1} s^{n-1} + s^n} = \frac{\hat{N}(s)}{\hat{D}(s)}$$

$$\boldsymbol{a} = (a_0, a_1, a_2, \ldots, a_{n-1}), \quad \boldsymbol{b} = (b_0, b_1, b_2, \ldots, b_{n-1})$$

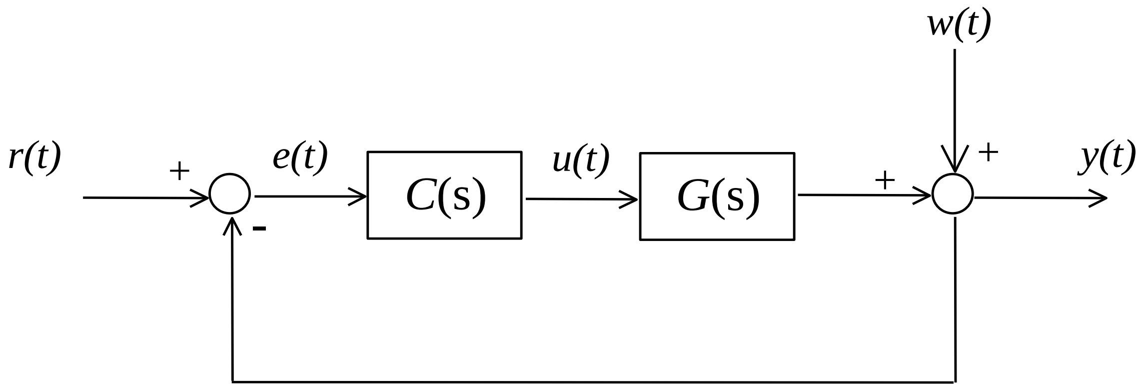
$$\boldsymbol{A} = \{\boldsymbol{a} : a_i \in [a_i^-, a_i^+], \forall i = 0,1,2,\cdots,n-1\},$$

$$\boldsymbol{B} = \{\boldsymbol{b} : b_i \in [b_i^-, b_i^+], \forall i = 0,1,2,\cdots,n-1\}$$

- Controller

$$C(s,\boldsymbol{p},\boldsymbol{q}) = \frac{q_0 + q_1 s + q_2 s^2 + \cdots + q_m s^m}{p_0 + p_1 s + p_2 s^2 + \cdots + p_m s^m} \equiv \frac{q(s)}{p(s)}$$

# Problem Formulation




$$G_{cl}(s,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b}) = \frac{C(s,\boldsymbol{p},\boldsymbol{q})G(s,\boldsymbol{a},\boldsymbol{b})}{1+C(s,\boldsymbol{p},\boldsymbol{q})G(s,\boldsymbol{a},\boldsymbol{b})} = \frac{q(s)\hat{N}(s)}{p(s)\hat{D}(s)+q(s)\hat{N}(s)}$$

$$\equiv \frac{N(s,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})}{D(s,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})}$$

**Design objectives:**

- **Integral of Squared Error (ISE)**
- **Disturbance rejection**

# Stability Test

- Robust stability

$$D(s,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b}) = p(s)\hat{D}(s) + q(s)\hat{N}(s)$$

- Generalized Kharitonov segment polynomials

$$\begin{aligned}\Delta_E(s,\lambda) \equiv & \left\{N_i(s)q(s) + \left((1-\lambda)D_j(s) + \lambda D_k(s)\right)p(s)\right\} \\ & \cup \\ & \left\{\left((1-\lambda)N_j(s) + \lambda N_k(s)\right)q(s) + D_i(s)p(s)\right\}\end{aligned}$$

where

$$i \in \{1,2,3,4\}$$

$$(j,k) \in \{(1,2),(1,3),(2,3),(3,4)\}$$

$$\lambda \in [0,1]$$

- **Integral of Squared Error (ISE)**

$$J_1(\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b}) = \int_0^{\infty} e^2(t,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})dt$$

$$J_1(\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b}) = \|e(t)\|^2 = \int_0^{\infty} e^2(t,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})dt = \sum_{l=1}^{n} \frac{\left(\beta_l(\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})\right)^2}{2\alpha_l(\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})}$$

$$\min_{\substack{\boldsymbol{p}\in\boldsymbol{P}\\ \boldsymbol{q}\in\boldsymbol{Q}}} \max_{\substack{\boldsymbol{a}\in\boldsymbol{A}\\ \boldsymbol{b}\in\boldsymbol{B}}} J_1(\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})$$

