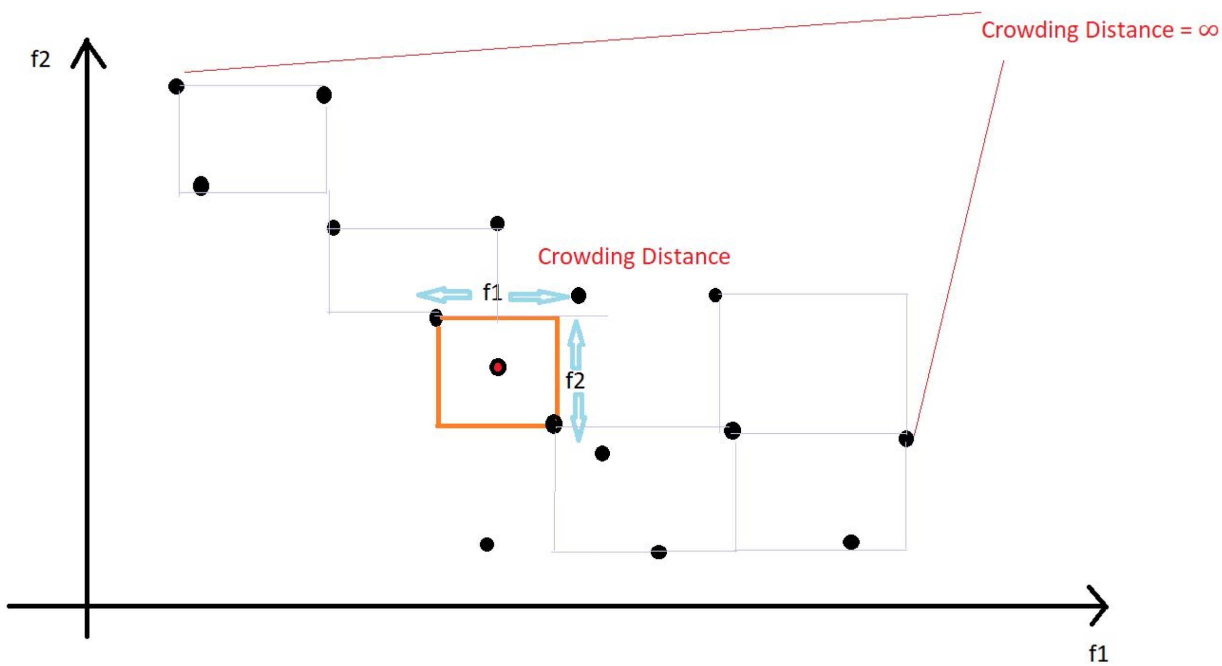
Submitted by: Ranji Raj

Assignment 23 (Crowding Distance)

- Explain the purpose of this operator. What is it used for in NSGA-II and how does it work?

Purpose: To select the solutions and maintain **diversity** among them.



- To select among solutions in a front which cannot all fit into the next population.
- Compute the distances to the **closest** neighbours in each objective separately, and sum up.
- Solutions with **larger** CD's are preferred.
- The **extreme** points get the **largest** CD values.
- No archive present only works upon the population.

