Non-Dominated Sorting

Sumit Mishra

Department of Computer Science & Engineering
Indian Institute of Technology Patna

sumitmishra@iitp.ac.in

December 2016

IIT Patna, Patna

Outline

1. Non-Dominated Sorting
   - Solution Representation
   - Dominance Relationship
   - Problem Definition

2. Approaches
   - Naive Approach
   - Fast Non-Dominating Sorting
   - Efficient Non-Dominating Sorting
   - Divide and Conquer based Non-Dominating Sorting

Outline

1. Non-Dominated Sorting
   - Solution Representation
   - Dominance Relationship
   - Problem Definition
2. Approaches
   - Naive Approach
   - Fast Non-Dominating Sorting
   - Efficient Non-Dominating Sorting
   - Divide and Conquer based Non-Dominating Sorting

Solution Representation

Solution Representation

A solution '*sol*' in M-dimensional space is represented as

$$sol = \{f_1(sol), f_2(sol), \ldots, f_M(sol)\} \quad (1)$$

where $f_i(sol), 1 \leq i \leq M$ is the value of solution '*sol*' in i-th dimension.

Representation of 5 solutions

$sol_1 = \{1, 1\}$
$sol_2 = \{1, 2\}$
$sol_3 = \{3, 1\}$
$sol_4 = \{2, 3\}$
$sol_5 = \{4, 2\}$

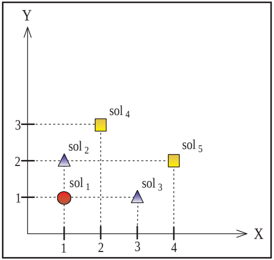


Figure 1 : Solutions in 2-dimensional space.

Outline

1. Non-Dominated Sorting
   - Solution Representation
   - Dominance Relationship
   - Problem Definition

2. Approaches
   - Naive Approach
   - Fast Non-Dominating Sorting
   - Efficient Non-Dominating Sorting
   - Divide and Conquer based Non-Dominating Sorting

Dominance

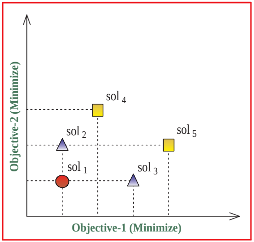
Definition: Dominance (for minimization problem)

A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

In the Figure



Dominance

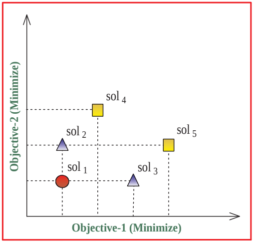
Definition: Dominance (for minimization problem)

A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

In the Figure
$sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$



Dominance

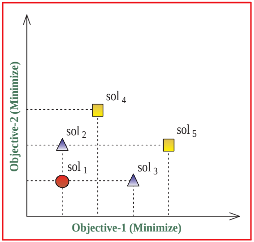
Definition: Dominance (for minimization problem)

A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

In the Figure
$sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$
$sol_2 \prec \{sol_4, sol_5\}$



Dominance

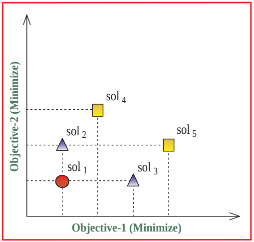
Definition: Dominance (for minimization problem)

A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

In the Figure
$sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$
$sol_2 \prec \{sol_4, sol_5\}$
$sol_3 \prec \{sol_5\}$



Dominance

Definition: Dominance (for minimization problem)

A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

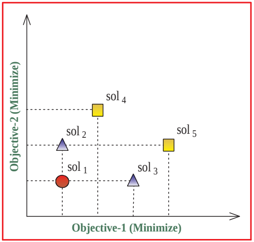
In the Figure

$sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$

$sol_2 \prec \{sol_4, sol_5\}$

$sol_3 \prec \{sol_5\}$

$sol_2 \preceq sol_3$



Dominance

Definition: Dominance (for minimization problem)

A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

In the Figure

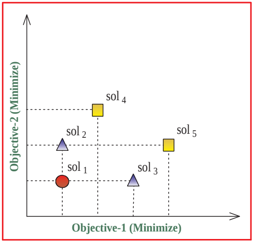
$sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$

$sol_2 \prec \{sol_4, sol_5\}$

$sol_3 \prec \{sol_5\}$

$sol_2 \preceq sol_3$

$sol_3 \preceq sol_4$



Dominance

Definition: Dominance (for minimization problem)

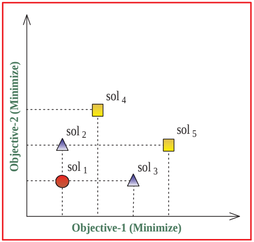
A solution $sol_i = \{f_1(sol_i), f_2(sol_i), \ldots, f_M(sol_i)\}$ dominates another solution $sol_j = \{f_1(sol_j), f_2(sol_j), \ldots, f_M(sol_j)\}$ denoted as $sol_i \prec sol_j$ *iff*

1. $f_m(sol_i) \leq f_m(sol_j) \quad \forall m \in \{1, 2, \ldots, M\}$
2. $f_m(sol_i) < f_m(sol_j) \quad \exists m \in \{1, 2, \ldots, M\}$

sol_i and sol_j are non-dominated represented as $sol_i \preceq sol_j$ *iff* neither $sol_i \prec sol_j$ nor $sol_j \prec sol_i$

In the Figure

$sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$
$sol_2 \prec \{sol_4, sol_5\}$
$sol_3 \prec \{sol_5\}$
$sol_2 \preceq sol_3$
$sol_3 \preceq sol_4$
$sol_4 \preceq sol_5$



Outline

Definition: Non-dominated Sorting [1]

Non-Dominated Sorting is to divide the population $\mathbb{P}$ in $K(1 \leq K \leq N)$ fronts. Let the set of these K fronts in decreasing order of their dominance (increasing order of non-domination level) is $\mathcal{F} = \{F_1, F_2, \ldots, F_K\}$. The division of the solutions in fronts is such that

1. $\forall sol_i, sol_j \in F_k$: $sol_i \preceq sol_j$ $\quad 1 \leq k \leq K$
2. $\forall sol \in F_k$, $\exists sol' \in F_{k-1}$: $sol' \prec sol$ $\quad 2 \leq k \leq K$

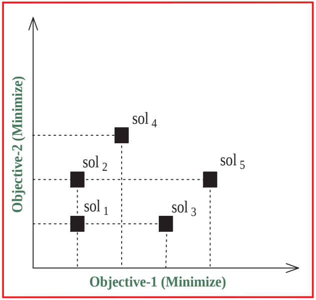


Figure 2 : Solutions.

