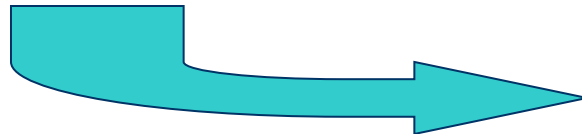




Non-Elitist MOEA

Elitist MOEA



Fundamental Concept of Elitism

- Elite-preserving operator favors the elitist of a population by giving them an opportunity to be directly carried over to the next generation.
- In single objective optimization, crossover is one possible design.
- This makes sure that the fitness of the population-best solution does not deteriorate. In this design, a good solution found early on in the run will never be lost unless a better solution is discovered.
- Elitism can be implemented to different degrees. For example, keep track of the best solution in a population but not use the elitist in any genetic operations. Or, all elites present in the current population can be carried over to the new population. The interesting choice will be an intermediate degree of elitism.

Elitist Non-dominated Sorting GA

- NSGA II has both elite-preservation strategy and an *explicit* diversity-preserving mechanism.
- After creating an offspring population Q_t from the parent population P_t , the two are combined and then a non-dominated sorting is used to classify the entire population R_t of size $2N$. This allows global non-domination check among the offsprings and parent solutions.
- Then the new population of size N is filled with solutions of different non-dominated fronts one at a time starting with the best non dominated front. All fronts which could not be accommodated are just deleted.

- When the last allowed front is being considered, there may exist more solutions than the number of remaining slots in the new population, so a niching strategy is used to choose the members which reside in the least crowded region in that front.
- The above especially useful in the later stages when most of the solutions lie in the first non dominated front, ensuring better spread in the Pareto-optimal front.
- Binary tournament selection (with a crowded tournament operator), recombination and mutation operator are used to create a offspring population.
- Code is available at www.iitk.ac.in/kangal/soft.htm

NSGA II Pseudo Codes

Step 1: Combine parent and offspring populations and create $R_t = P_t \cup Q_t$. Perform a non-dominated sorting to R_t and identify different fronts:

$$F_i, i = 1, 2, \dots, \text{etc.}$$

Step 2: Set new population $P_{t+1} = \emptyset$ Set a counter $i=1$.
Until $|P_{t+1}| + |F_i| < N$, perform $P_{t+1} = P_{t+1} \cup F_i$ and $i=i+1$.

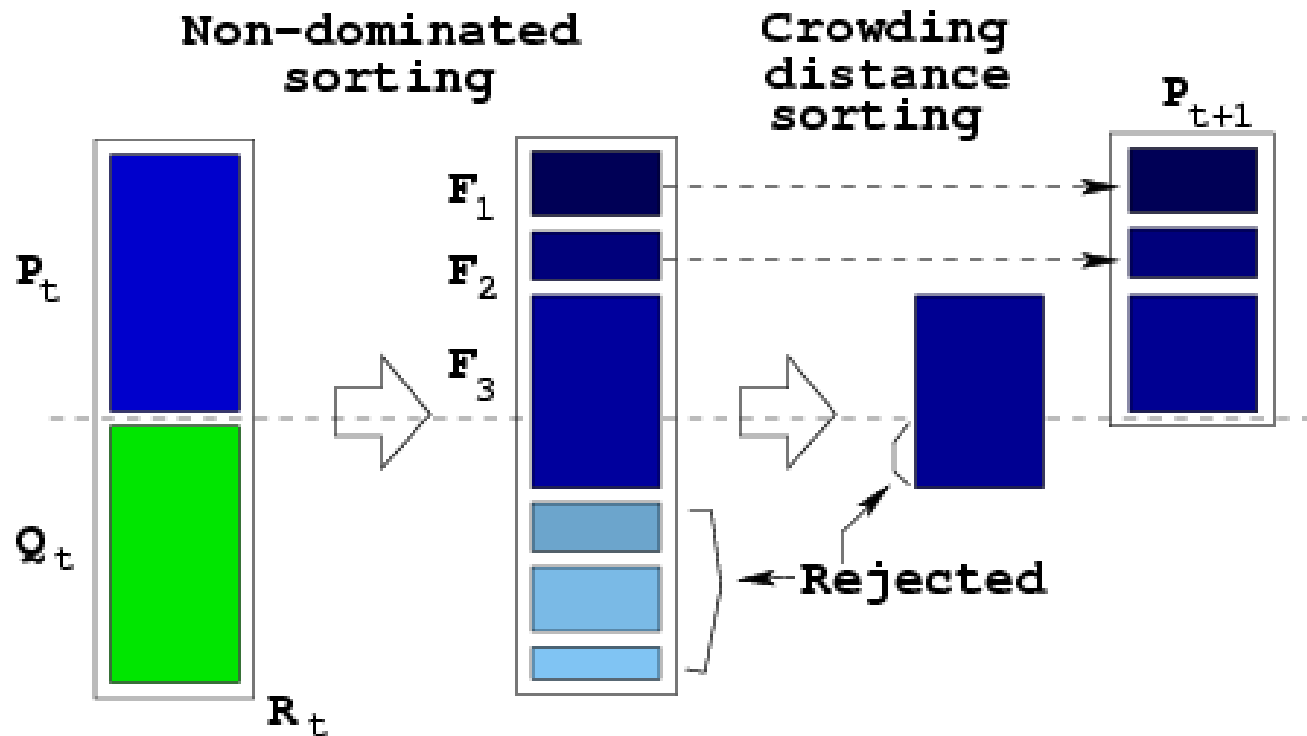
Step 3: Perform the Crowding-sort $(F_i, <_c)$ procedure and include the most widely spread $(N - |P_{t+1}|)$ solutions by using the crowding distance values in the sorted F_i to P_{t+1}

Step 4: Create offspring population Q_{t+1} from P_{t+1} by using the crowded tournament selection, crossover and mutation operators.

NSGA II Procedure

Elites are preserved

Non-dominated solutions are emphasized



Crowded Tournament Selection

- The crowded comparison operator ($<_c$) compares two solutions and returns the winner of the tournament based on the non-dominated rank r_i and the local crowding distance d_i .
- A solution i wins a tournament with another solution j if any of the following conditions are true:
 - $r_i < r_j$
 - $r_i = r_j$ and $d_i > d_j$

Crowding Distance

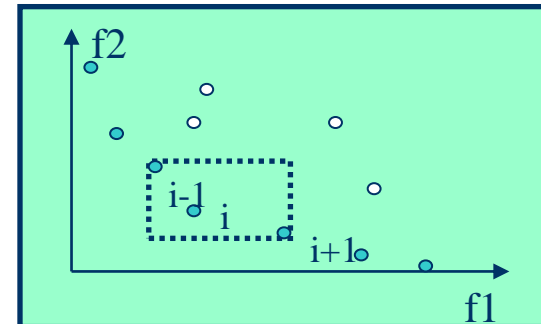
- Crowding-sort ($<_c$)

Step C1: Call the number of solutions in F as $I = |F|$. For each i in the set, first assign $d_i = 0$

Step C2: For each objective function $m = 1, 2, \dots, M$, sort the set in worse order of f_m or, find the sorted indices vector: $I^m = \text{sort}(f_m, >)$.

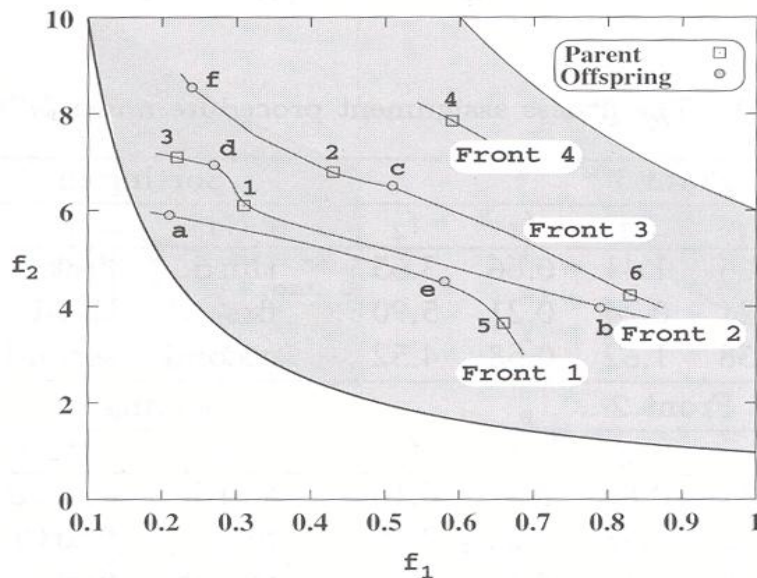
Step C3: For $m=1, 2, \dots, M$, assign a large distance to the boundary solutions, or ∞ , and for all other solutions $j = 2$ to $(I-1)$, assign

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



NSGA II Example

Parent population, P_t					Offspring population, Q_t				
Solution	x_1	x_2	f_1	f_2	Solution	x_1	x_2	f_1	f_2
1	0.31	0.89	0.31	6.10	a	0.21	0.24	0.21	5.90
2	0.43	1.92	0.43	6.79	b	0.79	2.14	0.79	3.97
3	0.22	0.56	0.22	7.09	c	0.51	2.32	0.51	6.51
4	0.59	3.63	0.59	7.85	d	0.27	0.87	0.27	6.93
5	0.66	1.41	0.66	3.65	e	0.58	1.62	0.58	4.52
6	0.83	2.51	0.83	4.23	f	0.24	1.05	0.24	8.54



$R_t = \{1, 2, 3, 4, 5, 6, a, b, c, d, e, f\}$

Four non-dominated fronts of the combined population R_t ;

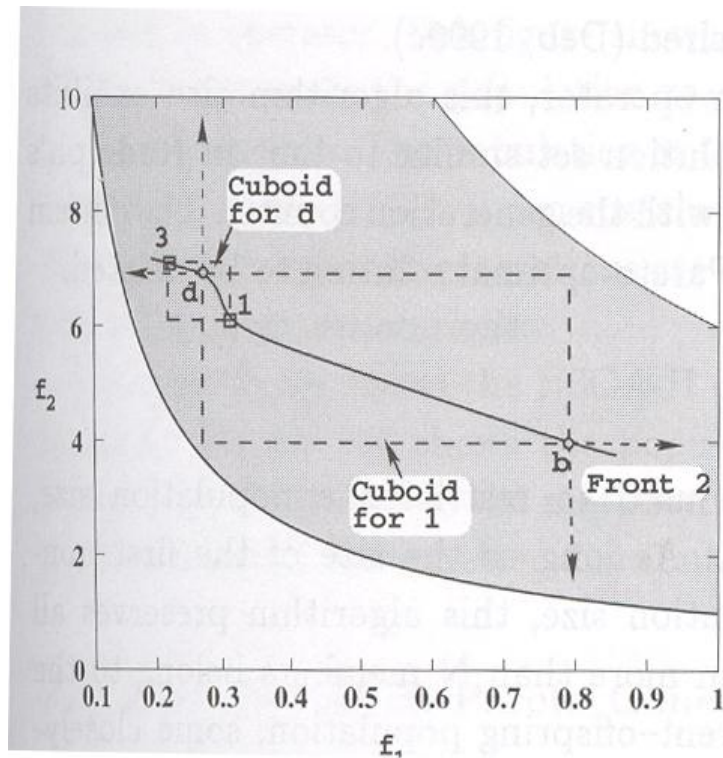
$F_1 = \{5, a, e\} \rightarrow P_{t+1}$

$F_2 = \{1, 3, b, d\}$ stop!

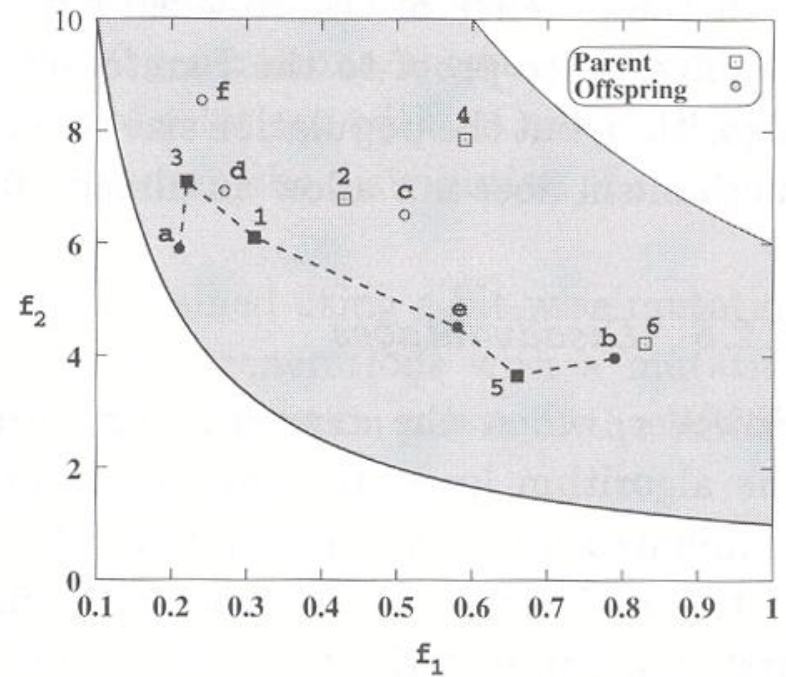
$F_3 = \{2, 6, c, f\}$

$F_4 = \{4\}$

Front 1					Sorting in		Distance
Solution	x_1	x_2	f_1	f_2	f_1	f_2	
5	0.66	1.41	0.66	3.65	third	first	∞
a	0.21	0.24	0.21	5.90	first	third	∞
e	0.58	1.62	0.58	4.52	second	second	0.54
Front 2					Sorting in		Distance
Solution	x_1	x_2	f_1	f_2	f_1	f_2	
1	0.31	0.89	0.31	6.10	third	second	0.63
3	0.22	0.56	0.22	7.09	first	fourth	∞
b	0.79	2.14	0.79	3.97	fourth	first	∞
d	0.27	0.87	0.27	6.93	second	third	0.12

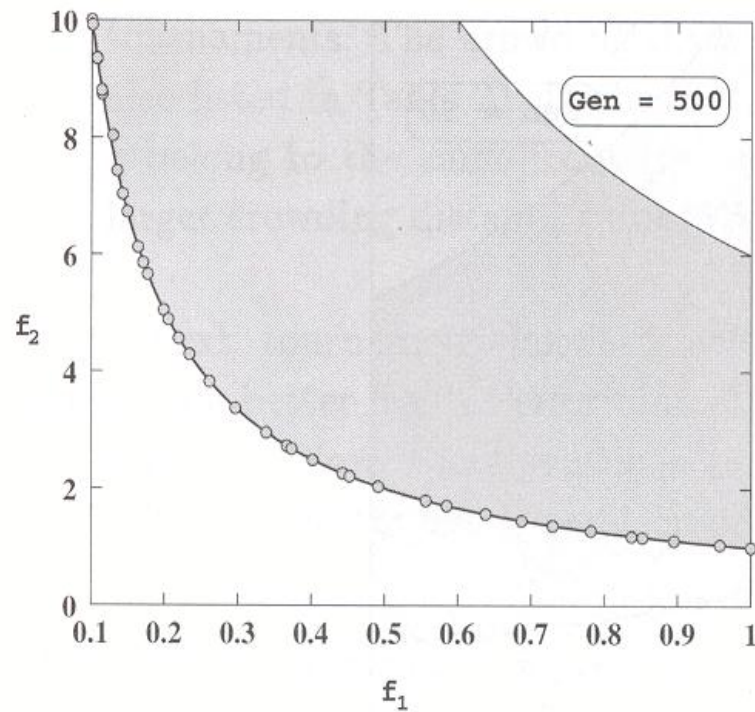


The cuboids of solutions 1 and d.
The cuboids of 3 and b extend to infinity

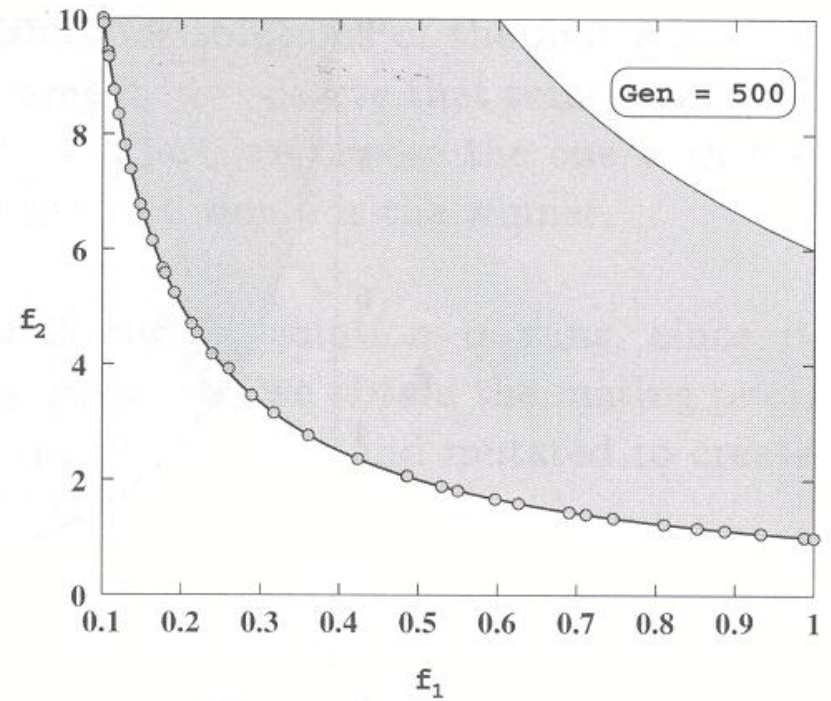


The parent population P_{t+1}
joined by line.