Subject to $\Delta_E(s,\lambda)$ are Hurwitz stable.

- **Disturbance rejection**

$$\max_{d(t)\in L_2} \frac{\|y\|_2}{\|d\|_2} = \left\| \frac{1}{1+C(s,\boldsymbol{p},\boldsymbol{q})G(s,\boldsymbol{a},\boldsymbol{b})} \right\|_{\infty}$$

$$\left\| \frac{1}{1+C(s,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}})G(s,\boldsymbol{a},\boldsymbol{b})} \right\|_{\infty} = \max_{\substack{\omega\in[0,\infty)\\ \boldsymbol{a}\in\boldsymbol{A}\\ \boldsymbol{b}\in\boldsymbol{B}}} \left( \frac{1}{(1+C(j\omega,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}})G(j\omega,\boldsymbol{a},\boldsymbol{b}))(1+C(-j\omega,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}})G(-j\omega,\boldsymbol{a},\boldsymbol{b}))} \right)^{0.5}$$

$$= \max_{\substack{\omega\in[0,\infty)\\ \boldsymbol{a}\in\boldsymbol{A}\\ \boldsymbol{b}\in\boldsymbol{B}}} \left(\alpha(\omega,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b})\right)^{0.5}$$

$$\min_{\substack{\boldsymbol{p}\in\boldsymbol{P}\\ \boldsymbol{q}\in\boldsymbol{Q}}} \max_{\substack{\omega\in[0,\infty)\\ \boldsymbol{a}\in\boldsymbol{A}\\ \boldsymbol{b}\in\boldsymbol{B}}} J_2(\omega,\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b})$$

Subject to $\Delta_E(s,\lambda)$ are Hurwitz stable.