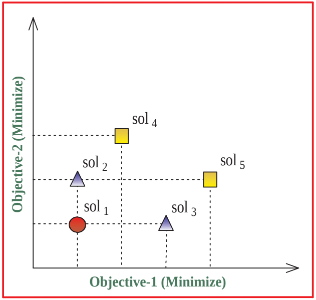


Figure 3 : Non-dominated fronts.

Different Approaches

1. Naive Approach
2. Fast Non-Dominating Sorting [1]
3. Deductive Sort [3]
4. ENS Approach [6]
5. DCNS Approach [4]
6. BOS Sort [5]

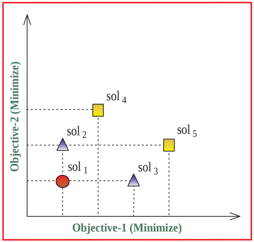
Outline

Domination Count

DEFINITION: Domination Count

Domination count of a solution *'sol'* in population $\mathbb{P}$ is the number of solutions in $\mathbb{P}$ which dominates solution *'sol'*.

In the Figure

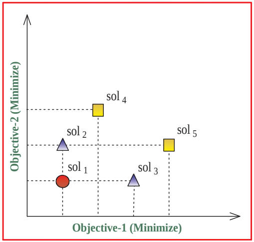


Domination Count

DEFINITION: Domination Count

Domination count of a solution '*sol*' in population $\mathbb{P}$ is the number of solutions in $\mathbb{P}$ which dominates solution '*sol*'.

In the Figure
Domination Count of $sol_1 = 0$

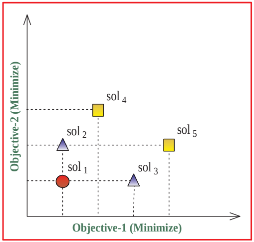


Domination Count

Definition: Domination Count

Domination count of a solution '*sol*' in population $\mathbb{P}$ is the number of solutions in $\mathbb{P}$ which dominates solution '*sol*'.

In the Figure
Domination Count of $sol_1 = 0$
Domination Count of $sol_2 = 1$

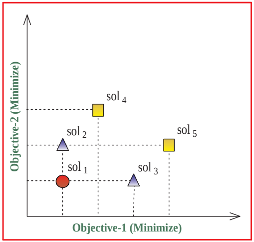


Domination Count

DEFINITION: Domination Count

Domination count of a solution *'sol'* in population $\mathbb{P}$ is the number of solutions in $\mathbb{P}$ which dominates solution *'sol'*.

In the Figure
Domination Count of $sol_1 = 0$
Domination Count of $sol_2 = 1$
Domination Count of $sol_3 = 1$

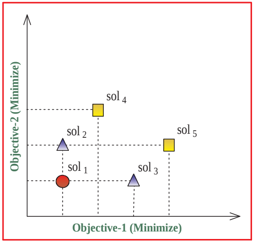


Domination Count

DEFINITION: Domination Count

Domination count of a solution '*sol*' in population $\mathbb{P}$ is the number of solutions in $\mathbb{P}$ which dominates solution '*sol*'.

In the Figure
Domination Count of $sol_1 = 0$
Domination Count of $sol_2 = 1$
Domination Count of $sol_3 = 1$
Domination Count of $sol_4 = 2$

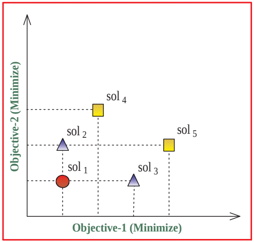


Domination Count

Definition: Domination Count

Domination count of a solution *'sol'* in population $\mathbb{P}$ is the number of solutions in $\mathbb{P}$ which dominates solution *'sol'*.

In the Figure
Domination Count of $sol_1 = 0$
Domination Count of $sol_2 = 1$
Domination Count of $sol_3 = 1$
Domination Count of $sol_4 = 2$
Domination Count of $sol_5 = 3$



Naive Approach : Example

1. For solution sol_1
   - $sol_1 \prec \{sol_2, sol_3, sol_4, sol_5\}$
2. For solution sol_2
   - $sol_2 \prec \{sol_4, sol_5\}$
   - $sol_2 \preceq \{sol_3\}$
   - $sol_1 \prec \{sol_2\}$
3. For solution sol_3
   - $sol_3 \prec \{sol_5\}$
   - $sol_3 \preceq \{sol_2, sol_4\}$
   - $sol_1 \prec \{sol_3\}$
4. For solution sol_4
   - $sol_4 \preceq \{sol_3, sol_5\}$
   - $sol_1, sol_2 \prec \{sol_4\}$
5. For solution sol_5
   - $sol_5 \preceq \{sol_4\}$
   - $sol_1, sol_2, sol_3 \prec \{sol_5\}$

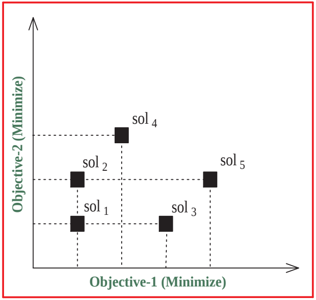


Figure 4 : Solutions

$n_{sol_1} = 0$ $\quad n_{sol_2} = 1$
$n_{sol_3} = 1$ $\quad n_{sol_4} = 2$
$n_{sol_5} = 3$

