# RM-MEDA* and MOEA/D+:
# Two New Multiobjective Evolutionary Algorithms

**Qingfu ZHANG**

Department of Computing & Electronics Systems,

University of Essex

CO4 3SQ, Colchester, UK

* **Qingfu Zhang, Aimin Zhou and Yaochu Jin**, *RM-MEDA: A Regularity Model Based Multiobjective Estimation of Distribution Algorithm, IEEE Trans. on Evolutionary Computation, 2008, in press.*

+ **Qingfu Zhang and Hui Li**, *MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Trans. on Evolutionary Computation, Dec/2007. .*

University of Essex

# RM-MEDA* and MOEA/D+: Two New Multiobjective Evolutionary Algorithms

**Qingfu ZHANG**

Department of CES,

University of Essex

CO4 3SQ, Colchester, UK

***Qingfu Zhang, Aimin Zhou and Yaochu Jin**, *RM-MEDA: A Regularity Model Based Multi-objective Estimation of Distribution Algorithm, IEEE Trans. on Evolutionary Computation, 2008, in press.*

+**Qingfu Zhang and Hui Li**, *MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Trans. on Evolutionary Computation, 2007, Dec/2007.*

# Contents

- Multiobjective Optimisation
- RM-MEDA: Regularity-Model Based Multiobjective Estimation of Distribution Algorithm
- MOEA/D: Multiobjective Evolutionary Algorithm Based on Decomposition
- Conclusion

# Contents

- Multiobjective Optimisation
- RM-MEDA: Regularity-Model Based Multiobjective Estimation of Distribution Algorithm
- MOEA/D: Multiobjective Evolutionary Algorithm Based on Decomposition
- Conclusion

# Multiobjective Optimisation Problem (MOP)

$$\min F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$$
$$s.t \quad x \in D$$

where:

$D$: decision (variable) space.

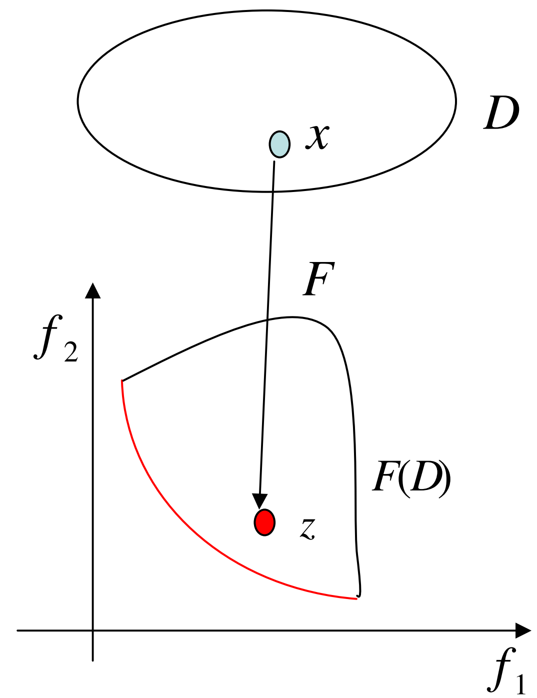
$R^m$: objective space.

$f_i : D \to R$, objective function

$F : D \to R^m$, objective vector function

$F(D) = \{F(x) \mid x \in D\}$: attainable objective set

- Applications: too long to list
  google (ca. 80k results)




$$z = (z_1, z_2) = F(x)$$
$$z_1 = f_1(x),\ z_2 = f_2(x)$$

# What is Optimal in MOPs?

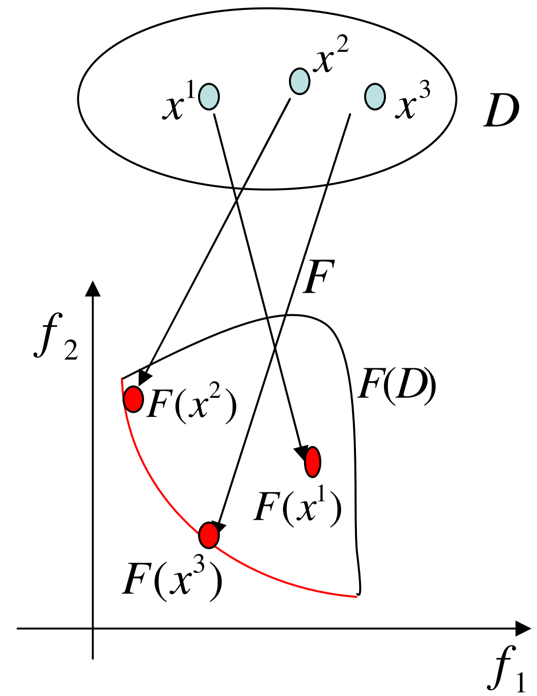
For minimization problem:

Let $x, y \in \mathrm{D}$,

$x$ dominates $y$ (or $F(x)$ dominates $F(y)$)

$$\Updownarrow$$

$f_i(x) \le f_i(y)$ for all $i$ and $f_j(x) < f_j(y)$ for at least one index $j$.




- "Dominate" = " be better than".
- Example:

$x^3$ ($F(x^3)$) dominates $x^1$ ($F(x^1)$).

$x^2$ ($F(x^2)$) and $x^1$ ($F(x^1)$) cannot be compared with each other.

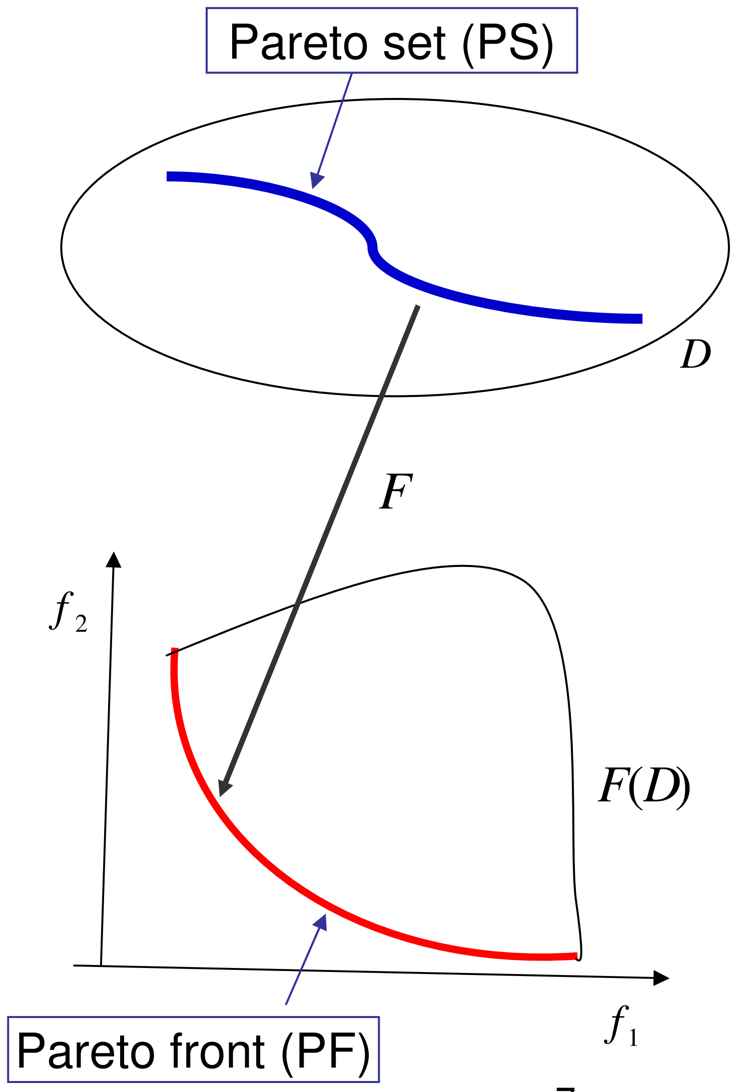
Domination is a partial ordering

⇧

Why MOPs are harder than single opt. problems

# What is Optimal in MOPs?

- **Pareto Optimal Solution (1896):**
  a solution cannot be dominated by any other solutions.
- **Pareto Set** (PS): the set of all the Pareto optimal solutions (in the decision space).
- **Pareto front** (PF): PF=*F*(PS) (in the objective space).




The PF is the southwest boundary of F(D).

# Task of MOEAs

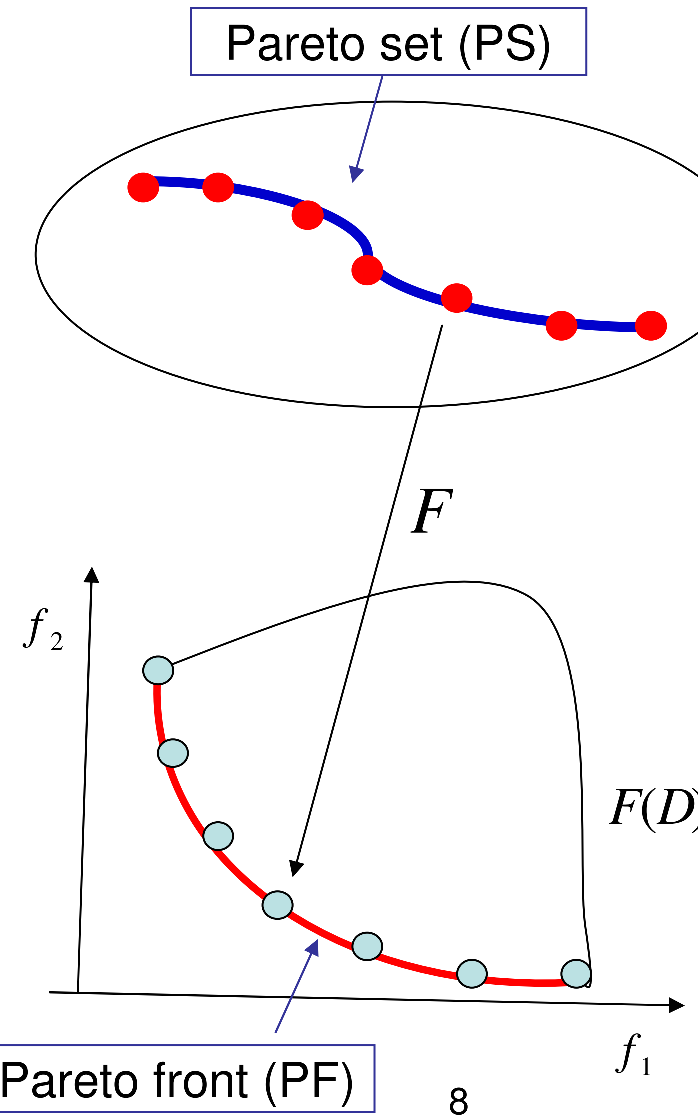
Solving a MOP requires interactions between decision makers and engineers.

Very often, A decision maker wants:

A representative set of Pareto optimal solutions

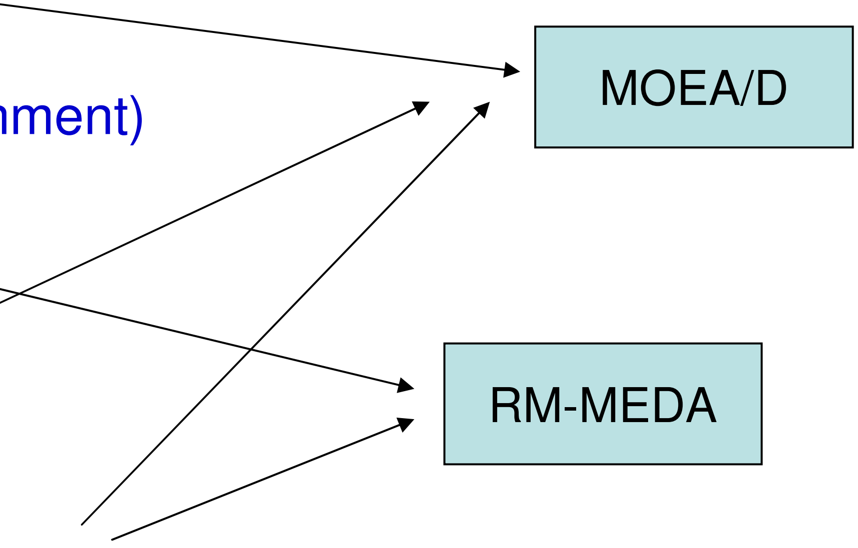
(uniformly distributed along the PF or PS )

The task of most Multiobjective Evolutionary Algorithms (MOEAs)



# Major issues in the current MOEA research

- Framework
- Selection (Fitness Assignment)
- Reproduction Operators
- Local Search
- Problem-specific Knowledge
- Other tricks: external population, mating strategy.



