NSGA-II

Douglas Nunes de Oliveira

Índice

- Fast Non-Dominated Sort – Parte I
- Fast Non-Dominated Sort – Parte II
- Crowding Distance Assignment
- Main Loop

Fast Non-Dominated Sort – Parte I

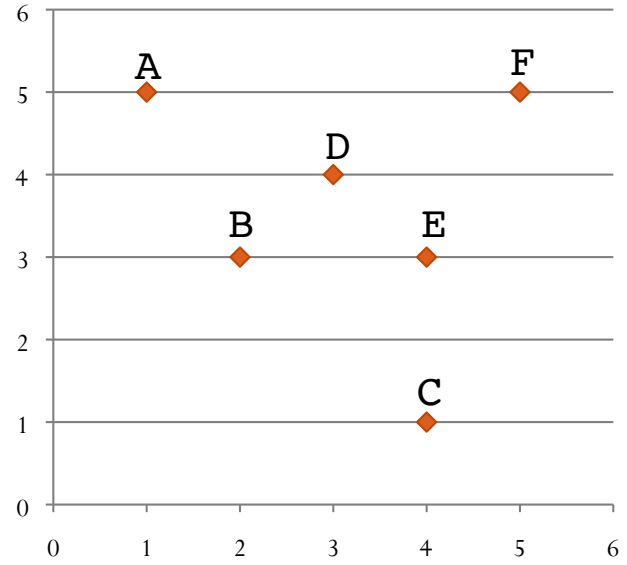
Definição da primeira fronteira e das dominâncias.

p = A

```
P = {A, B, C, D, E, F}
p = A
q = B
A → np=0; Sp={}; rank=0;
q = C
A → np=0; Sp={}; rank=0;
q = D
A → np=0; Sp={}; rank=0;
q = E
A → np=0; Sp={}; rank=0;
q = F
A → np=0; Sp={F}; rank=0;
A → np=0; Sp={F}; rank=1;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={}; rank=0;
C → np=0; Sp={}; rank=0;
D → np=0; Sp={}; rank=0;
E → np=0; Sp={}; rank=0;
F → np=0; Sp={}; rank=0;

F1 → {A}
```

```
1 for each p : P
2   Sp = {};
3   np = 0;
4   for each q : P
5       if (p < q) then
6           Sp = Sp U {q};
7       else if (q < p) then
8           np = np + 1;
9   if (np = 0) then
10      pRank = 1;
11      F1 = F1 U {p}
```

