```
procedure SGA(f : 𝕏 ↦ ℝ⁺, ps, cr)                              ▷ for maximization!
    x_B ← ∅;  y_B ← −∞;                                          ▷ best-so-far solution
    for j ∈ 1 . . . ps do                                        ▷ random initial population
        randomly sample S_0[j].x from 𝕏;  S_0[j].y ← f(S_0[j].x);
        if S_0[j].y > y_B then x_B ← S_0[j].x;  y_B ← S_0[j].y;
    for i ∈ 0 . . . ∞ do                                         ▷ iterate "generations"
        for j ∈ 1 . . . ps do          ▷ new pop. via mutation and crossover
            if ℜ_0^1 < cr then N_i[j].x ← binary(S_i[⌊ℜ_1^ps⌋].x, S_i[⌊ℜ_1^ps⌋].x);
            else N_i[j].x ← move(S_i[⌊ℜ_1^ps⌋].x);
            N_i[j].y ← f(N_i[j].x);
            if N_i[j].y > y_B then x_B ← N_i[j].x;  y_B ← N_i[j].y;
        S_{i+1} ← Roulette Wheel: select ps records from P_i = S_i ∪ N_i
                 such that, for each of the ps slots, the probability
                 of P_i[j] to be chosen is proportional to P_i[j].y.
    return x_B, y_B
```