# Contents

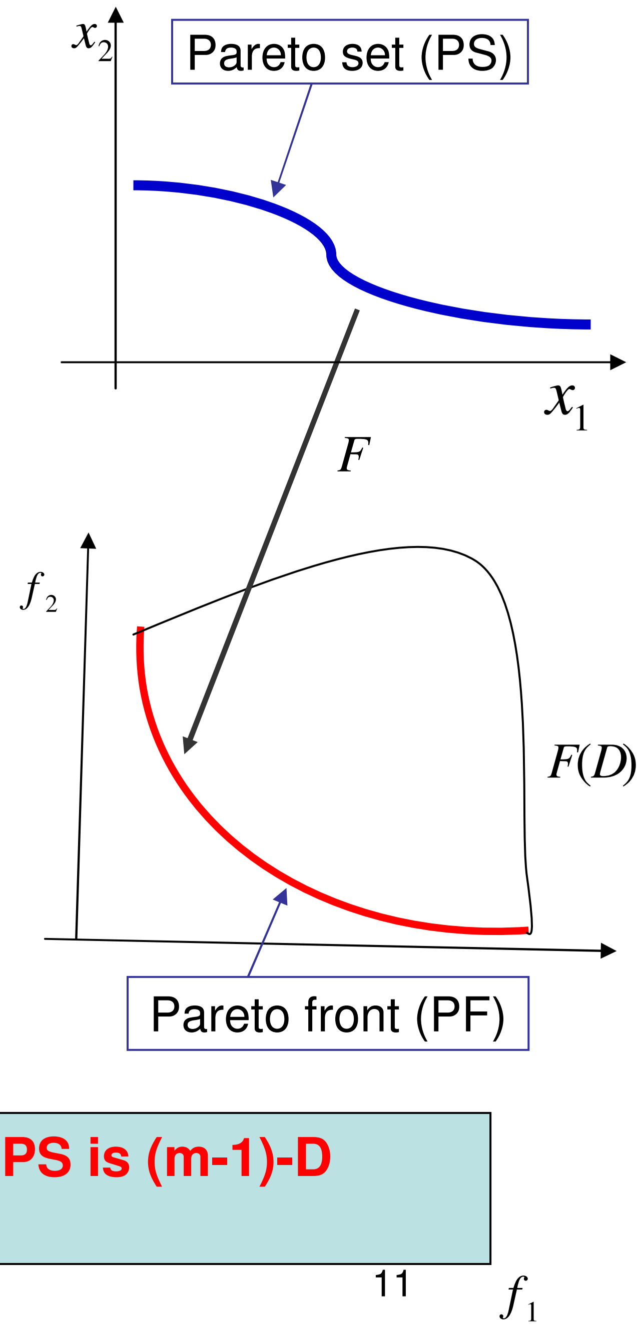
- Multiobjective Optimisation
- RM-MEDA: Regularity-Model Based Multiobjective Estimation of Distribution Algorithm
- MOEA/D: Multiobjective Evolutionary Algorithm Based on Decomposition
- Conclusion

# Motivations

- Regularity of continuous MOP:

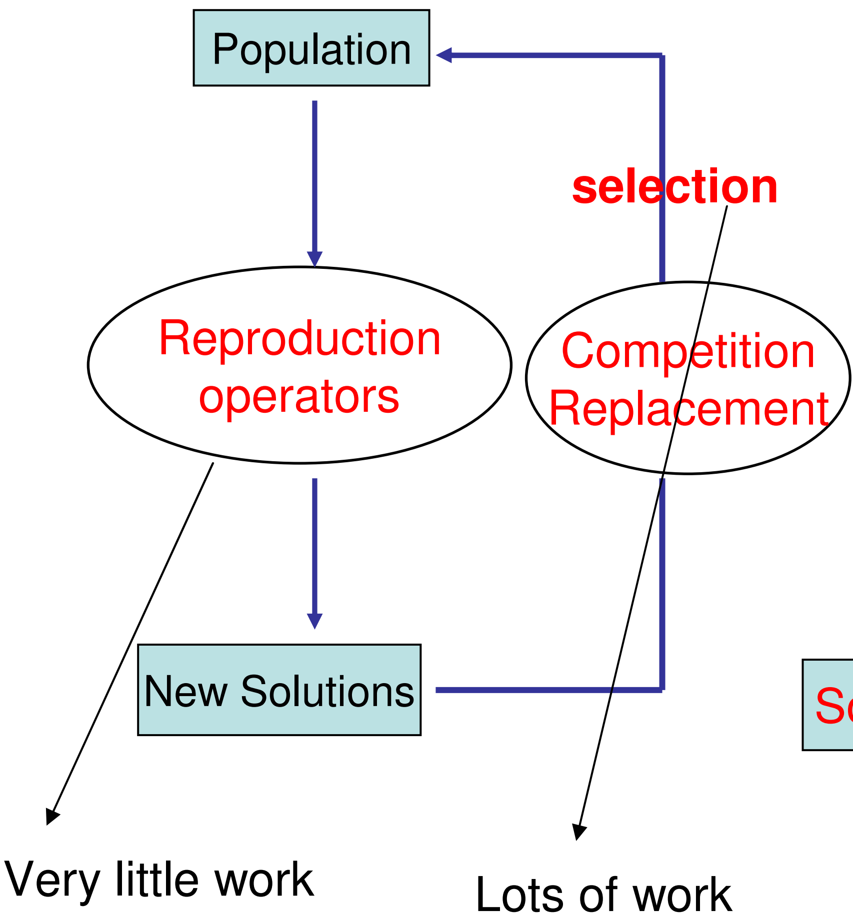
**Under certain conditions, the PS (PF) is a (m-1)-dimensional piecewise continuous manifold in decision (objective) space.**
Where m is the # of the objs.

- This property has been ignored by MOEA researchers.
- The PSs in most commonly-used test problems are too simple (a line or a plane).
- There is very little research on the shape of PSs.

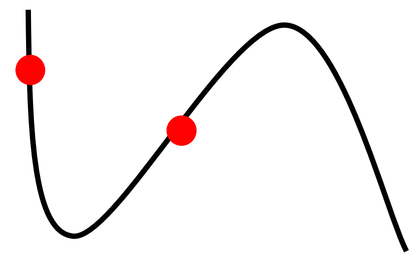



**How can we deal with a continuous MOP if its PS is (m-1)-D piecewise continuous manifold?**

- Suppose we use the following commonly-used framework:




- Why commonly-used genetic operators do not work well for complicated PSs?

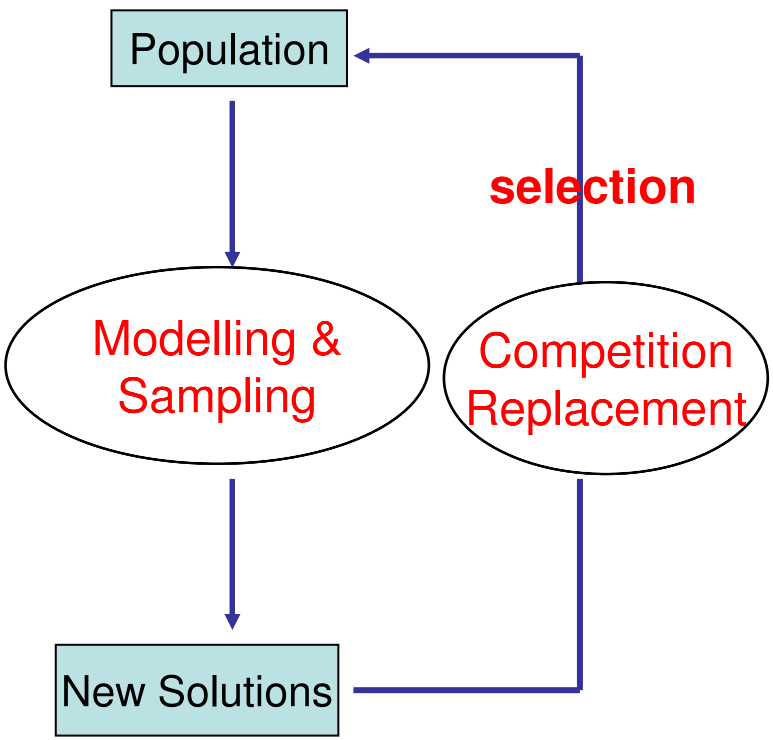


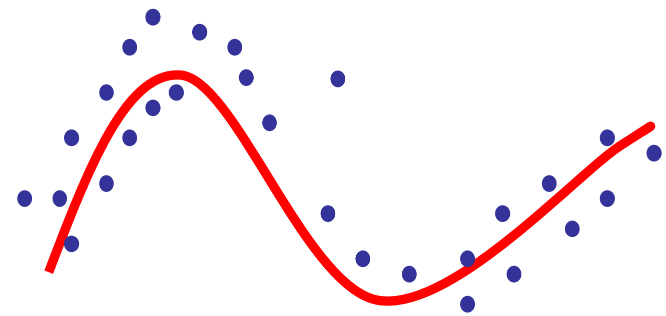
PS in decision Space.

  - When two parents are in the PS, their offspring may not be close to the PS.
  - The PS is not an equilibrium.

So we resort to EDA (Modelling and Sampling).

# Basic Idea

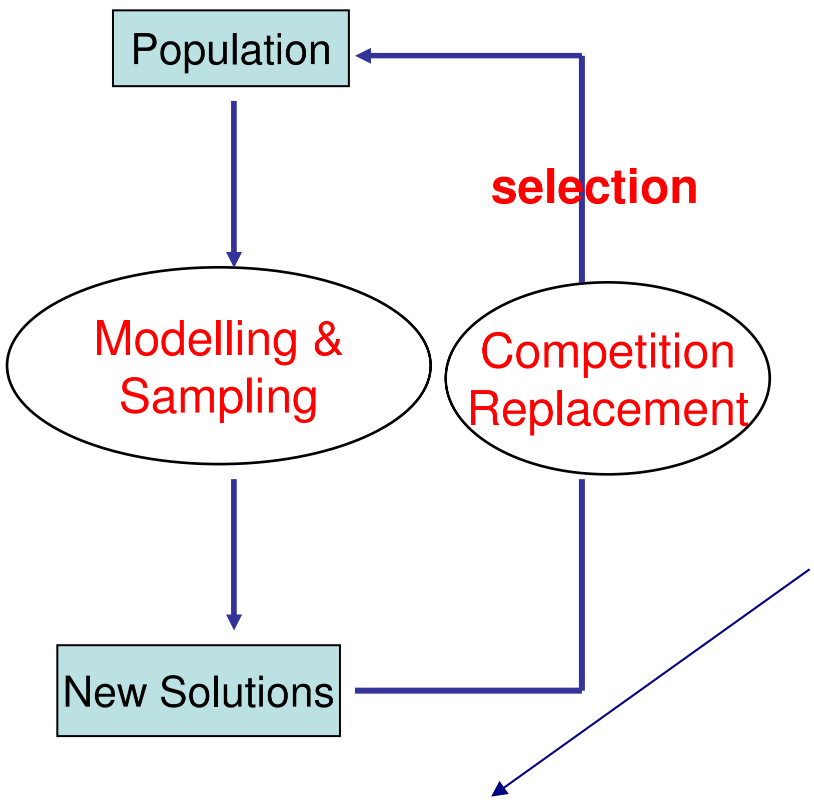


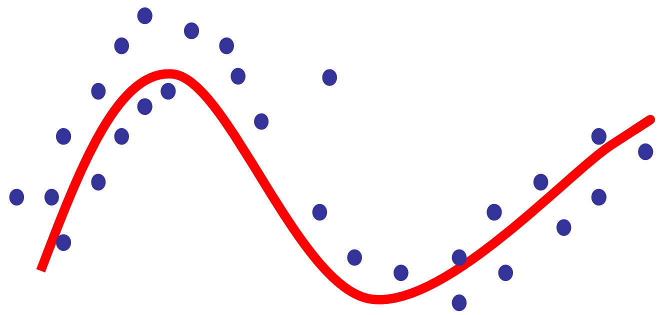



- In the case of 2 objs
  - The PS is  a 1-D curve.
  - If the algorithm works well.

    Population → PS

  - The principal (centroid) curve of the population could be an approximation to the PS.



Each point in the current population is regarded as a sample of

$$x = \xi + \varepsilon$$

where

$\xi$ is uniformly distributed on a 1-D curve $C$.

$\varepsilon$ is a $n$-D Gaussian noise.

This model is different from other models in EDAs.

How to model $C$ and $\varepsilon$?