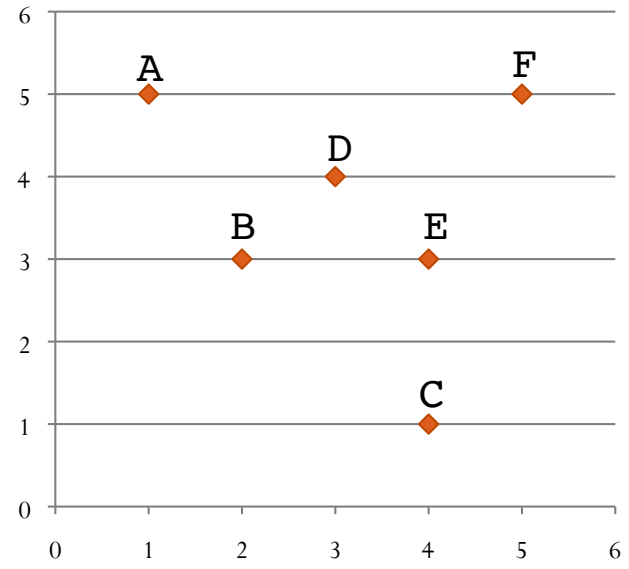


p = B

```
P = {A, B, C, D, E, F}
p = B
q = A
B → np=0; Sp={}; rank=0;
q = C
B → np=0; Sp={}; rank=0;
q = D
B → np=0; Sp={D}; rank=0;
q = E
B → np=0; Sp={D,E}; rank=0;
q = F
B → np=0; Sp={D,E,F}; rank=0;
B → np=0; Sp={D,E,F}; rank=1;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={}; rank=0;
D → np=0; Sp={}; rank=0;
E → np=0; Sp={}; rank=0;
F → np=0; Sp={}; rank=0;

F1 → {A,B}
```

```
1 for each p : P
2   Sp = {};
3   np = 0;
4   for each q : P
5       if (p < q) then
6           Sp = Sp U {q};
7       else if (q < p) then
8           np = np + 1;
9   if (np = 0) then
10      pRank = 1;
11      F1 = F1 U {p}
```

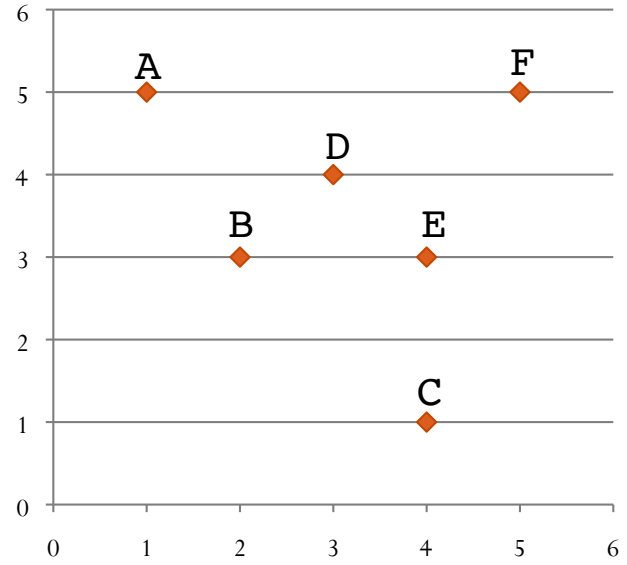


p = C

```
P = {A, B, C, D, E, F}
p = C
q = A
C → np=0; Sp={}; rank=0;
q = B
C → np=0; Sp={}; rank=0;
q = D
C → np=0; Sp={}; rank=0;
q = E
C → np=0; Sp={E}; rank=0;
q = F
C → np=0; Sp={E,F}; rank=0;
C → np=0; Sp={E,F}; rank=1;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=0; Sp={}; rank=0;
E → np=0; Sp={}; rank=0;
F → np=0; Sp={}; rank=0;

F1 → {A,B,C}
```

```
1 for each p : P
2   Sp = {};
3   np = 0;
4   for each q : P
5       if (p < q) then
6           Sp = Sp U {q};
7       else if (q < p) then
8           np = np + 1;
9   if (np = 0) then
10      pRank = 1;
11      F1 = F1 U {p}
```