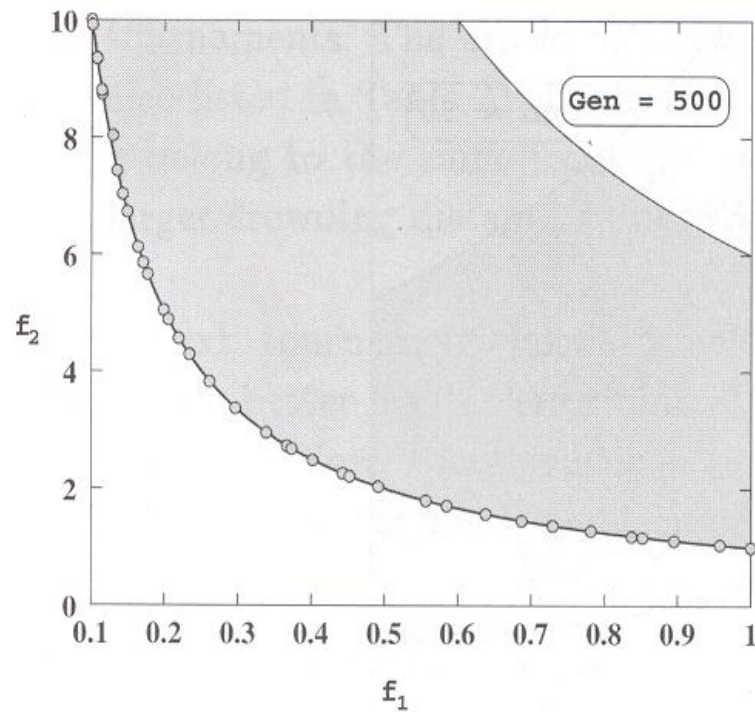
- Tournament selection to generate the next offspring
 - Assume we pair the solutions (5,e), (a,3), (1,b), (a,1), (e,b), and (3,5), so that each solution participates in exactly two tournaments.
 - In the first tournament, we observe that solution 5 and e belongs to the same front. Thus we choose the one with the larger crowding distance value. Solution 5 is the winner.
 - In the next, solution a wins. We obtain the mating pool: {5, a, b, a, e, 5}. Now, these solutions can be mated pair-wise and mutated to create Q_{t+1} . This complete one generation of the NSGA-II



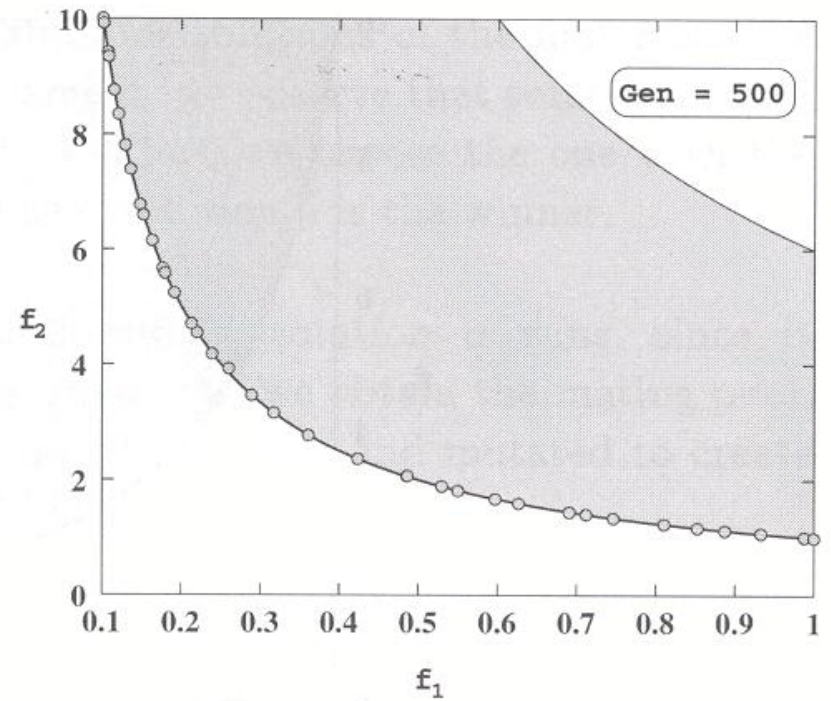
Population after 500 generations
without mutation.



Population after 500 generations
with mutation



Population after 500 generations
without mutation.

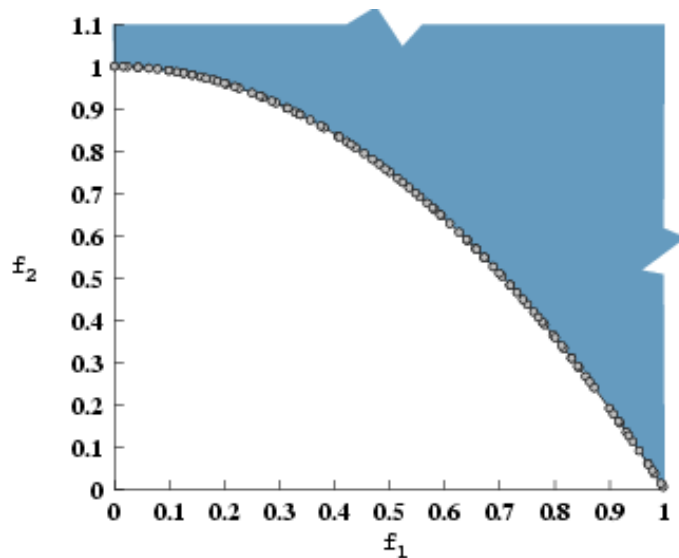


Population after 500 generations
with mutation

$$(\text{Min}) \quad f_1(x) = x_1$$

$$(\text{Min}) \quad f_2(x) = g \left[1 - \left(\frac{f_1}{g} \right)^2 \right]$$

$$\text{Where} \quad g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

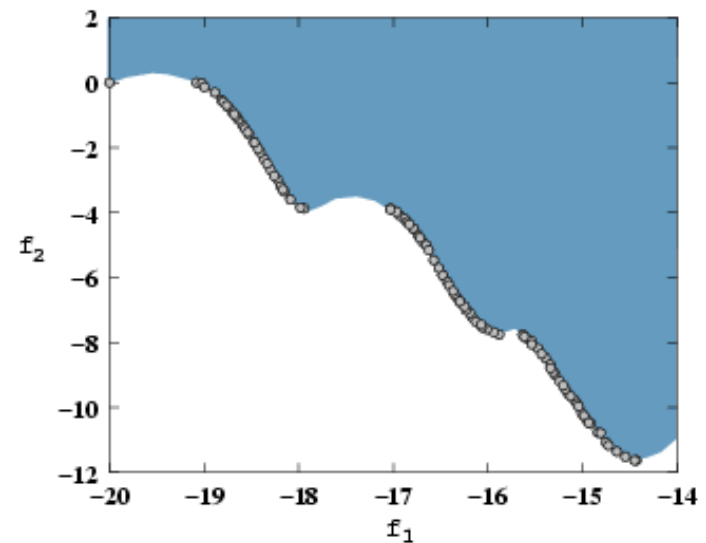


$$(\text{Min})$$

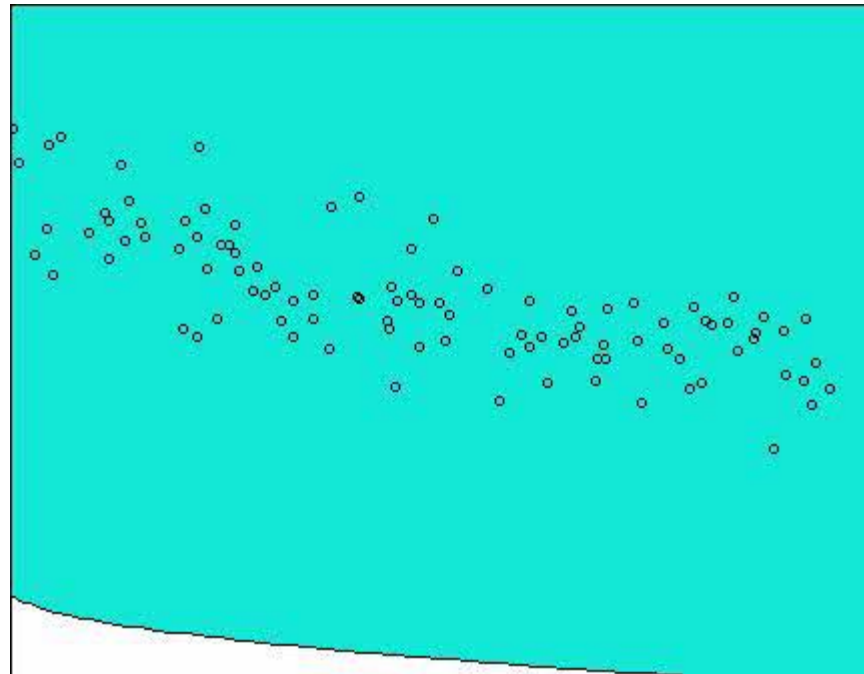
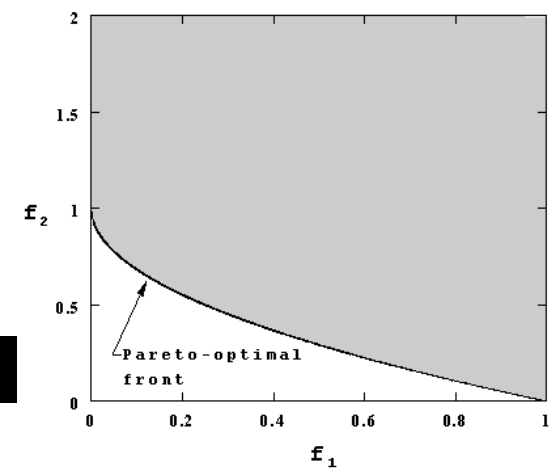
$$f_1(x) = x_1$$

$$(\text{Min}) \quad f_2(x) = g \left[1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \sin(10\pi f_1) \right]$$

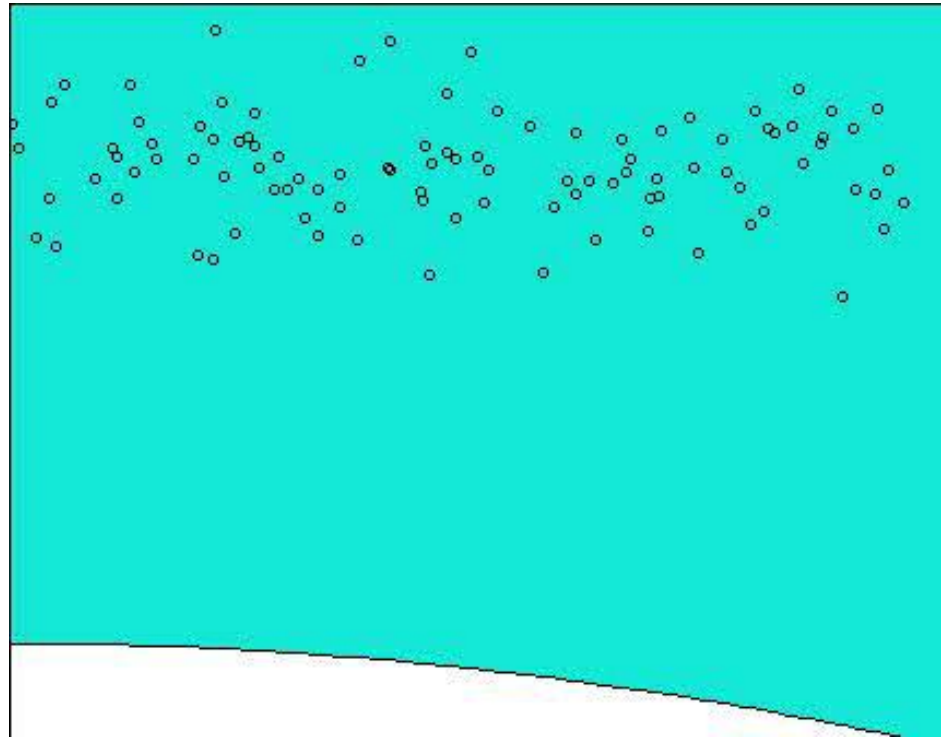
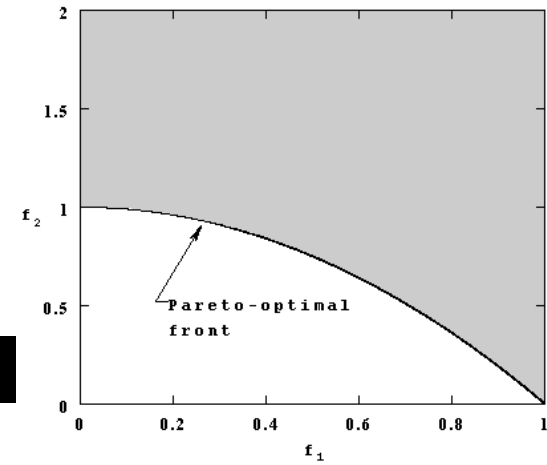
$$\text{Where} \quad g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$



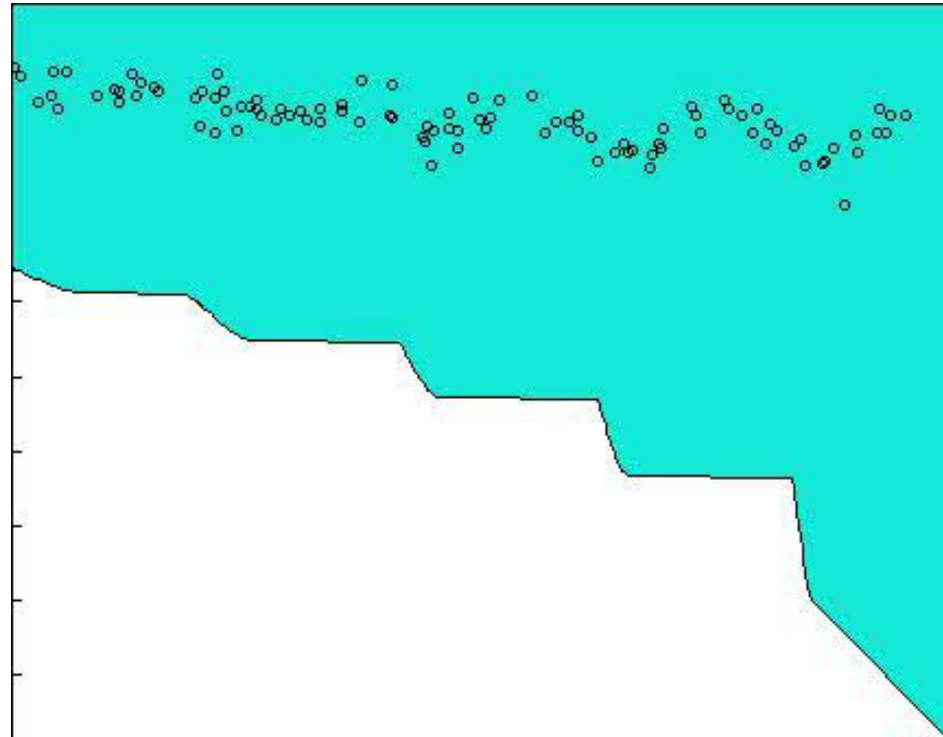
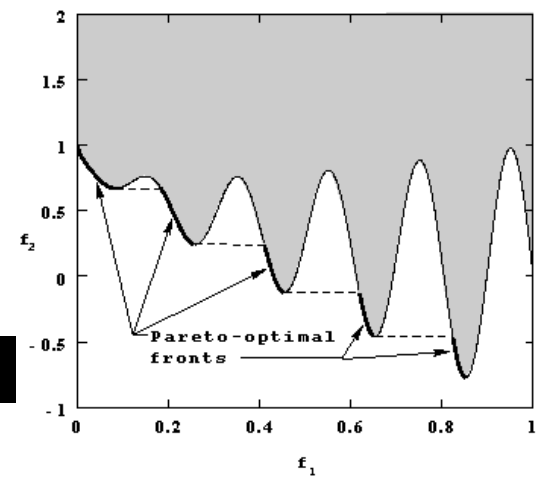
NSGA II for ZDT1