# **Modelling:** How to model *C* and $\mathcal{E}$?

**We assume that**:

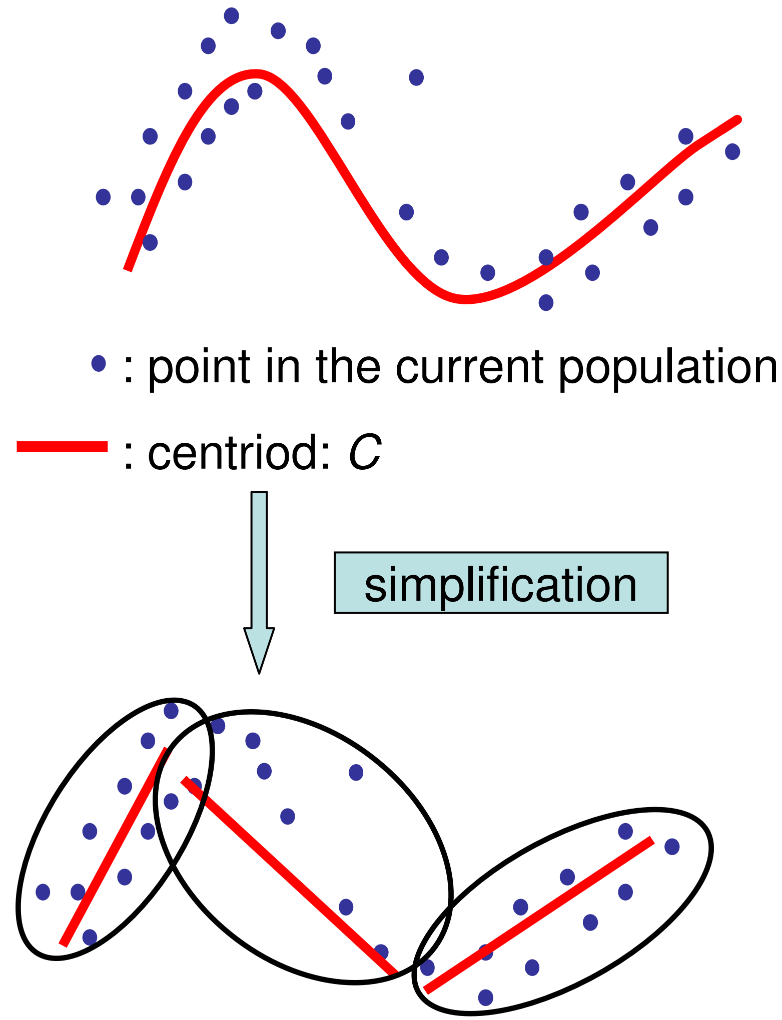
- o The centroid curve *C* consists of several line segments. This assumption makes *C* computable.

❑ How to model *C*

- o Divide the population into several clusters by local PCA.
- o Compute the central line of each cluster.

❑ How to model $\mathcal{E}$

- o the deviation of the points in each cluster to its central line.



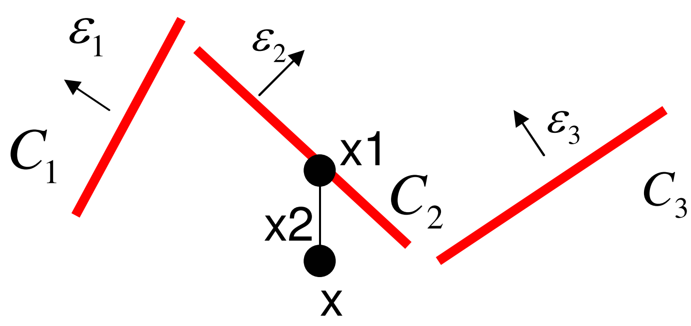

The number of clusters needs to be preset.

# Sampling

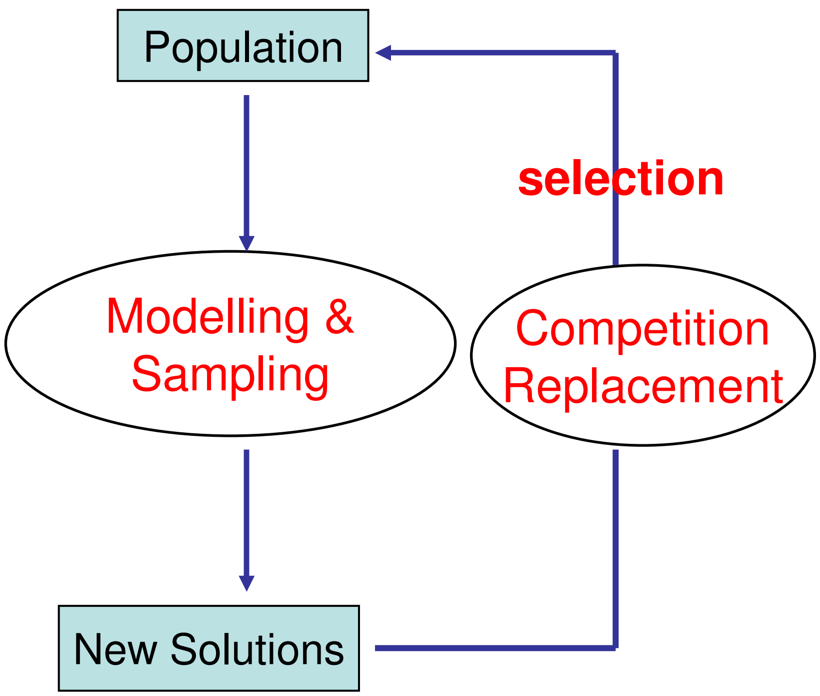
- How to sample new solutions:
  - The number of new solutions sampled around $C_i$ :

$$\#\_of\_new\_solutions \times \frac{L(C_i)}{L(C_1)+L(C_2)+L(C_3)}$$

  - Sampling around $C_i$
    - ✓ Uniformly randomly pick a point x1 in $C_i$
    - ✓ x2~N(0, $\varepsilon_i I_{n\times n}$),
    - ✓ x=x1+x2.



# Selection




- We use the non-dominated sorting (Deb et al) in selection.
  - Convergence: prefer solutions close to the PF.
  - Diversity: prefer “isolated” solutions

# Experiments: Test Problems we did't use

- ZDT test instance, ZDT1:

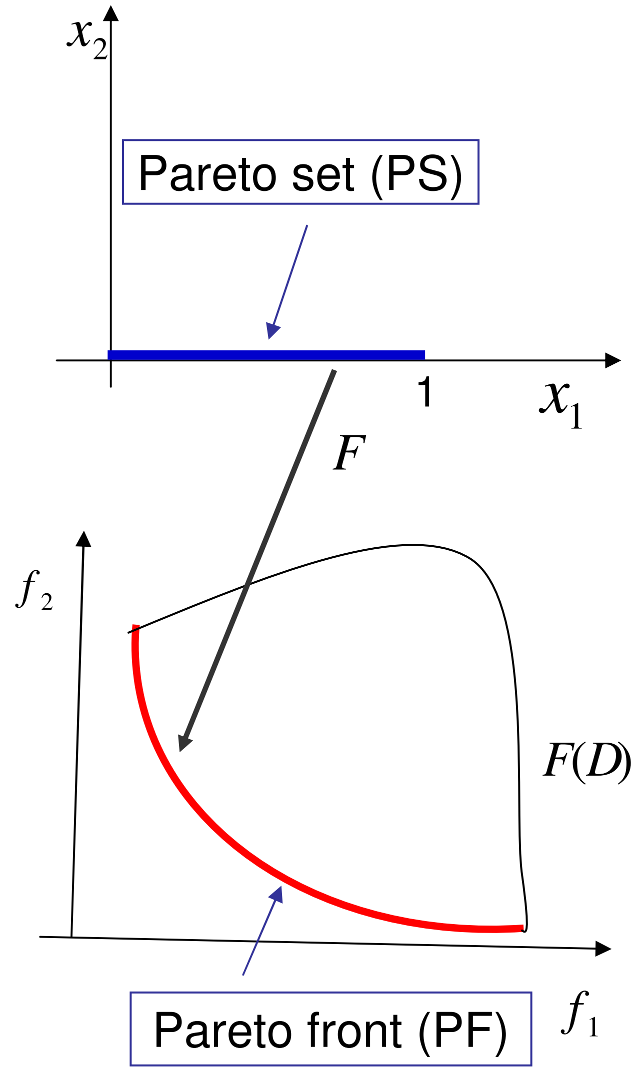
$$\min\ f_1(x) = x_1$$
$$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$$
$$\text{where } g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$$
$$0 \le x_i \le 1, \text{ for } i = 1,2,\ldots,n.$$

- In all the ZDTs with two objectives, the PS is

$$0 \le x_1 \le 1,\quad x_2 = x_3 = \ldots = x_n = 0.$$

- It is too friendly for commonly-used crossovers:

If two solutions are Pareto optimal, then its children is more likely to be Pareto optimal.




So, we will not use these test instances.

# Experiments: Test Problems we used

- Test instances with 2 objs used in our experiments: modified versions of ZDT.
  - PS

$$0 \le x_1 \le 1, \quad x_1 = x_2 = \ldots = x_n$$

Variable transformation used:

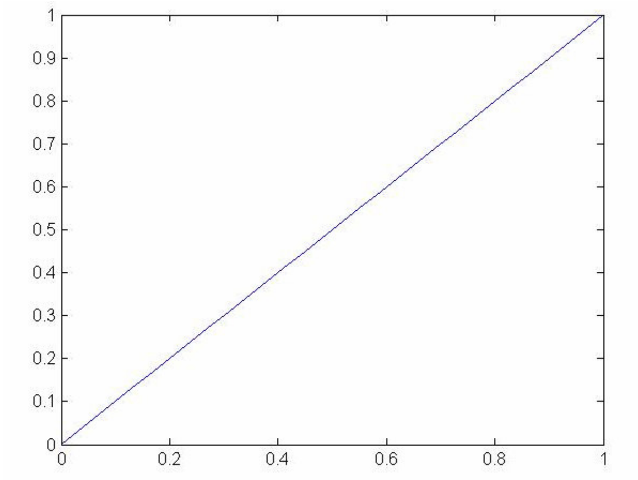
$$x_1 \to x_1, \; x_i \to x_i - x_1 \;\; (i = 2, \ldots, n)$$

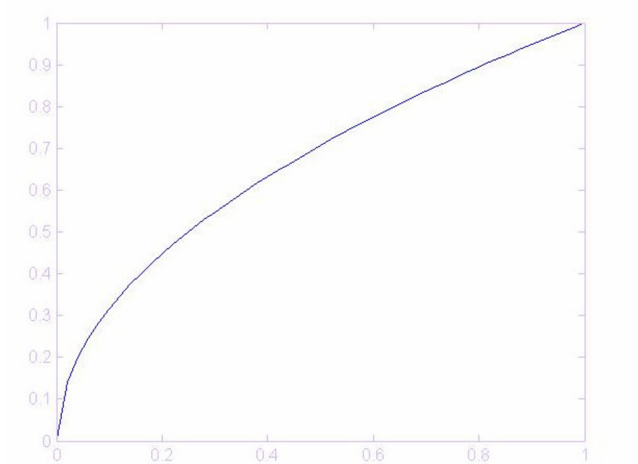
  - PS

$$0 \le x_1 \le 1, \quad x_1 = x_2^2 = \ldots = x_n^2$$

Transformation:

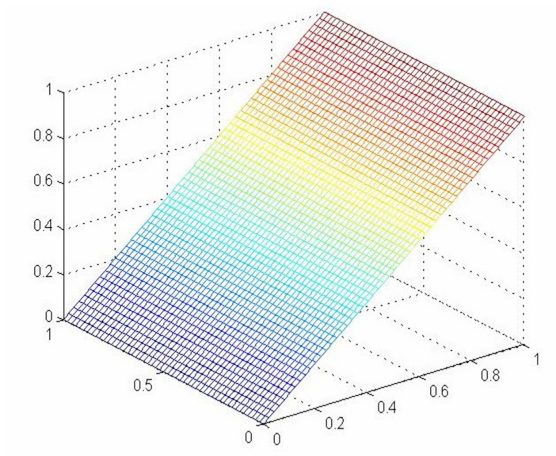
$$x_1 \to x_1, \; x_i \to x_i^2 - x_1 \;\; (i = 2, \ldots, n)$$



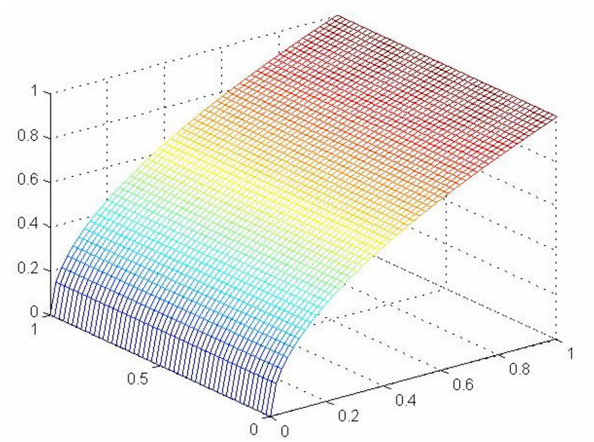
- Test instances with 3 objs used in our experiments: modified versions of DTLZ with variable transformation.
  - o DTLZ-1 with PS

$$0 \le x_1, x_2 \le 1, \quad x_1 = x_3 = \ldots = x_n$$




  - o DTLZ-2 with PS

$$0 \le x_1, x_2 \le 1, \quad x_1 = x_3^2 = \ldots = x_n^2$$



# Experiments: Parameter setting

- The number of decision variables=30.
- Pop_Size=100 for two objs, and 200 for three objs for all the algorithms
- The number of clusters in Local PCA=5.
- All the statistics are based on 20 independent runs.