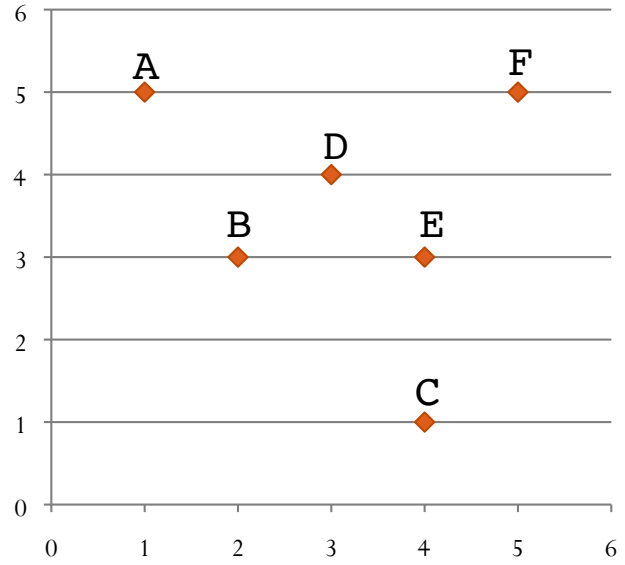


p = D

```
P = {A, B, C, D, E, F}
p = D
q = A
D → np=0; Sp={}; rank=0;
q = B
D → np=1; Sp={}; rank=0;
q = C
D → np=1; Sp={}; rank=0;
q = E
D → np=1; Sp={}; rank=0;
q = F
D → np=1; Sp={F}; rank=0;
D → np=1; Sp={F}; rank=0;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=1; Sp={F}; rank=0;
E → np=0; Sp={}; rank=0;
F → np=0; Sp={}; rank=0;

F1 → {A,B,C}
```

```
1 for each p : P
2   Sp = {};
3   np = 0;
4   for each q : P
5       if (p < q) then
6           Sp = Sp U {q};
7       else if (q < p) then
8           np = np + 1;
9   if (np = 0) then
10      pRank = 1;
11      F1 = F1 U {p}
```

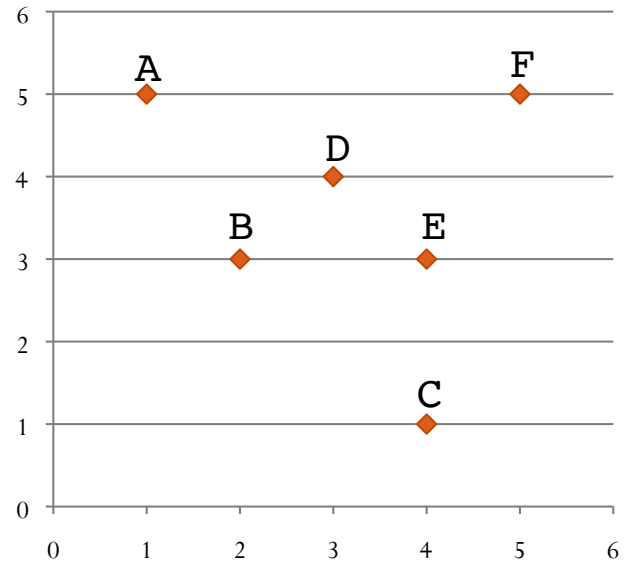


p = E

```
P = {A, B, C, D, E, F}
p = E
q = A
E → np=0; Sp={}; rank=0;
q = B
E → np=1; Sp={}; rank=0;
q = C
E → np=2; Sp={}; rank=0;
q = D
E → np=2; Sp={}; rank=0;
q = F
E → np=2; Sp={F}; rank=0;
E → np=2; Sp={F}; rank=0;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=1; Sp={F}; rank=0;
E → np=2; Sp={F}; rank=0;
F → np=0; Sp={}; rank=0;

F1 → {A,B,C}
```

```
1 for each p : P
2   Sp = {};
3   np = 0;
4   for each q : P
5       if (p < q) then
6           Sp = Sp U {q};
7       else if (q < p) then
8           np = np + 1;
9   if (np = 0) then
10      pRank = 1;
11      F1 = F1 U {p}
```

