Evolutionary Computation (6560) HW01

Alex Darwiche

September 7, 2024

Answers

Q1: Subset Problem

(a) **Representation:**

(1) Binary Vector of Length N

(2) 0 means not in Subset P

(3) 1 mean in Subset P

(b) **Fitness Function:**

(1) Distance from 21, to be minimized.

(2) Or the abolsute value of sum of all elements in P, minus 21.

(3) Fitness Formula = Min($|\sum p - 21|$)

(4)

X = {12, 17, 3, 24, 6}

12	17	3	24	6		Fitness
0	0	0	0	0		21
0	0	0	1	0		3
1	0	1	0	1		0

(c) **Mutation:**

(1) Bit-Flip Mutation with probability 1/N

(d) **Crossover:**

(1) 1-point crossover between two parents

(e) **Repair:**

(1) If the $\sum p > 21$ then flip a random 1 to 0.

(2) If the $\sum p < 21$ then flip a random 0 to 1.

(3) Computationally Heavy Option: Calculate the distance from 21 of $\sum p$, and only swap lowest value that is less than the distance from either 0 to 1 or 1 to 0, depending on if $\sum p > 21$ or < 21.

(f) **Termination:**

(1) Fitness = 0 or reach 10,000 steps.

Q2: Graph K-colording Problem

(a) **Representation:**

(1) Integer Representation Vector of Length N

(2) Each integer corresponds to a color from 1 to k

(3) 0 means Red

(4) 1 means Blue

(5) ...

(6) k means k-th color

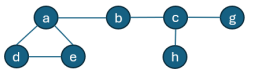
(b) **Fitness Function:**

(1) Minimize the number of conflicted edges.

(2) A conflicted edge is one that has the same color on each vertex.

(3) Fitness Formula = Min(Count(Conflicted Edges of E))

(3)



a	b	c	d	e	f	g	Fitness
1	1	1	1	1	1	1	7
1	2	1	2	3	2	3	0
1	2	3	2	3	1	2	0

(c) **Mutation:**

(1) Flip Mutation with probability 1/N, changing to one of (K-1) other color options

(d) **Crossover:**

(1) 1-point crossover between two parents

(e) **Repair:**

(1) If a conflicted edge exists, change the color of the edge with the least amount of neighbors to a random different color.

(f) **Termination:**

(1) Fitness = 0 or reach 10,000 steps.

Q3: Minimum Vertex Cover

(a) **Representation:**

(1) Binary Vector of Length N

(2) 0 means not in X

(3) 1 mean in X

(b) **Fitness Function:**

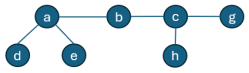
(1) Maximize (N*Good edges) + (N-Size of X).

(2) Good Edges: This is an edge that is only touched by only 1 vertex in X. We multiply it by N to ensure finding good edges is the first priority, then the second term helps to ensure we find the minimum vertex cover among feasible options.

(3) Bad Edges: This is an edge that is touched by 2 or 0 vertices in X.

(4) (N-Size of X): This incentivizes finding a smaller solution set.

(3)



a	b	c	d	e	f	g	Fitness
0	0	0	0	0	0	0	7
1	1	1	1	1	1	1	0
1	0	1	0	0	0	0	47

(c) **Mutation:**

(1) Bit-Flip Mutation with probability 1/N

(d) **Crossover:**

(1) 1-point crossover between two parents

(e) **Repair:**

(1) Swap the value of a vertex if it is part of a "bad" edge. This can be either changing from 1 to 0 or 0 to 1 depending on if the edge is bad because it is touched by no vertex in X, or if its touched by 2 vertices in X.

(f) **Termination:**

(1) Fitness = (N*M + N) or Reach 10,000 steps.

(2) (N*M + (N-1)) is the maximum possible value of fitness. Essentially all edges are good AND only 1 vertex is used in the cover. This assumes that there exists at least 1 edge.