“It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is most adaptable to change. ”
Charles Darwin

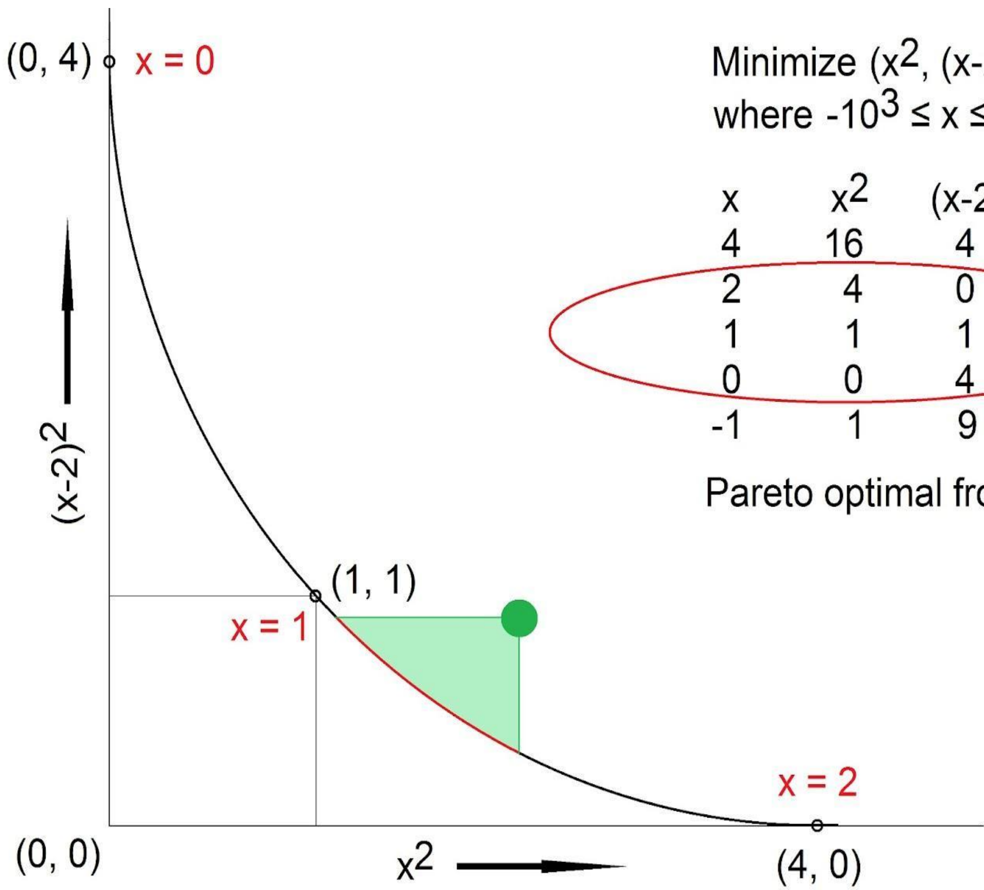
Multi-Objective Evolutionary Algorithms

Dr Sujit Das

Multi-Objective Optimization!...

- Have more than one objectives.
- Objectives are conflicting in nature.
- Most of the real life problems are multi-objective in nature.
    - Bye a car
        - Mercedes Benz or Tata Nano!!!...
            - Comfort or cost...

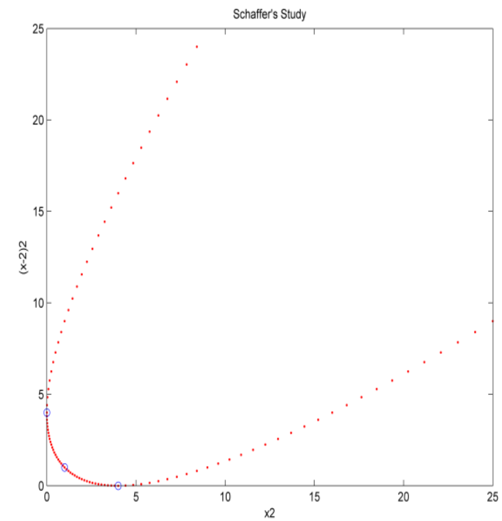
How does it look mathematically?...
- Schaffer's study