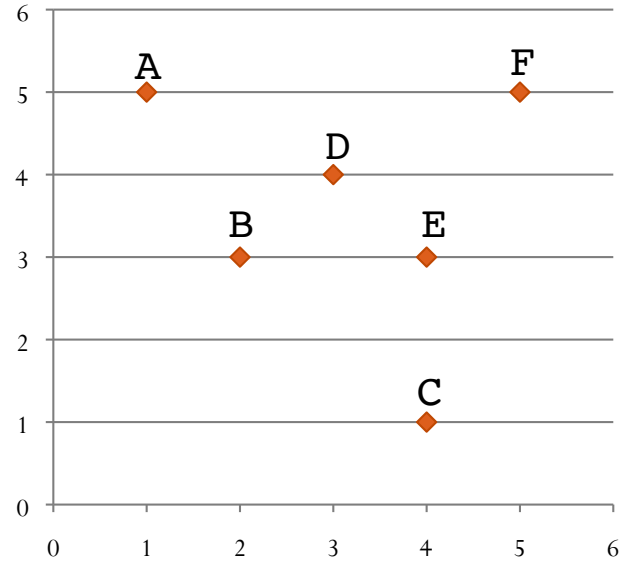


p = F

```
P = {A, B, C, D, E, F}
p = F
q = A
F → np=1; Sp={}; rank=0;
q = B
F → np=2; Sp={}; rank=0;
q = C
F → np=3; Sp={}; rank=0;
q = D
F → np=4; Sp={}; rank=0;
q = E
F → np=5; Sp={}; rank=0;
F → np=5; Sp={}; rank=0;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=1; Sp={F}; rank=0;
E → np=2; Sp={F}; rank=0;
F → np=5; Sp={}; rank=0;

F1 → {A,B,C}
```

```
1 for each p : P
2   Sp = {};
3   np = 0;
4   for each q : P
5       if (p < q) then
6           Sp = Sp U {q};
7       else if (q < p) then
8           np = np + 1;
9   if (np = 0) then
10      pRank = 1;
11      F1 = F1 U {p}
```



Fast Non-Dominated Sort – Parte II

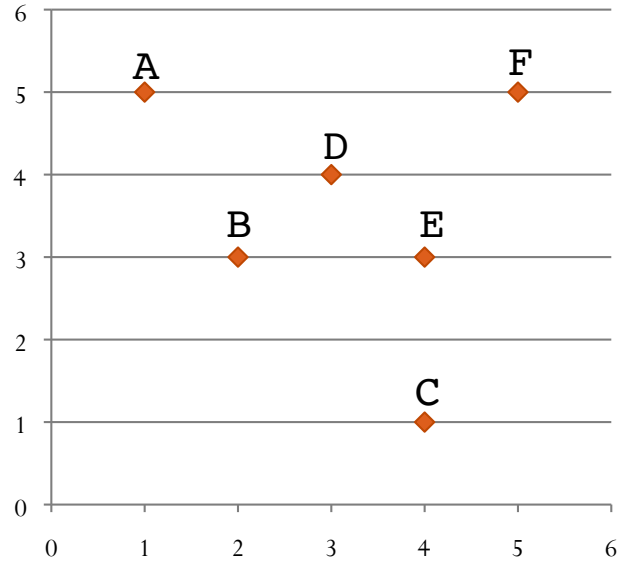
As demais fronteiras.

Fi = F1; p = A

```
Q = {};
p = A;
Sp = {F}
q = F
F → np=4; Sp={}; rank=0;
```

```
1 i = 1;
2 while Fi <> {}
3   Q = {};
4   for each p : Fi
5       for each q : Sp
6           nq = nq - 1;
7           if nq = 0 then
8               q.rank = i + 1;
9               Q = Q U {q};
10  i = i + 1;
11  Fi = Q;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=1; Sp={F}; rank=0;
E → np=2; Sp={F}; rank=0;
F → np=4; Sp={}; rank=0;

F1 → {A,B,C}
```