# Constrained multi-objective minimax optimization problem

## Design problem:

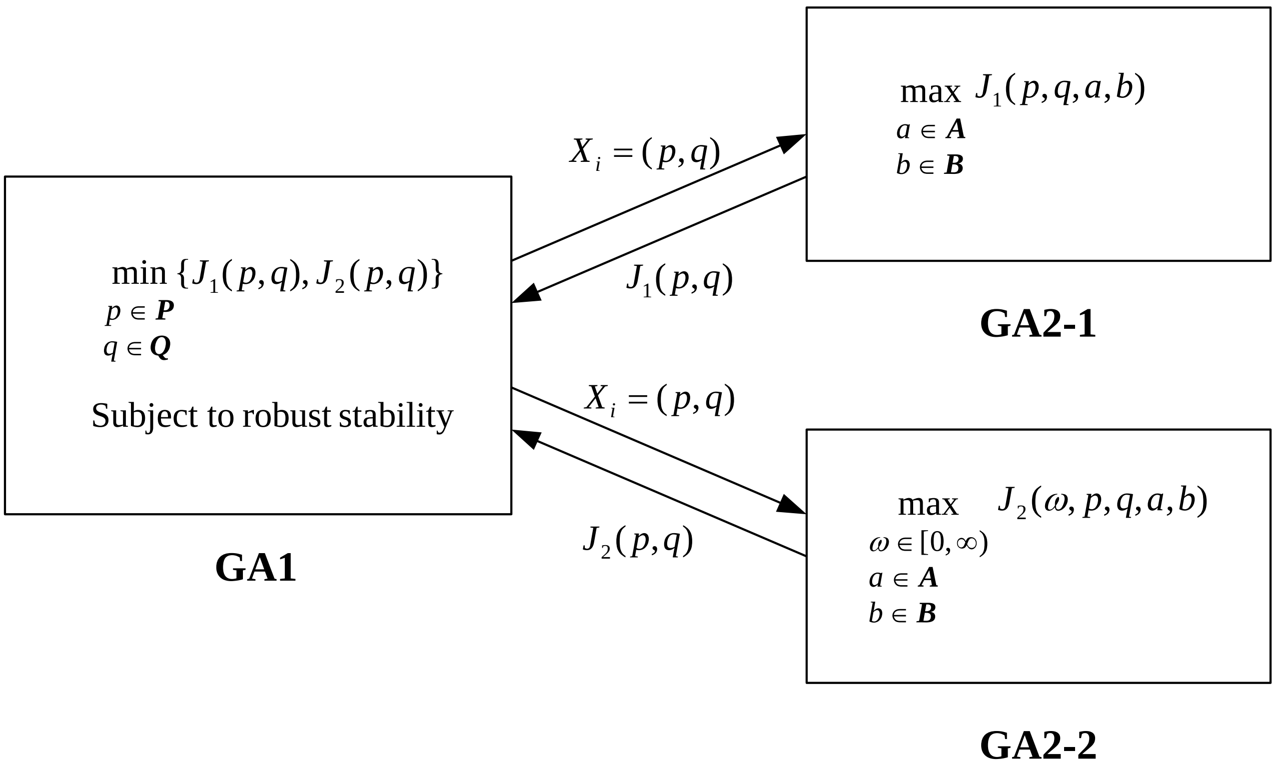
$$\min_{\substack{\boldsymbol{p}\in\boldsymbol{P}\\ \boldsymbol{q}\in\boldsymbol{Q}}} \max_{\substack{\boldsymbol{a}\in\boldsymbol{A}\\ \boldsymbol{b}\in\boldsymbol{B}}} J_h(\boldsymbol{p},\boldsymbol{q},\boldsymbol{a},\boldsymbol{b}),\ \ h=\{1,2\}$$

$$\text{Subject to} \quad \Delta_E(s,\lambda)\ \text{are Hurwitz stable.}$$

where

$$\Delta_E(s,\lambda) \equiv \left\{N_i(s)q(s) + \left(\left(1-\lambda\right)D_j(s) + \lambda D_k(s)\right)p(s)\right\} \cup \left\{\left(\left(1-\lambda\right)N_j(s) + \lambda N_k(s)\right)q(s) + D_i(s)p(s)\right\}$$

# Multi-objective evolutionary scheme



- Representation of solutions

$$\boldsymbol{X_i} = (\boldsymbol{p},\boldsymbol{q}) = \begin{bmatrix} p_0 & p_1 & \cdots & p_m & q_0 & q_1 & \cdots & q_m \end{bmatrix}$$

$$p_j \in \left[p_j^{\,-}, p_j^{\,+}\right] \;,\; q_j \in \left[q_j^{\,-}, q_j^{\,+}\right] \;,\; j = 0,1,2,\cdots,m.$$

- Fitness functions

$$F_{2-1}(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b}) = J_1(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b}) = \int_0^\infty e^2(t,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b})dt$$

$$F_{2-2}(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b}) = J_2(\omega,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b}) = \alpha(\omega,\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\boldsymbol{a},\boldsymbol{b})^{0.5}$$

$$F_{2-1}(p,q) = \begin{cases} J_1, & \text{if } X_i = (p,q) \text{ is feasible} \\ \infty, & \text{if } X_i = (p,q) \text{ is infeasible} \end{cases}$$

$$F_{2-2}(p,q) = \begin{cases} J_2, & \text{if } X_i = (p,q) \text{ is feasible} \\ \infty, & \text{if } X_i = (p,q) \text{ is infeasible} \end{cases}$$