Minimize (x^2, $(x-2)^2$)
where $-10^3 \leq x \leq 10^3$

x	x^2	$(x-2)^2$
4	16	4
2	4	0
1	1	1
0	0	4
-1	1	9

Pareto optimal front: $0 \leq x \leq 2$



Spaces

1. Genotypic Space/ Variable Space
2. Phenotypic Space/ Objective Space

In Schaffer's Study:

x is the variable

x^2 and $(x-2)^2$ are the objectives

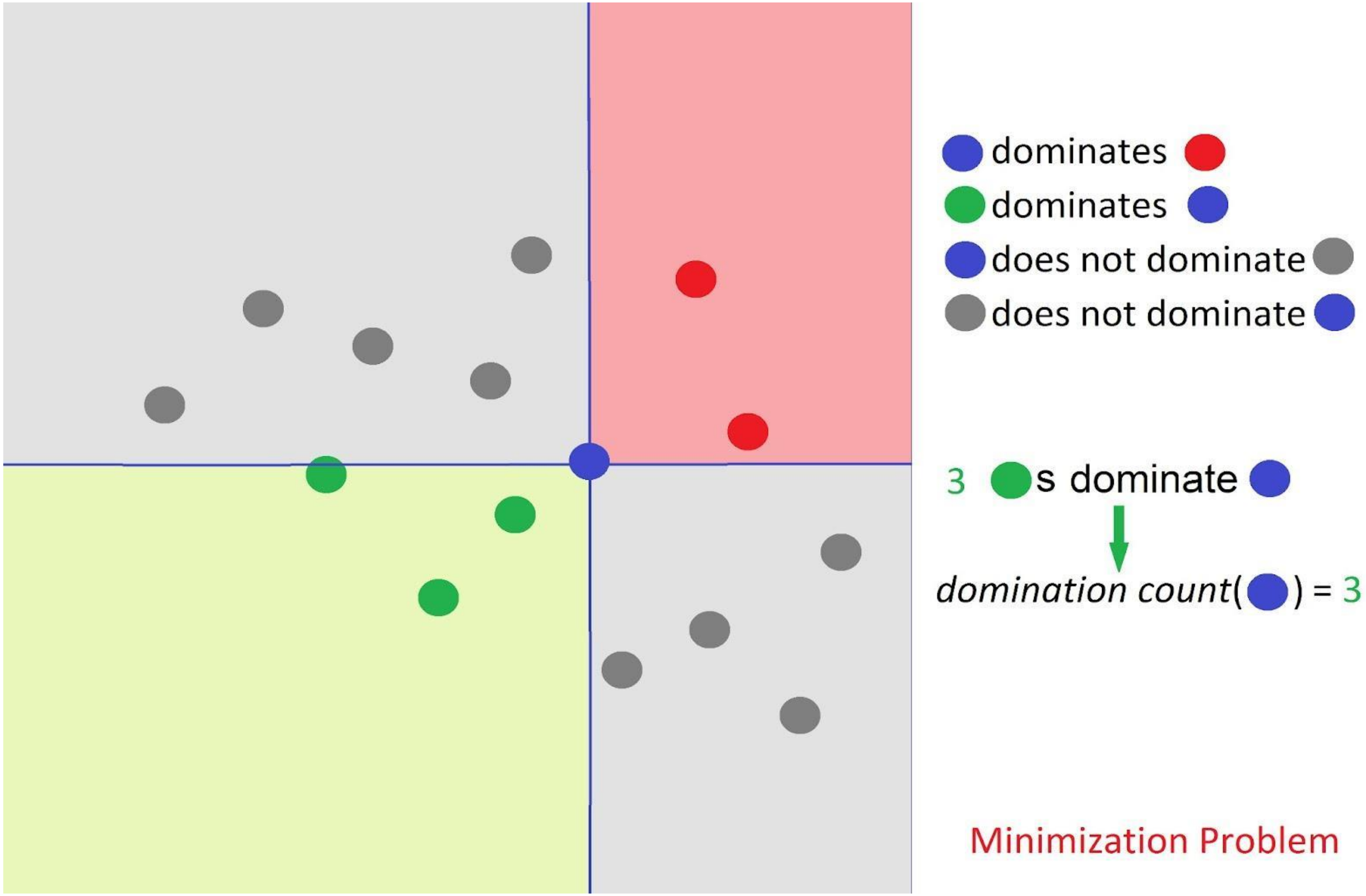
Remembering Schaffer's study!...

minimize $(x^2, (x-2)^2)$, where $-1000 <= x <= 1000$

"*Dominates*": what is that!?!..