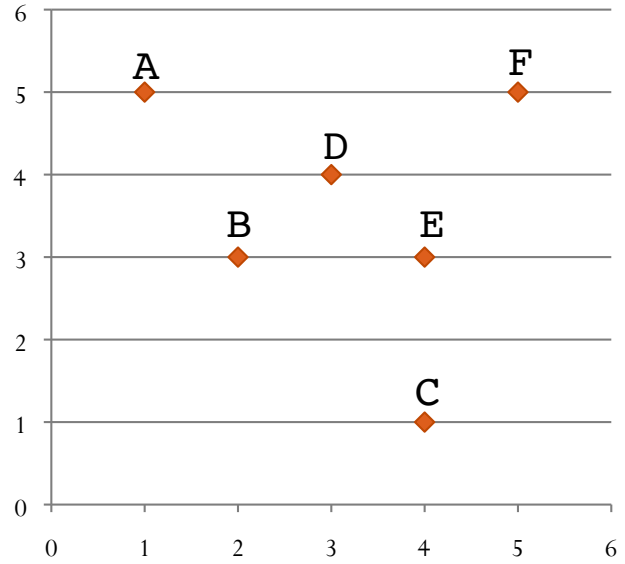


Fi = F1; p = B

```
Q = {};
p = B;
Sp = {D,E,F};
q = D;
D → np=0; Sp={F}; rank=2;
Q = {D};
q = E;
E → np=1; Sp={F}; rank=0;
q = F
F → np=3; Sp={}; rank=0;
```

```
1 i = 1;
2 while Fi <> {}
3   Q = {};
4   for each p : Fi
5       for each q : Sp
6           nq = nq - 1;
7           if nq = 0 then
8               q.rank = i + 1;
9               Q = Q U {q};
10  i = i + 1;
11  Fi = Q;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=0; Sp={F}; rank=2;
E → np=1; Sp={F}; rank=0;
F → np=3; Sp={}; rank=0;

F1 → {A,B,C}
```

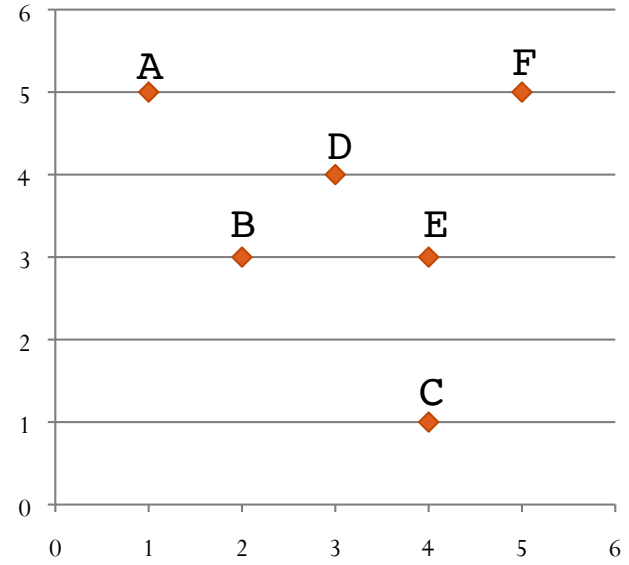


Fi = F1; p = C

```
Q = {D};
p = C;
Sp = {E,F};
q = E;
E → np=0; Sp={F}; rank=2;
Q = {D,E};
q = F
F → np=2; Sp={}; rank=0;
i = 2;
F2 = Q;
F2 = {D,E};
```

```
1 i = 1;
2 while Fi <> {}
3   Q = {};
4   for each p : Fi
5       for each q : Sp
6           nq = nq - 1;
7           if nq = 0 then
8               q.rank = i + 1;
9               Q = Q U {q};
10  i = i + 1;
11  Fi = Q;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=0; Sp={F}; rank=2;
E → np=0; Sp={F}; rank=2;
F → np=2; Sp={}; rank=0;

F1 → {A,B,C}
F2 → {D,E}
```