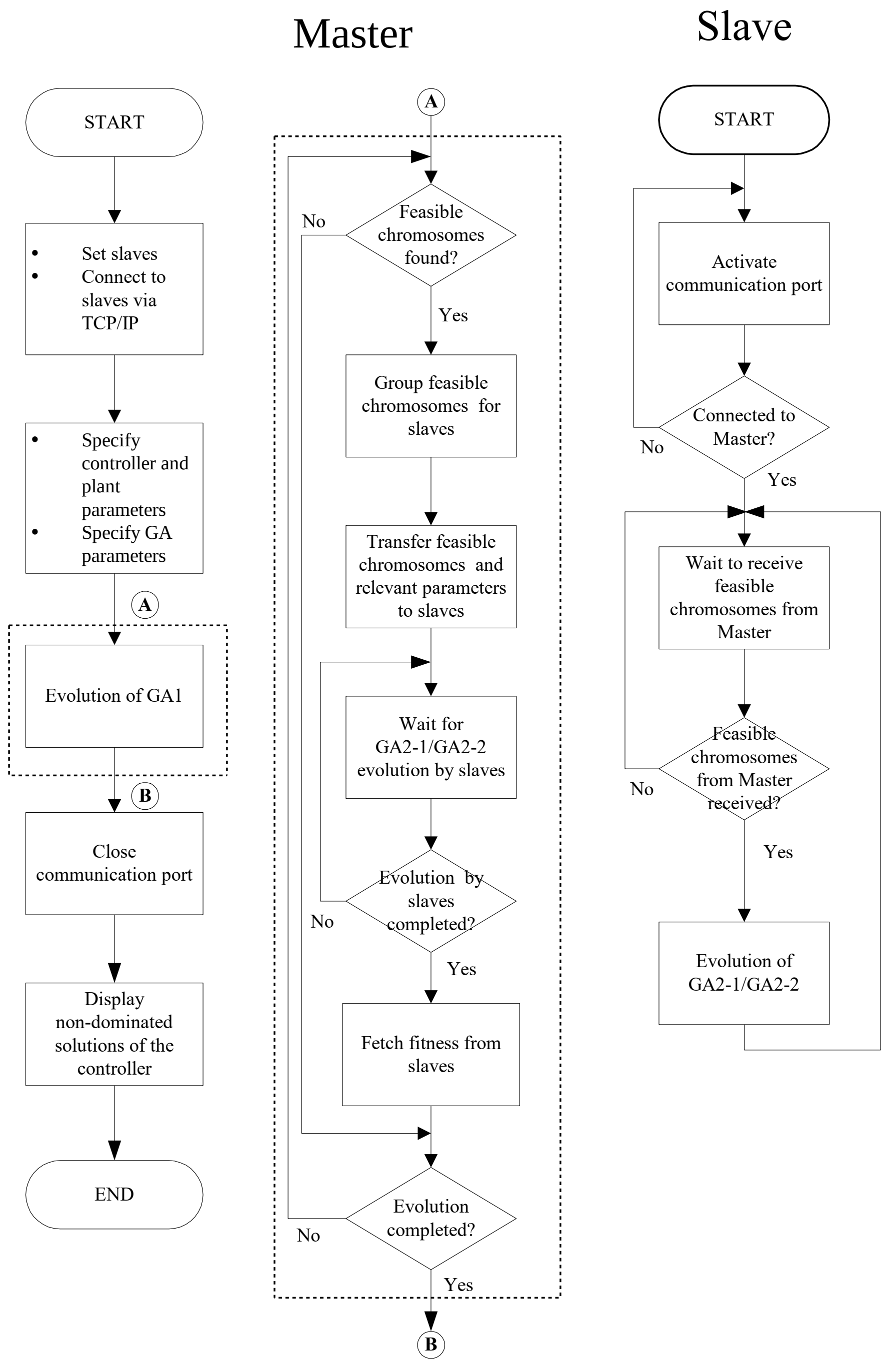
$$F_3(p,q) = \begin{cases} 0, & \text{if } X_i = (p,q) \text{ is feasible} \\ \phi, & \text{if } X_i = (p,q) \text{ is infeasible} \end{cases}$$