The algorithms in comparison:

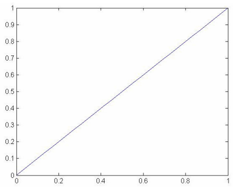
- PCX-NSGA-II, New solution generator: PCX.
- GDE 3 New solution generator: DE
- MIEDA: New solution generator: ordinary EDA

All these algorithms use the same selection as RM-MEDA.
The only difference is how to generate new solutions.

# Experiments: Results

- Test Instance

$$f_1(x) = x_1$$
$$f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$$
$$g(x) = 1 + 9(\sum_{i=2}^{n}(x_i - x_1)^2)/(n-1)$$

  - PS:  PF: convex

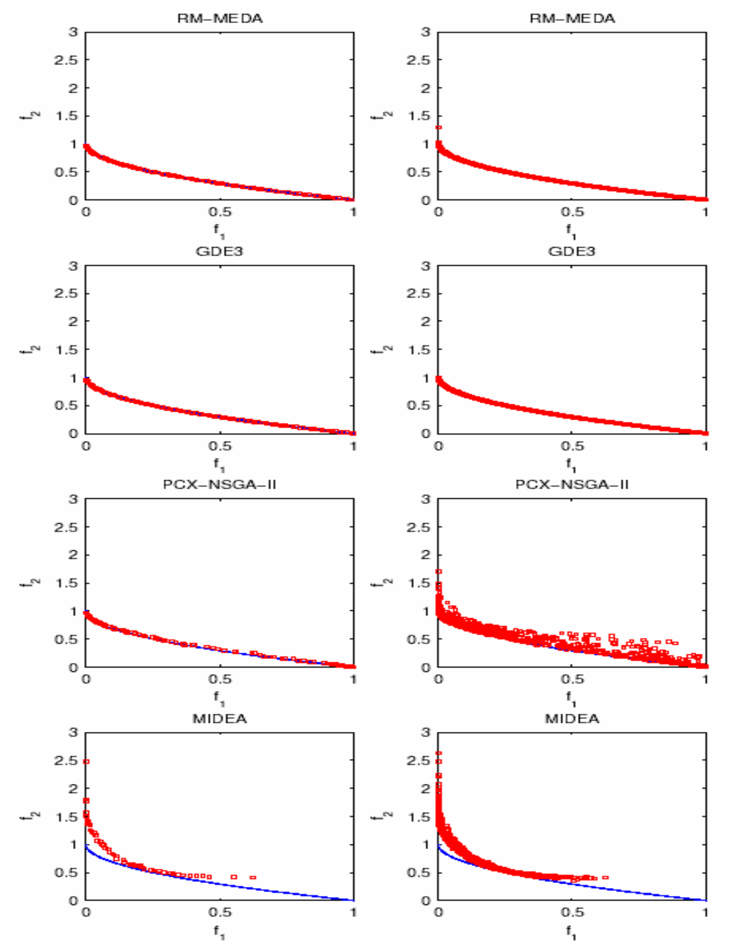



Fig. 7. The final nondominated fronts found by each algorithm on F1. The left panels show the nondominated fronts with the lowest $D$-metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

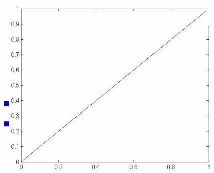
- RM-MEDA and GDE3 are better than two others. Why?

  If all the parents are in the PS,

  - ✓ In RM-MEDA and GDE3, the offspring will be in (very close to) the PS.
  - ✓ In the 2 others, the offspring will not.

## ❑ Test Instance

$$f_1(x) = 1 - exp(-4x_1)sin^6(6\pi x_1)$$
$$f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$$
$$g(x) = 1 + 9[\sum_{i=2}^{n}(x_i - x_1)^2/9]^{0.25}$$

- PS: 
- Optimal solutions are not uniformly distributed.

## ❑ RM-MEDA>GDE3, Why

- If all the parents are in the PS, but not uniformly distributed.
  - ✓ In RM-MEDA, the offspring will be uniformly distributed in the PF.
  - ✓ In GDE3, the offspring will not be uniformly distributed.

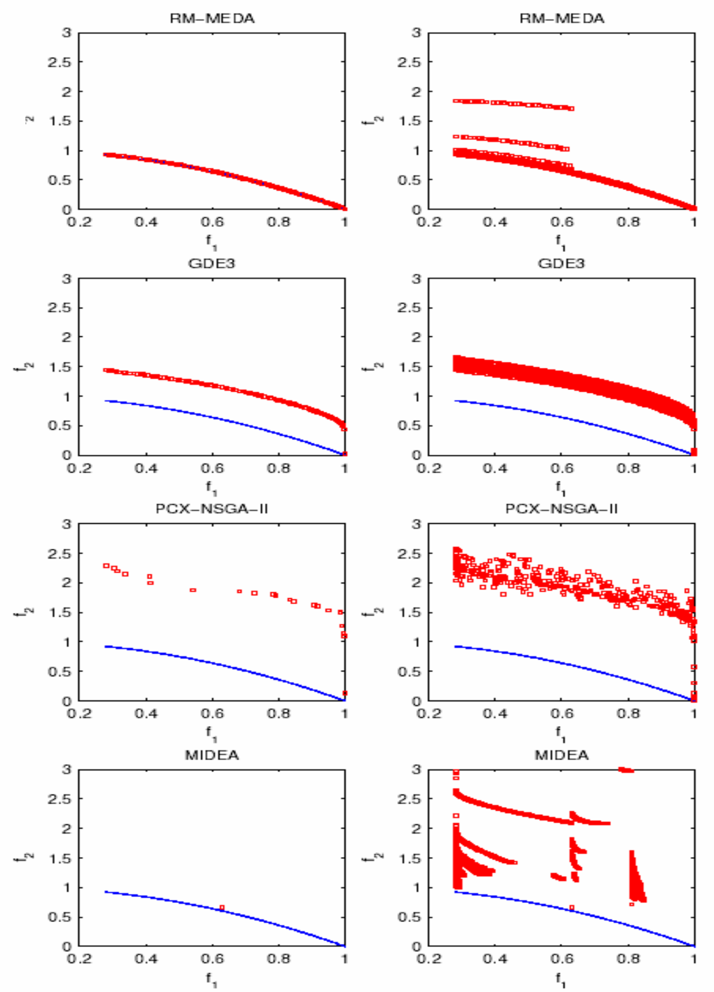
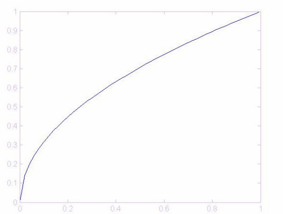



Fig. 9. The final nondominated fronts found by each algorithm on F3. The left panels show the nondominated fronts with the lowest $D$-metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

- **Test Instance**

$$f_1(x) = 1 - exp(-4x_1)sin^6(6\pi x_1)$$
$$f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$$
$$g(x) = 1 + 9[\sum_{i=2}^{n}(x_i^2 - x_1)^2/9]^{0.25}$$

  - PS:



- **RM-MEDA>others, Why**

  If all the parents are in the PF,

  - ✓ In RM-MEDA, the offspring will be very close to the PF.
  - ✓ In all the other algorithms, the offspring will be far away from the PF.

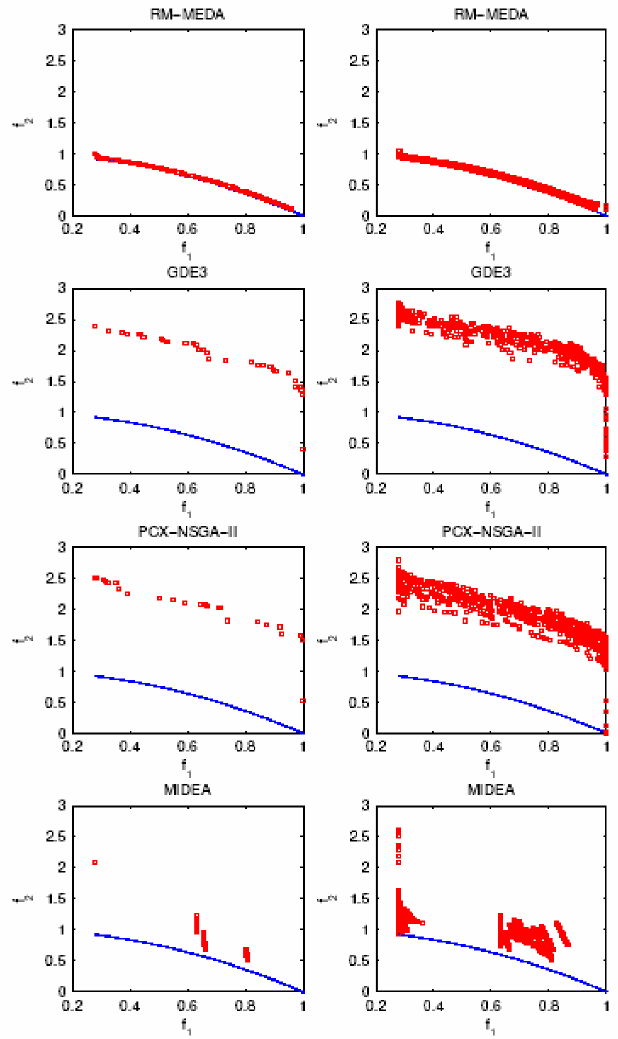
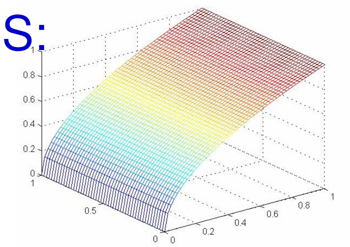



Fig. 17. The final nondominated fronts found by each algorithm on F7. The left panels show the nondominated fronts with the lowest $D$-metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

- Test Instance

$$f_1(x) = cos(\frac{\pi}{2}x_1)cos(\frac{\pi}{2}x_2)(1+g(x))$$
$$f_2(x) = cos(\frac{\pi}{2}x_1)sin(\frac{\pi}{2}x_2)(1+g(x))$$
$$f_3(x) = sin(\frac{\pi}{2}x_1)(1+g(x))$$
$$g(x) = \sum_{i=3}^{n}(x_i^2 - x_1)^2$$

  - PS:



- RM-MEDA>others, Why

  If all the parents are in the PF,

  - ✓ In RM-MEDA, the offspring will be very close to the PF.
  - ✓ In all the other algorithms, the offspring will be far away from the PF.

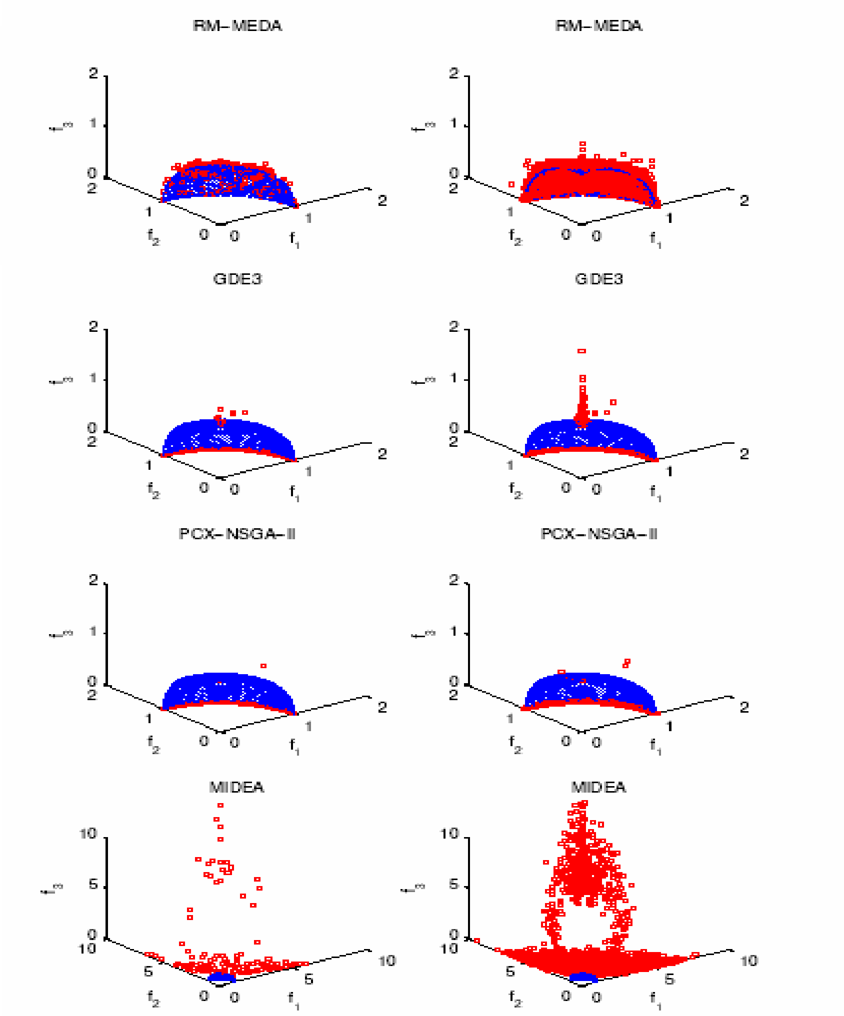



Fig. 18. The final nondominated fronts found by each algorithm on F8. The left panels show the nondominated fronts with the lowest $D$-metric obtained by each algorithm, while the right panels plot all the 20 fronts together found by each algorithm.