Naive Approach : Example

1. For solution sol_2
   - $sol_2 \prec \{sol_4, sol_5\}$
   - $sol_2 \preceq \{sol_3\}$
2. For solution sol_3
   - $sol_3 \prec \{sol_5\}$
   - $sol_3 \preceq \{sol_2, sol_4\}$
3. For solution sol_4
   - $sol_4 \preceq \{sol_3, sol_5\}$
   - $sol_2 \prec \{sol_4\}$
4. For solution sol_5
   - $sol_5 \preceq \{sol_4\}$
   - $sol_2, sol_3 \prec \{sol_5\}$

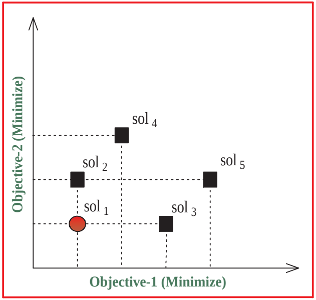


Figure 5 : Solutions

$n_{sol_2} = 0$ $n_{sol_3} = 0$

$n_{sol_4} = 1$ $n_{sol_5} = 2$

Naive Approach : Example

1. For solution sol_4
   - $sol_4 \preceq \{sol_5\}$
2. For solution sol_5
   - $sol_5 \preceq \{sol_4\}$

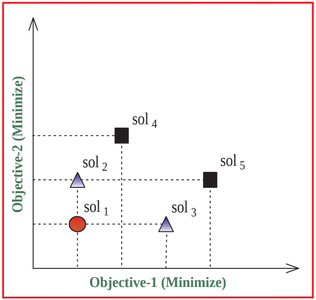


Figure 6 : Solutions

$n_{\text{sol}_4} = 0$ $n_{\text{sol}_5} = 0$

Final Sorted Solutions

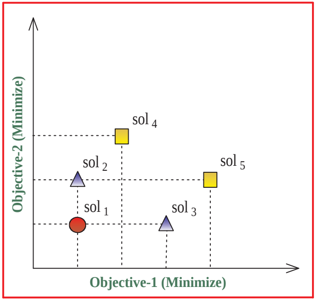


Figure 7 : Sorted Solutions

Naive Approach: Algorithm

Algorithm 1 Naive Approach

Input: $\mathbb{P}$: Population
Output: Ranked solutions

```
rank ← 1
repeat
    Initialize an array of size |ℙ| to store domination count
    for each solution sol ∈ ℙ do
        n_p ← 0                                        // Domination count
        for each solution sol′ ∈ ℙ do
            if sol is dominated by sol′ then
                n_p ← n_p + 1
        Dom_count[sol] ← n_p
    for each solution sol whose domination count is 0 do
        sol_rank ← rank
        ℙ ← ℙ \ {sol}
    rank ← rank + 1
until ℙ becomes empty
```

Complexity Analysis

Space Complexity

- Domination count of each of the solutions is stored.
- Initially the number of solutions is N.
- Space complexity $= \mathcal{O}(N)$

Time Complexity

- Each solution is compared with all other solutions.
  - Time required $= \mathcal{O}(MN^2)$
- This process may be repeated maximum N times.
  - Happens when N solutions are in N different fronts.
- Time complexity $= \mathcal{O}(MN^3)$

Outline

Fast Non-Dominating Sorting: Example

1. For solution sol_1
   - $S_{sol_1} = \{sol_2, sol_3, sol_4, sol_5\}$
   - $n_{sol_1} = 0$
2. For solution sol_2
   - $S_{sol_2} = \{sol_4, sol_5\}$
   - $n_{sol_2} = 1$ $\quad sol_2 \succ \{sol_1\}$
3. For solution sol_3
   - $S_{sol_3} = \{sol_5\}$
   - $n_{sol_3} = 1$ $\quad sol_3 \succ \{sol_1\}$
4. For solution sol_4
   - $S_{sol_4} = \{\}$
   - $n_{sol_4} = 2$ $\quad sol_4 \succ \{sol_1, sol_2\}$
5. For solution sol_5
   - $S_{sol_5} = \{\}$
   - $n_{sol_5} = 3$ $\quad sol_5 \succ \{sol_1, sol_2, sol_3\}$

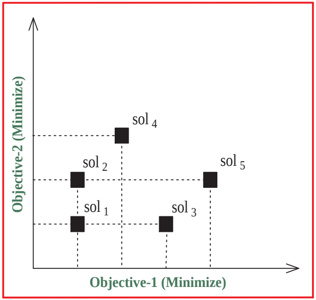


Figure 8 : Solutions

Fast Non-Dominating Sorting: Example

n_{sol_1}	$= 0$	S_{sol_1}	$= \{sol_2, sol_3, sol_4, sol_5\}$
n_{sol_2}	$= 1$	S_{sol_2}	$= \{sol_4, sol_5\}$
n_{sol_3}	$= 1$	S_{sol_3}	$= \{sol_5\}$
n_{sol_4}	$= 2$	S_{sol_4}	$= \{\}$
n_{sol_5}	$= 3$	S_{sol_5}	$= \{\}$

$\mathbf{n}_{sol_1} = \mathbf{0}$

- $\text{sol}_{1_{\text{rank}}} = 1$
- $F_1 = \{\text{sol}_1\}$

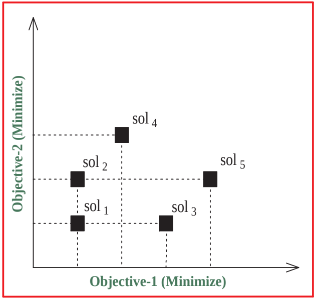


Figure 9 : Solutions

Fast Non-Dominating Sorting: Example

$$
\begin{array}{ll}
n_{sol_1} = 0 & S_{sol_1} = \{sol_2, sol_3, sol_4, sol_5\} \\
n_{sol_2} = 0 & S_{sol_2} = \{sol_4, sol_5\} \\
n_{sol_3} = 1 & S_{sol_3} = \{sol_5\} \\
n_{sol_4} = 2 & S_{sol_4} = \{\} \\
n_{sol_5} = 3 & S_{sol_5} = \{\}
\end{array}
$$

$$
\begin{array}{ll}
n_{sol_1} = 0 & S_{sol_1} = \{sol_2, sol_3, sol_4, sol_5\} \\
n_{sol_2} = 0 & S_{sol_2} = \{sol_4, sol_5\} \\
n_{sol_3} = 0 & S_{sol_3} = \{sol_5\} \\
n_{sol_4} = 2 & S_{sol_4} = \{\} \\
n_{sol_5} = 3 & S_{sol_5} = \{\}
\end{array}
$$