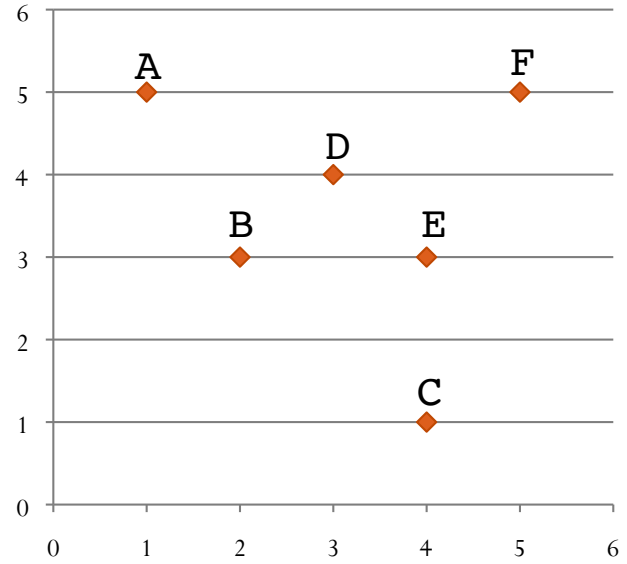


Fi = F2; p = D

```
Q = {};
p = D;
Sp = {F};
q = F;
F → np=1; Sp={}; rank=0;
```

```
1 i = 1;
2 while Fi <> {}
3   Q = {};
4   for each p : Fi
5       for each q : Sp
6           nq = nq - 1;
7           if nq = 0 then
8               q.rank = i + 1;
9               Q = Q U {q};
10  i = i + 1;
11  Fi = Q;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=0; Sp={F}; rank=2;
E → np=0; Sp={F}; rank=2;
F → np=1; Sp={}; rank=0;

F1 → {A,B,C}
F2 → {D,E}
```

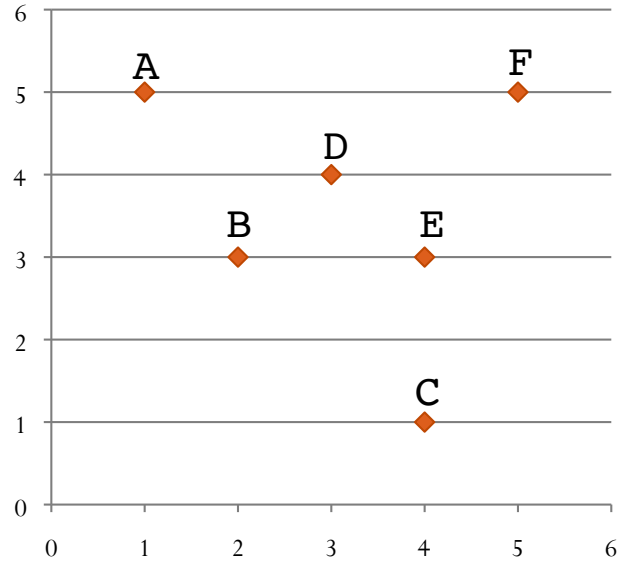


Fi = F2; p = E

```
Q = {};
p = E;
Sp = {F};
q = F;
F → np=0; Sp={}; rank=3;
Q = {F};
i = 3;
F3 = Q;
F3 = {F};
```

```
1 i = 1;
2 while Fi <> {}
3   Q = {};
4   for each p : Fi
5       for each q : Sp
6           nq = nq - 1;
7           if nq = 0 then
8               q.rank = i + 1;
9               Q = Q U {q};
10  i = i + 1;
11  Fi = Q;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=0; Sp={F}; rank=2;
E → np=0; Sp={F}; rank=2;
F → np=0; Sp={}; rank=3;

F1 → {A,B,C}
F2 → {D,E}
F3 → {F}
```