# Penalty function

$$\phi(\boldsymbol{p},\boldsymbol{q}) = \sum_{i=1}^{N}\sum_{j=1}^{32}\sum_{k=1}^{v} C_{jk}, \quad v \text{ is the order of the characteristic polynomial } D(s)$$

$$C_{jk} = \begin{cases} 0, & \text{if } r_{jk} < 0 \\ m_1, & \text{if } r_{jk} = 0 \\ r_{jk}, & \text{if } r_{jk} > 0 \end{cases}$$

- $m_1$ is a sufficiently small positive number
- $r_{jk}$ is the real part of the $k_{th}$ root of the $j_{th}$ generalized Kharitonov segment polynomial with $i_{th}$ assignment of
- $N$ is the number of segmentation of $\lambda$



Master
Slave
START
Set slaves
Connect to slaves via TCP/IP
Specify controller and plant parameters
Specify GA parameters
A
Evolution of GA1
B
Close communication port
Display non-dominated solutions of the controller
END
A
Feasible chromosomes found?
No
Yes
Group feasible chromosomes for slaves
Transfer feasible chromosomes and relevant parameters to slaves
Wait for GA2-1/GA2-2 evolution by slaves
Evolution by slaves completed?
No
Yes
Fetch fitness from slaves
Evolution completed?
No
Yes
B
START
Activate communication port
Connected to Master?
No
Yes
Wait to receive feasible chromosomes from Master
Feasible chromosomes from Master received?
No
Yes
Evolution of GA2-1/GA2-2

# Illustrated Example

- High-order interval plant

$$G(s) = \frac{[0.9 \quad 1.1]s^2 + [2.4 \quad 2.6]s + [1.4 \quad 1.6]}{s^5 + [16 \quad 17]s^4 + [75 \quad 77]s^3 + [103 \quad 105]s^2 + [33 \quad 35]s + [119 \quad 121]}$$

- Third order controller

$$C(s,q) = \frac{q_1 s^2 + q_2 s + q_3}{s(q_4 s^2 + q_5 s + q_6)}$$

$$\begin{aligned} &q_1 \in [-1200 \quad 1200], \quad q_2 \in [-200 \quad 200], \quad q_3 \in [-500 \quad 500], \\ &q_4 \in [-200 \quad 200], \quad q_5 \in [-200 \quad 200], \quad q_6 \in [-200 \quad 200]. \end{aligned}$$