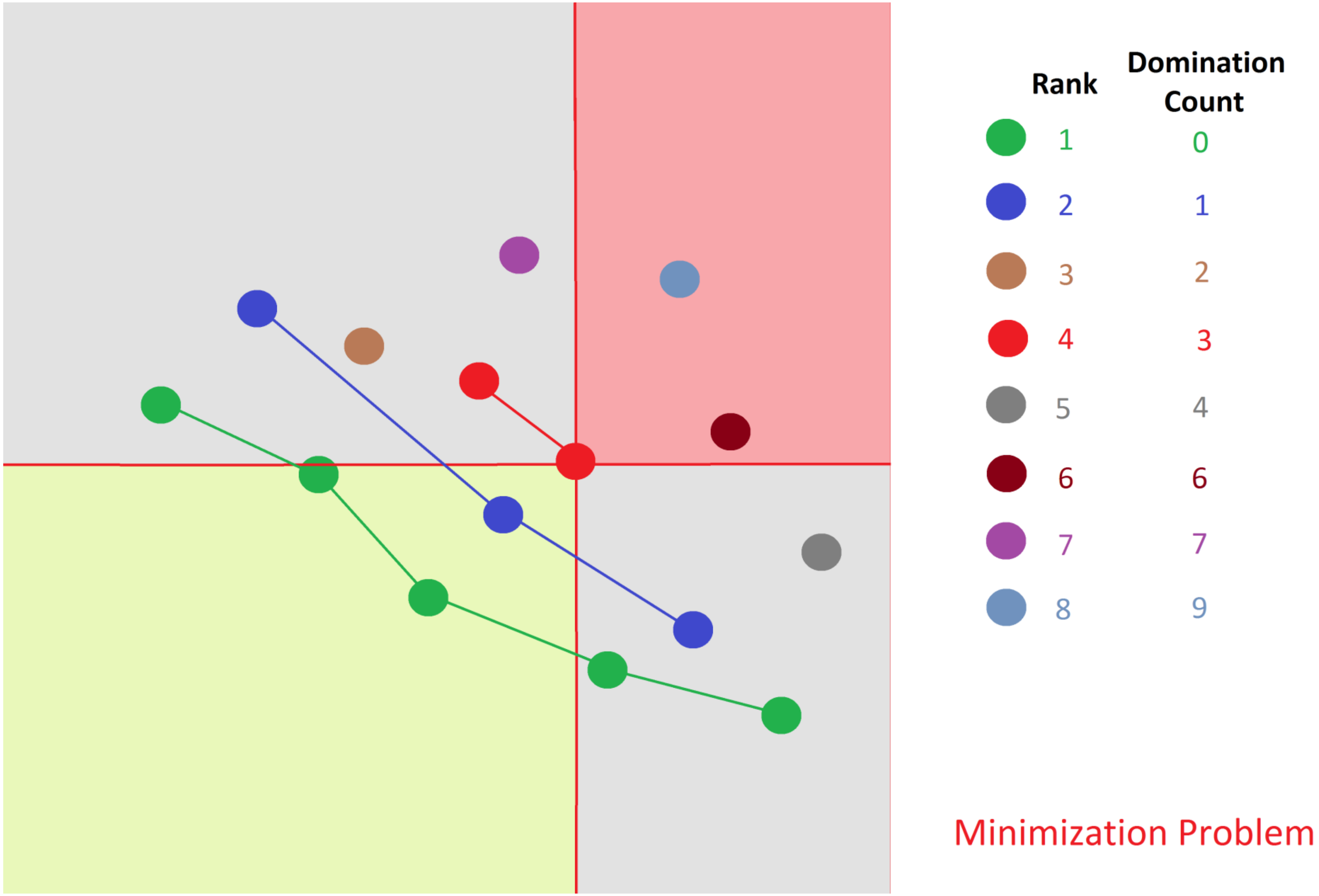
- solution p dominates solution q if all objectives of p is not poor than that of q and for at least one objective p is better than q.
- All optimization problems can be restated as minimization problems.