# Conclusions on RM-MEDA

- Based on regularity property.
- New solution generator is problem-specific for continuous MOPs.
- Very promising experimental results.
- Downsides:
  - o Local PCA: a little bit complicated.
  - A simple yet efficient version of RM-MEDA is under development

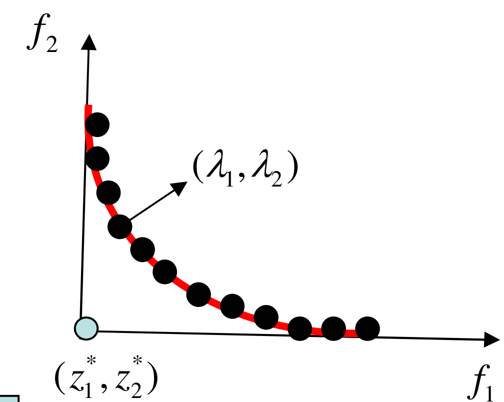
- Future Work: how to get a math description of PS.

# Contents

- Multiobjective Optimisation
- RM-MEDA: Regularity-Model Based Multiobjective Estimation of Distribution Algorithm
- MOEA/D: Multiobjective Evolutionary Algorithm Based on Decomposition
- Conclusion

# Background

- Most current MOEAs are based on Pareto domination.
  - o Very hard to make its solutions uniformly distributed along the PF, particularly if the population size is small.
- Ideas in traditional math program methods: Decomposition (aggregation)




Weight Sum Approach:

$$\min g^{ws}(x, \lambda) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$

Techbycheff Approach

$$\min g^{T}(x, \lambda) = \max\{\lambda_1 | f_1(x) - z_1^* |, \lambda_2 | f_2(x) - z_2^* |\}$$

where $\lambda_1 + \lambda_2 = 1$, and $\lambda_1, \lambda_2 \geq 0$.

Finding a set of N uniformly distributed Pareto optimal solutions ⇒

$\min \ g(x, \lambda^1)$

$\min \ g(x, \lambda^2)$

⋮

$\min \ g(x, \lambda^N)$

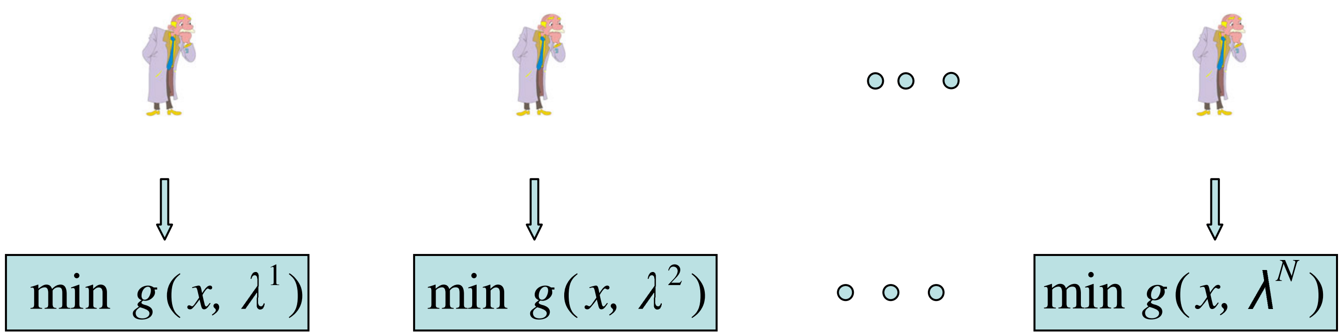
N problemS.

Not a N-obj opt problem!

Solve these N problems one by one.
The distribution of final solutions could be very uniform if g( ) and λ are properly chosen.

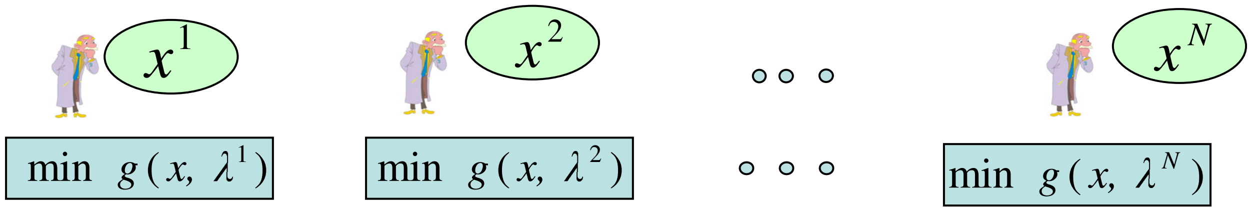
# Idea

- These problems are related with each other.
  - If $\lambda^i$ and $\lambda^j$ are close, we can call $g(x,\lambda^i)$ and $g(x,\lambda^j)$ neighbours.
  - neighbouring problems should have similar solutions.
- N agents are used for solving these N problems




- During the search, neighbouring agents can help each other.

# Algorithm Framework




- Agent i records $x^i$, the best solution he has found so far for his problem.
- At each generation, each agent i does the following:
  - Randomly select several neighbours and obtain their best solutions.
  - Apply genetic operators on these selected solutions and generate a new solution $\bar{y}$ .
  - Apply single opt. local search on $\bar{y}$ to optimise its obj $g(x, \lambda^i)$ and obtain $y$ .
  - Replace $x^i$ by $y$ if $g(y, \lambda^i) < g(x^i, \lambda^i)$ .
  - Let its neighbours replace their best solutions by $y$ if $y$ is better than their current best solutions (measured by their individual objectives).

# More Details of MOEA/D

- **Initialisation**: each agent can initialise $x^i$ randomly or by using problem-specific knowledge, e.g.
  - o Randomly generate a point in the decision space and then use a single obj LS to improve it.
- **Decomposition Method:**
  - o Any methods should do. We have tried weight sum approach, Tchebycheff approach and Penalty based boundary intersection (PBI) approach (which we proposed by modifying the NBI method).
  - o The setting of $\lambda^1,\ldots\lambda^N$ : uniformly distributed in

$$\{\lambda = ((\lambda_1,\ldots,\lambda_m) \mid \sum_{i=1}^{m} \lambda_i = 1; \lambda \geq 0\}$$

  Other methods can also be considered, e.g., dynamically tuning.
  - o Reference point (needed in PBI and Tchebycheff approaches) is estimated from the previous search.

# Remarks on MOEA/D

- **Diversity:** "Diversity" in subproblems will lead to the diversity among $\{x^1, x^2, \ldots, x^N\}$.
- **Mating Restriction:** Solutions have a chance to mate only when they are for neighbouring problems (can be relaxed).
- **Complexity:** much lower than NSGA-II and MOGLS (detailed analysis can be found in the paper).
  - Suppose that both NSGA-II and MOEA/D uses the same population size, then at each generation, the ratio of computational complexity between NSGA-II and MOEA/D is:

$$O\left(\frac{T}{Pop\_Size}\right)$$

where T is the number of neighbours for each subproblem.

## Two Other Related Methods

- **Cell Multi-objective Genetic Algorithm (cMOGA) (Murata et al, 2001 )**
  - **is based on the concept of cells. Cell ≈subproblem.**
  - **uses the neighbourhood relationship among cells.**
  - **needs Pareto domination and the external population, otherwise, it will not work well based on our recent experiments**
  - **is much poorer than MOEA/D based on our recent experiments.**

- **MOGLS (Jaszkiewicz, 2002)**
  - **use decomposition methods.**
  - **Tries to optimise all the weighted objectives.**
  - **Does not use the concept of neighbourhoods.**

# Comparison With MOGLS: Test Problem

- Multiobjective 0-1 knapsack problem:

$$\max \quad f_i(x) = \sum_{j=1}^{n} p_{ij} x_j \qquad i = 1,2,\ldots,m.$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_{ij} x_j \le c_i \qquad i = 1,2,\ldots,m.$$

$$x = (x_1,\ldots,x_n) \in \{0,1\}^n.$$

$$p_{ij}, w_{ij} \ge 0.$$

m objs
m constraints

- NP-hard
- 9 test instances have been widely-used in evolutionary computation.
- MOGLS outperforms a number of MOEAs on these test instances (2002).

# Comparison With MOGLS: Experiment Setting

- MOGLS: The code and setting are from its author's paper and web.
  - external population is used to record all the non-dominated solutions found in the search.
  - Xover: one-point
  - Mutation rate: 0.01.
  - Local search is applied each new solution.
  - Decomposition: weight sum and Tchebycheff methods.
  - The size of its internal population dynamically increases. its upper bound is from 3,000 to 7,000.

- MOEA/D:
  - External population, Xover, mutation rate, local search, decomposition methods: the same as in MOGLS.
  - The size of internal population: 150~450.
  - The size of neighbourhood=20.

The maximal no of calls of local search for both algorithms:
500X The size of internal population.

# Comparison With MOGLS: Experimental Results

AVERAGE CPU TIME (IN SECOND) USED BY MOEA/D AND MOGLS

| Decomposition Method | | | Tchebycheff | | Weighted Sum | |
|---|---|---|---|---|---|---|
| | $m$ | $n$ | MOEA/D | MOGLS | MOEA/D | MOGLS |
| Instance | 2 | 250 | 3.93 | 26.47 | 3.70 | 29.37 |
| | 2 | 500 | 10.40 | 70.80 | 9.40 | 72.97 |
| | 2 | 750 | 20.13 | 127.30 | 17.87 | 129.60 |
| | 3 | 250 | 7.00 | 81.70 | 6.53 | 78.17 |
| | 3 | 500 | 17.50 | 147.70 | 15.30 | 137.13 |
| | 3 | 750 | 31.90 | 219.30 | 26.73 | 202.47 |
| | 4 | 250 | 17.33 | 188.80 | 19.60 | 186.17 |
| | 4 | 500 | 42.60 | 292.10 | 45.17 | 287.60 |
| | 4 | 750 | 75.17 | 425.33 | 70.93 | 396.20 |

- Pentium (R), 3.2 GHZ, 1.00GB

MOGLS needs much less CPU time

In the paper, we analysed the complexity of MOGLS and MOEA/D. The above results confirm our analysis.

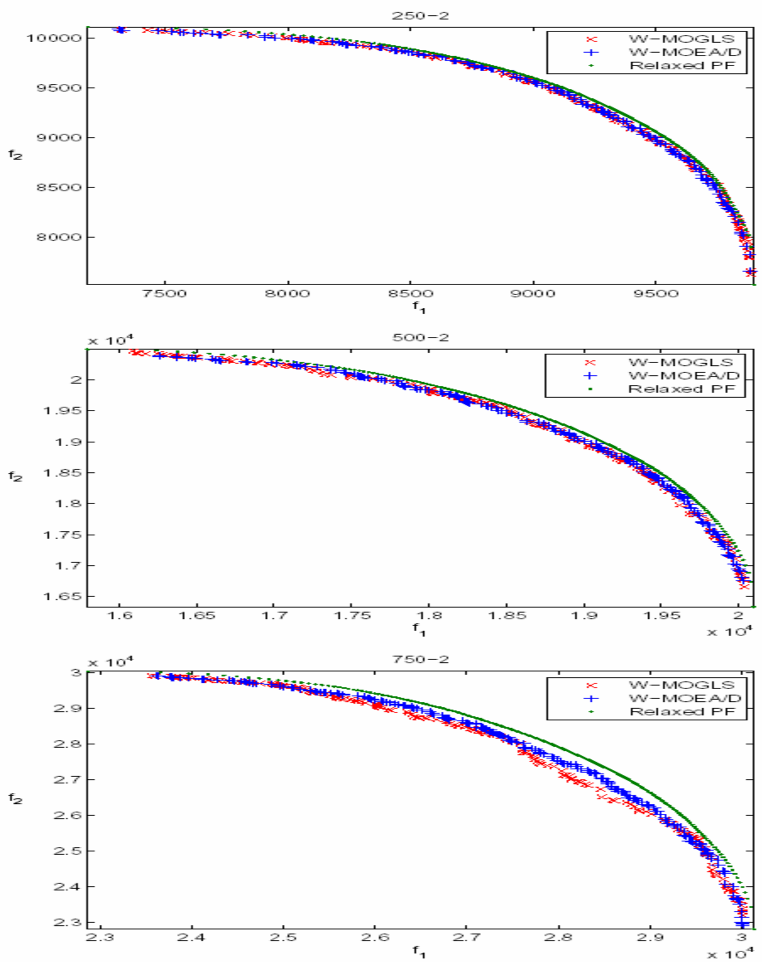



Fig. 4. Plots of the nondominated solutions with the lowest $D$-metric in 30 runs of MOEA/D and MOGLS with the weighted sum approach for all the 2-objective MOKP test instances.