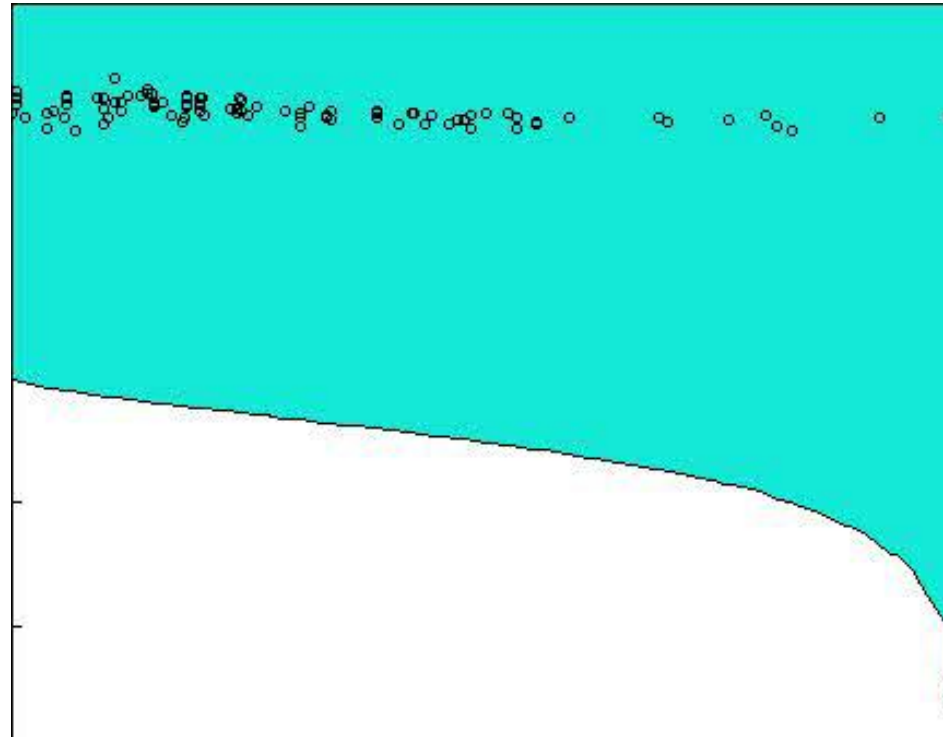
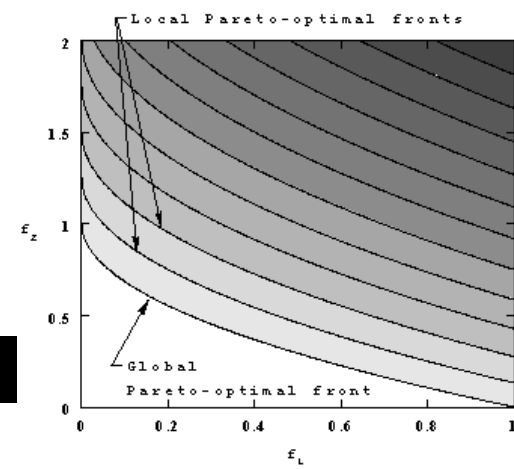
NSGA II for ZDT2



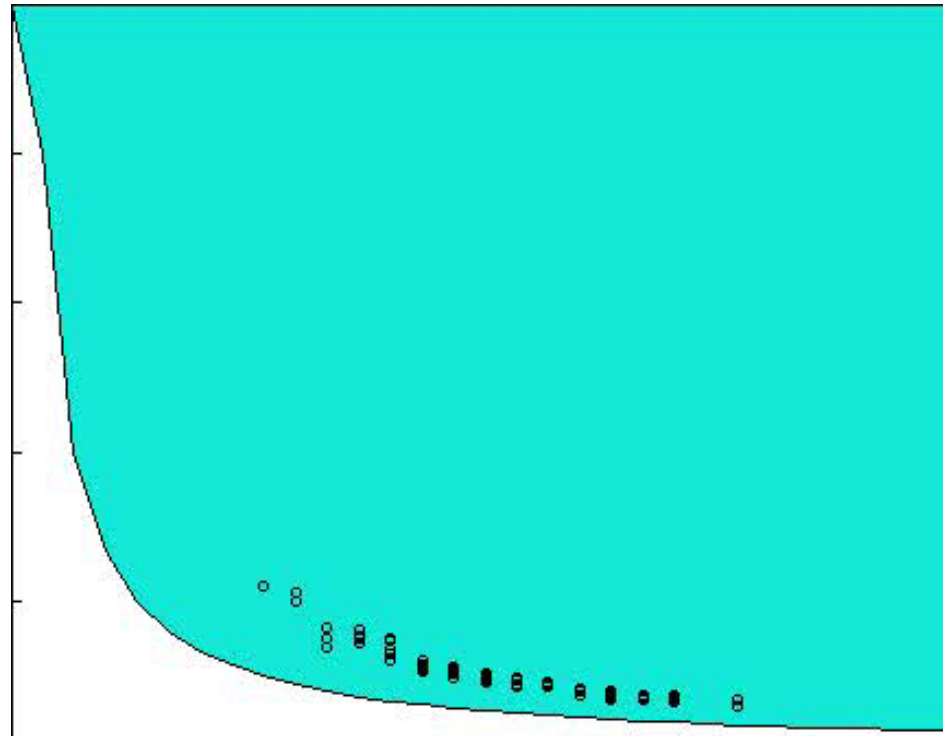
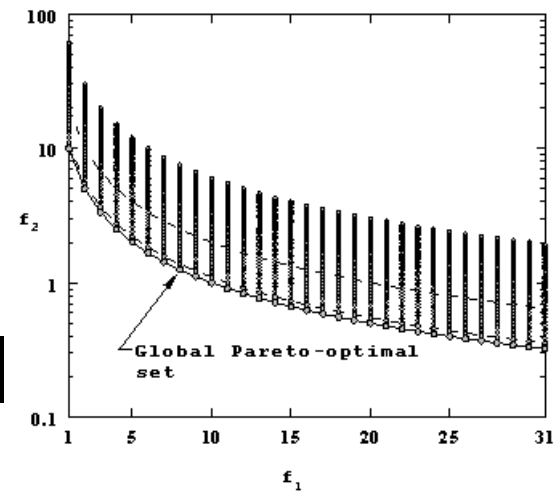
NSGA II for ZDT3



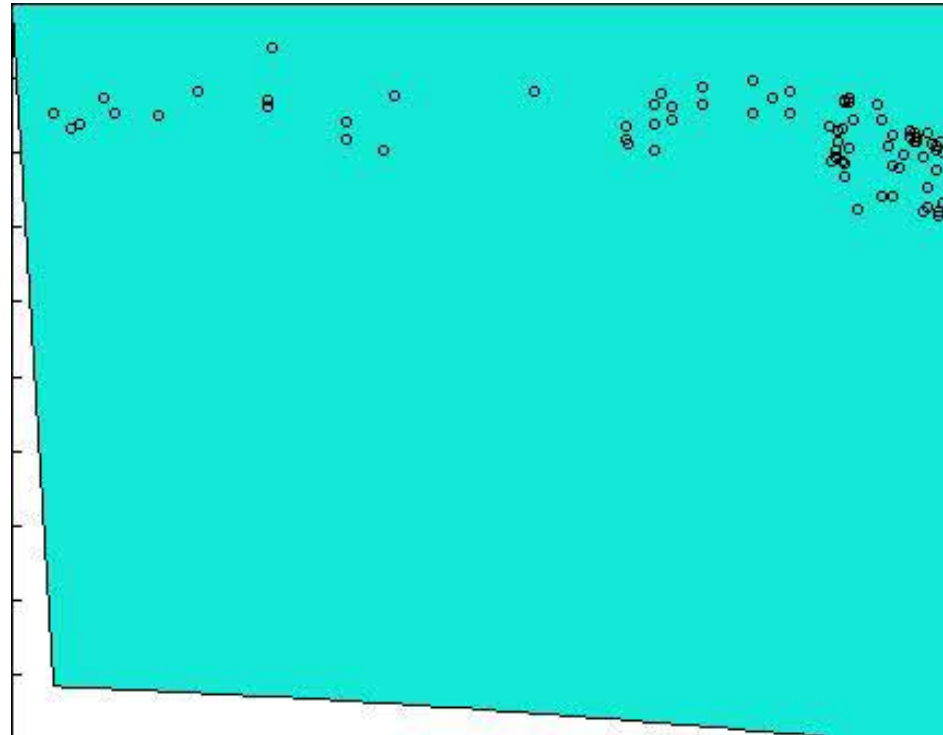
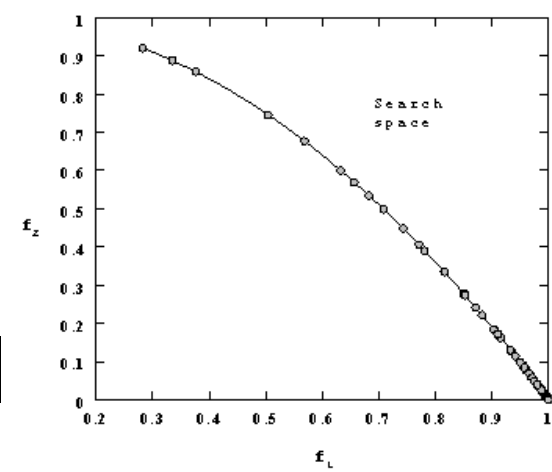
NSGA II for ZDT4



NSGA II for ZDT5



NSGA II for ZDT6



NSGA II Advantages

- The diversity among non-dominated solutions is introduced by using the crowding comparison procedure which is used with the tournament selection and in the population reduction phase.
- Since solutions compete with their crowding distances no sharing parameter is required.
- The elitism mechanism does not allow an already found Pareto-optimal solution to be deleted.

NSGA II Disadvantages

- The crowded comparison used to restrict the population size, reduces the convergence property of the algorithm in the case when the population of the first Non-dominated exceeds the population size.
- Closely packed Pareto-optimal solutions may be eliminated for other non-dominated but non-Pareto-optimal solutions, and again these may be dominated by Pareto-optimal solutions in later generations. This cycle may keep continuing sometimes.
- The non dominated sorting need to be performed on a population size of $2N$ rather than a size of N as in most other algorithms.
- Computational complexity: $O(MN^2)$

Distance Based Pareto GA (DPGA)

- Using one fitness measure to emphasize the progress toward the Pareto-optimal front and the diversity along the obtained front.
- Two populations: one standard GA population P_t where genetic operations are performed and another elite population E_t containing all non-dominated solutions found thus far.
- A non-dominated solution lying a large distance away from the existing elite set gets a large (better) fitness. This helps in two ways. First, if the new solution dominates a few members of the elite set, the fitness assignment procedure helps in emphasizing solutions closer to the Pareto-optimal set. A distance solution here means a solution distant from the existing elite set but closer to the Pareto-optimal set. Assigning a large fitness to such a solution helps to progress towards the Pareto-optimal front. On the other hand, if the new solution lies in the same non-dominated front along with the elite solutions, the fitness assignment procedure helps in maintaining diversity among them. A distance solution here means an isolated solution on the same front. Assigning a large fitness to an isolated solution helps to maintain diversity among obtained non-dominated solutions.

DPGA Pseudo Code

Step 1: Create an initial random population P_0 of size N and set the fitness of the first solution as F_1 . Set generation counter $t = 0$.

Step 2: If $t = 0$, insert the first element of P_0 in an elite set $E_0 = \{1\}$. For each population member $j \geq 2$ for $t = 0$ and $j \geq 1$ for $t > 0$, perform the following steps to assign a fitness value.

Step 2a: Calculate the distance $d_j^{(k)}$ with each elite member k (with fitness $e_m^{(k)}$, $m=1, 2, \dots, M$) as follows:

$$d_j^{(k)} = \sqrt{\sum_{m=1}^M \left(\frac{e_m^{(k)} - f_m^{(j)}}{e_m^{(k)}} \right)^2}$$

Step 2b: Find the minimum distance and the index of the elite member closest to solution j :

$$d_j^{\min} = \min_k d_j^{(k)}$$

$$k_j^* = \{k : d_j^{(k)} = d_j^{\min}\}$$

- Step 2c: If any elite member dominates solution j , the fitness of j is

$$F_j = \max \left[0, F(e^{(k_j^*)}) - d^{\min} \right]$$

Otherwise, the fitness of j is:

$$F_j = F(e^{(k_j^*)}) + d^{\min}$$

and j is included in E_t by eliminating all elite members that are dominated by j .

Step 3: Find the maximum fitness value of all elite members:

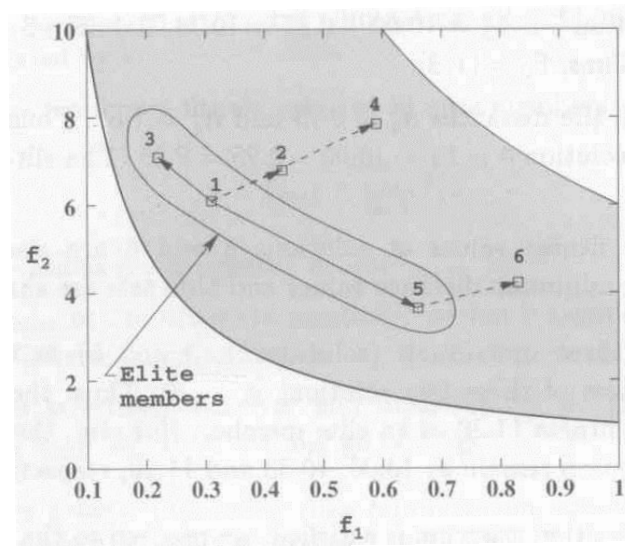
$$F_{\max} = \max_k F_i$$

All elite solutions are assigned a fitness F_{\max} .

Step 4: If $t < t_{\max}$ or any other termination criterion is satisfied, the process is complete. Otherwise, go to Step 5.

Step 5: Perform selection, crossover, and mutation on P_t and create a new population P_{t+1} . Set $t = t+1$ and go to Step 2.

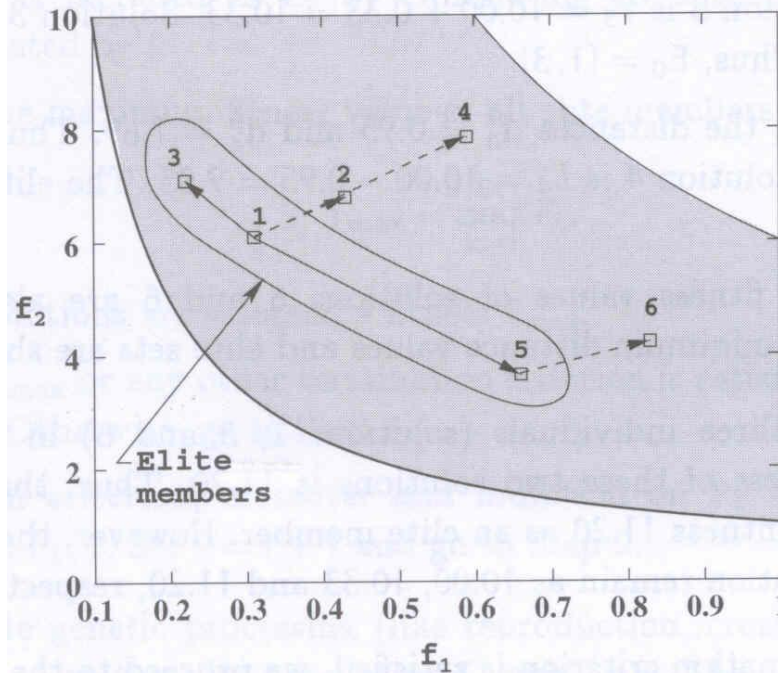
DPGA Example



Solution	x_1	x_2	f_1	f_2
1	0.31	0.89	0.31	6.10
2	0.43	1.92	0.43	6.79
3	0.22	0.56	0.22	7.09
4	0.59	3.63	0.59	7.85
5	0.66	1.41	0.66	3.65
6	0.83	2.51	0.83	4.23

Solution (j)	GA population						Fitness	E_0
	x_1	x_2	f_1	f_2	d_j^{\min}			
1	0.31	0.89	0.31	6.10	—		10.00	{1}
2	0.43	1.92	0.43	6.79	0.40		9.60	{1}
3	0.22	0.56	0.22	7.09	0.33		10.33	{1,3}
4	0.59	3.63	0.59	7.85	0.95		9.05	{1,3}
5	0.66	1.41	0.66	3.65	1.20		11.20	{1,3,5}
6	0.83	2.51	0.83	4.23	0.30		10.90	{1,3,5}

Fitness Assignment in DPGA



$\text{fitness}(\text{ind}_1) = 10.0$; $\text{fitness}(\text{ind}_2) = 9.60$; $\text{fitness}(\text{ind}_3) = 10.33$;
 $\text{fitness}(\text{ind}_4) = 9.05$; $\text{fitness}(\text{ind}_5) = 11.20$; $\text{fitness}(\text{ind}_6) = 10.90$.

DPGA Advantage & Disadvantage

- For some sequence of fitness evaluations, both goals of progressing towards the Pareto-optimal front and maintaining diversity among solutions are achieved without an explicit niching method.
- The elite size is not restricted and is allowed to grow to any size. This will increase the computational complexity of the algorithm (restricting the elite size?).
- DPGA fitness assignment scheme is sensitive to the ordering of individuals in a population.
- Computational complexity: $O(M\eta^2)$

Strength Pareto EA (SPEA)

- Algorithm maintains elitism by explicitly maintaining an external population \bar{P}
- This population stores a fixed number of non-dominated solutions.
- At every generation, newly found non-dominated solutions are compared with the existing external population and the resulting non-dominated solutions are preserved
- Elites participate in genetic operations along with the current population in the hope of steering population towards the Pareto front

- The algorithm begins with a randomly created population P_0 of size N and an empty external population P'_0 with a maximum capacity of N'
- In any generation t the best non-dominated solutions (belonging to the first non-dominated front) of the population P_t are copied to the external population P'_t .
- Then the dominated solutions in the modified external population are found and deleted from the external population. \Rightarrow previously found elites which are now dominated by a new elite solution are deleted from the external population.
- Therefore external population contains the best non-dominated solutions of both old and new elites.
- To avoid external population overcrowded, the size of the external population is bounded to a limit, N' . That is, when the size of the external population is less than N' , all elites are kept in the population. When the size exceed N' , not all elites can be accommodated in the external population. \Rightarrow Elites which are less crowded in the non-dominated front are kept. A clustering method is designed.