Domination



dominates
dominates
does not dominate
does not dominate
3 s dominate
domination count( ) = 3
Minimization Problem

Domination count and Rank



Pareto Optimality

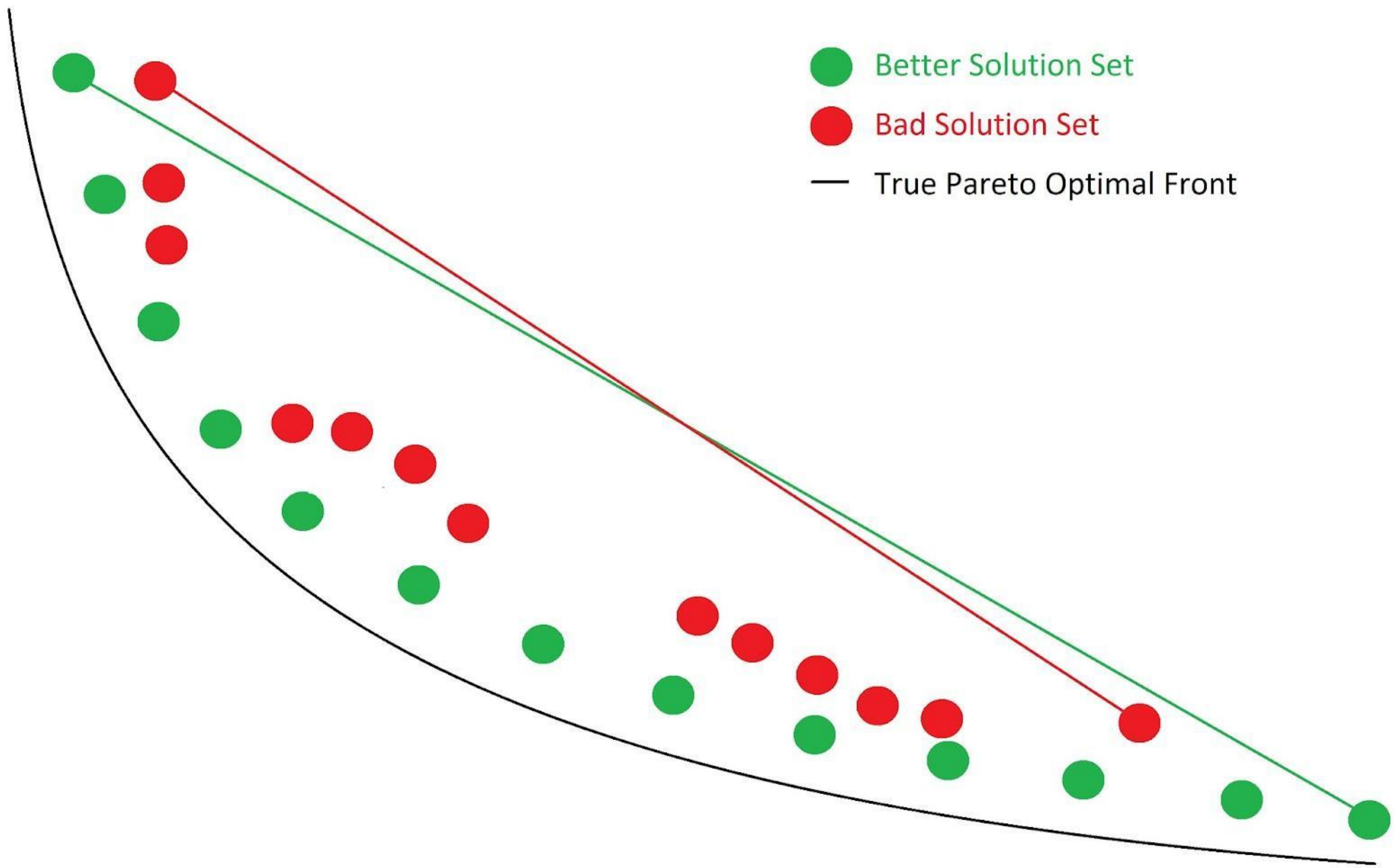
- A solution is Pareto optimal if no solution dominates it.
- Set of all Pareto optimal solutions (points in variable space) is called Pareto Set.
- Set of all Pareto objective vectors is called Pareto Front.