Algorithm NSGA-II

Input: Search Space S, size of the population N

Output: $P(t)$

begin

$t = 0$

initialize a random population $P(t)$

repeat

Evaluate $P(t)$

$M(t) =$ Apply tournament selection $(p(t))$

crossover: $M'(t) = c(M(t))$

mutation: $Q(t) = m(M'(t))$

$R(t) = P(t) \cup Q(t)$

$\mathcal{F}$ = fast-non-dominated-sort $R(t)$

$P(t+1) = \emptyset$ and $i = 1$

until $|P(t+1)| + |\mathcal{F}_i| \leq N$

$P(t+1) = P(t+1) \cup \mathcal{F}_i$

$i = i + 1$

end

$\boxed{P(t+1) = P(t+1) \cup \mathcal{F}_i[1{:}N - |P(t+1)|]}$

$t = t + 1$

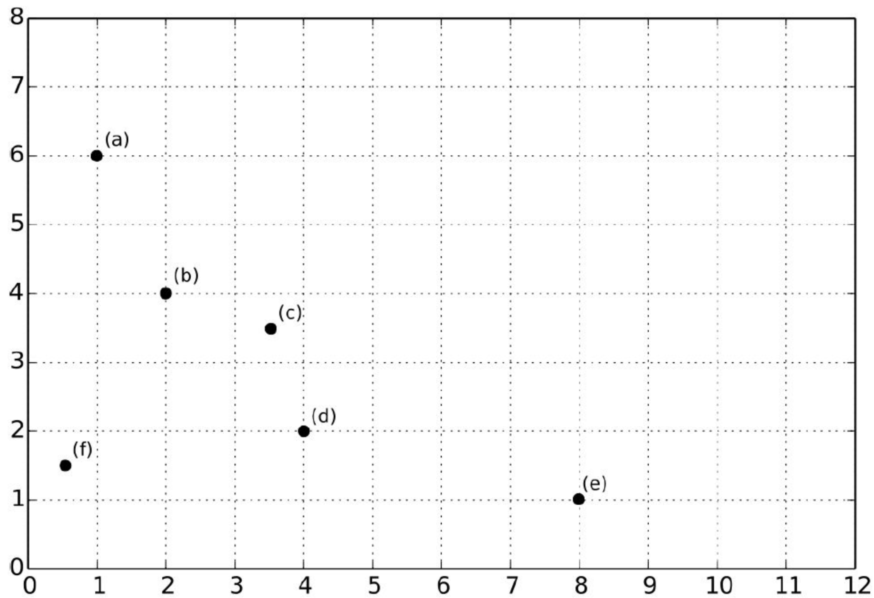
until *termination-condition*

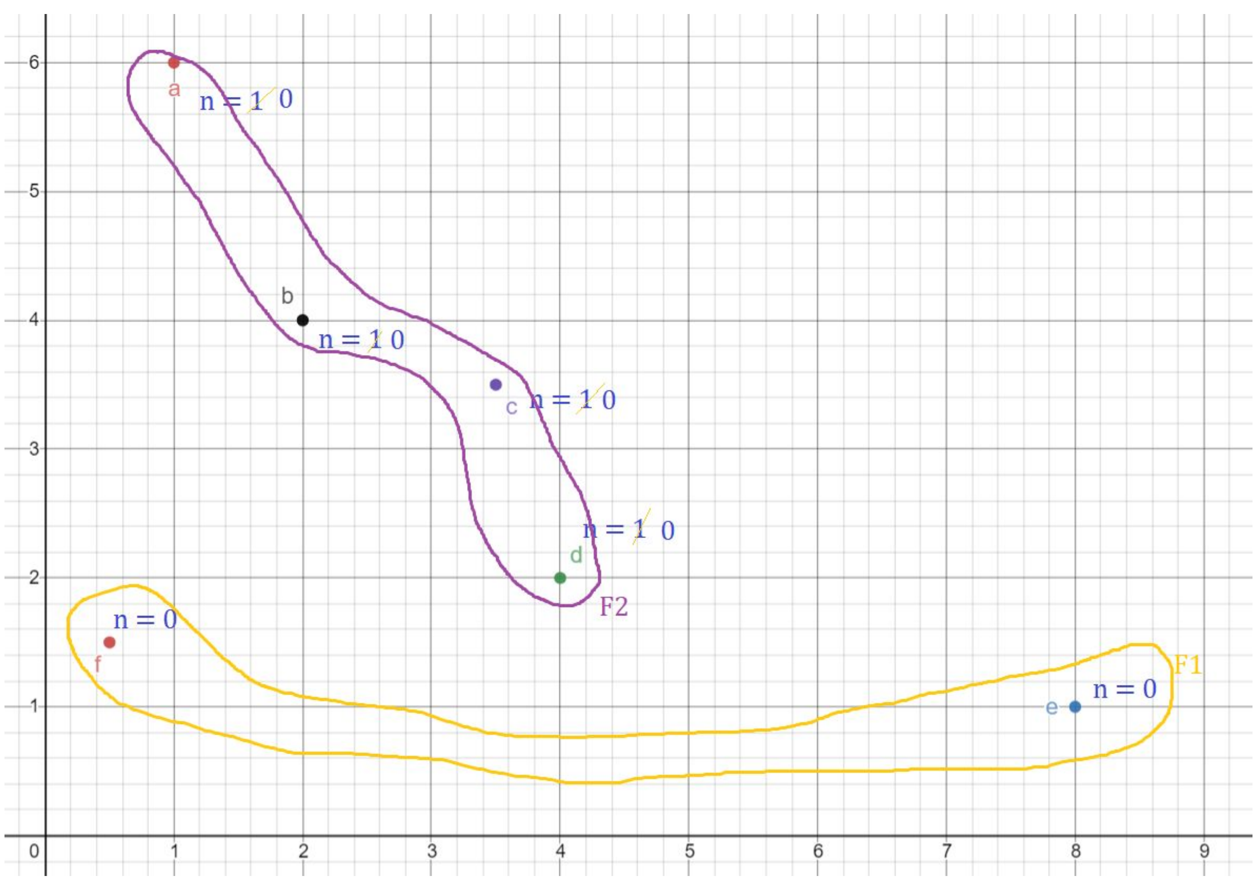
End

Source: Lecture Slides

Submitted by: Ranji Raj

- Given the following set of solutions in a multi-objective optimization problem, where both objectives should **be minimized**. If NSGA-II is used with a population size of 5, which of the solutions will survive to the next generation?





After applying the fast non-dominated sorting we obtain the front F_1 which contains solutions {f, e} this survives to the next generation.

In the second round to find the next best front, we end up having F_2 with 4 solutions F_2 = {a, b, c, d}

Since we are given the population size as 5 we apply the crowding distance operator to front F_2. We then calculate the euclidean distance to all other points in it
d(c,a) = 3.53; *d*(c,b) = 1.58; *d*(c,d) = 1.58 we observe distance of c to a is the highest and to the points b and d it is equidistant.

So we select solutions {c, b, d} as the one which moves to the next generation.

Final population P(t+1) = {b, c, d, e, f}