- Once the new elites are preserved for the next generation, the algorithm then turns to the current population and uses genetic operators to find a new population.
- In addition to the assigning of fitness to the current population members, fitness is also assigned to external population members. In fact, SPEA assigns a fitness (called the strength) S_i to each member i of the external population first. The strength is proportional to the number (n_i) of current population members that are external solution i dominates

$$S_i = \frac{n_i}{N + 1}$$

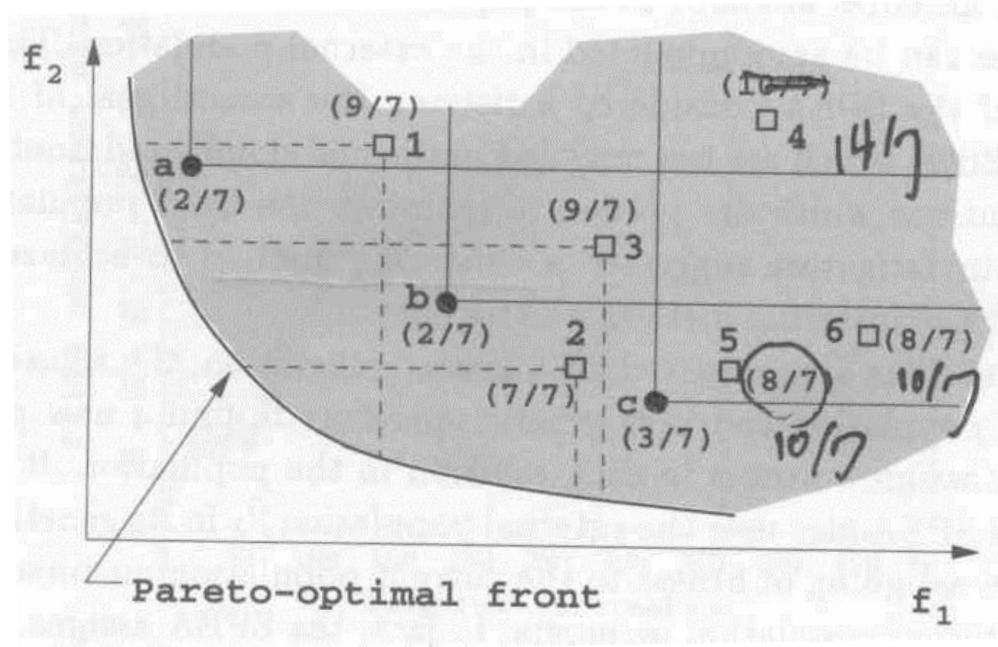
Fitness of the external population
- In other words, SPEA assigns more strength to an elite which dominates more solutions in the current population. Division by $(N+1)$ ensures that the maximum value of the strength of any external population members is never one or more. A non-dominated solution dominating a fewer solutions has a smaller (or better) fitness.

- Therefore, the fitness of a current population member j is assigned as one more than the sum of the strength values of all external population members which dominate j

$$F_j = 1 + \sum_{i \in \bar{P}_t \wedge i \leq j} S_i$$

Fitness of the
main population

- The addition of one makes the fitness of any current population member P_t to be more than the fitness of any external population member P'_t . This method of fitness assignment suggests that a solution with a smaller fitness is better.
- With these fitness values, a binary tournament selection procedure is applied to the combined $(P'_t \cup P_t)$ population to choose solution with smaller fitness values. As usual, crossover and mutation operators are applied to the mating pool, and a new population P_{t+1} of size N is created.
- Computational complexity: $O(MN^2)$



SPEA Pseudo Codes

Step 1: Find the best non-dominated set of the current population P_t (i.e., $F_1(P_t)$) and copy it to the external population P'_t , or perform $P'_t = P'_t \cup F_1(P_t)$

Step 2: Find the best non-dominated solutions of the modified external population P'_t (i.e., $F_1(P'_t)$) and delete all dominated solutions from the external population (i.e., $P'_t = F_1(P'_t)$)

Step 3: If $|P'_t| > N'$, use a clustering technique to reduce the population size to N' . Otherwise, keep P'_t unchanged. The resulting population is the external population P'_{t+1} of the next generation.

Step 4: Assign fitness to each elite solution $i \in P'_{t+1}$ by using $S_i = \frac{n_i}{N+1}$ and for the current population using $F_j = 1 + \sum_{i \in P'_{t+1} \wedge i \leq j} S_i$

Step 5: Apply a binary tournament selection with these fitness values, a crossover and a mutation operator to create the new population P_{t+1} of size N from the combined population $(P'_{t+1} \cup P_t)$ of size $(N' + N)$.

Clustering Algorithm

Step C1: Initially, each solution belongs to a distinct cluster or $C_i = \{i\}$, so that

$$C = \{C_1, C_2, \dots, C_{N'}\}$$

Step C2: If $|C| \leq N'$, go to Step 5, otherwise go to Step 3

Step C3: For each pair of clusters, calculate the cluster distance by using

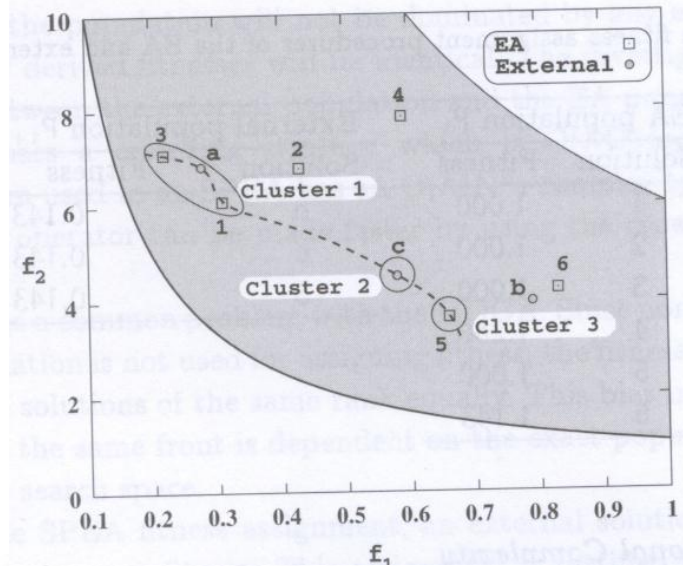
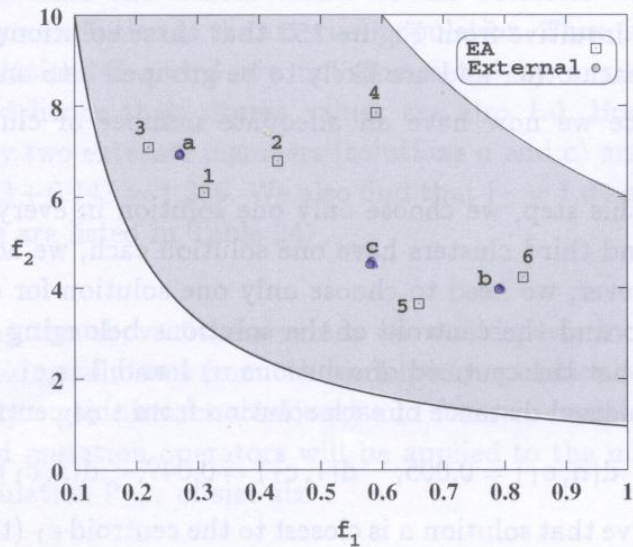
$$d_{12} = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d(i, j)$$

where $d(i, j)$ is the Euclidian distance between the points i, j and find the pair (i_1, i_2) which corresponds to the minimum cluster distance.

Step C4: Merge the two clusters C_1 and C_2 together. This reduces the size of C by one, go to step C2.

Step C5: Choose only one solution from each cluster and remove the others from the clusters. The solution having the minimum average distance from other solutions in the cluster can be chosen as the representative solution of a cluster.

SPEA Example



EA population P_t					External population \bar{P}_t				
Solution	x_1	x_2	f_1	f_2	Solution	x_1	x_2	f_1	f_2
1	0.31	0.89	0.31	6.10	a	0.27	0.87	0.27	6.93
2	0.43	1.92	0.43	6.79	b	0.79	2.14	0.79	3.97
3	0.22	0.56	0.22	7.09	c	0.58	1.62	0.58	4.52
4	0.59	3.63	0.59	7.85					
5	0.66	1.41	0.66	3.65					
6	0.83	2.51	0.83	4.23					

- $P_t = \{1, 2, 3, 4, 5, 6\}$ and $P'_t = \{a, b, c\}$, $N=6$, $N'=3$.
- Step 1: Find the non-dominated solutions of P_t , $F_1(P_t) = \{1, 3, 5\}$.
 $P'_t = \{a, b, c, 1, 3, 5\}$
- Step 2: Find the non-dominated solutions of this modified population, $F_1(P'_t) = \{a, c, 1, 3, 5\}$. This is the new external population.
- Step 3: Since the size of P'_t is 5, which is greater than the allowed space $N' = 3$, we need to use the clustering algorithm to find which three will remain in the external population.
 - Step C1: Initially, all five solutions belong to a separate cluster.
 - Step C2: Since there are 5 clusters, we move to Step C3.
 - Step C3: We use $f_1^{\max}=1$, $f_1^{\min}=0.1$, $f_2^{\max}=60$, $f_2^{\min}=1$. and find $d_{12}=0.35$, $d_{13}=0.05$, $d_{14}=0.06$, $d_{15}=0.44$, $d_{23}=0.30$, $d_{24}=0.40$, $d_{25}=0.09$, $d_{34}=0.09$, $d_{35}=0.39$, $d_{45}=0.49$
 we observe that minimum cluster distance occurs between the first and the third clusters.

- Step C4: we merge these cluster together and have only four clusters.
- Step C2: Since there are four clusters, we move to Step C3 to reduce one more cluster.
- Step C3: Now average distance between the first and second clusters is the average distance between the two pairs of solutions (a, c) and (1, c). The distance between a and c is 0.35 and that between solutions 1 and c is 0.30. Thus, the average distance $d_{12}=0.325$. Similarly, we find $d_{12}=0.325$, $d_{13}=0.075$, $d_{14}=0.415$, $d_{23}=0.400$, $d_{24}=0.090$, and $d_{34}=0.490$. The minimum distance occurs between clusters 1 and 3.
- Step C4: The we merge clusters 1 and 3 and have the following three clusters: $C_1=\{a, 1, 3\}$, $C_2=\{c\}$, and $C_3=\{5\}$
- Step C5: We choose only one solution in every cluster. Since the second and third clusters have only one solution each, we accept them as they are. However, we need to choose only one solution for cluster 1. The first step is to find the centroid of the solutions belong to the cluster, $c_1=(0.27, 6.71)$. Now the normalized distance of each solution from this centroid is as follows: $d(a,c_1)=0.005$, $d(1,c_1)=0.049$, and $d(3,c_1)=0.052$.

- We observe that solution a is closest to the centroid. Thus we choose solution a and delete solutions 1 and 3 from this cluster. There the new external population is $P_{t+1}' = \{a, c, 5\}$.

- Step 4: We assign fitness values to the solutions of populations P_t and P_{t+1}' . First we concentrate on the external population. We observe that solution a dominates only one solution (solution 4) in P_t . Thus, its fitness is assigned as $F_a = 1/(6+1) = 0.143$ ($n_a = 1$). Similarly, we find $n_c = n_5 = 1$, and their fitness values are also $F_c = F_5 = 0.143$.
- Next we calculate the fitness values of the solutions of P_t . Solution 1 is dominated by no solution in the external population. Thus, its fitness is $F_1 = 1.0$. Similarly, solutions 2 and 3 are not dominated by any external population members and hence their fitness values are also 1.0. However, solution 4 is dominated by two external members (solutions a and c) and hence its fitness is $F_4 = 1 + 0.143 + 0.143 = 1.286$. We also find that $F_5 = 1.0$ and $F_6 = 1.143$.

- Step 5: Now, using the above fitness values we would perform six tournament s by randomly picking solutions from the combined populations of size nine and form the mating pool. Thereafter, crossover and mutation operators will be applied to the mating pool to create the new population P_{t+1} of size six.

Advanced Strength Pareto EA (SPEA2)

- Areas for improvements:
 1. *Improved fitness assignment scheme*, which takes each individual into account how many individuals it dominates and it is dominated by.
 2. A *nearest neighbor density technique* is incorporated which allows a more precise guidance of the search process.
 3. New *archive truncation method* (as opposed to clustering algorithm used in SPEA) guarantees the preservation of boundary solutions.

Fitness Assignment

- To avoid individuals dominated by the same archive members having identical fitness values, both dominating and dominated solutions are taken into account.
- In detail, each individual i in the archive P_t' and the population P_t is assigned a strength value $S(i)$, representing the number of solutions it dominates

$$S(i) = \left| \{j \mid j \in P_t + \overline{P_t'} \wedge i \succ j\} \right|$$

where $|\cdot|$ denotes the cardinality of a set, $+$ stands for multiset union and the symbol \succ corresponds to the Pareto dominance relation.

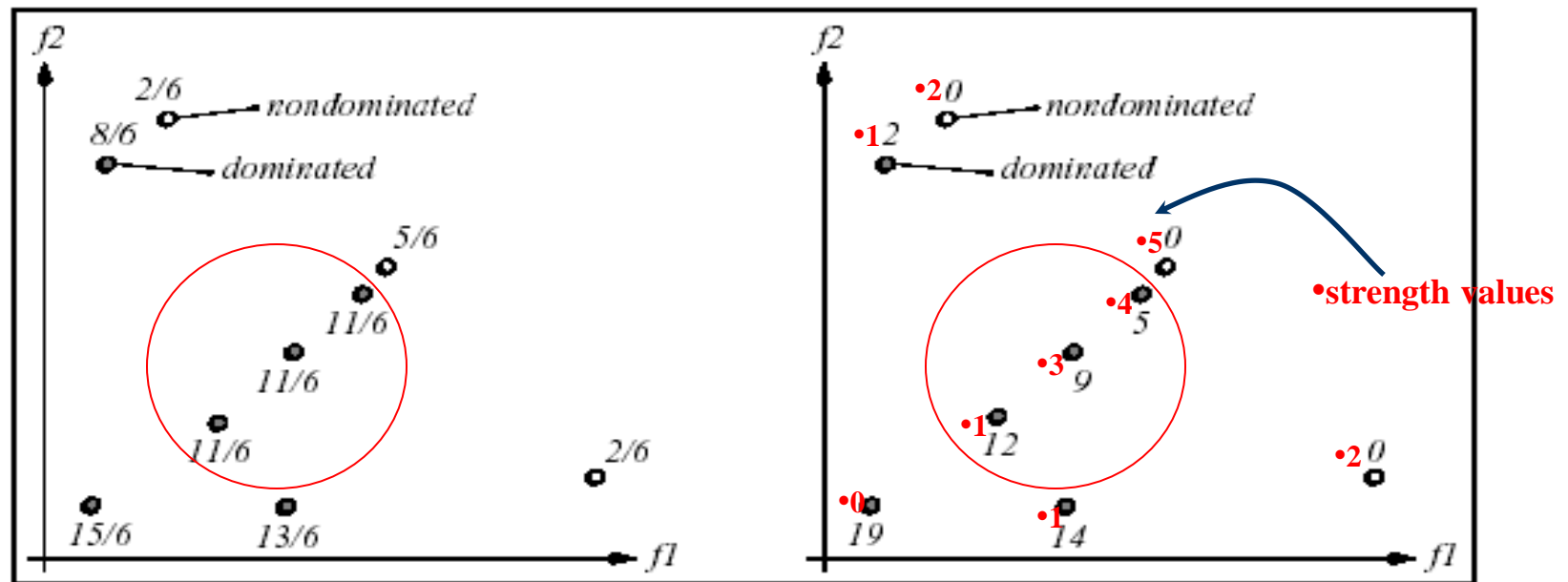
SPEA2 Design Procedure

- On the basis of the S values, the raw fitness $R(i)$ of an individual i is calculated as

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j \succ i} S(j)$$

- Fitness is determined by the strengths of its dominators in both archive and population, as opposed to SPEA where only archive members are considered in this context.
- Fitness is to be minimized here, i.e., $R(i) = 0$ corresponds to a non-dominated individual, while a high $R(i)$ value means that i is dominated by many individuals (which in turn dominate many individuals).
- SPEA2 is available at www.tik.ee.ethz.ch/pisa

SPEA vs. SPEA2 (maximization problem)



On the left, the fitness values for a given population according to the SPEA Scheme. On the right, the raw SPEA2 fitness values for the same population are depicted. No individual dominated by the same archive members will have the identical fitness values.