# Control parameters of GA

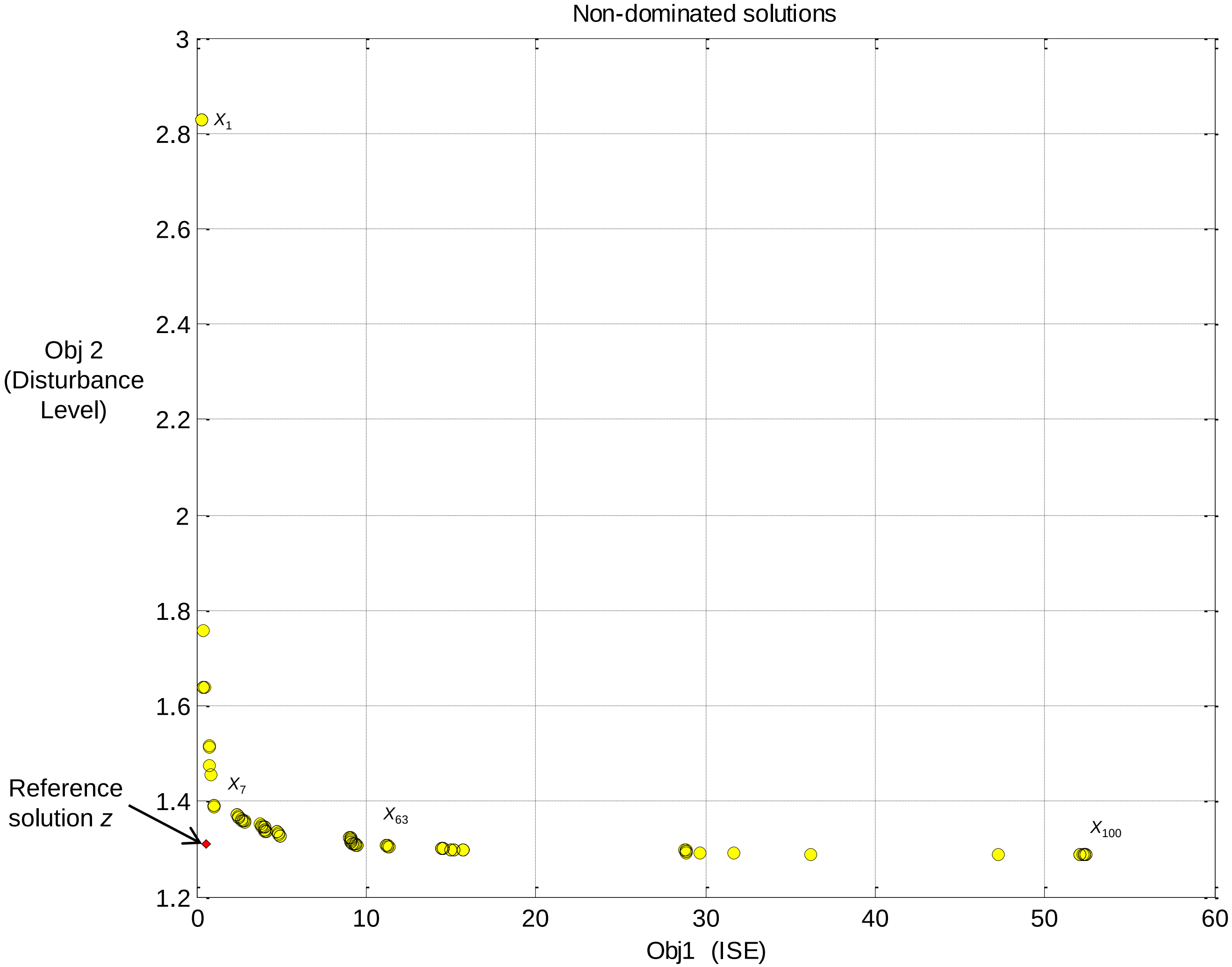
GA1:

population size=50, *pc*=0.9, *pm*=0.15, tournament size *k*=2, and distribution indices for crossover and mutation operators $\eta_c = 2$ and $\eta_m = 2$

GA2-1 (GA2-2):

population size=50, *pc*=0.9, tournament size *k*=2, *pc*=0.9 , *pm*(boundary)=0.05 (GA2-1), *pm*(nonuniform)=0.1 (GA2-2)