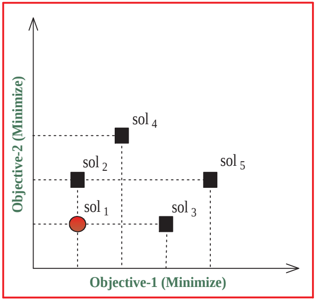


Figure 10 : Solutions

Fast Non-Dominating Sorting: Example

$$
\begin{aligned}
n_{sol_1} &= 0 & S_{sol_1} &= \{\color{red}{sol_2}, \color{red}{sol_3}, \color{red}{sol_4}, sol_5\} \\
n_{sol_2} &= \color{red}{0} & S_{sol_2} &= \{sol_4, sol_5\} \\
n_{sol_3} &= \color{red}{0} & S_{sol_3} &= \{sol_5\} \\
n_{sol_4} &= \color{red}{1} & S_{sol_4} &= \{\} \\
n_{sol_5} &= 3 & S_{sol_5} &= \{\}
\end{aligned}
$$

$$
\begin{aligned}
n_{sol_1} &= 0 & S_{sol_1} &= \{\color{red}{sol_2}, \color{red}{sol_3}, \color{red}{sol_4}, \color{red}{sol_5}\} \\
n_{sol_2} &= \color{red}{0} & S_{sol_2} &= \{sol_4, sol_5\} \\
n_{sol_3} &= \color{red}{0} & S_{sol_3} &= \{sol_5\} \\
n_{sol_4} &= \color{red}{1} & S_{sol_4} &= \{\} \\
n_{sol_5} &= \color{red}{2} & S_{sol_5} &= \{\}
\end{aligned}
$$

$\mathbf{n}_{sol_2} = \mathbf{0}$, $\mathbf{n}_{sol_3} = \mathbf{0}$

- $\text{sol}_{2_{rank}} = \text{sol}_{3_{rank}} = 2$
- $F_2 = \{\text{sol}_2, \text{sol}_3\}$

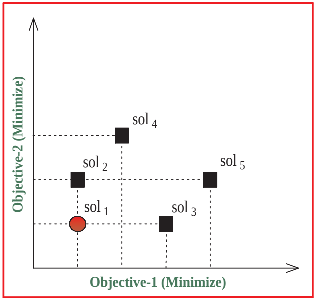


Figure 11 : Solutions

Fast Non-Dominating Sorting: Example

$$
\begin{array}{ll}
n_{sol_1} = 0 & S_{sol_1} = \{sol_2, sol_3, sol_4, sol_5\} \\
n_{sol_2} = 0 & S_{sol_2} = \{sol_4, sol_5\} \\
n_{sol_3} = 0 & S_{sol_3} = \{sol_5\} \\
n_{sol_4} = 0 & S_{sol_4} = \{\} \\
n_{sol_5} = 2 & S_{sol_5} = \{\}
\end{array}
$$

$$
\begin{array}{ll}
n_{sol_1} = 0 & S_{sol_1} = \{sol_2, sol_3, sol_4, sol_5\} \\
n_{sol_2} = 0 & S_{sol_2} = \{sol_4, sol_5\} \\
n_{sol_3} = 0 & S_{sol_3} = \{sol_5\} \\
n_{sol_4} = 0 & S_{sol_4} = \{\} \\
n_{sol_5} = 1 & S_{sol_5} = \{\}
\end{array}
$$

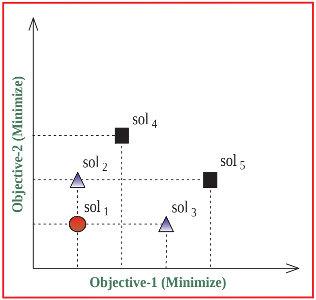


Figure 12 : Solutions

Fast Non-Dominating Sorting: Example

n_{sol_1}	$= 0$	S_{sol_1}	$= \{sol_2, sol_3, sol_4, sol_5\}$
n_{sol_2}	$= 0$	S_{sol_2}	$= \{sol_4, sol_5\}$
n_{sol_3}	$= 0$	S_{sol_3}	$= \{sol_5\}$
n_{sol_4}	$= 0$	S_{sol_4}	$= \{\}$
n_{sol_5}	$= 0$	S_{sol_5}	$= \{\}$

$\mathbf{n}_{sol_4} = \mathbf{0}, \ \mathbf{n}_{sol_5} = \mathbf{0}$

- $\text{sol}_{4_{\text{rank}}} = \text{sol}_{5_{\text{rank}}} = 3$
- $F_3 = \{\text{sol}_4, \text{sol}_5\}$