Nearest Neighbor Density Technique

- Additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique used is an adaptation of the k -th nearest neighbor method (Silverman 1986), where the density at any point is a (decreasing) function of the distance to the k -th nearest data point. Here, we simply take the inverse of the distance to the k -th nearest neighbor as the density estimate.
- For each individual i the distances (in objective space) to all individuals j in archive and population are calculated and stored in a list in increasing order, the k -th element gives the distance sought, denoted as σ_i^k and commonly $k = \sqrt{N + \bar{N}}$

The density $D(i)$ corresponding to i is defined by $D(i) = \frac{1}{\sigma_i^k + 2}$
 $0 < D(i) < 1$

Fitness of individual i is given by $F(i) = R(i) + D(i)$

Archive Update

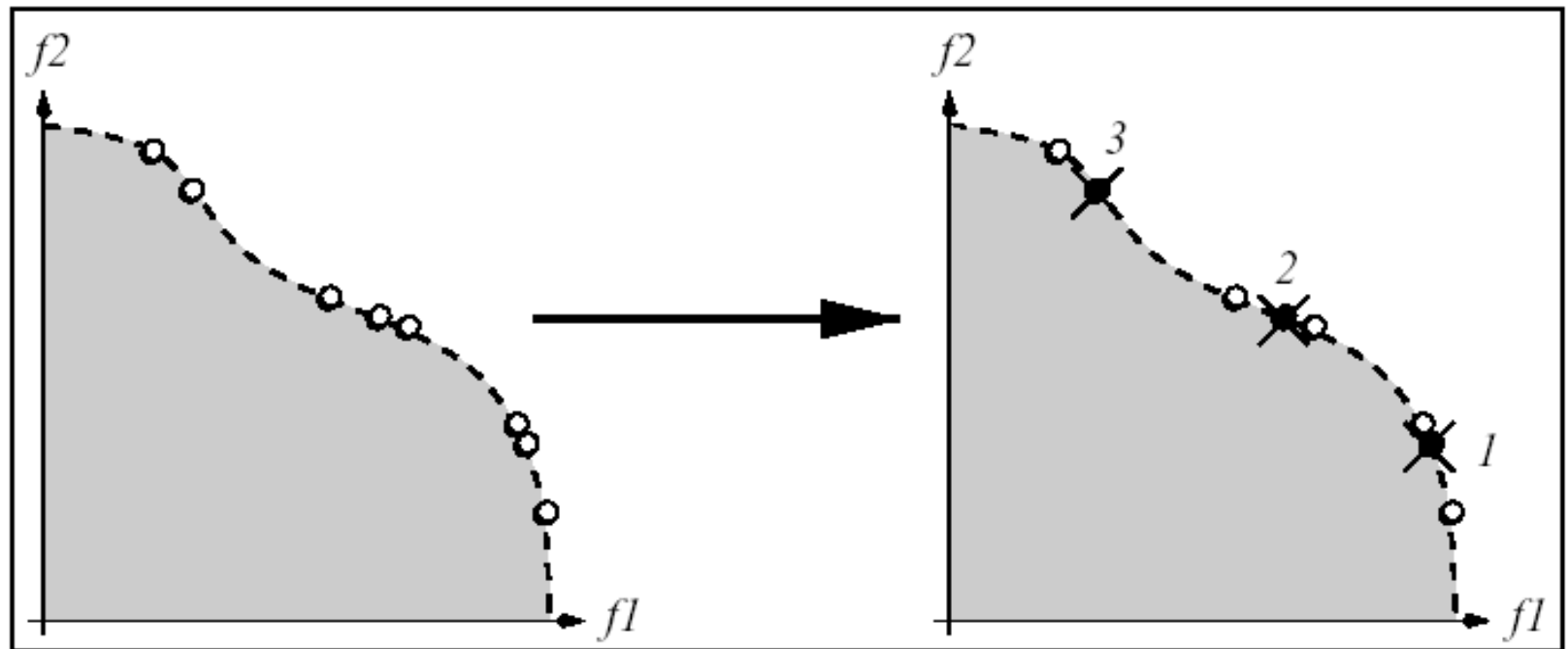
- The archive update operation in SPEA2 differs from the one in SPEA in two respects:
 - the number of individuals contained in the archive is constant over time
 - the truncation method prevents boundary solutions being removed.

Archive Truncation

- When the size of the current non-dominated set needs to be reduced, the archive truncation procedure is used, where at each iteration that individual i is chosen for removal for which for all $j \in \bar{P}_{t+1}$ with $i \leq_d j$

$$i \leq_d j \iff \begin{aligned} &\forall 0 < k < |\bar{P}_{t+1}| : \sigma_i^k = \sigma_j^k \vee \\ &\exists 0 < k < |\bar{P}_{t+1}| : \left[\left(\forall 0 < l < k : \sigma_i^l = \sigma_j^l \right) \wedge \sigma_i^k < \sigma_j^k \right] \end{aligned}$$

where σ_i^k denotes the distance of i to its k th nearest neighbor in $|\bar{P}_{t+1}|$, this means the individual which has the minimum distance to another individual is chosen at each stage; if there are several individuals with minimum distance the tie is broken by considering the second smallest distances and so forth.



On the left, a non-dominated set is shown. On the right, it is depicted which solutions are removed in which order by the truncate operator (assuming that $N' = 5$).

SPEA2 Pseudo Codes

Step 1: *Initialization*: Generate an initial population P_0 and create the empty archive (external set) $P_0' = \emptyset$; Set $t = 0$.

Step 2: *Fitness assignment*: Calculate fitness values of individuals in P_t and P_t'

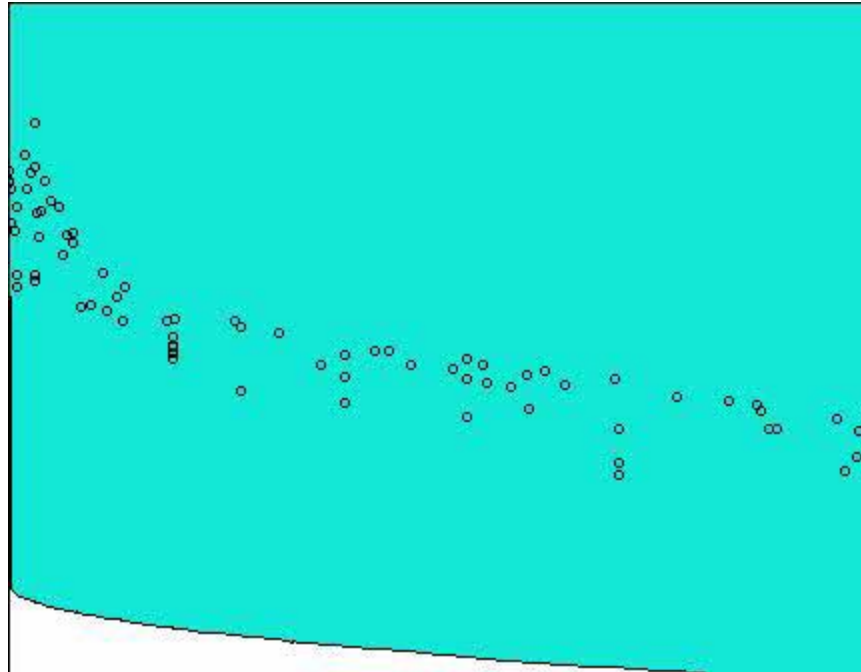
Step 3: *Environmental selection*: Copy all non-dominated individuals in P_t and P_t' to P_{t+1}' . If size of P_{t+1}' exceeds N' then reduce P_{t+1}' by means of the truncation operator, otherwise if size of P_{t+1}' is less than N' then fill P_{t+1}' with dominated individuals in P_t and P_t' .

Step 4: *Termination*: If $t < T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in P_{t+1}' . Stop.

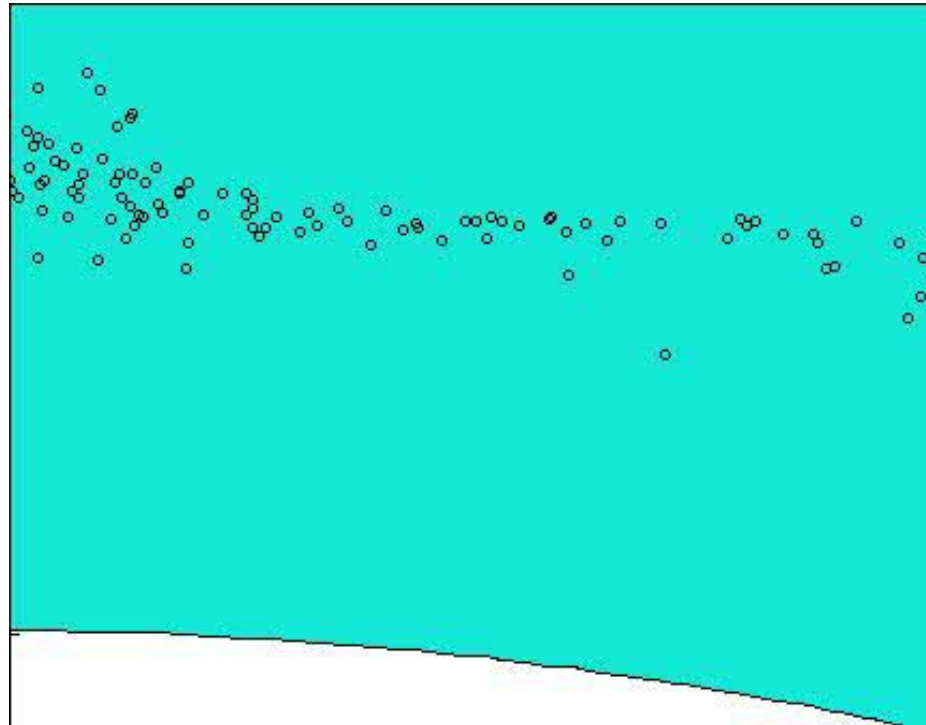
Step 5: *Mating selection*: Perform binary tournament selection with replacement on P_{t+1}' in order to fill the mating pool.

Step 6: *Variation*: Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t = t + 1$) and go to Step 2.