- weight sum approach used.
- 3 bi-obj test instances.
- The best approximation found by each algorithm.

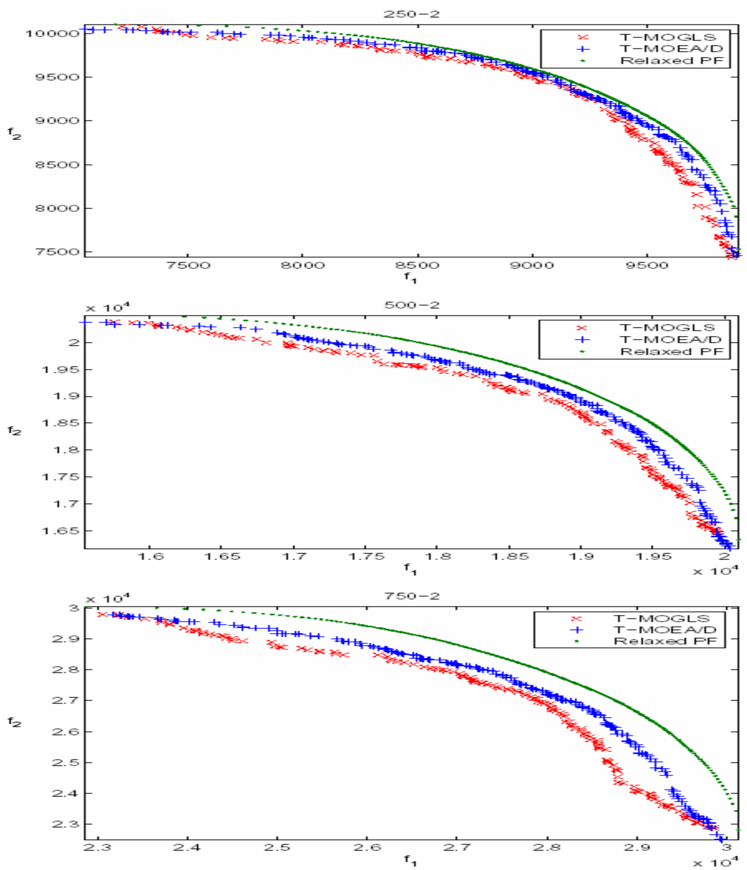



Fig. 5. Plots of the nondominated solutions with the lowest $D$-metric in 30 runs of MOEA/D and MOGLS with the Tchebycheff approach for all the 2-objective MOKP test instances.

- Techebycheff method.
- 3 bi-obj test instances.
- the best approximation found by each algorithm.

# Comparison With MOGLS: Performance Metric

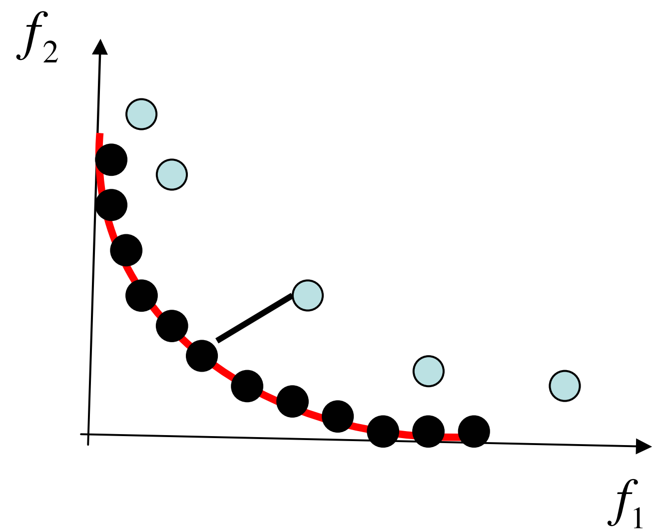
- The inverted generational distance (IGD)
  - Let P be a set of uniformly distributed points along the PF. Let S be the set of the approximated solutions found by an algorithm.

$$D(P \rightarrow S) = \frac{\sum_{v \in P} d(v, S)}{\#\ \text{of the points in } P.}$$

where

$$d(v, S) = \min_{u \in S}\{d(v, u)\}$$

  - If P is large enough, then $D(P \rightarrow S)$ could measure both the diversity and convergence of S.




: solutions found by algorithm
: Pareto solutions uniformly distributed along the PF.
P: all the black points.
S: all the green points.

TABLE IV

$D$-METRIC VALUES OF THE SOLUTIONS FOUND BY MOEA/D AND MOGLS. THE NUMBERS IN PARENTHESES REPRESENT THE STANDARD DEVIATION.

| Decomposition Method | | | Tchebycheff | | Weighted Sum | |
|---|---|---|---|---|---|---|
| | $m$ | $n$ | MOEA/D | MOGLS | MOEA/D | MOGLS |
| Instance | 2 | 250 | 54.10 (4.57) | 96.38 (6.03) | 37.17(2.98) | 38.18(3.21) |
| | 2 | 500 | 184.85 (11.88) | 328.00 (19.85) | 79.07(5.41) | 98.63 (9.16) |
| | 2 | 750 | 437.07 (21.19) | 765.66 (44.95) | 166.04(13.63) | 274.20 (22.70) |
| | 3 | 250 | 158.71 (6.58) | 217.25 (8.45) | 97.75(7.23) | 141.21 (12.25) |
| | 3 | 500 | 489.27 (15.91) | 701.70(27.40) | 270.31 (11.92) | 419.15 (23.86) |
| | 3 | 750 | 960.17 (23.62) | 1378.64 (54.63) | 446.12(19.12) | 768.30 (31.86) |
| | 4 | 250 | 253.23 (7.17) | 301.32 (6.33) | 176.52 (7.25) | 266.17 (9.27) |
| | 4 | 500 | 763.96 (14.81) | 964.41 (24.85) | 431.94(11.59) | 725.16 (24.57)) |
| | 4 | 750 | 1546.44 (27.78) | 1994.78 (72.84) | 761.57 (17.29) | 1246.54 (29.16) |

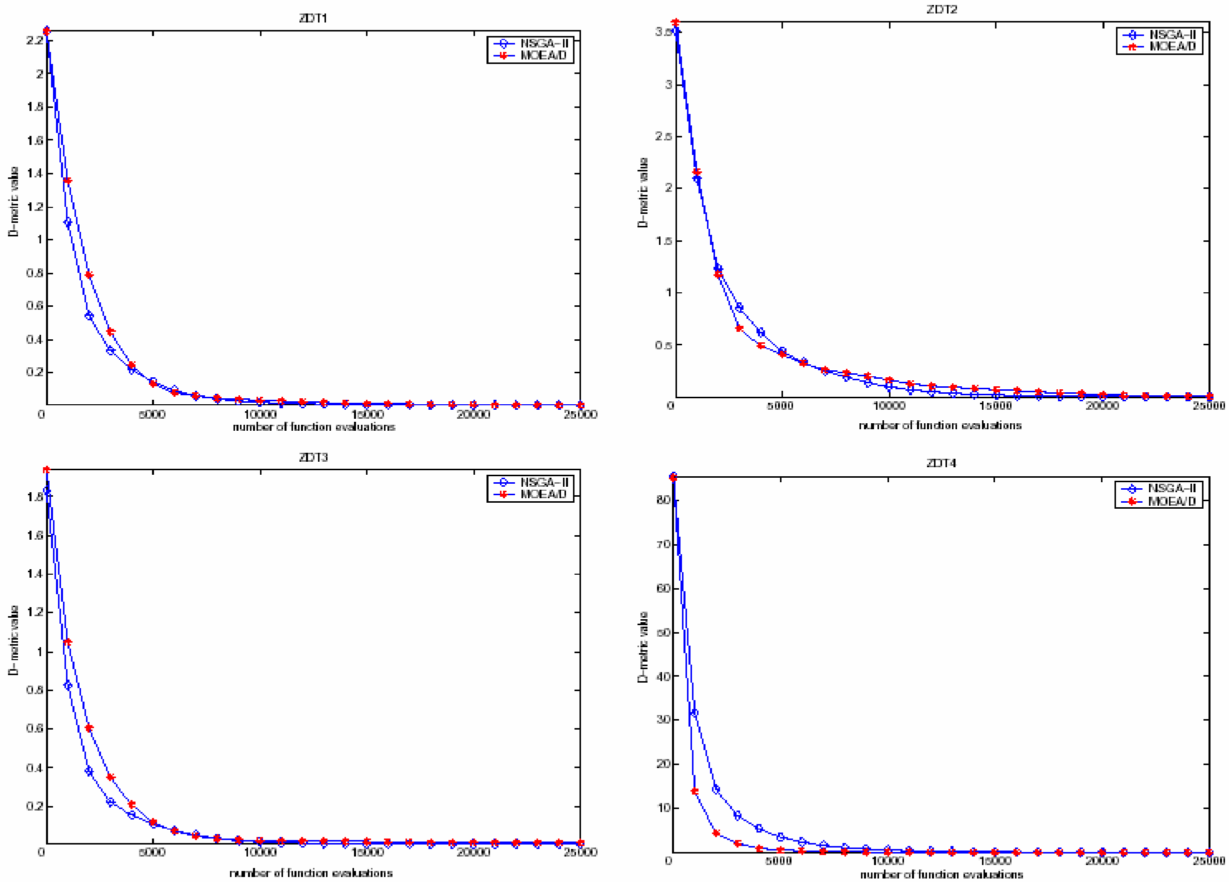
MOEA/D outperforms MOGLS in terms of IGD metric in all the instances.

# Comparison With NSGA-II: Test Problem

- Continuous Test Instances:
  - Bi-objective: ZDT 1, 2, 3 (n=30).
  - Bi-objective: ZDT 4 and 6 (n=10).
  - Tri-objective: DTLZ1 and 2 (n=10).
- The setting of NSGA-II is from its authors' paper.
  - Without external population
  - Pop_size=100 for 2 objs and 300 for 3objs.
  - Xover: SBX
  - Mutation: polynomial mutation.
- The setting of MOEA/D
  - Tchebycheff Method
  - Without external population
  - Pop_size, Xover and mutation, as the same as in NSGA-II.
  - The size of neighbour=20.