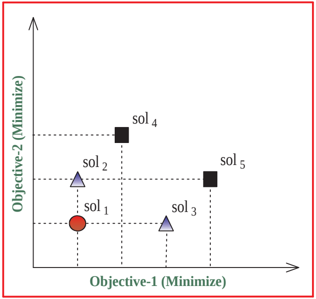


Figure 13 : Solutions

Final Sorted Solutions

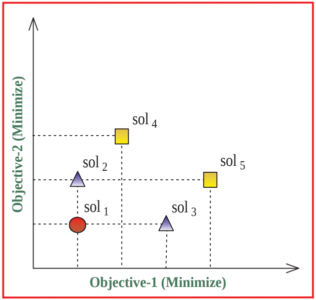


Figure 14 : Solutions

Fast Non-Dominating Sorting: Algorithm

Algorithm 2 Non-dominated Sorting

for each sol $\in \mathbb{P}$ **do**
    $S_{\text{sol}} \leftarrow \Phi,\ n_{\text{sol}} \leftarrow 0$
    for each sol$'$ $\in \mathbb{P}$ **do**
        if sol $\prec$ sol$'$ **then**
            $S_{\text{sol}} \leftarrow S_{\text{sol}} \cup \{\text{sol}'\}$ // Add sol$'$ to the set of solutions dominated by sol
        else if sol$'$ $\prec$ sol **then**
            $n_{\text{sol}} \leftarrow n_{\text{sol}} + 1$ // Increment the domination counter of sol
    if $n_{\text{sol}} = 0$ **then**
        $\text{sol}_{\mathrm{rank}} \leftarrow 1$
        $F_1 \leftarrow F_1 \cup \{\text{sol}\}$
$i \leftarrow 1$ // Initialize the front counter
while $F_i \neq \Phi$ **do**
    $Q \leftarrow \Phi$ // Used to store the members of the next front
    for each sol $\in F_i$ **do**
        for each sol$'$ $\in S_{\text{sol}}$ **do**
            $n_{\text{sol}'} \leftarrow n_{\text{sol}'} - 1$
            if $n_{\text{sol}'} = 0$ **then**
                $\text{sol}'_{\mathrm{rank}} \leftarrow i + 1$
                $Q \leftarrow Q \cup \{\text{sol}'\}$
    $i \leftarrow i + 1$
    $F_i \leftarrow Q$

Complexity Analysis

Space Complexity

- Domination count of each of the solutions is stored.
  - Storage requirement $= \mathcal{O}(N)$
- The set of solutions dominated by all the solutions is stored.
  - Storage requirement $= \mathcal{O}(N^2)$
- Space complexity $= \mathcal{O}(N^2)$

Time Complexity

- Each solution is compared with all other solutions exactly once.
  - Time required $= \mathcal{O}(MN^2)$
- The set of dominated solution sis traversed once and domination count value is reduced.
  - Time required $= \mathcal{O}(N^2)$
- Time complexity $= \mathcal{O}(MN^2)$

Outline

1. Non-Dominated Sorting
   - Solution Representation
   - Dominance Relationship
   - Problem Definition

2. Approaches
   - Naive Approach
   - Fast Non-Dominating Sorting
   - Efficient Non-Dominating Sorting
   - Divide and Conquer based Non-Dominating Sorting

ENS Approach: Basic Idea

- Existing approaches usually compare a solution with all other solutions in the population before assigning it to a front.
- ENS compares it only with those that have already been assigned to a front.
- This is made possible because in ENS, the population is sorted in one objective before actual sorting.
- Thus, a solution added to the fronts cannot dominate any solutions that are added before.
- As a result, ENS can avoid a large number of redundant dominance comparisons.
- Significantly improves the computational efficiency.

ENS Approach

First Phase: Pre-Sorting

The solutions are sorted in ascending order based on the objectives [2].
Advantage: When two solutions sol_i and $sol_j, i > j, 1 \leq i, j \leq N$ are compared, only two possibilities

- sol_i is non-dominated with sol_j
- sol_i is dominated by sol_j.

Second Phase: Assignment

Sorted solutions are assigned to their respective front.

Example

Solution	Objectives
sol_1	1,1
sol_2	1,2
sol_3	3,1
sol_4	2,3
sol_5	4,2

(a)

Solution	Objectives
sol_1	1,1
sol_2	1,2
sol_4	2,3
sol_3	3,1
sol_5	4,2

(b)

Table 1 : (a). A set of 5 solutions where two objectives are associate with each solution. (b). Solutions in sorted order based on objectives.

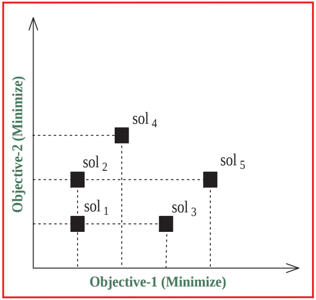


Figure 15 : Solutions

Example

- Rank 1 $= \{sol_1\}$

Solution	Objectives
sol_2	1,2
sol_4	2,3
sol_3	3,1
sol_5	4,2

Table 2 : un-assigned solutions

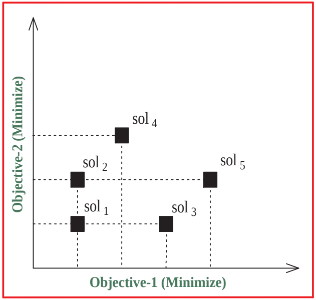


Figure 16 : Solutions

Example

- Rank 1 $= \{sol_1\}$
- Rank 2 $= \{sol_2\}$

Solution	Objectives
sol_4	2,3
sol_3	3,1
sol_5	4,2

Table 3 : un-assigned solutions

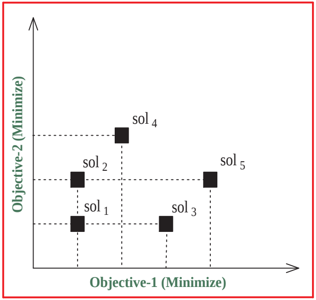


Figure 17 : Solutions

Example

- Rank 1 = $\{sol_1\}$
- Rank 2 = $\{sol_2\}$
- Rank 3 = $\{sol_4\}$

Solution	Objectives
sol_3	3,1
sol_5	4,2

Table 4 : un-assigned solutions

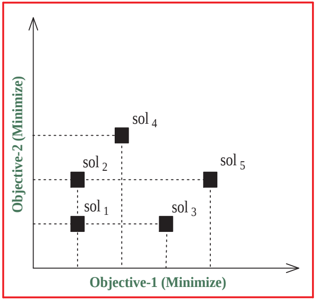


Figure 18 : Solutions

Example

- Rank 1 $= \{sol_1\}$
- Rank 2 $= \{sol_2, sol_3\}$
- Rank 3 $= \{sol_4\}$

Solution	Objectives
sol_5	4,2

Table 5 : un-assigned solutions

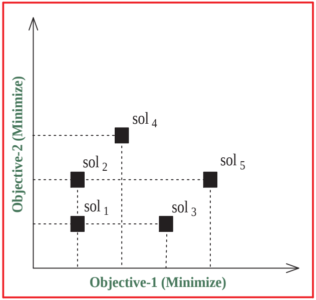


Figure 19 : Solutions

Example

- Rank 1 $= \{sol_1\}$
- Rank 2 $= \{sol_2, sol_3\}$
- Rank 3 $= \{sol_4, sol_5\}$

