Parallel Cellular Algorithms

Custom-function (2):

Preturn Sum (x ** 2)

Initialize - population (grid - side, dim, lower, upper)

population < random (sower, upper, (grid-size, dim))

oreturn population

Evaluate - fitness (population, fitness-function)
fitness = geros (population shape [-1])
for i in range (population shape [0])
for j in range (population shape [1])

fitness[i][j]

Custom - function

(population [i,j])

setus fitnes.

update-states (pop, fit, radius, lower, upper):

grid-size, -olen - pop shape

now-pon - pop-copy()

new-pop <- pop-copy()

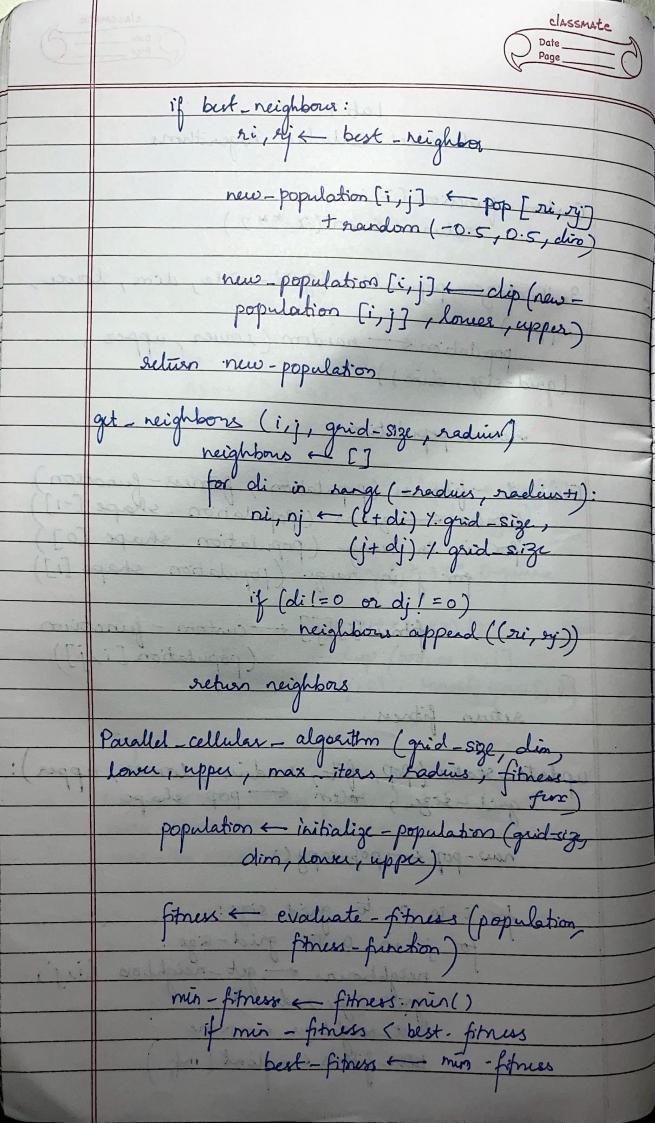
for i in 0 to grid-size!

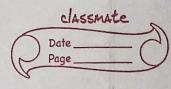
for j in 0 to grid-size!

neighbours = get-neighbors (i,j)

grid-size, radius)

best-fitnes (float (inf





best_sol + None & do! best-fit ← ∞ for iter in max-iters pop - update - states (population, fitners, radius, lower, upper) fitness (evaluate - fitness (pop , fitnes-function) min-fitness - fitness-min () if min-fitness (best-fitness: best-fitness + min-fitness best-sol ← population (np. unravel-index (fitness. min ()) Tologod (00) fitners-shape)] print (Heration, fitness) return best-sil, best-fit. Inputs: Down many world and in the aprid-size=10 loner = -5 max_ cters = 100 eadins = 1 pg - lately