Computational Intelligence

Master in Artificial Intelligence

2021-22

Multi-objective optimization

Nondominated Sorting Genetic Algorithm II (NSGA-II)

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan

IEEE Trans. on Evolutionary Computation, Vol.6, No.2, 2002

Soft Computing Research Group

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Multi-objective optimization

Problem

Let X be a search space of feasible solutions, also called a *feasible set*. Let $f : X \longrightarrow \mathbb{R}^M$, $f(x) = (f_1(x), \ldots, f_M(x))$, $M \geq 2$, be the *multi-objective function* to optimize. We seek $\max_{x \in X}(f_1(x), \ldots, f_M(x))$.

Domination relation and Pareto optimality

A feasible solution x_1 is said to *dominate* another solution x_2, denoted $x_1 \succ x_2$, iff

1. $f_i(x_1) \geq f_i(x_2)$, $\forall i : 1 \leq i \leq M$, and
2. $f_j(x_1) > f_j(x_2)$ for at least one $j \in \{1, \ldots, M\}$.

A solution $x^* \in X$ is *Pareto optimal* if $\nexists x' \in X : x' \succ x^*$.
Intuitively, a solution is called nondominated or Pareto optimal if none of the objective functions can be improved in value without degrading some of the other objective values.
All Pareto optimal solutions are considered equally good.

Pareto fronts and nondominated sort algorithm

Pareto fronts

- The first Pareto front $\mathcal{F}_1$ of a set $P \subseteq X$ is the subset of all the nondominated solutions in P.
- For $i > 1$, the i-th Pareto front $\mathcal{F}_i$ is the subset of all the nondominated solutions in $P - \bigcup_{j<i} \mathcal{F}_j$.
- The naive method to compute sequentially the Pareto fronts of a set P following the above definition has a computational cost $O(MN^3)$, where $N = |P|$.
- The *fast nondominated sort algorithm* [Deb *et al.*, 2002] is a clever method to compute all the Pareto fronts of a set P with computational cost $O(MN^2)$.

Fast-nondominated-sort(P) - First part

$\mathcal{F}_1 = \emptyset$

for p in P:

  $S_p = \emptyset$ # solutions dominated by p

  $n_p = \emptyset$ # domination counter of p

  for q in P:

    if $p \succ q$: # p dominates q

      $S_p = S_p \cup \{q\}$

    elif $q \succ p$: # q dominates p

      $n_p = n_p + 1$

  if $n_p == 0$:

    $p_{rank} = 1$ # p belongs to the first front

    $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$

Fast-nondominated-sort(P) - Second part

```
i = 1
while F_i ≠ ∅:
    Q = ∅          # members of the next front
    for p in F_i:
        for q in S_p:
            n_q = n_q − 1
            if n_q == 0:    # q belongs to the next front
                q_rank = i + 1
                Q = Q ∪ {q}
    i = i + 1
    F_i = Q
```

Crowding distance and crowded-comparison operator

Crowding distance

- Within a Pareto front $\mathcal{F}_i$, solutions with a less crowded surrounding should be given priority in the selection process, to guide the search toward a uniformly spread-out Pareto-optimal front.
- The *crowding distance* of p in $\mathcal{F}_i$ is the sum of the (normalized) lengths of the edges of a hyper-rectangle of M-dim. formed by the two nearest neighbors of p in $\mathcal{F}_i$ along each of the functions f_m.
- Hence, a greater crowding distance of a solution p means a lesser crowded region around p in $\mathcal{F}_i$.

Crowded-comparison operator

Given two individuals p, q (feasible solutions) of a set $P \subseteq X$, the *crowded-comparison operator* $\succ_n$ is a partial order defined as:

$p \succ_n q$ iff

$p_{rank} < q_{rank}$ (i.e. $p \succ q$) or

$(p_{rank} == q_{rank}$ and $p_{distance} > q_{distance})$

Crowding-distance-assignment(S)

$l = |S|$ # number of solutions in set S

for i in range(1,$l+1$): # $S[i]$ is an individual, $1 \le i \le l$
    $S[i]_{distance} = 0$

for m in range(1,$M+1$): # f_m is an objective function, $1 \le m \le M$
    S =sort(S,m) # sort using m-th objective function
    $S[1]_{distance} = S[l]_{distance} = \infty$ # extreme values must be kept

    for i in range(2,l): # for every non-extreme value, $2 \le i \le l-1$
        if $S[i]_{distance} \ne \infty$:
            $S[i]_{distance} = S[i]_{distance} +$ # sum the normalized edge for m
                $(S[i+1].m - S[i-1].m)\ /\ (f_m^{max} - f_m^{min})$

NSGA-II algorithm loop body

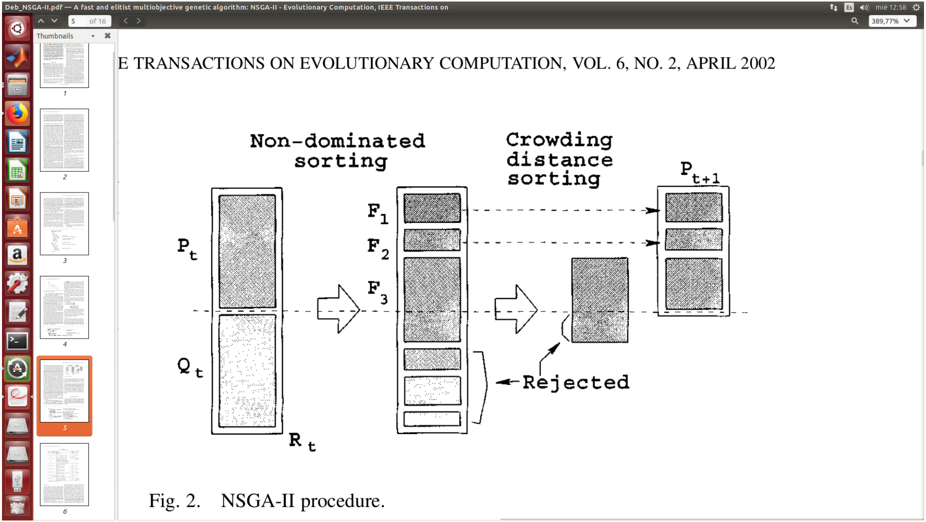
Start with current population P_t at generation t

Q_t = make-offspring(P_t) # by tournament selection, recombination
and mutation ($\lambda = \mu$)
$R_t = P_t \cup Q_t$ # mixed population to perform ($\mu + \lambda$) replacement
$\mathcal{F}$ = fast-nondominated-sort(R_t) # $\mathcal{F} = (\mathcal{F}_1, \ldots)$ set of Pareto fronts
$P_{t+1} = \emptyset$
$i = 1$

while $|P_{t+1}| + |\mathcal{F}_i| < \mu$:
    crowding-distance-assignment($\mathcal{F}_i$) # for later $\succ_n$ comparison
    $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$
    $i = i + 1$

crowding-distance-assignment($\mathcal{F}_i$)
$\mathcal{F}_i$ = sort($\mathcal{F}_i$,$\succ_n$) # sort in decreasing order using crowding distance
$P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : \mu - |P_{t+1}|]$
$t = t + 1$

NSGA-II algorithm loop body (graphically)



taken from K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". *IEEE Trans. on Evolutionary Computation*, Vol.6, No.2, 2002.