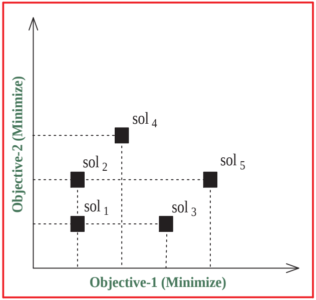


Figure 20 : Solutions

Searching Techniques

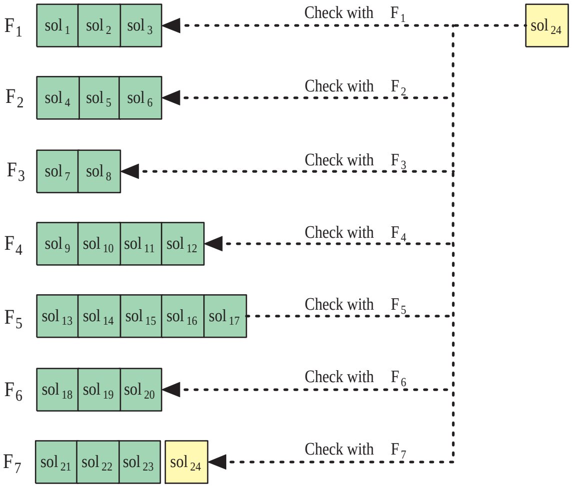
Sequential

To obtain the position of un-assigned solution, sequential search is used in the sorted set of fronts.

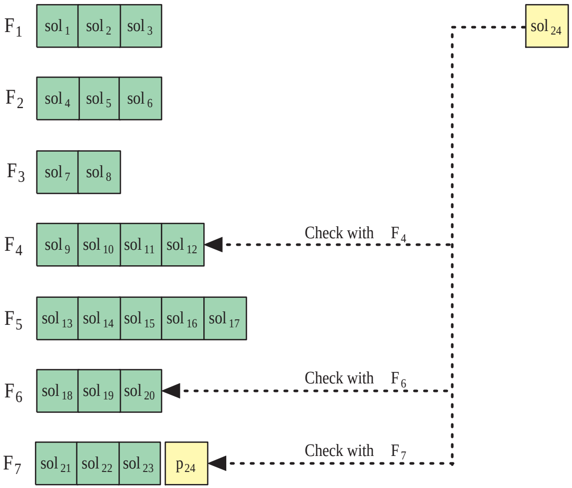
Binary

To obtain the position of un-assigned solution, binary search is used in the sorted set of fronts.

Sequential Search



Binary Search



Complexity Analysis: First Phase

- Solutions are sorted based on objective.
- Heap sort is used. $\mathcal{O}(N \log N)$.
- While comparing two solutions minimum 1 and maximum M objective may be considered.
- So Time Complexity $= \mathcal{O}(MN \log N)$.

Comparisons with Other fronts

- Let N solutions are divided into K fronts.
- The number of solutions in front $F_k = N_k (1 \leq k \leq K)$.
- So $N_1 + N_2 + N_3 + \ldots + N_K = N$
- If a solution sol_i belongs to front F_k, then sol_i should be dominating by at-least one solution belonging to fronts $F_1, F_2, \ldots, F_{k-1}$.
- This means that at-least $i - 1$ comparisons are needed for solution sol_i which is from different fronts.
- Number of solutions in front F_k is N_k, so a total of $(k-1)N_k$ comparisons are needed for front F_k.
- Therefore, the total number of necessary dominance comparisons between solutions in different fronts is:

$$\#Comp_1 = \sum_{k=1}^{K} (k-1) N_k \tag{2}$$

Comparisons with same front

- Each of the N_k solutions in front F_k should be compared with the other solutions in front F_k, which needs a total of $N_k(N_k - 1)/2$ comparisons, the total number of comparisons between solutions in the same front is:

$$\#Comp_2 = \sum_{k=1}^{K} \frac{N_k(N_k - 1)}{2} \tag{3}$$

Total comparisons

$$\#Comp = \#Comp_1 + \#Comp_2 \tag{4}$$

Complexity Analysis: Worst case

When all N solutions in the population are non-dominated with each other.

$$\#Comp_1 = \sum_{k=1}^{1} (k-1)N_k = 0 \tag{5}$$

For the N solutions in the same front, the number of comparisons needed in the sequential search strategy is

$$\#Comp_2 = \sum_{k=1}^{K} \frac{N_k(N_k-1)}{2} = \frac{N(N-1)}{2} \tag{6}$$

Total comparisons

$$\#Comp = \#Comp_1 + \#Comp_2 = \frac{N(N-1)}{2} \tag{7}$$

Complexity Analysis: Best case – Sequential

When N solutions belong to $\lceil\sqrt{N}\rceil$ fronts, each of which roughly has $\lceil\sqrt{N}\rceil$ solutions, and each solution in a front is dominated by all solutions in the preceding front.

$$\#Comp_1 = \sum_{k=1}^{\lceil\sqrt{N}\rceil}(k-1)\lceil\sqrt{N}\rceil = \frac{\lceil\sqrt{N}\rceil^2(\lceil\sqrt{N}\rceil - 1)}{2} \qquad (8)$$

In each of the $\lceil\sqrt{N}\rceil$ fronts, any two solutions of the $\lceil\sqrt{N}\rceil$solutions in the same front need to be compared.

$$\#Comp_2 = \sum_{k=1}^{\lceil\sqrt{N}\rceil}\frac{\lceil\sqrt{N}\rceil(\lceil\sqrt{N}\rceil - 1)}{2} = \frac{\lceil\sqrt{N}\rceil^2(\lceil\sqrt{N}\rceil - 1)}{2} \qquad (9)$$

Total comparisons

$$\#Comp = \#Comp_1 + \#Comp_2 = \lceil\sqrt{N}\rceil^2(\lceil\sqrt{N}\rceil - 1) \qquad (10)$$