- Non-dominated solutions

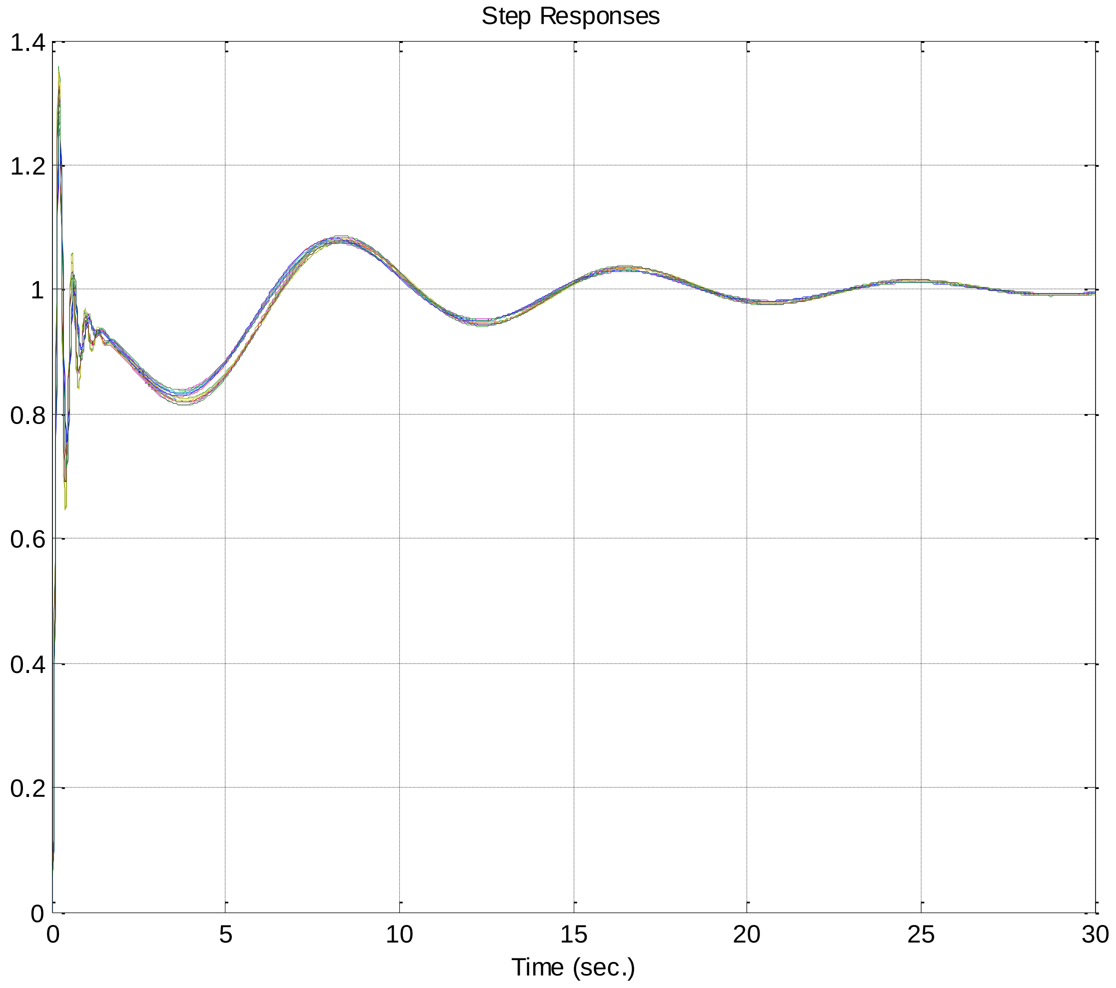


## Table 1 Non-dominated solutions

| | $X_1$ | $X_2$ | $X_3$ | … | $X_7$ | … | $X_{63}$ | … | $X_{100}$ |
|---|---|---|---|---|---|---|---|---|---|
| **Obj1** | **0.2844** | **0.3866** | **0.4514** | … | **0.4534** | … | **11.2035** | … | **52.4371** |
| **Obj2** | **2.8288** | **1.7582** | **1.6401** | … | **1.6372** | … | **1.3078** | … | **1.2874** |

- Desired solutions among the non-dominated set

$$C(s) = \frac{1066s^2 + 189.6s + 497}{s(0.0002891s^2 + 0.0995s + 3.993)}$$




Step responses of the closed-loop system with Kharitonov plants and the optimal controller X1 derived by the proposed MOGA approach

# Conclusions

- Optimal controllers satisfying performance criteria of minimum tracking error and disturbance level for interval systems
- Constrained multi-objective optimization problem
- Solved via a MOGA-based framework with two GAs
- Constraint handling based on generalized Kharitonov segment polynomials
- No restrictive condition under which the proposed approach is developed
- Other performance specifications can be easily incorporated into
- No constraints on plant order or controller order
- Easy to use
- Parallel computation scheme

**The End!**