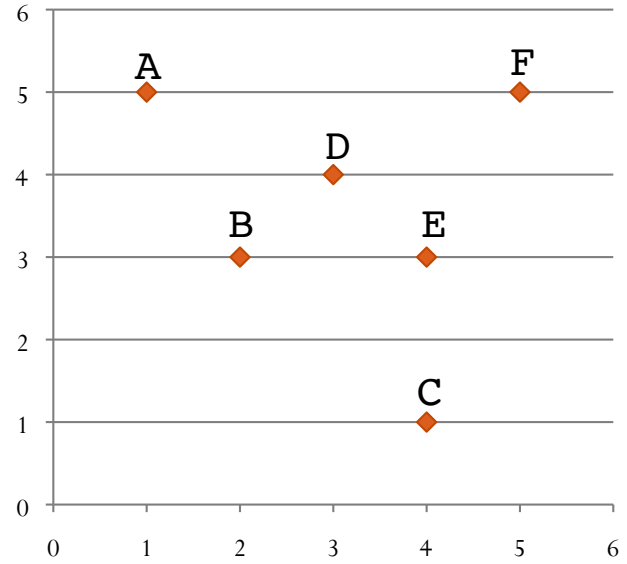


Fi = F3; p = F

```
Q = {};
p = E;
Sp = {};
i = 4;
F4 = Q;
F4 = {};
```

```
1 i = 1;
2 while Fi <> {}
3   Q = {};
4   for each p : Fi
5       for each q : Sp
6           nq = nq - 1;
7           if nq = 0 then
8               q.rank = i + 1;
9               Q = Q U {q};
10  i = i + 1;
11  Fi = Q;
```

```
A → np=0; Sp={F}; rank=1;
B → np=0; Sp={D,E,F}; rank=1;
C → np=0; Sp={E,F}; rank=1;
D → np=0; Sp={F}; rank=2;
E → np=0; Sp={F}; rank=2;
F → np=0; Sp={}; rank=3;

F1 → {A,B,C}
F2 → {D,E}
F3 → {F}
F4 → {}
```



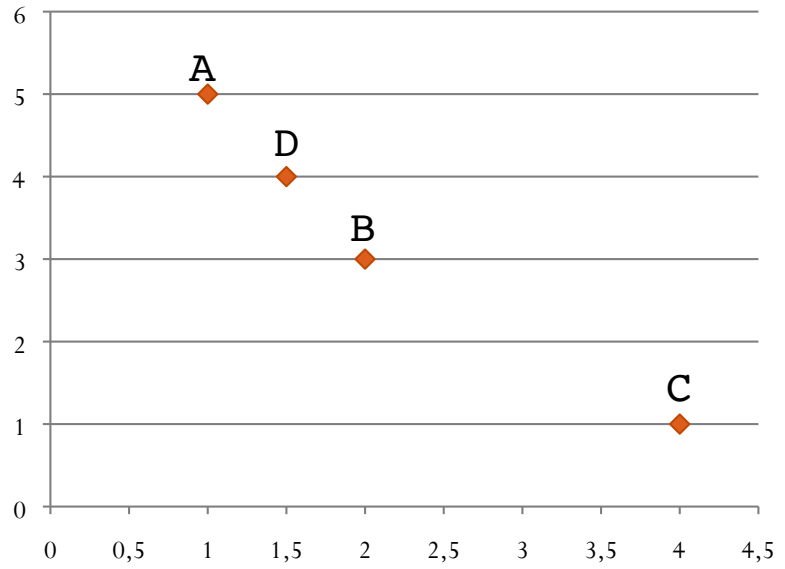
Crowding Distance Assignment

Objetivo 1

```
T = {A, B, C, D};
l = 4
m = 1;
T = sort(T, m);
T = {A, D, B, C}
T[1] = inf+; T[4] = inf+;

i = 2; \\to(l-1)=3
aux = (2-1)/(4-1) = 0.333;
D.d = 0 + 0.333;

i = 3; \\to(l-1)=3
aux = (4-1.5)/(4-1) = 0.833;
B.d = 0 + 0.833;
```

```
A → distance=inf+; m1=1; m2=5;
D → distance=0.333; m1=1.5; m2=4;
B → distance=0.833; m1=2; m2=3;
C → distance=inf+; m1=4; m2=1;
```

```
l = |T|;
for each i, set T[i].d = 0;

for each objective m
    T = sort(T, m);
    T[1].distance = inf+;
    T[l].distance = inf+;
    for i = 2 to (l-1)
        aux = (T[i+1].m-T[i-1].m)/
              (fm_max-fm_min);
        T[i].d = T[i].d + aux;
```