Complexity Analysis: Best case – Binary

When N solutions in the population belong to N different fronts. In this case, no solution in any of the N fronts needs to be compared with the solution to be assigned.

$$\#Comp_1 = \sum_{k=1}^{N} \lceil \log k \rceil \tag{11}$$

$$\#Comp_2 = \sum_{k=1}^{N} \frac{1(1-1)}{2} = 0 \tag{12}$$

Total comparisons

$$\#Comp = \#Comp_1 + \#Comp_2 = \sum_{k=1}^{N} \lceil \log k \rceil = \mathcal{O}(N \log N) \tag{13}$$

Outline

1. Non-Dominated Sorting
   - Solution Representation
   - Dominance Relationship
   - Problem Definition
2. Approaches
   - Naive Approach
   - Fast Non-Dominating Sorting
   - Efficient Non-Dominating Sorting
   - Divide and Conquer based Non-Dominating Sorting

Framework

First Phase: Pre-sorting

The solutions are sorted in ascending order based on the objectives [2], [6].
Advantage: When two solutions sol_i and $sol_j, i > j, 1 \leq i, j \leq N$ are compared, only two possibilities

- sol_i is non-dominated with sol_j
- sol_i is dominated by sol_j.

Framework

First Phase: Pre-sorting

The solutions are sorted in ascending order based on the objectives [2], [6].
Advantage: When two solutions sol_i and $sol_j, i > j, 1 \leq i, j \leq N$ are compared, only two possibilities

- sol_i is non-dominated with sol_j
- sol_i is dominated by sol_j.

Second Phase: Assignment Phase

- The actual assignment of solutions to their corresponding front is performed.
- Each solution is considered as a set of fronts.

Framework

Algorithm 3 DCNS framework for non-dominating sorting

Input: $\mathbb{P}$: Population
Output: Set of non-dominated fronts

1: Sort $\mathbb{P}$ in ascending order of objective // First phase
   // Assign the sorted solutions to $\mathcal{F}$
2: **for** $i \leftarrow 1$ to N **do**
3: $\quad F_{i1} \leftarrow \{\mathbb{P}(i)\}$ // Consider a solution as a front
4: $\quad \mathcal{F}_i \leftarrow \{F_{i1}\}$ // Consider a front as a set of front
5: $\quad \mathcal{F} \leftarrow \mathcal{F} \cup \{\mathcal{F}_i\}$
6: Perform non-dominated sorting on $\mathcal{F}$ // Second phase
7: **return** $\mathcal{F}(1)$ // Sorted solutions are in $\mathcal{F}(1)$

Pre-sorting

Example

$\mathbb{P} = \{sol_1, sol_2, sol_3, sol_4, sol_5, sol_6, sol_7\}$.

Solution	Objectives
sol_1	1,2,2
sol_2	2,1,4
sol_3	1,3,1
sol_4	2,1,3
sol_5	1,2,1
sol_6	3,1,5
sol_7	1,1,1

(a)

Solution	Objectives
sol_7	1,1,1
sol_5	1,2,1
sol_1	1,2,2
sol_3	1,3,1
sol_4	2,1,3
sol_2	2,1,4
sol_6	3,1,5

(b)

Table 6 : (a). A set of 7 solutions where three objectives are associate with each solution. (b). Solutions in sorted order based on objectives.

Assignment Phase: Serial Execution

Set of Fronts	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_4$	$\mathcal{F}_5$	$\mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$
Index	1	2	3	4	5	6	7	8

$1 \downarrow 2$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_4$	$\mathcal{F}_5$	$\mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$
Index	1	2	3	4	5	6	7

$2 \downarrow 3$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2$	$\mathcal{F}_3, \mathcal{F}_4$	$\mathcal{F}_5$	$\mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$
Index	1	2	3	4	5	6

$3 \downarrow 4$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2$	$\mathcal{F}_3, \mathcal{F}_4$	$\mathcal{F}_5, \mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$
Index	1	2	3	4	5

$4 \downarrow 5$

– – – – – – – – – – **LEVEL-1** – – – – – – – – – –

$1 \downarrow 2$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$	$\mathcal{F}_5, \mathcal{F}_6$	$\mathcal{F}_7, \mathcal{F}_8$
Index	1	2	3

$2 \downarrow 3$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$	$\mathcal{F}_5, \mathcal{F}_6, \mathcal{F}_7, \mathcal{F}_8$
Index	1	2

– – – – – – – – – – **LEVEL-2** – – – – – – – – – –

$1 \downarrow 2$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6, \mathcal{F}_7, \mathcal{F}_8$
Index	1

– – – – – – – – – – **LEVEL-3** – – – – – – – – – –

Figure 21 : Illustration of the proposed serial approach. $i \downarrow j$ indicates that the set of fronts at index position i and j are merged.

Assignment Phase: Parallel Execution

Set of Fronts	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathcal{F}_4$	$\mathcal{F}_5$	$\mathcal{F}_6$	$\mathcal{F}_7$	$\mathcal{F}_8$
Index	1	2	3	4	5	6	7	8

$1 \downarrow 2$ $3 \downarrow 4$ $5 \downarrow 6$ $7 \downarrow 8$

Set of Fronts	$\mathcal{F}_1, \mathcal{F}_2$		$\mathcal{F}_3, \mathcal{F}_4$		$\mathcal{F}_5, \mathcal{F}_6$		$\mathcal{F}_7, \mathcal{F}_8$	
Index	1	2	3	4	5	6	7	8

– – – – – – – – – – – – – – – – – – **LEVEL-1** – – – – – – – – – – – – – – – – – –

$1 \downarrow 3$ $5 \downarrow 7$

Set of Fronts	$\mathcal{F}_1 - \mathcal{F}_4$				$\mathcal{F}_5 - \mathcal{F}_8$			
Index	1	2	3	4	5	6	7	8

– – – – – – – – – – – – – – – – – – **LEVEL-2** – – – – – – – – – – – – – – – – – –

$1 \downarrow 5$

Set of Fronts	$\mathcal{F}_1 - \mathcal{F}_8$							
Index	1	2	3	4	5	6	7	8

– – – – – – – – – – – – – – – – – – **LEVEL-3** – – – – – – – – – – – – – – – – – –

Figure 22 : Illustration of the proposed parallel approach. $i \downarrow j$ indicates the set of fronts at index position i and j are merged. All the merge operations at the same level are performed simultaneously and no set of fronts is removed.

