INF3490 exercise answers - week 4 2015

Problem 1

In the island model, every individual has a uniform low probability of migrating to any other island, while in the diffusion model individuals can in effect only migrate one step each generation. With a 3×5 grid it would take $(3-1)+(5-1) = 6$ generation to migrate to the other corner with 4 neighbors, and $\max\{(3-1), (5-1)\} = 4$ generations with 8 neighbors.

Problem 2

a

The $(1+4)$ ES would generate four candidate solutions from the same origin before potentially changing parent instead of one. This would make for a more informed choice in which direction to move in the search space. The higher λ is, the more information is gathered about the neighborhood of the current solution. Recall that greedy search checks out all neighbors before making a move, while hill climbing on makes one. So we would expect the $(1+\lambda)$ ES to behave increasingly like a greedy search as we increase λ.

b

An adaptive search strategy will, in most cases, increase the convergence rate of the search especially in the late stages. However, it does not by itself help avoid getting stuck in local optima.

c

When strategy parameters are mutated first, the change in strategy has immediate effect on the new solution that is created. Thus the fitness of this solution also indirectly rates the strategy to some degree. If the strategy parameters are mutated after the solution parameters this link is weaker, and we would expect the strategies to adapt slower, if at all.

Problem 3

a

See file "inf3490ex4P3a.py"

b

The best solution will always survive as an offspring, so it just needs to be created at least once in the first place. If parents are drawn without replacement the probability is zero - 4 can only be one of the parents. Otherwise the probability of any one offspring having 4 as both parents is $\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$. The chance of none of the eight offspring having 4 as both parents is $\left(\frac{15}{16}\right)^8 \approx 0.597$ and so the probability of the solution surviving is $1 - 0.597 = 0.403$.

c

See file "INF3490ex4p3c.py"

Problem 4

One way would be to randomly deselect items until the solution is under budget. One might also consider a greedy approach that iteratively deselects the items of least worth until the budget is met. One could even consider this as a smaller 0-1 knapsack problem from only the selected items and optimize that using some potentially completely different algorithm.