What do we try to obtain?

A set of solutions:

1. Well diverse.
2. Close to Pareto front.
3. Covers the whole spectrum of Pareto front.

Properties of Good Solution Set

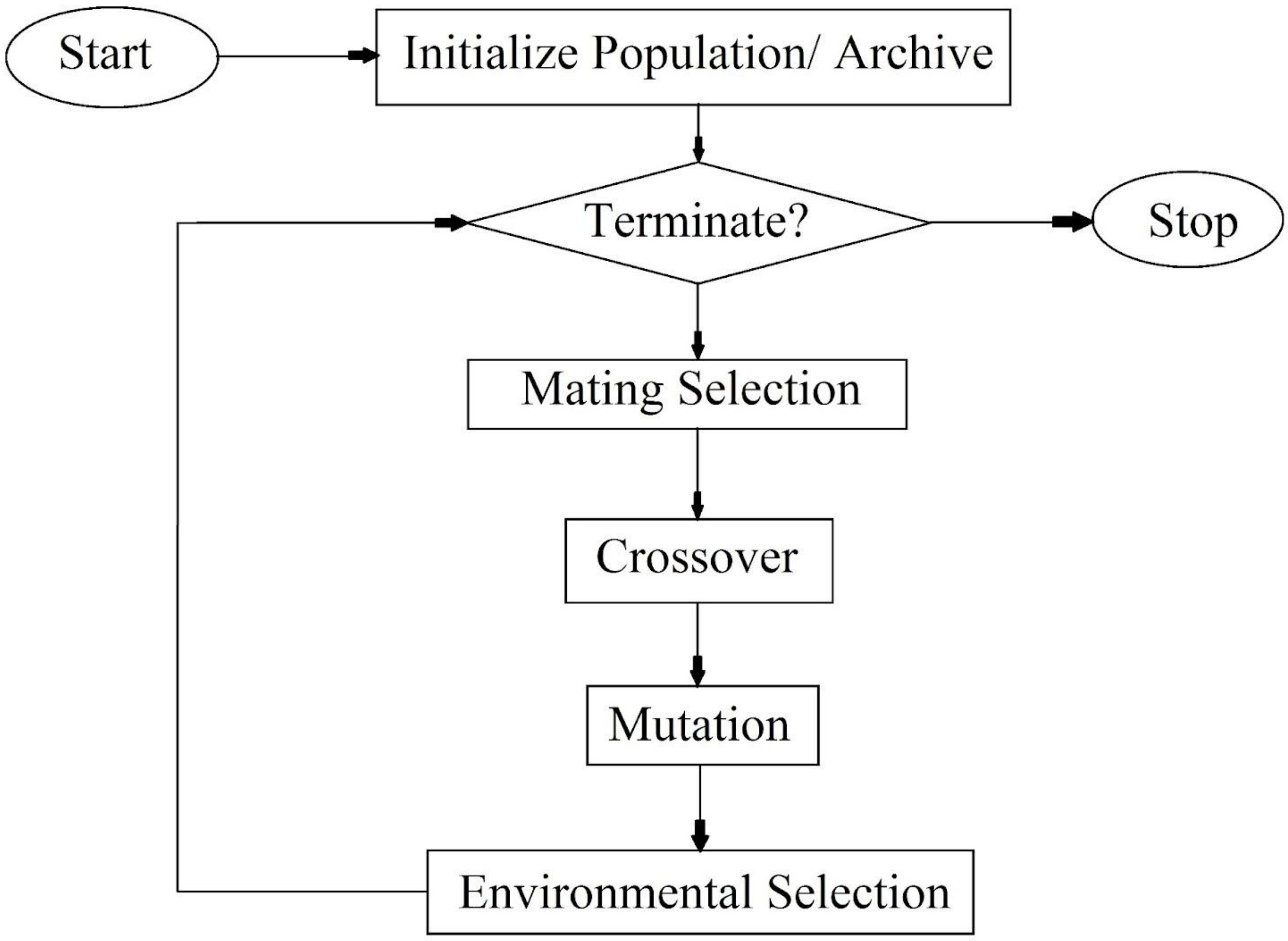


Multi-Objective Evolutionary Algorithms...