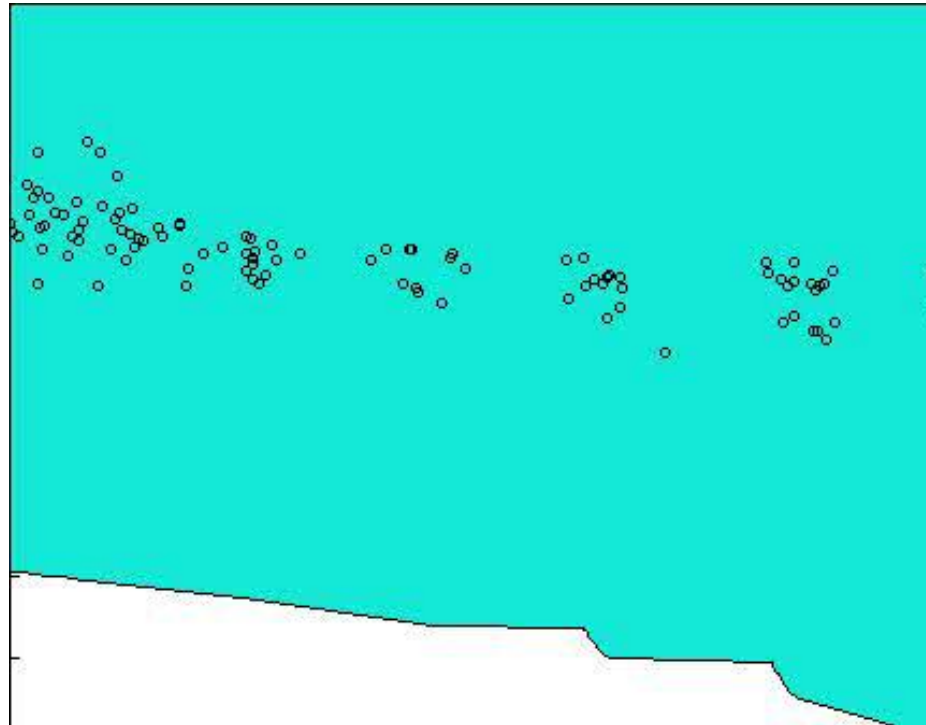
SPEA2 for ZDT1



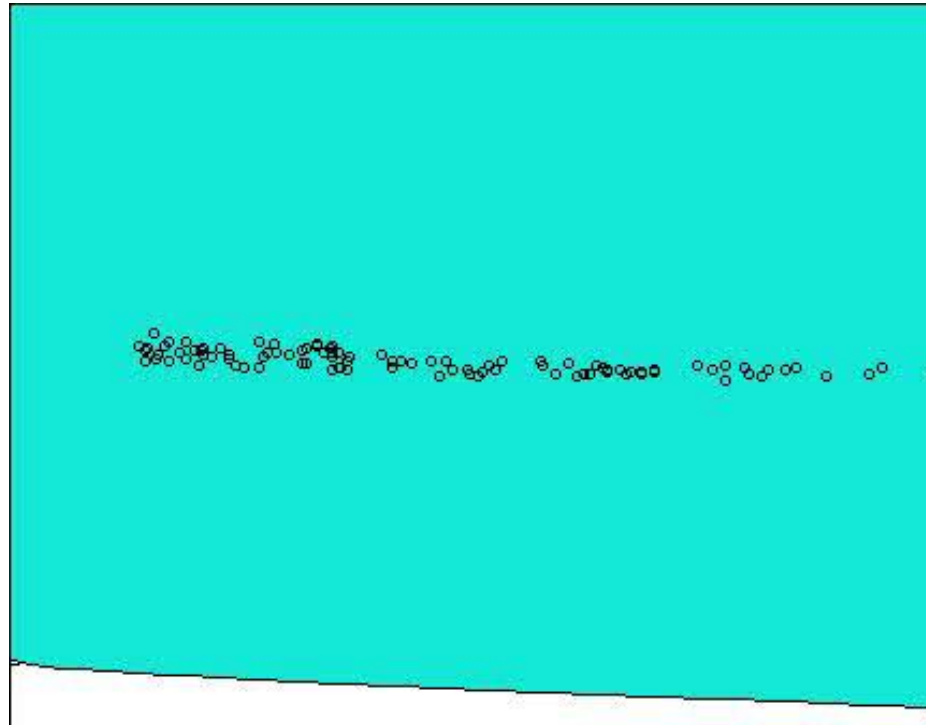
SPEA2 for ZDT2



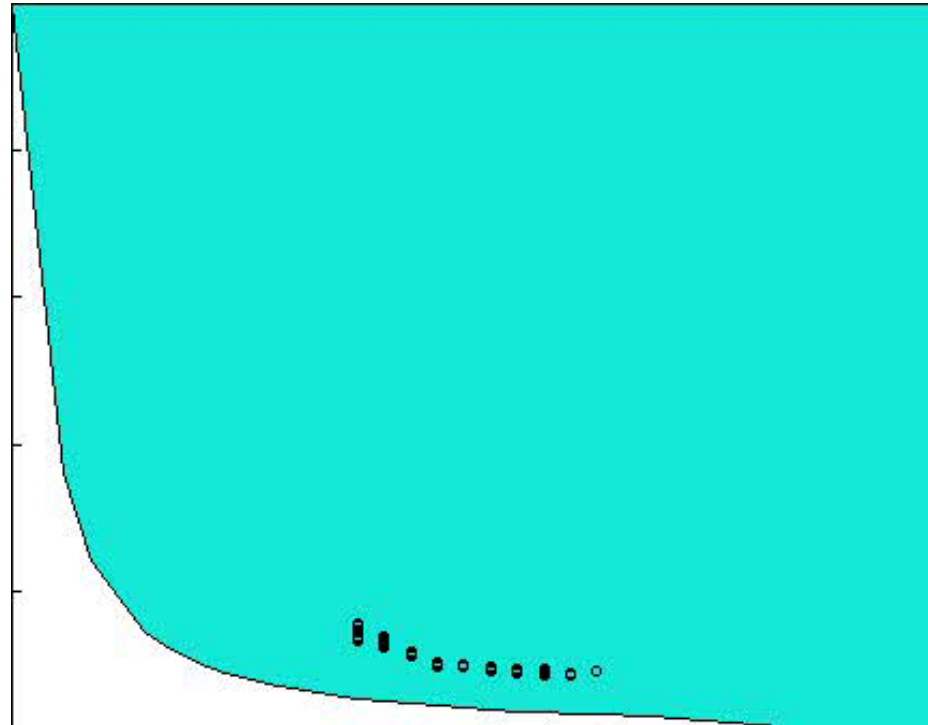
SPEA2 for ZDT3



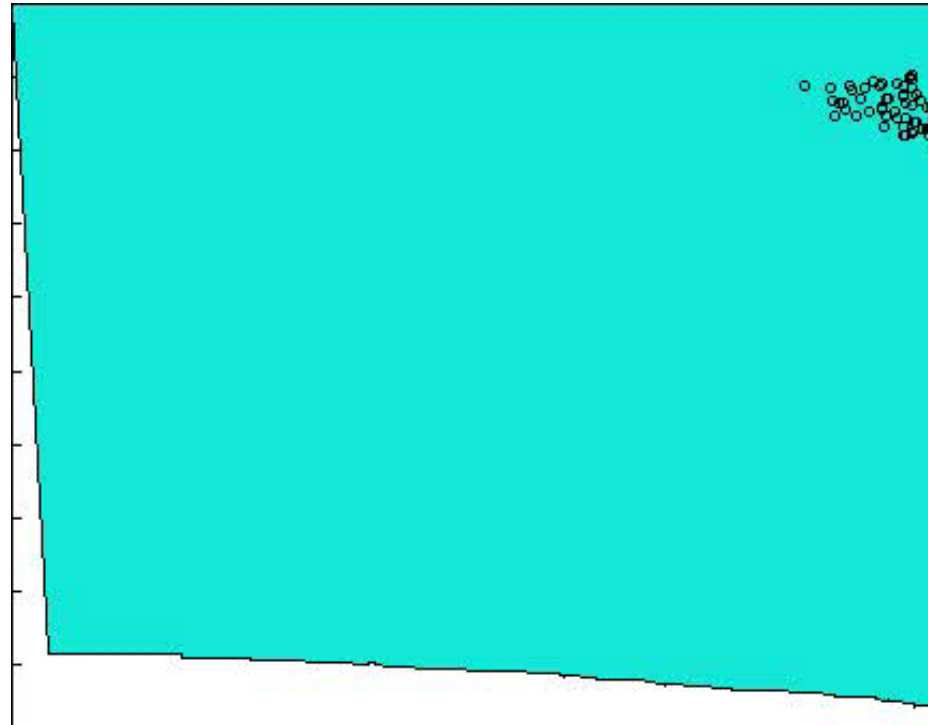
SPEA2 for ZDT4



SPEA2 for ZDT5

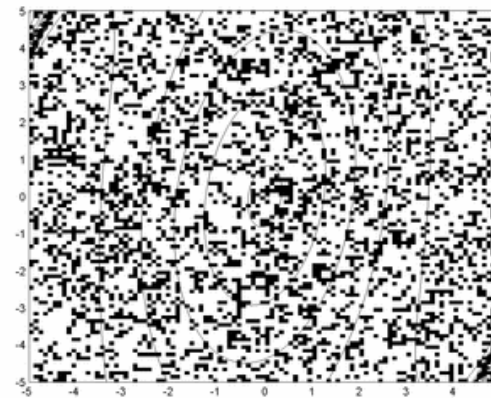
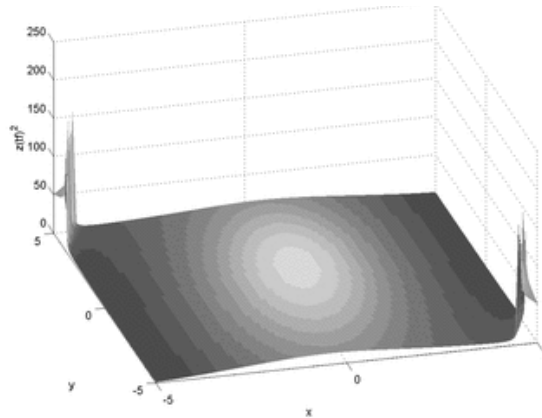


SPEA2 for ZDT6



DYNAMIC POPULATION STRATEGY IN RDGA

.....



Dynamic Population Strategy

- Goal: to exploit and avoid “genetic drift” phenomenon simultaneously
- If the predetermined population size is *too small*, there will not be enough schemas to be exploited, resulting into a non-uniformly distributed Pareto front. If the population size is *too large*, it may require unnecessarily large computational resources and result into an extremely long running time
- A fixed population size will have great difficulty in obtaining a Pareto front with a desired resolution because the size and shape of the true Pareto front is unknown *a priori* for most of the MOPs
- In most MOEAs, a “guessed” population size is used and a replacement method is usually designed to keep the population size
- A dynamic population size will be more reasonable for MOEA if the computational effort can be adaptively adjusted based on the complexity of the problem

Cost Benefit

- optimal population size depends on the complexity of the problem
- automatic stopping criteria
- time/computational complexity
- biological evidence

Cell-Based Rank & Density Calculation

- An adaptive grid density estimation approach is used. The length of adaptive grid cell of the i 'th dimension in objective space is computed as

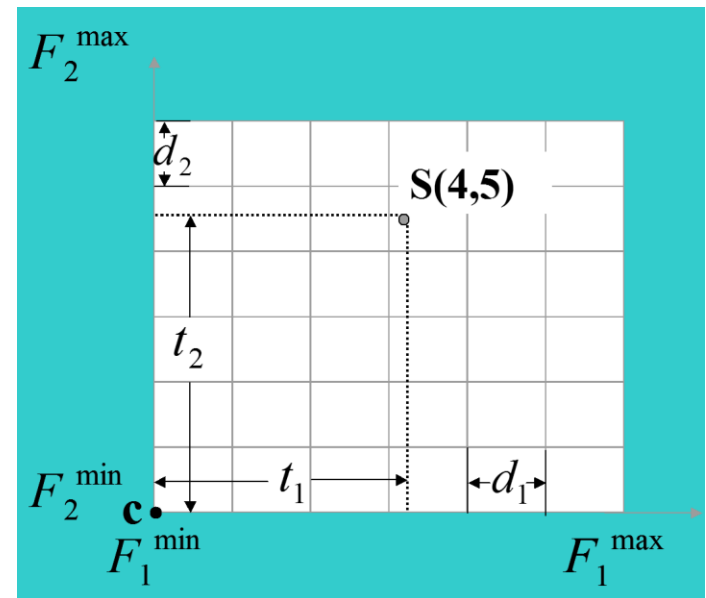
$$d_i = \frac{\max_{\mathbf{x} \in X} f_i(\mathbf{x}) - \min_{\mathbf{x} \in X} f_i(\mathbf{x})}{K_i}$$

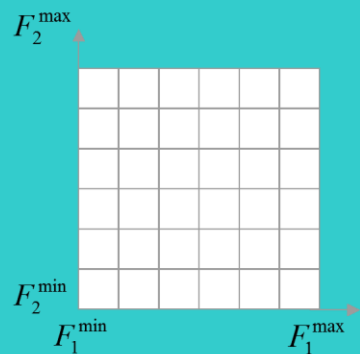
$$i = 1, \dots, n$$

$$t_i = s_i - F_i^{\min}$$

- Home address of individual S

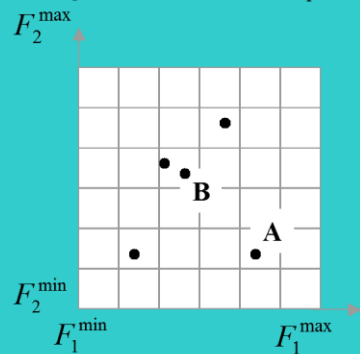
$$h_i = \text{mod}(t_i, d_i) + 1$$





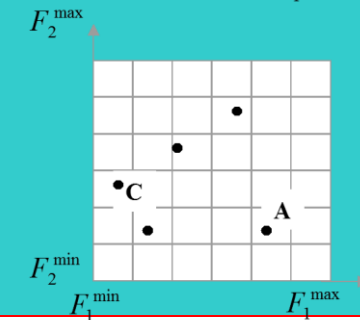
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0



1	2	4	5	6	6
1	2	4	4	6	6
1	2	2	4	5	5
1	2	2	2	3	3
1	1	2	2	2	3
1	1	1	1	1	1

0	0	0	0	0	0
0	0	0	1	0	0
0	0	2	0	0	0
0	0	0	0	0	0
0	1	0	0	1	0
0	0	0	0	0	0

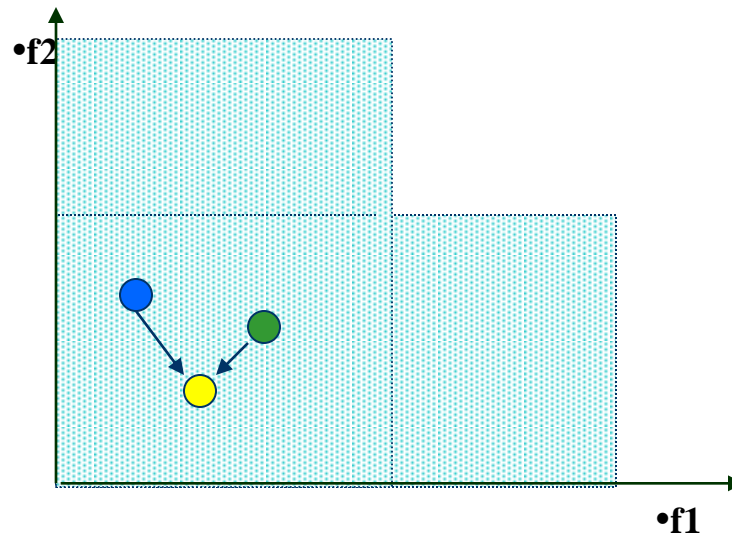


2	3	4	5	6	6
2	3	4	4	6	6
2	3	3	4	5	5
1	3	3	3	4	4
1	1	2	2	2	3
1	1	1	1	1	1

0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	0
1	0	0	0	0	0
0	1	0	0	1	0
0	0	0	0	0	0

Population Growing Strategy

- Goal: to focus on pure population increment to ensure that each individual survives enough generations so that it can contribute its valuable schemas.



Population Declining Strategy

- Goal: to prevent the population size from growing unboundedly for an individual c at generation n , we introduce:

- Health indicator

$$H(c, n) = \frac{1}{\text{rank}(c, n)}$$

- Global density indicator

$$DG(n) = \frac{D_{avg}}{ppv}$$

- Local density indicator

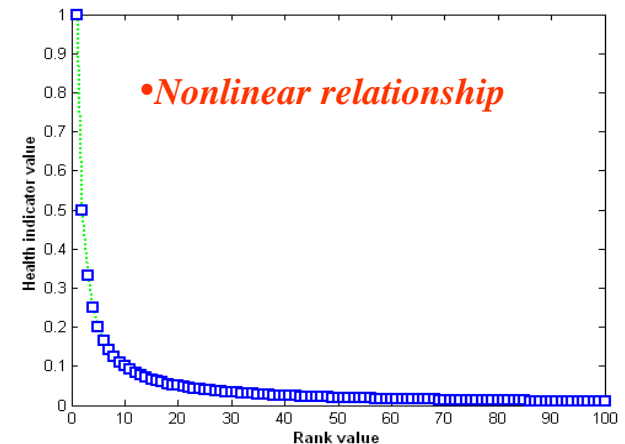
$$DL(c, n) = \begin{cases} \frac{\text{density}(c, n)}{ppv}, & \text{if } \text{density}(c, n) > ppv \\ 0, & \text{otherwise} \end{cases}$$

where, ppv denotes the desired number of individuals in each cell and D_{avg} is the average density value of the current population

- Age indicator

$$A(y, n) = \frac{\text{age}(y, n) - A_{th}}{n}, \text{ where } A_{th} \text{ denotes the}$$

predetermined threshold of the age



- Probabilities of removing an individual

- Likelihood of removing the **most unhealthy** individuals

$$l_1^i = (1 - H(c_i, n))^2 \times A(y_i, n)$$

$$H(c_i, n) = 1 / r_{\max}$$

- Likelihood of removing the **unhealthy** individuals in the **most crowded** cells

$$l_2^i = (1 - H(c_i, n))^2 \times DG(n) \times (DL(c_i, n) - 1) \times A(y_i, n)$$

- Likelihood of removing a **nondominated** individual from the **most crowded** cells *after the entire population converges*

$$l_3^i = (DL(c_i, n) - 1) \times DG(n) \times A(y_i, n)$$

- To determine whether an individual y_i will be eliminated, three random numbers between [0, 1] are generated to compare with the concerned likelihoods according to the situation of the given individual.

Observations

- Because the age indicator $A(y_i, n)$ influences all of three likelihoods, l_1 , l_2 and l_3 will be 0 if the age of the concerned individual is less than the age threshold A_{th} . This implies that if an individual is not old enough, it will not be eliminated from the population no matter how high its rank and density values are.
- At each generation, DMOEA will remove those *most unhealthy* individuals according to likelihood l_1 , based on their rank values and ages. Assume the age indicator of an individual y is $A(y, n)=1$, the relationship between its rank value and l_1 value is an exponentially growing function. Without considering the effects of other indicators, when an *unhealthy* individual in the set has a very high rank value, it will have a very high likelihood (l_1) to be eliminated, since it is too far away from the current Pareto front. Moreover, as r_{max} drops and gets closer to 1, l_1 will decrease very fast, and the concerned individual will not be removed easily because it is very likely to be evolved into an elitist in the future. Therefore, this “shell removing” strategy will keep eliminating the individuals located on the outside layer with an adaptive probability until the entire population converges into a non-dominated set.

- Because all the individuals in the same cell share the fixed computation resource (or “living resource”), the individuals located in a crowded cell have to compete much harder for the limited resource than those located in a sparse cell. Therefore, another elimination scheme based on a *crowdedness* indicator is designed in DMOEA in order to remove some *unhealthy* individuals that stay in the most crowded areas. From I_2 likelihood Equation, at each generation, if an individual belongs to the set Y_{dr} , it will have the likelihood of I_2 to be eliminated based on its *age*, *health*, and local rank value and density condition. From this scheme, the population tends to be distributed homogeneously by eliminating the redundant individuals.
- After every individual has converged into a Pareto point, another elimination scheme is implemented based on I_3 values. Therefore, the resulting trade-off hyperareas $A_{to}(n)$ are counted, and the final population is truncated to ensure that each cell contains *ppv* number of individuals; thus the optimal population size can be calculated by

$$dps(n) = ppv \times A_{to}(n)$$

Objective Space Compression Strategy

Goal— to ensure the precise of resulting Pareto front. Three criteria need to be met:

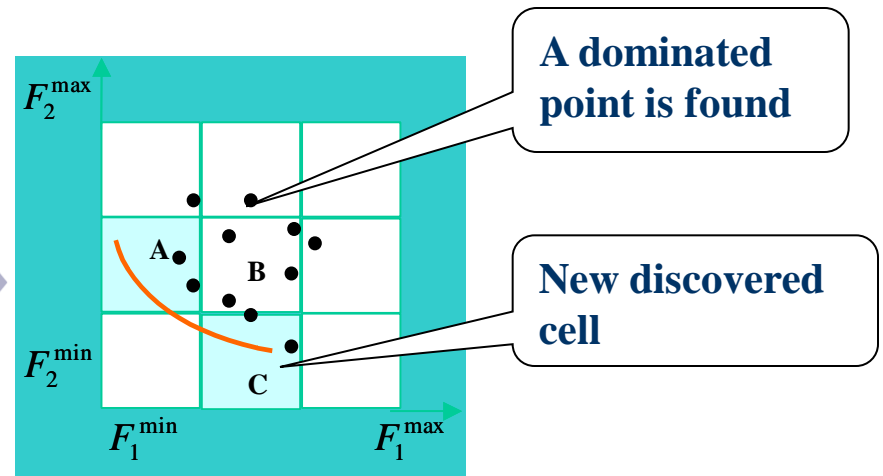
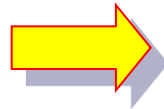
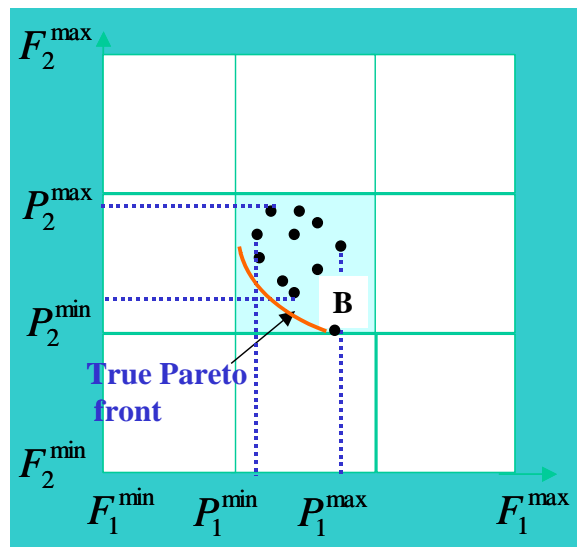
- maximum cell rank value of all the individuals is 1-- all discovered cells are non-dominated;
- $(F_i^{\max} - P_i^{\max}) > 0.1(F_i^{\max} - F_i^{\min})$ or $(P_i^{\min} - F_i^{\min}) > 0.1(F_i^{\max} - F_i^{\min})$ -- there is still some room for compression;
- minimum age value of all the individuals is greater than the predefined age threshold A_{th} -- no new generated individuals since most of the resulting points are Pareto optimal;

Therefore, new boundaries of the objective space will be

$$F_i^{\max} = (P_i^{\max} + F_i^{\max}) / 2$$

$$F_i^{\min} = (P_i^{\min} + F_i^{\min}) / 2$$

A Graphical Illustration



Stopping Criteria

- The rank values of all cells are 1s
- The objective space cannot be compressed anymore
- Each resulting non-dominated cell contains ppv individuals
- Afterwards, DMOEA is refined by using the traditional Pareto ranking scheme instead of cell-based ranking to ensure all the resulting individuals are Pareto points

F-4: High Dimensional Decision Space

- Deb (2001)