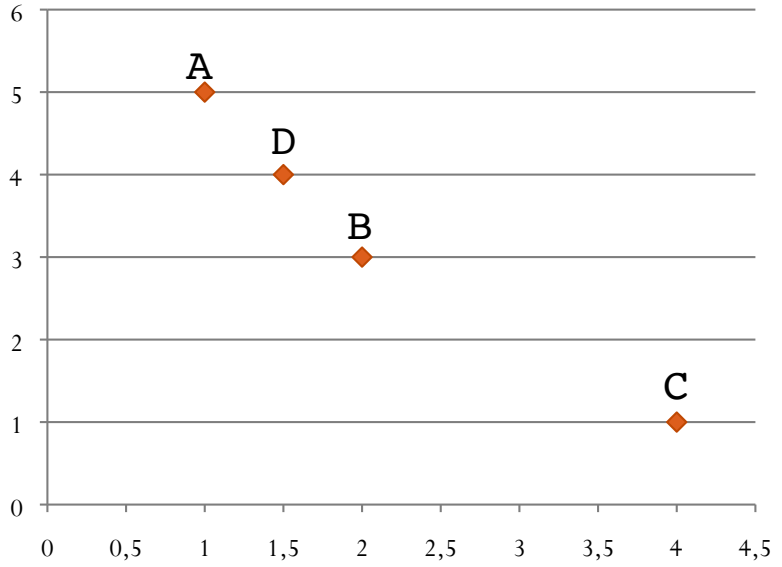


Objetivo 2

```
l = 4;

m = 2;
T = {A, D, B, C};
T = sort(T, m);
T = {C, B, D, A}
T[1] = inf+; T[4] = inf+;

i = 2; \\to(l-1)=3
aux = (4-1)/(5-1) = 0.750;
B.d = 0.833 + 0.750 = 1.583;

i = 3; \\to(l-1)=3
aux = (5-3)/(5-1) = 0.500;
D.d = 0.333 + 0.500 = 0.833;
```

```
C → distance=inf+; m1=4; m2=1;
B → distance=1.583; m1=2; m2=3;
D → distance=0.833; m1=1.5; m2=4;
A → distance=inf+; m1=1; m2=5;
```

```
l = |T|;
for each i, set T[i].d = 0;

for each objective m
    T = sort(T, m);
    T[1].distance = inf+;
    T[l].distance = inf+;
    for i = 2 to (l-1)
        aux = (T[i+1].m-T[i-1].m)/
              (fm_max-fm_min);
        T[i].d = T[i].d + aux;
```



Main Loop

N = 4

P_t = {A,C,D};
Q_t = {B,E,F};
R_t = {A,C,D,B,E,F};
P_{t+1} = {};
i = 1;
($|P_{t+1}|$ = 0 + $|F_1|$ = 3) <= 4
    cr-dist-assig (F_1);
    P_{t+1} = {A,B,C};
    i = 2;
($|P_{t+1}|$ = 3 + $|F_2|$ = 2) <= 4

Sort(F_2, $<_n$);
P_{t+1} = {A,B,C} U F_2[1:(N-$|P_{t+1}|$)];
P_{t+1} = {A,B,C} U F_2[1:(4-3)];
P_{t+1} = {A,B,C} U F_2[1:1];
P_{t+1} = {A,B,C} U {D};
P_{t+1} = {A,B,C,D};

F_1 → {A,B,C}
F_2 → {D,E}
F_3 → {F}
F_4 → {}

R_t = P_t U Q_t;
F = fast-non-dominated-sort(R_t);
P_{t+1} = {};
i=1;

while $|P_{t+1}|$ + $|F_i|$ <= N
    cr-dist-assig (F_i);
    P_{t+1} = P_{t+1} U F_i;
    i = i + 1;

Sort(F_i, $<_n$);
P_{t+1} = P_{t+1} U F_i[1:(N-$|P_{t+1}|$)];
Q_{t+1} = make-new-pop(P_{t+1});
t = t + 1;