Merge Procedure

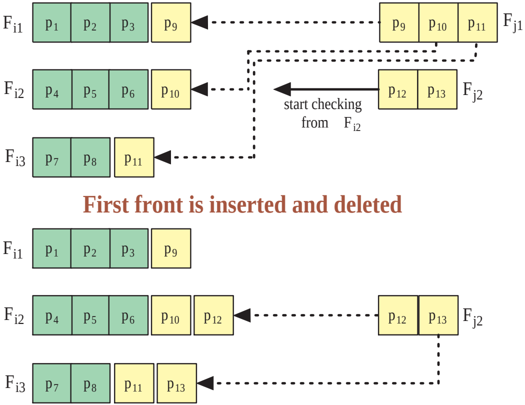


Figure 23 : Merge procedure

Insert Procedure

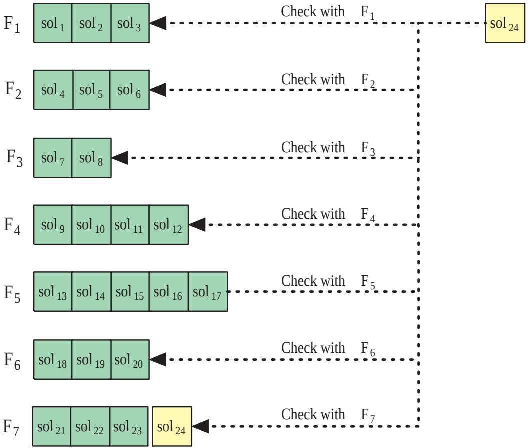


Figure 24 : Insert procedure: Sequential

Insert Procedure

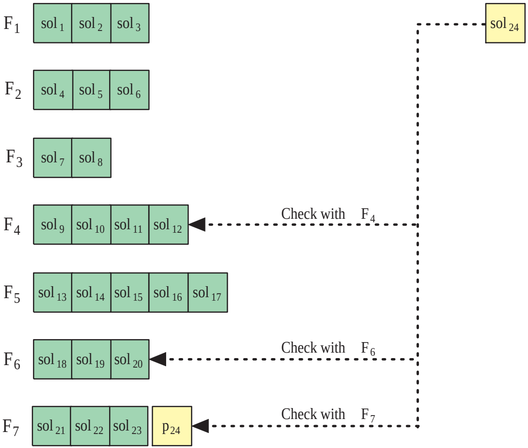


Figure 25 : Insert procedure: Binary

All solutions are non-dominated

$$\begin{aligned} \spadesuit_{SS} = \spadesuit_{BS} &= \sum_{i=1}^{\mathcal{L}} \frac{N}{2^i} \left[(2^{i-1})(2^{i-1}) + {}^{2^{i-1}}C_2 \right] \\ &= \frac{3}{4} N\,(N-1) - \frac{1}{4} N \log N \end{aligned} \tag{14}$$

All solutions are dominating

$$\spadesuit_{SS} = \spadesuit_{SSS} = \sum_{i=1}^{\mathcal{L}} \frac{N}{2^i}\left(2^{i-1}\right) = \frac{1}{2} N \log N \quad (15)$$

$$\spadesuit_{BS} = \spadesuit_{BSS} = \sum_{i=1}^{\mathcal{L}} \frac{N}{2^i}\lceil \log(2^{i-1}+1)\rceil \quad (16)$$

$\sqrt{N}$ solutions in $\sqrt{N}$ fronts

$$\spadesuit_{SS} = \sum_{i=1}^{\frac{\mathcal{L}}{2}} \frac{N}{2^i}\left[\left(2^{i-1}\right)\left(2^{i-1}\right) + {}^{2^{i-1}}C_2\right] + \sum_{i=\frac{\mathcal{L}}{2}+1}^{\mathcal{L}} \frac{N}{2^i} 2^{\frac{\mathcal{L}}{2}} 2^{i-\left(\frac{\mathcal{L}}{2}+1\right)}$$

$$= \frac{3}{4}N\left(\sqrt{N}-1\right) + \frac{1}{8}N\log N \tag{17}$$

$$\spadesuit_{BS} = \sum_{i=1}^{\frac{\mathcal{L}}{2}} \frac{N}{2^i}\left[(2^{i-1})(2^{i-1}) + {}^{2^{i-1}}C_2\right] + \sum_{i=\frac{\mathcal{L}}{2}+1}^{\mathcal{L}} \frac{N}{2^i} 2^{\frac{\mathcal{L}}{2}} \log\left\lceil 2^{i-\left(\frac{\mathcal{L}}{2}+1\right)}+1\right\rceil \tag{18}$$

References I

[1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan.
A fast and elitist multiobjective genetic algorithm: Nsga-ii.
Evolutionary Computation, IEEE Transactions on, 6(2):182–197, 2002.

[2] Mikkel T Jensen.
Reducing the run-time complexity of multiobjective eas: The nsga-ii and other algorithms.
Evolutionary Computation, IEEE Transactions on, 7(5):503–515, 2003.

[3] Kent McClymont and Ed Keedwell.
Deductive sort and climbing sort: New methods for non-dominated sorting.
Evolutionary computation, 20(1):1–26, 2012.

[4] Sumit Mishra, Sriparna Saha, and Samrat Mondal.
Divide and conquer based non-dominated sorting for parallel environment.
In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 4297–4304. IEEE, 2016.

[5] Proteek Chandan Roy, Md Monirul Islam, and Kalyanmoy Deb.
Best order sort: A new algorithm to non-dominated sorting for evolutionary multi-objective optimization.

[6] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin.
An efficient approach to nondominated sorting for evolutionary multiobjective optimization.
Evolutionary Computation, IEEE Transactions on, 19(2):201–213, 2015.

Thank you!