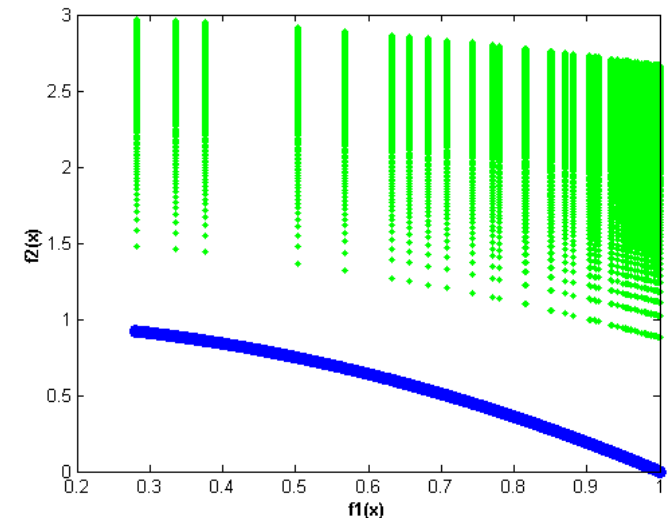
Minimize:

$$f_1(x) = 1 - e^{-4x_1} \sin^6(6\pi x_1)$$

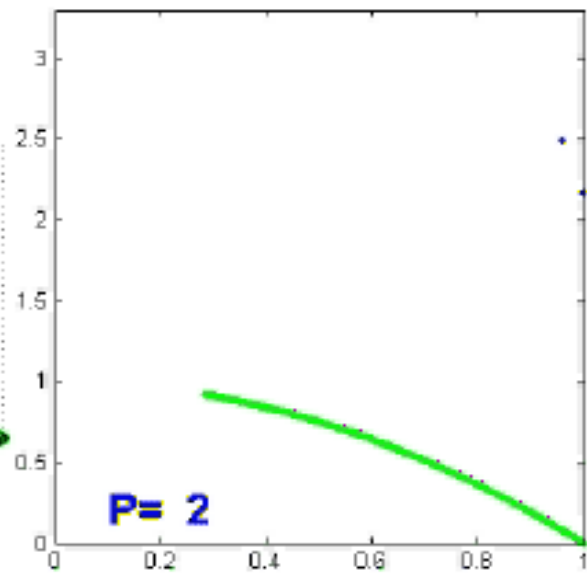
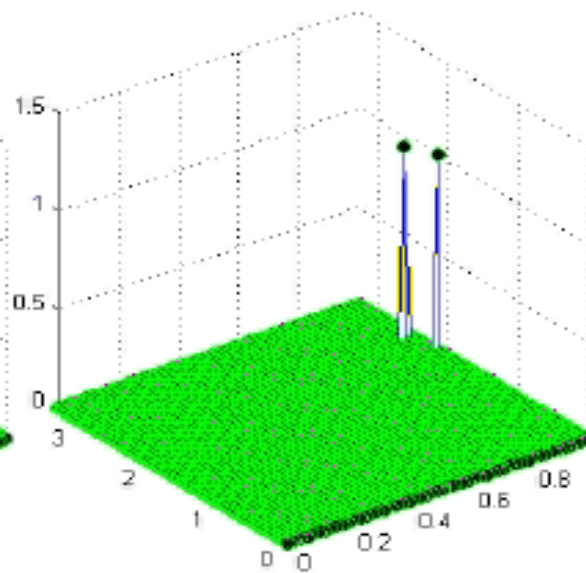
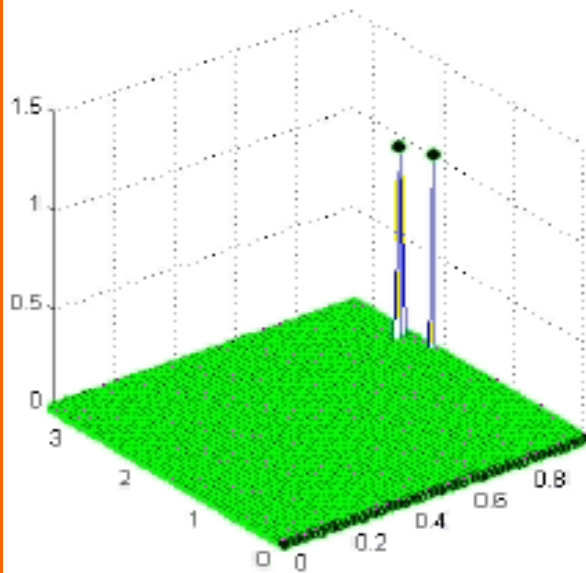
$$f_2(x) = g \left(1 - \left(\frac{f_1(x)}{g} \right)^2 \right)$$

Subject to

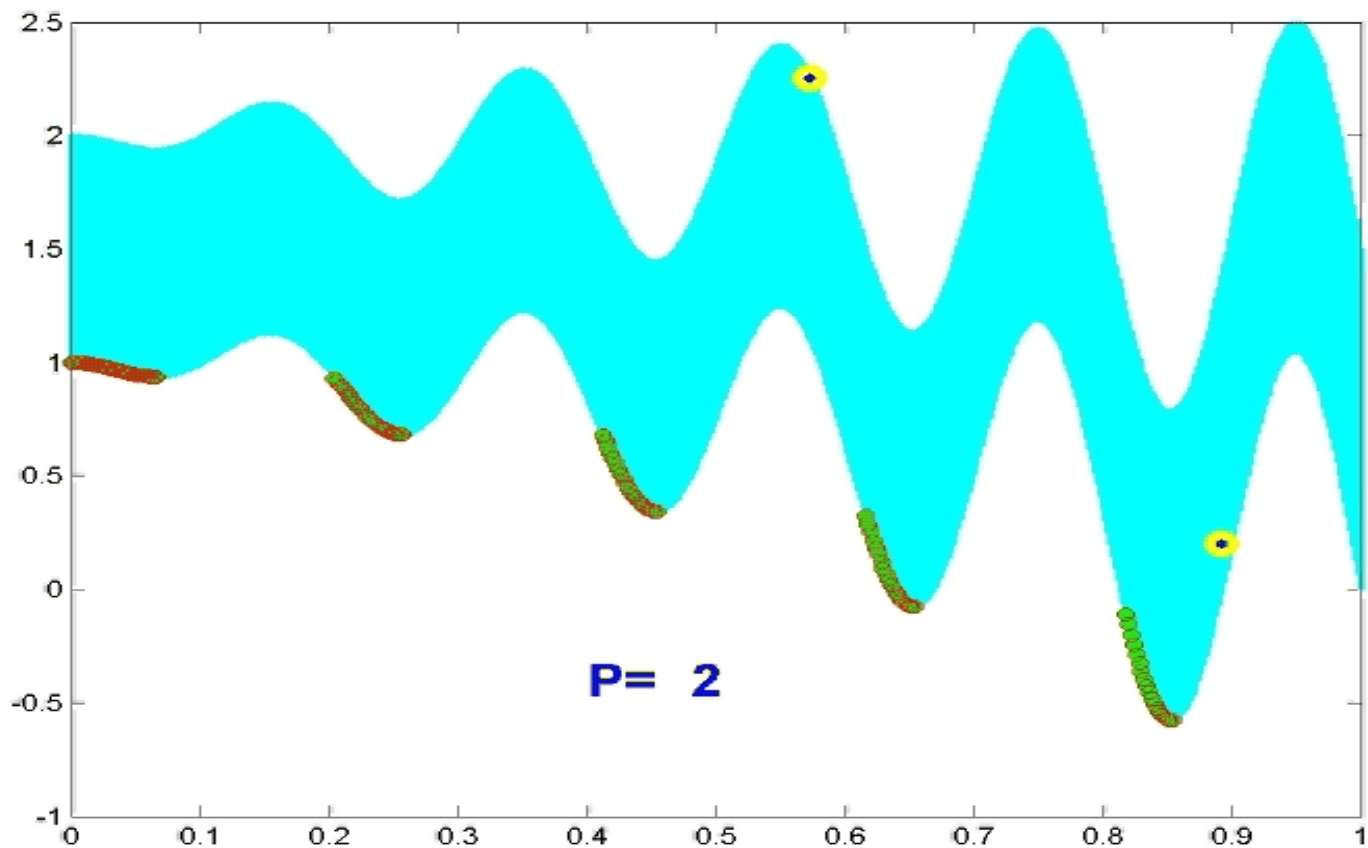
$$g(x) = 1 + 4 \left(\sum_{i=2}^5 x_i / 4 \right)^{0.25}, 0 \leq x_i \leq 1, i = 1, \dots, 5$$



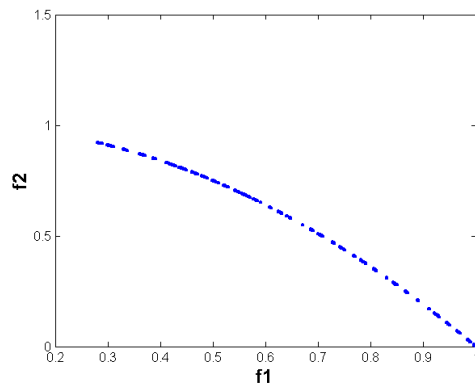
Demonstration: DMOEA for *F-4*



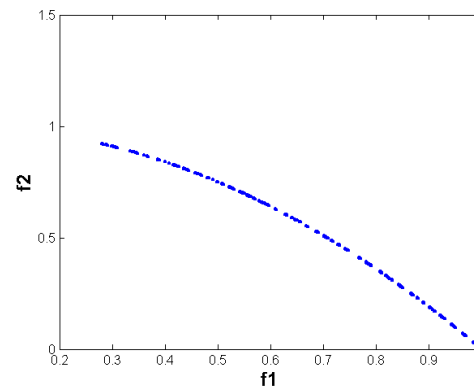
Demonstration: DMOEA



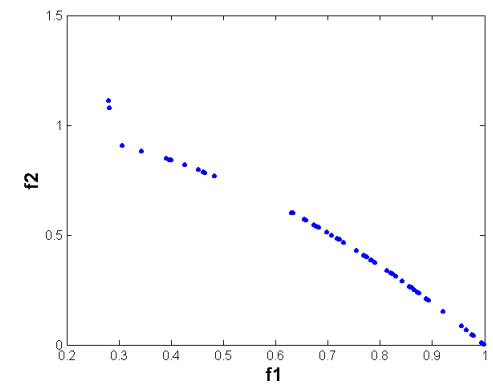
F-4: Resulted Pareto Fronts



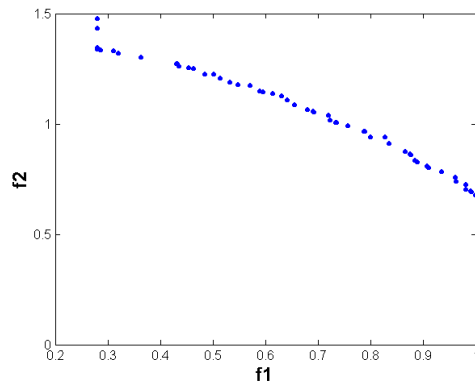
IMOEa



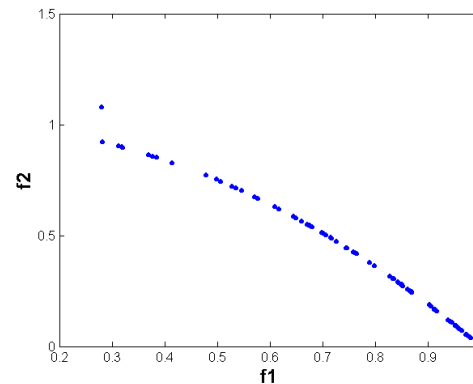
DMOEA



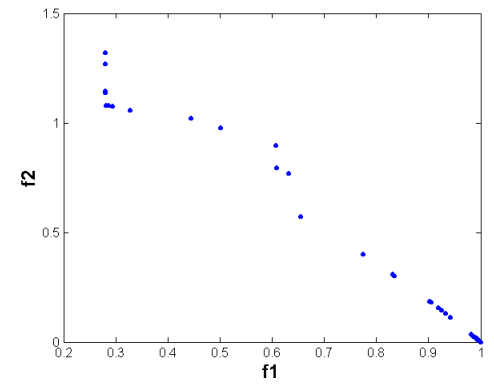
NSGA-II



PAES

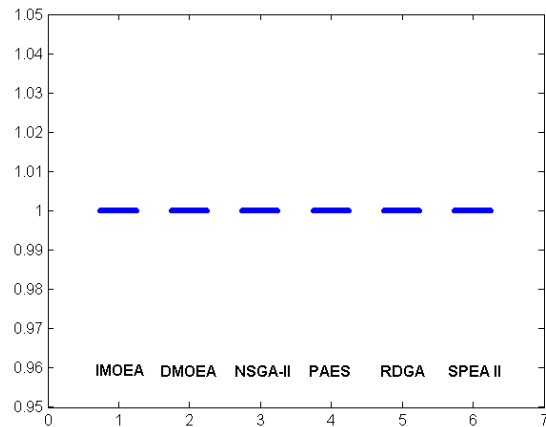


RDGA



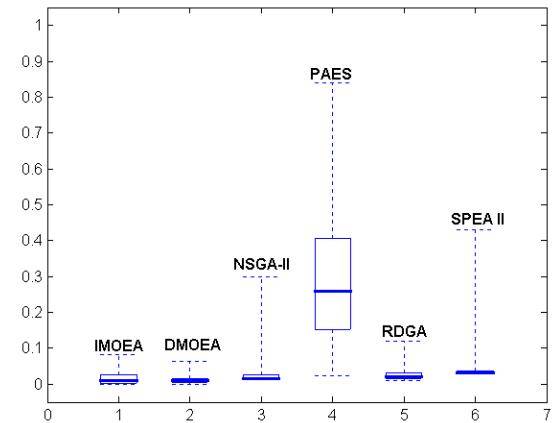
SPEA II

F-4: Rank, Density & Distance Measures

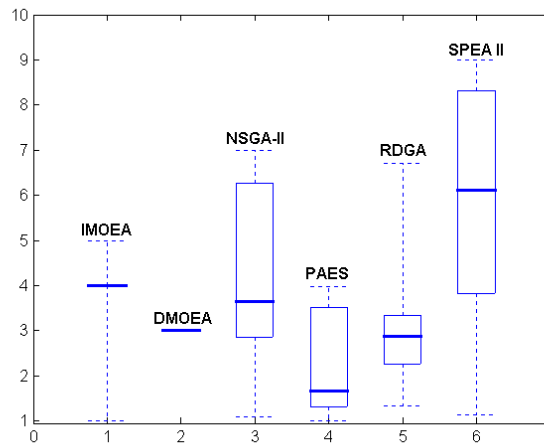


Rank

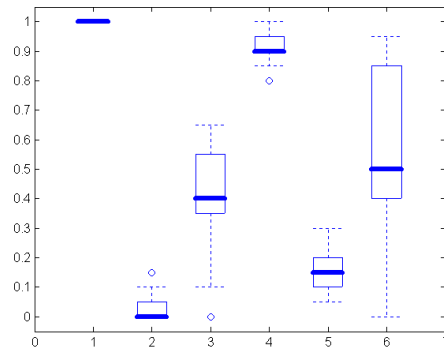
Density



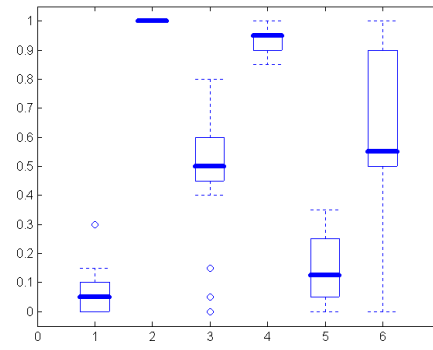
Distance



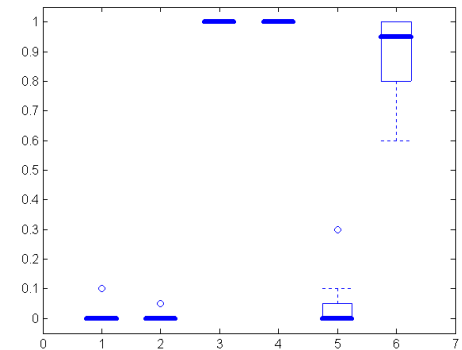
F-4: C Measure



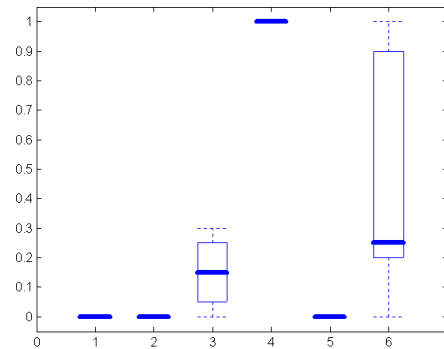
$C(X_1, X_{1-6})$



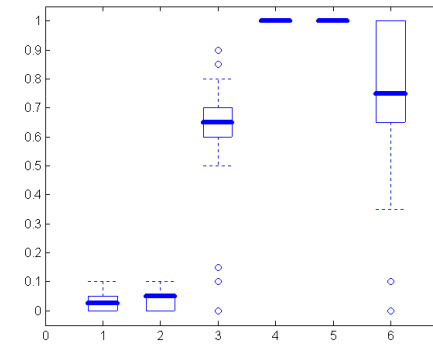
$C(X_2, X_{1-6})$



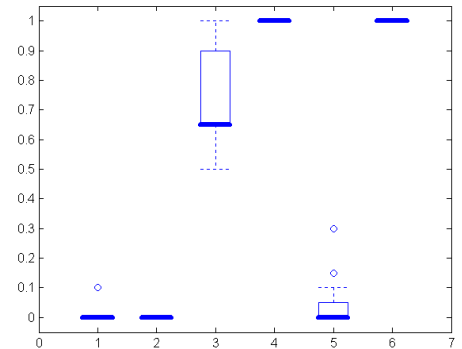
$C(X_3, X_{1-6})$



$C(X_4, X_{1-6})$



$C(X_5, X_{1-6})$



$C(X_6, X_{1-6})$

1.IMOEA; 2.DMOEA; 3.NSGA-II; 4.PAES; 5.RDGA; 6.SPEA II

F-6: High Dim Objective/Decision Spaces

- Deb (2002), one global and many local Pareto fronts.
- Minimize

$$f_1(x) = (1 + g(\mathbf{x})) \cos\left(\frac{\pi x_1}{2}\right) \cos\left(\frac{\pi x_2}{2}\right)$$