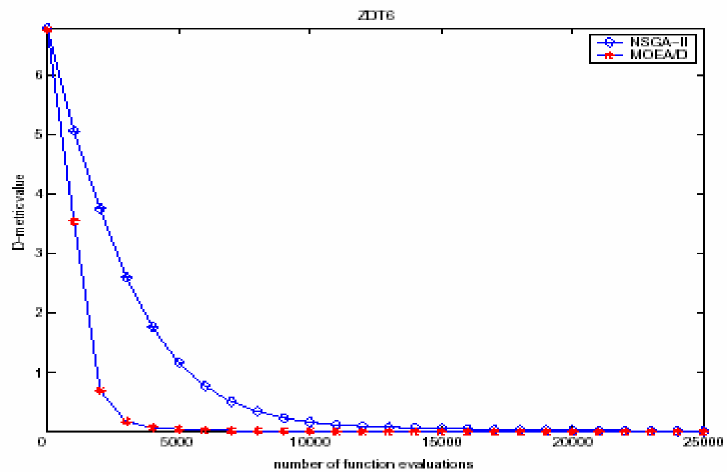
# Comparison With NSGA-II: Experimental Results

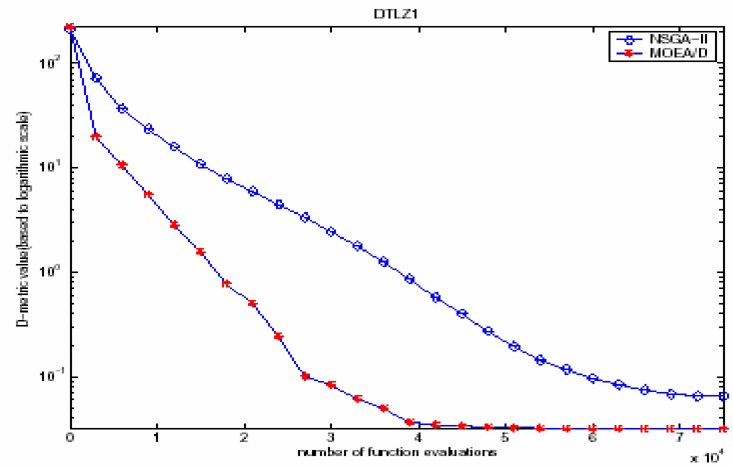
IGD-metric vs. no. of function evaluations

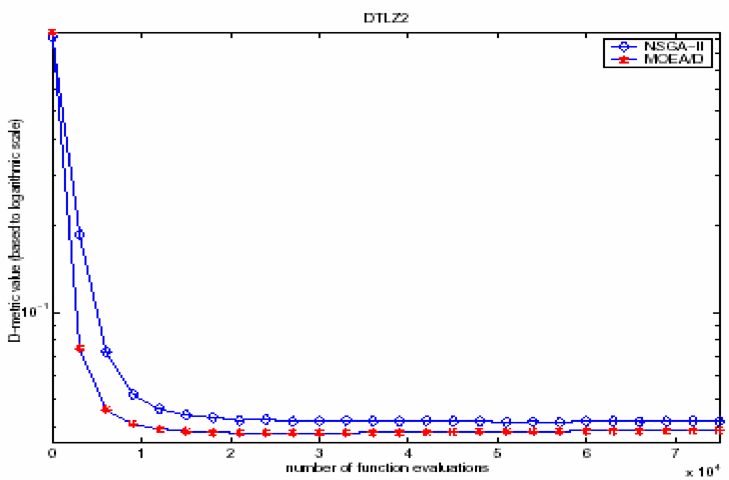



MOEA/D =NSGA-II on these 4 test instances in terms of IGD

# IGD-metric vs. no. of function evaluations

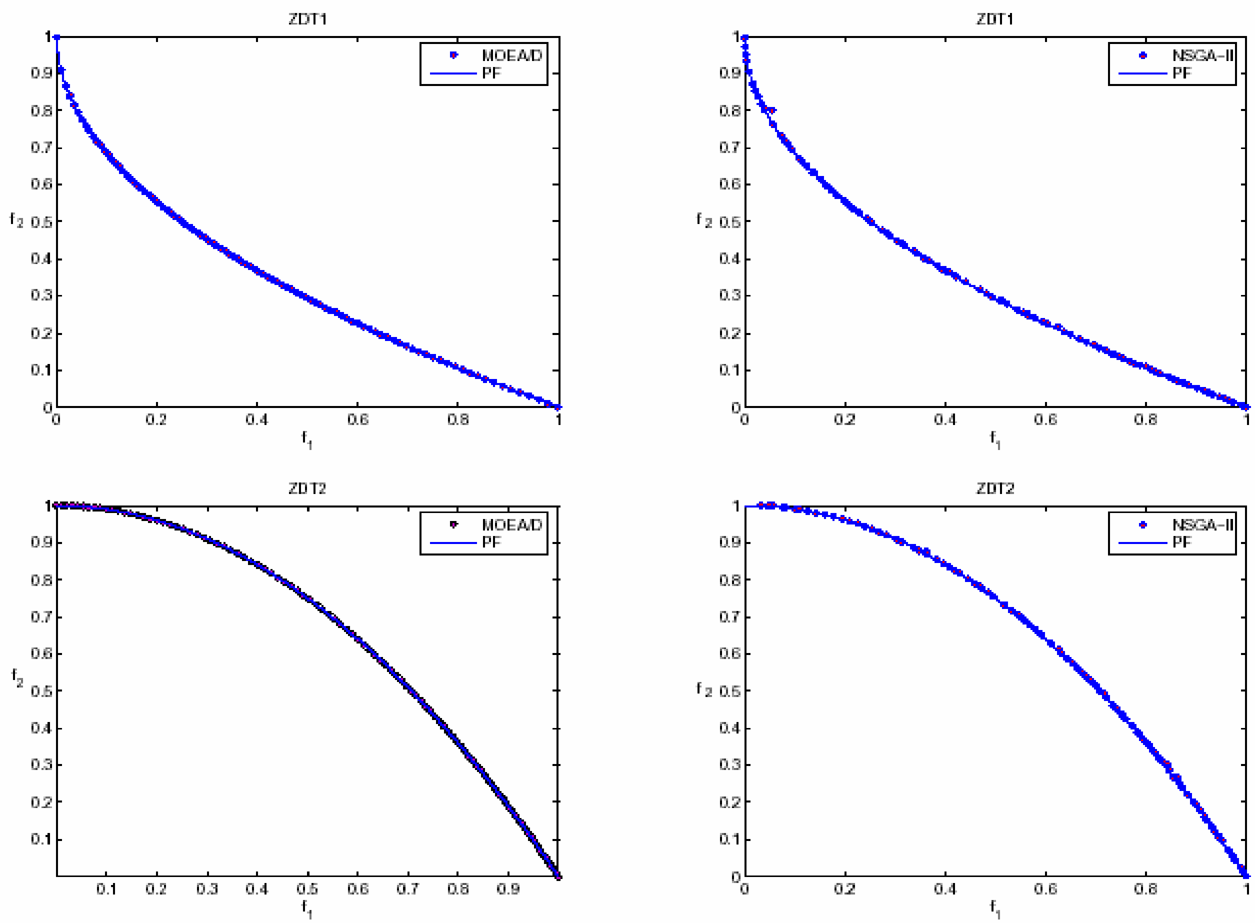


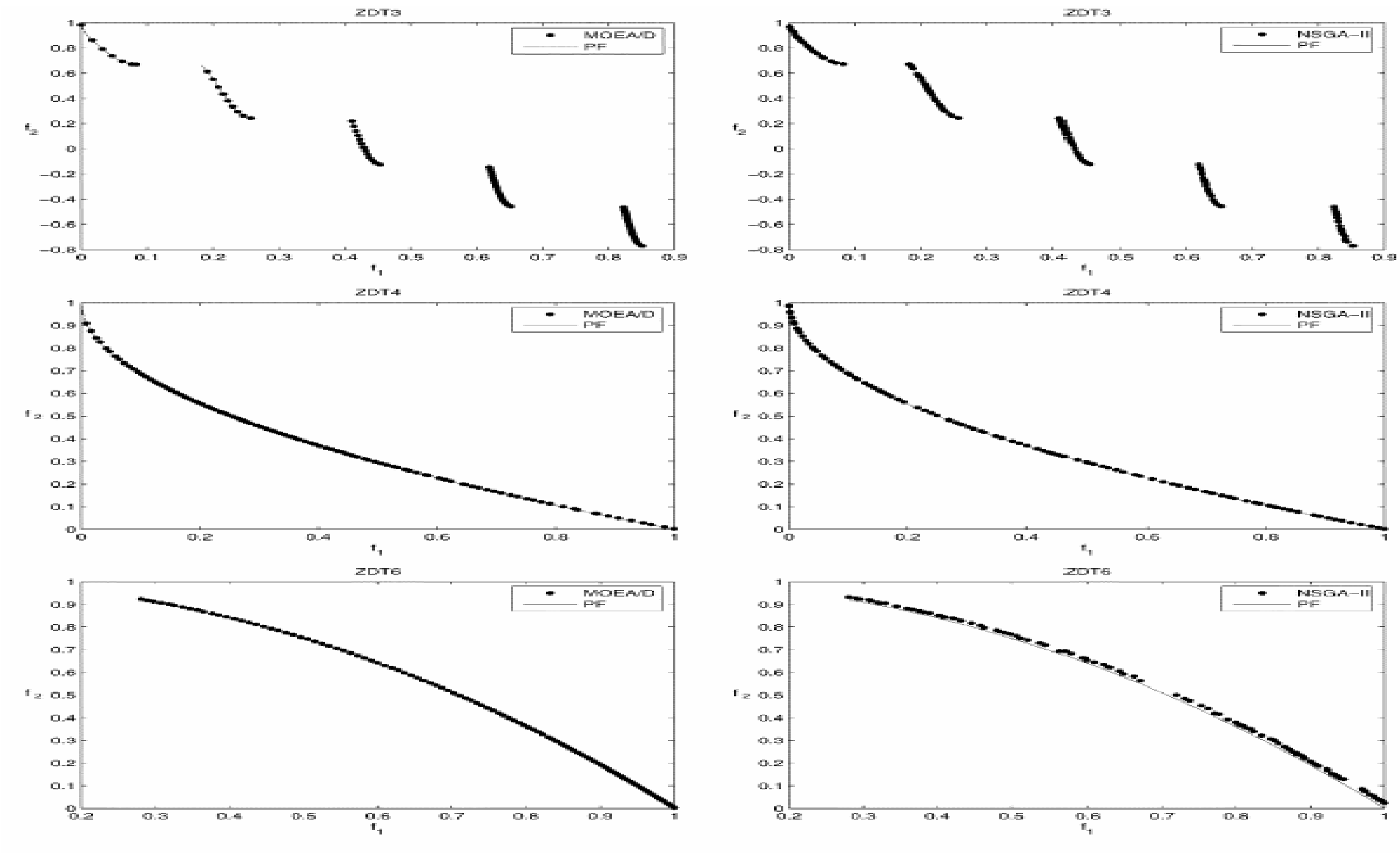




MOEA/D >NSGA-II on these 3 test instances in terms of IGD

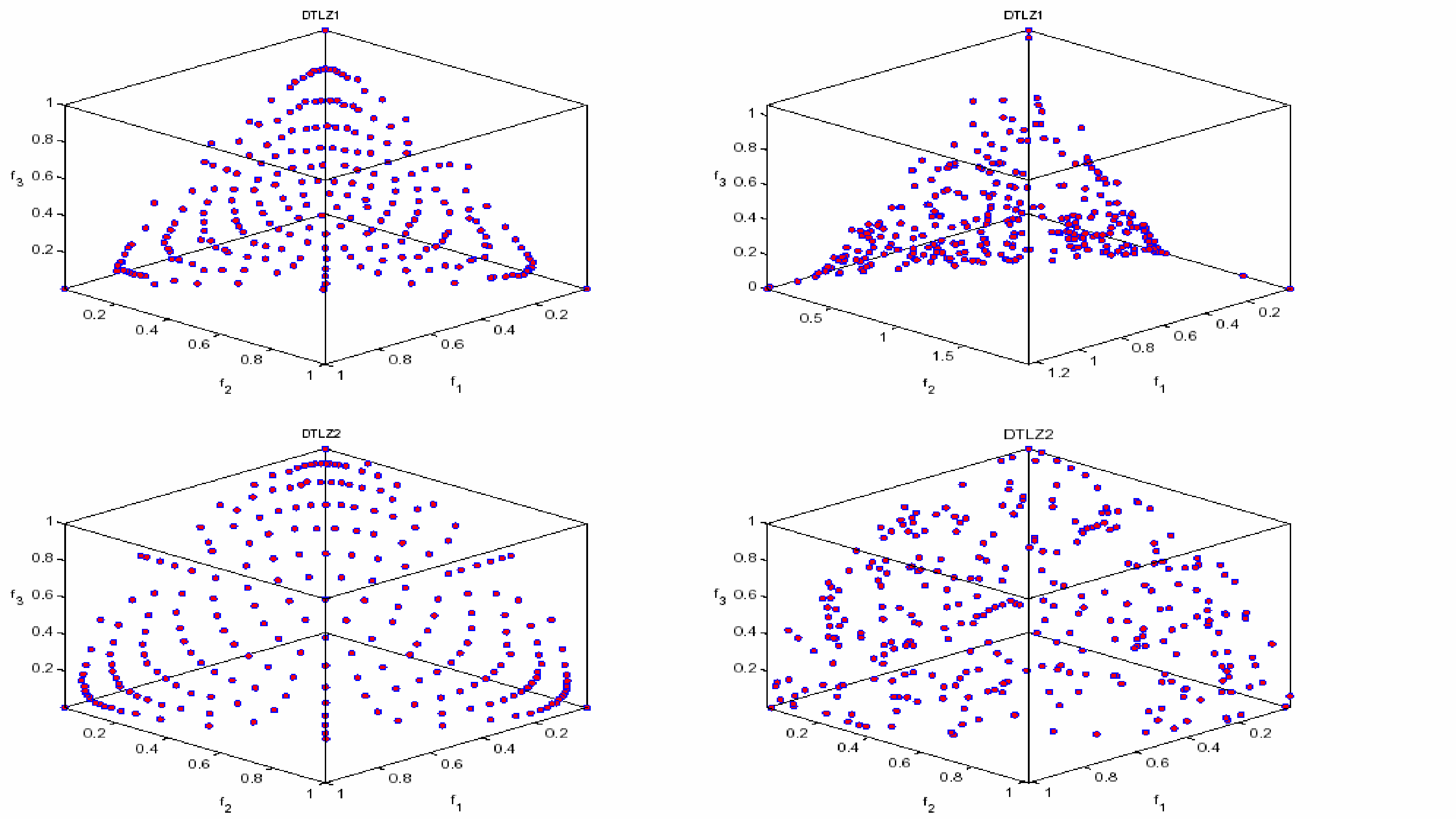
# The best approximation found by each algorithm




ZDT3
MOEA/D
PF
ZDT3
NSGA−II
PF
ZDT4
MOEA/D
PF
ZDT4
NSGA−II
PF
ZDT6
MOEA/D
PF
ZDT6
NSGA−II
PF

## The best approximation found by each algorithm

16




MOEA/D

NSGA-II

MOEA/D >NSGA-II on these 2 test instances

AVERAGE CPU TIME (IN SECOND) USED BY NSGA-II AND MOEA/D WITH THE TCHEBYCHEFF APPROACH.

| | | NSGA-II | MOEA/D |
|---|---|---|---|
| Instance | ZDT1 | 1.03 | 0.60 |
| | ZDT2 | 1.00 | 0.47 |
| | ZDT3 | 1.03 | 0.57 |
| | ZDT4 | 0.77 | 0.33 |
| | ZDT6 | 0.73 | 0.27 |
| | DTLZ1 | 10.27 | 1.20 |
| | DTLZ2 | 8.37 | 1.10 |

MOEA/D is faster than NSGA-II if both of them carry out the same number of function evaluations.