1. NSGA
2. NSGA-II
3. SPEA
4. SPEA2
5. GDE3
6. AMGA
7. AMGA2
8. MOEA/D

And many more...

An Overall Flowchart of Multi-Objective Genetic Algorithm



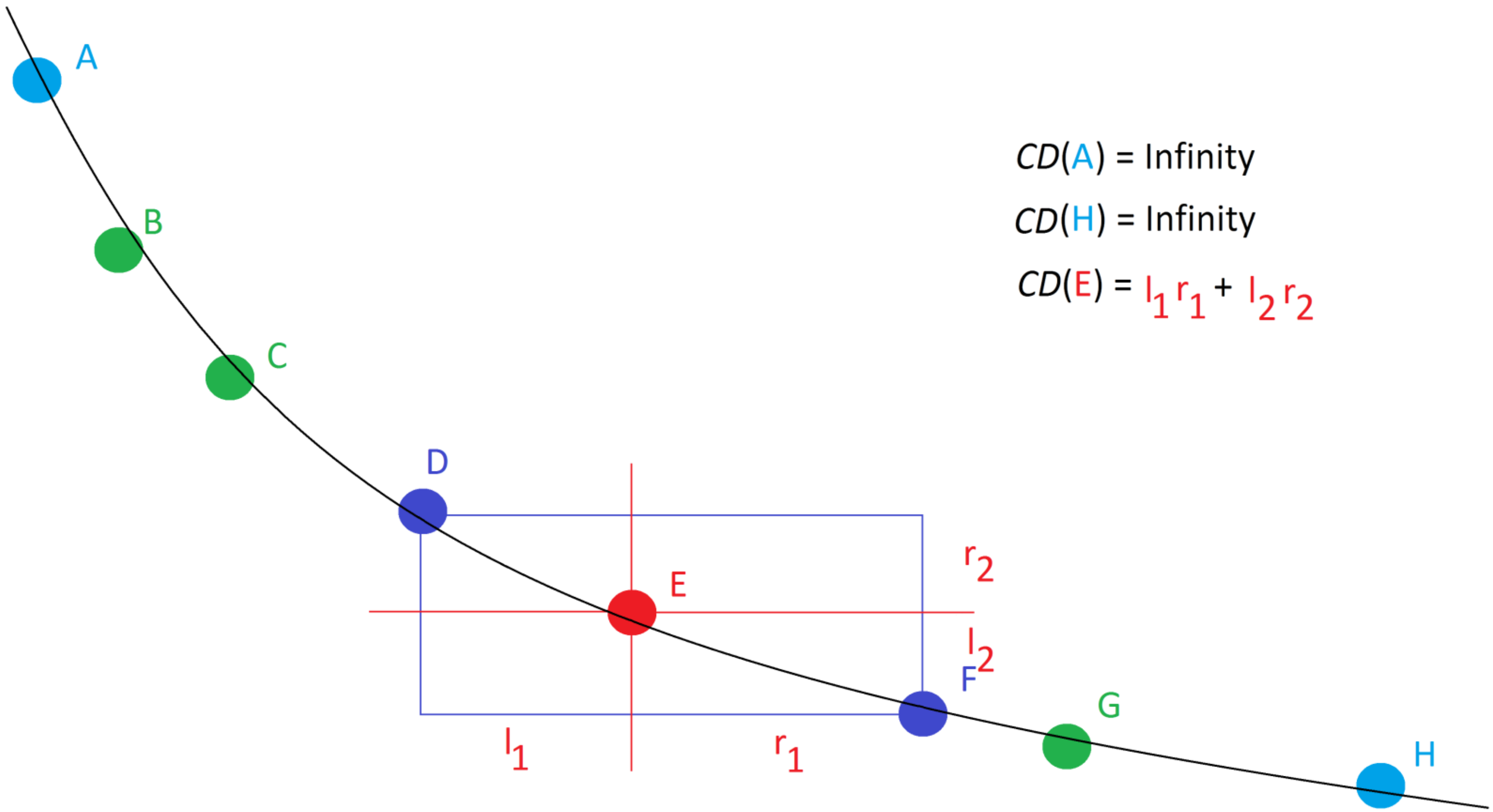
NSGA-II: The Main Loop

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \texttt{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $\lvert P_{t+1}\rvert + \lvert\mathcal{F}_i\rvert \leq N$	until the parent population is filled
$\texttt{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding-distance in $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
$\text{Sort}(\mathcal{F}_i, \prec_n)$	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \lvert P_{t+1}\rvert)]$	choose the first $(N - \lvert P_{t+1}\rvert)$ elements of $\mathcal{F}_i$
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

NSGA-II: *fast-non-dominated-sort*

```
fast-non-dominated-sort(P)
for each p ∈ P
    S_p = ∅
    n_p = 0
    for each q ∈ P
        if (p ≺ q) then                     If p dominates q
            S_p = S_p ∪ {q}                 Add q to the set of solutions dominated by p
        else if (q ≺ p) then
            n_p = n_p + 1                   Increment the domination counter of p
    if n_p = 0 then                         p belongs to the first front
        p_rank = 1
        F_1 = F_1 ∪ {p}
i = 1                                       Initialize the front counter
while F_i ≠ ∅
    Q = ∅                                   Used to store the members of the next front
    for each p ∈ F_i
        for each q ∈ S_p
            n_q = n_q − 1
            if n_q = 0 then                 q belongs to the next front
                q_rank = i + 1
                Q = Q ∪ {q}
    i = i + 1
    F_i = Q
```

NSGA-II: Crowding Distance



NSGA-II: Crowding Distance
How it is assigned…

`crowding-distance-assignment`($\mathcal{I}$)

$l = \lvert\mathcal{I}\rvert$	number of solutions in $\mathcal{I}$
for each i, set $\mathcal{I}[i]_{\text{distance}} = 0$	initialize distance
for each objective m	
$\quad\mathcal{I} = \text{sort}(\mathcal{I}, m)$	sort using each objective value
$\quad\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$	so that boundary points are always selected
$\quad$for $i = 2$ to $(l-1)$	for all other points
$\qquad\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i+1].m - \mathcal{I}[i-1].m)/(f_m^{\max} - f_m^{\min})$	

NSGA-II: Crowding comparator

1) nondomination rank (i_{rank});
2) crowding distance (i_{distance}).