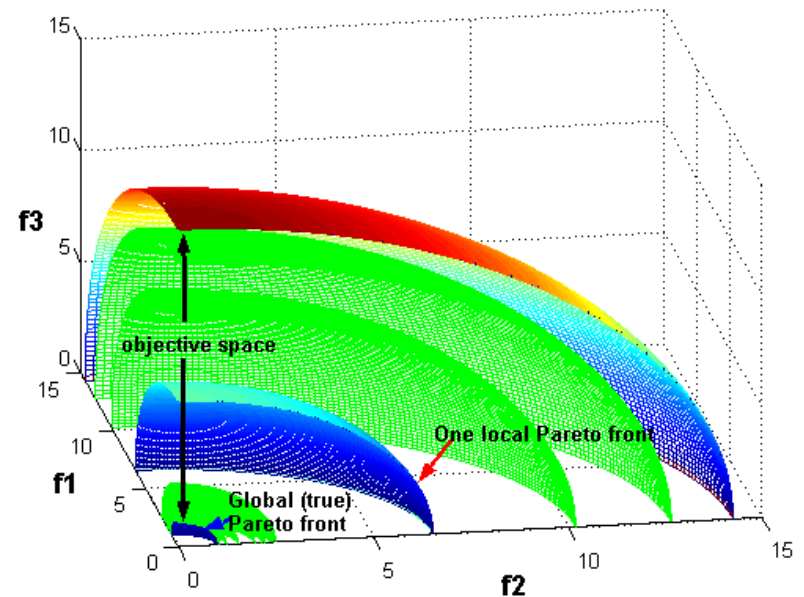
$$f_2(x) = (1 + g(\mathbf{x})) \cos\left(\frac{\pi x_1}{2}\right) \sin\left(\frac{\pi x_2}{2}\right)$$

$$f_3(x) = (1 + g(\mathbf{x})) \sin\left(\frac{\pi x_1}{2}\right)$$

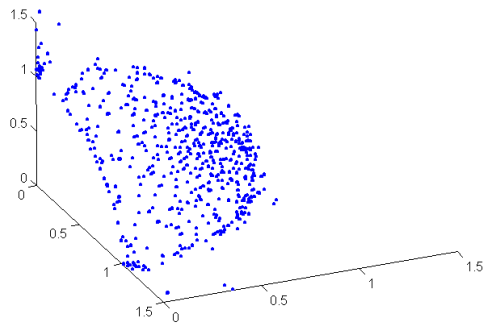
subject to

$$g(\mathbf{x}) = 12 + \sum_{i=1}^{12} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)),$$

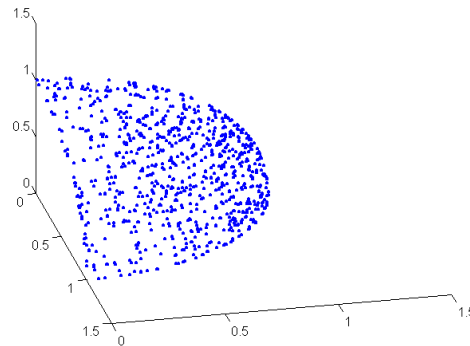
$$0 \leq x_i \leq 1, i = 1, \dots, 12$$



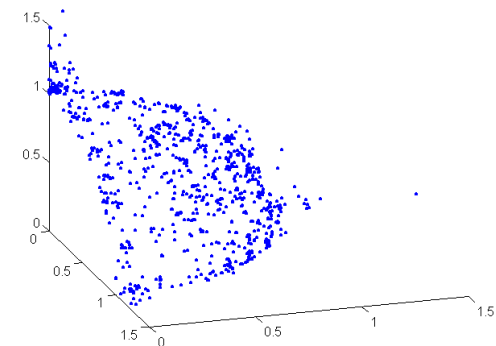
F-6: Resulted Pareto Fronts



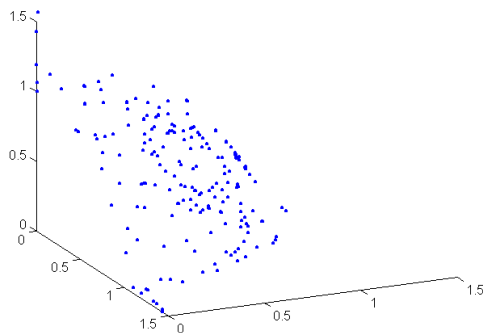
IMOEA



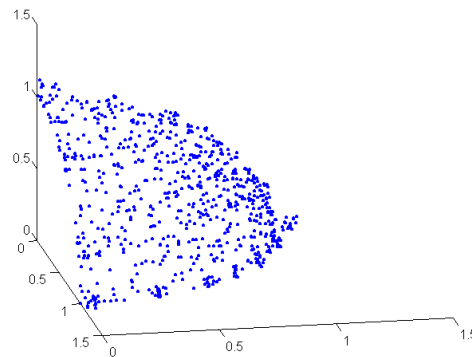
DMOEA



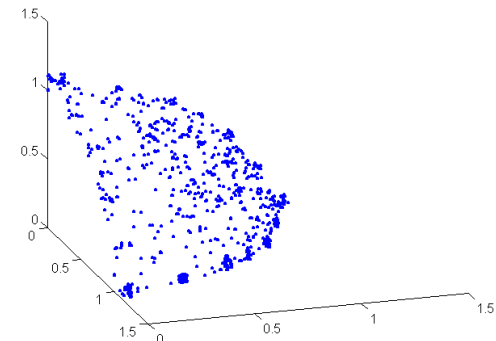
NSGA-II



PAES

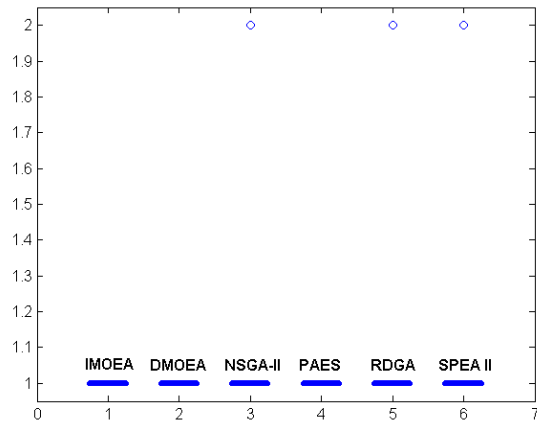


RDGA

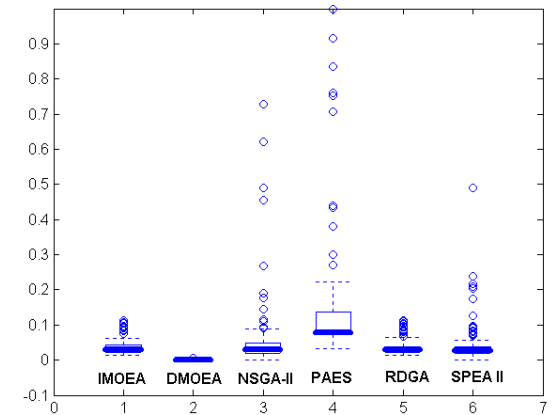


SPEA II

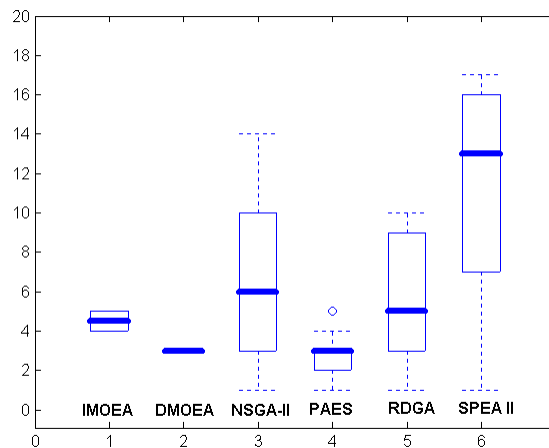
F-6: Rank, Density & Distance Measures



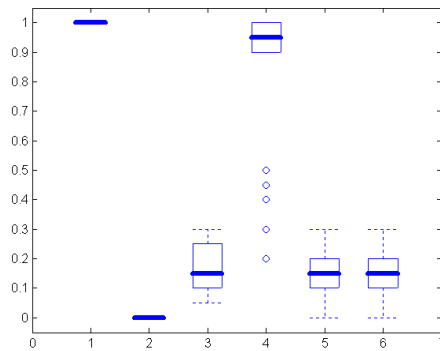
Rank



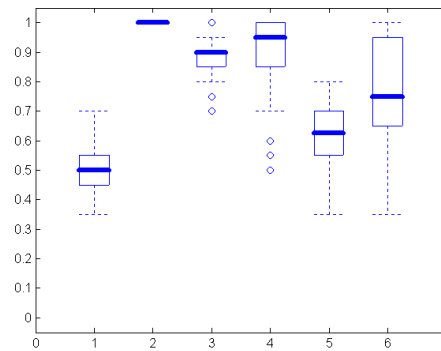
Distance



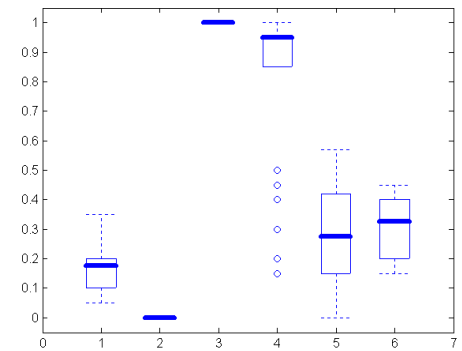
F-6: C Measure



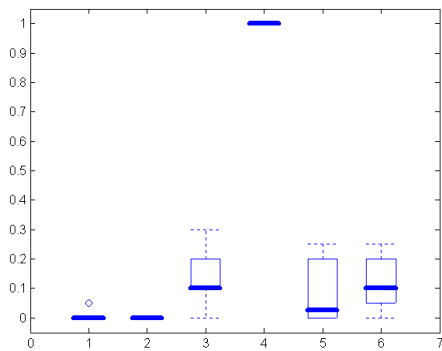
$C(X_1, X_{1-6})$



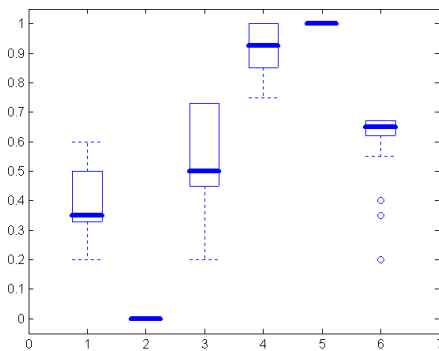
$C(X_2, X_{1-6})$



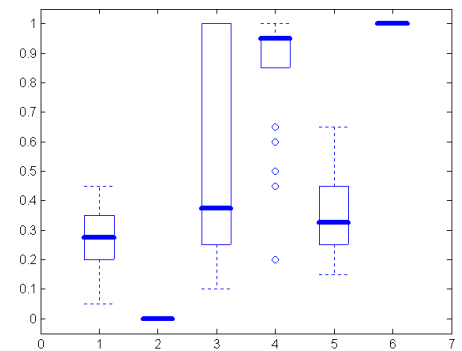
$C(X_3, X_{1-6})$



$C(X_4, X_{1-6})$



$C(X_5, X_{1-6})$



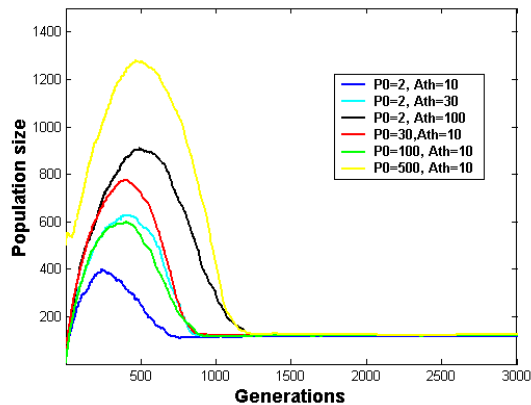
$C(X_6, X_{1-6})$

1.IMOEA; 2.DMOEA; 3.NSGA-II; 4.PAES; 5.RDGA; 6.SPEA II

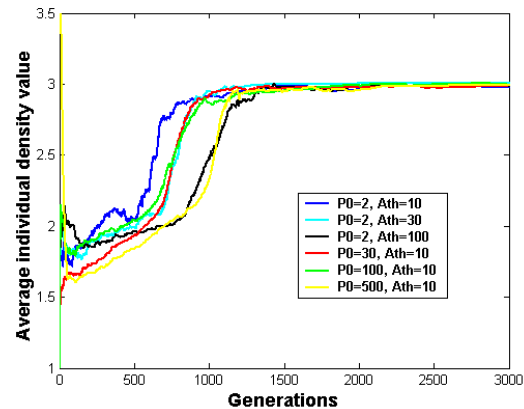
Robustness Study

Population size

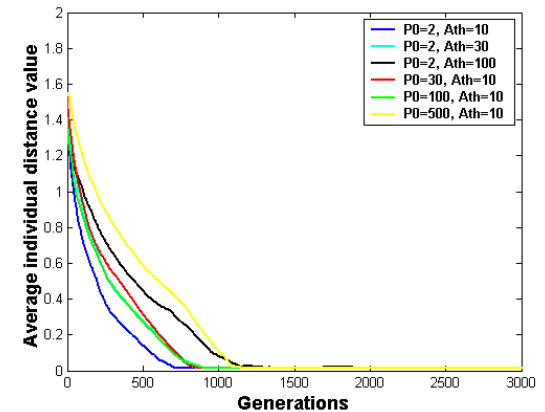
Function $F-4$:



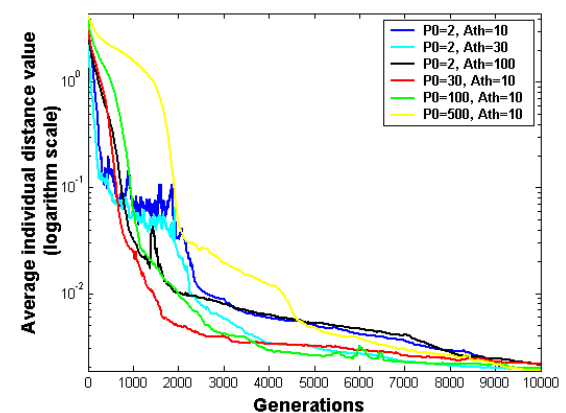
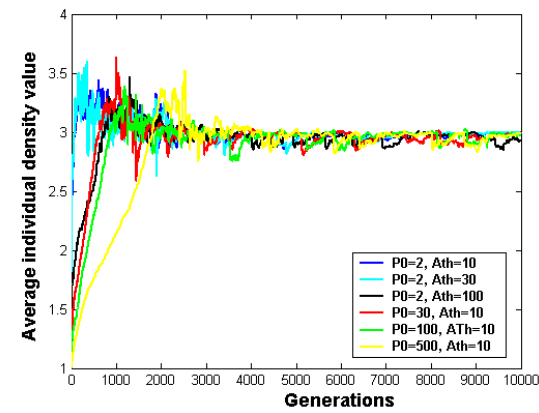
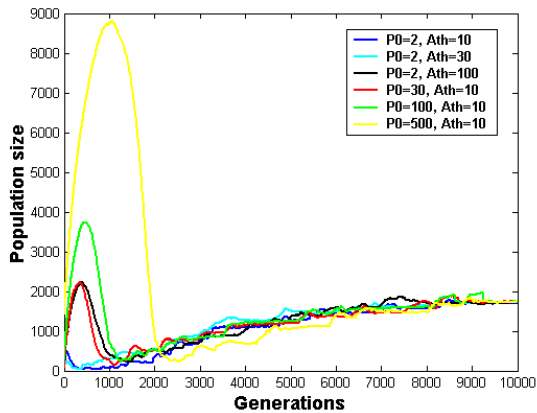
Density Value



Distance Value



Function $F-6$



Computational Complexity

- Efficiency of DMOEA— less time consuming comparing to the other advanced MOEAs.

	IMOEa	PAES	NSGA-II	RDGA	SPEA II	DMOEa (2,10)	DMOEa (2,30)	DMOEa (2,100)	DMOEa (30,10)	DMOEa (100,10)	DMOEa (500,10)
Time (min)	106	133	251	684	407	25	25	25	26	26	27

- DMOEa—potential in solving real-time complicated MOPs, which is still needed to be validated in future work