Overall, on continuous test problems,

- MOEA/D with Techebycheff approach is faster than NSGA-II on continuous test problems.
- MOEA/D with Techebycheff approach is slightly better than NSGA-II in terms of solution quality.

The next several slides are about when MOEA/D is much better than other algorithms in terms of solution quality?

# More about MOEA/D

- Advanced Decomposition Method

  Penalty Based Boundary Intersection (PBI)

$$\begin{aligned} \text{minimize} \quad & g^{bip}(x|\lambda, z^*) = d_1 + \theta d_2 \\ \text{subject to} \quad & x \in \Omega, \end{aligned}$$

where

$$d_1 = \frac{\|(z^* - F(x))^T \lambda\|}{\|\lambda\|}$$

and

$$d_2 = \|F(x) - (z^* - d_1 \lambda)\|.$$

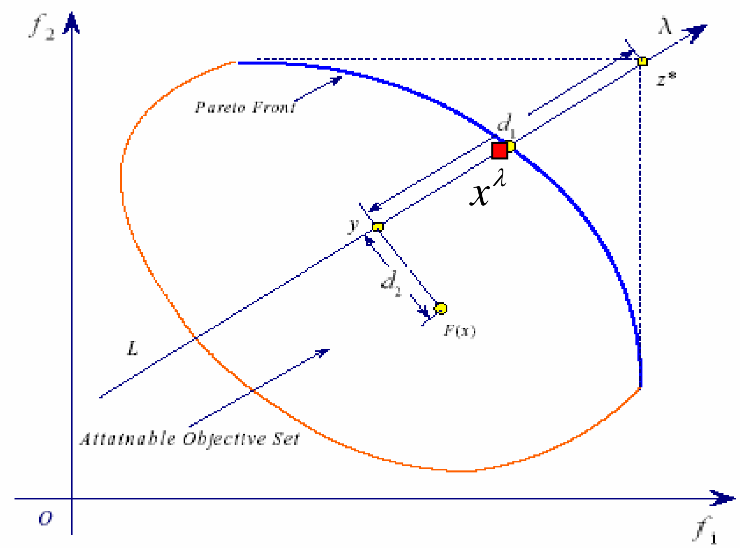



Fig. 2. Illustration of the Penalty based Boundary Intersection Approach.

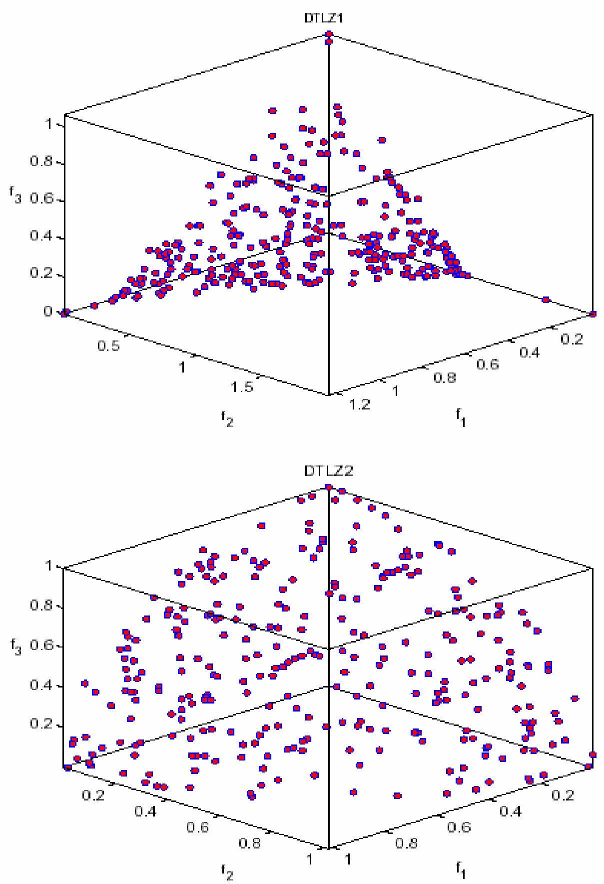
As $\theta \to \infty$, the optimal solution to the above problem $\to x^{\lambda}$

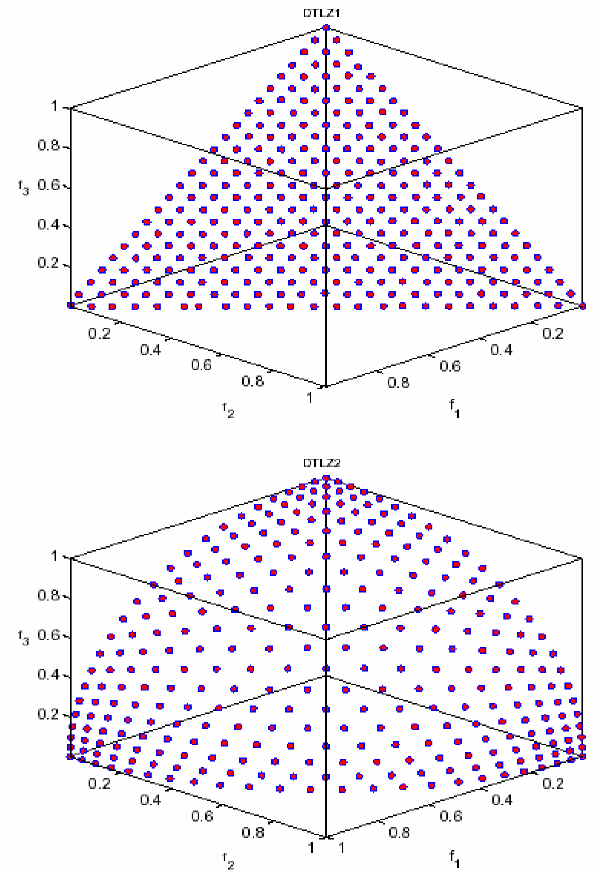
- It is not very sensitive to the shape of PF.
- A set of uniformly distributed direction vectors often can generate a set of uniformly distributed Pareto solutions.

NSGA-II

16

MOEA/D with PBI




MOEA/D with PBI>> NSGA-II

- Small Population
  - o Decision maker may not like large number of solutions.

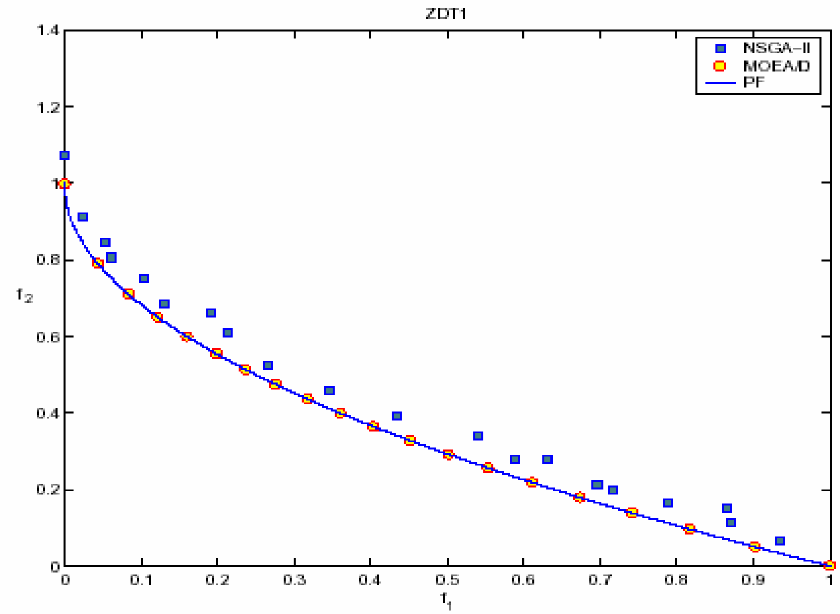



Fig. 13. Plot of the nondominated fronts found by MOEA/D and NSGA-II using small population ($N = 20$).

MOEA/D>>NSGA-II

Pop_size=20, # of generation=250.

- MOEA/D for expensive MOPs

Motivation

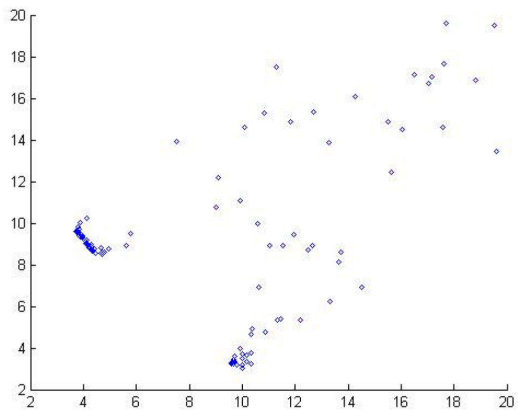
- In many applications, computing the value of each obj is very costly, say, needs one hour.
- How to find a small number of representative Pareto optimal solutions within,  a very small number of, say, 100 function evaluations?

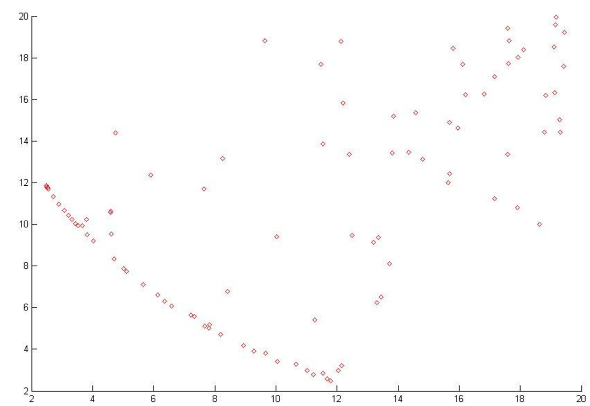
## MOEA/D with MetaModel

Try to solve N subproblems simultaneously.

- Step 0: a small number (say, 10) of points are selected by an experimental design method. Objs are evaluated at these points.
- Step 1: A meta-model are built for each subproblem based on the evaluated points. Then MOEA/D is used for optimise these surrogated subproblems and obtain N candidate points.
- Step 2: Select one (or two) point from these candidates and evaluate the objs at this point.
- If the stopping condition is not met, go to Step 1.

100 evaluations, KNO1 problem (2 obj and 2 variables)



ParEGO

J. Knowles, IEEE Trans on EC 2006

MOEA/D with Metamodel

For details, please refer to:
Wudong Liu ,Qingfu Zhang, Edward Tsang, Botond Virginas, On the Performance of Metamodel Assisted MOEA/D, ISICA 2007.

# Conclusions on MOEA/D

- Bridge between Math Programming and Evolutionary Algorithms
- Local Search can be easily used in MOEA/D

- Downsides:
  - Shape of PF: In case of disconnected, The solutions of some subproblems are not Pareto optimal.

# Contents

# Conclusion

- **Multiobjective optimisation is old.**
- **Population based methods for MOP are still in their very infancy.**
- **Traditional methods (ideas) + Population based algorithms works.**

**Papers and codes can be downloaded from:**

**http://cswww.essex.ac.uk/staff/qzhang/**

**Thanks!**