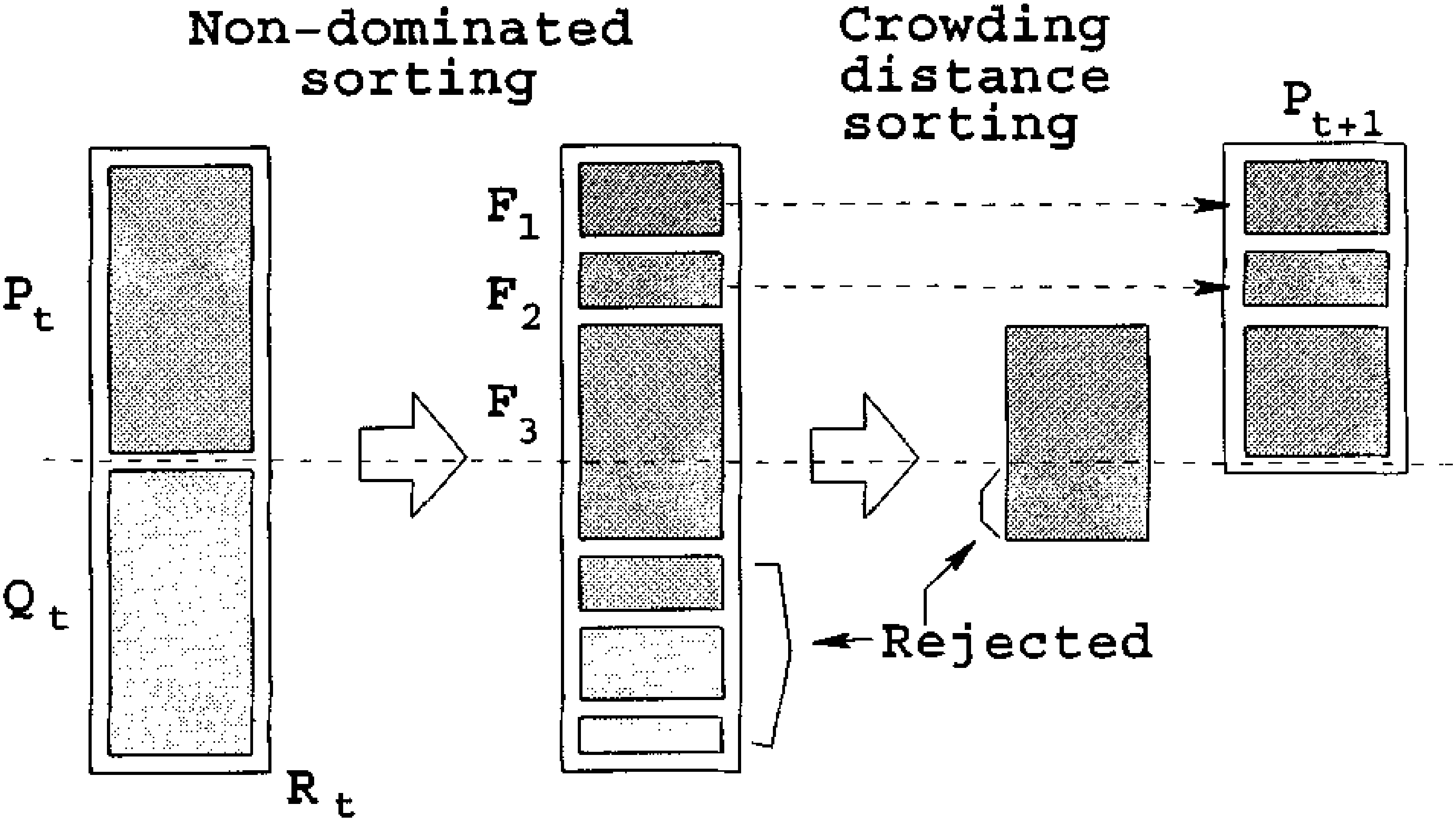
We now define a partial order $\prec_n$ as

$$\begin{aligned} i \prec_n j \quad & \text{if } (i_{\text{rank}} < j_{\text{rank}}) \\ & \text{or } ((i_{\text{rank}} = j_{\text{rank}}) \\ & \text{and } (i_{\text{distance}} > j_{\text{distance}})) \end{aligned} .$$

NSGA-II: Revisiting the Main Loop

$R_t = P_t \cup Q_t$	combine parent and offspring population
$\mathcal{F} = \texttt{fast-non-dominated-sort}(R_t)$	$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$, all nondominated fronts of R_t
$P_{t+1} = \emptyset$ and $i = 1$	
until $\lvert P_{t+1} \rvert + \lvert \mathcal{F}_i \rvert \leq N$	until the parent population is filled
$\texttt{crowding-distance-assignment}(\mathcal{F}_i)$	calculate crowding-distance in $\mathcal{F}_i$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i$	include ith nondominated front in the parent pop
$i = i + 1$	check the next front for inclusion
$\text{Sort}(\mathcal{F}_i, \prec_n)$	sort in descending order using $\prec_n$
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - \lvert P_{t+1} \rvert)]$	choose the first $(N - \lvert P_{t+1} \rvert)$ elements of $\mathcal{F}_i$
$Q_{t+1} = \texttt{make-new-pop}(P_{t+1})$	use selection, crossover and mutation to create a new population Q_{t+1}
$t = t + 1$	increment the generation counter

NSGA-II: How it works…



MOEA: Some Characteristics

- Based on use of Archive
  - Population based
    - NSGA-II
  - Archive based
    - AMGA, AMGA2
- Based on size of working Population
  - Micro genetic
    - AMGA, AMGA2
  - Macro genetic
    - NSGA-II

MOEA: Characteristics cont…

- Based on number of off-springs generated in each generation
  - Generational
    - NSGA-II
  - Steady-state
    - ssNSGA-II
  - In between Generation and Steady-state
    - AMGA-II

Discussion…