CMSI 2130 – Classwork 7

BONUS: This is an optional, bonus classwork exercise meant to help you study for the coming exam. Completing this assignment will award you a max of +2 Fornbucks (for fully-completed submissions). The solution for this classwork will be posted at the deadline, so late submissions will not be accepted.

Instructions:
This worksheet gives you some important practice with the fundamentals of local search, hill climbing, and genetic algorithms!

- Provide answers to each of the following questions and write your responses in the blanks. If you are expected to show your work in arriving at a particular solution, space will be provided for you.
- Place the names of your group members below:

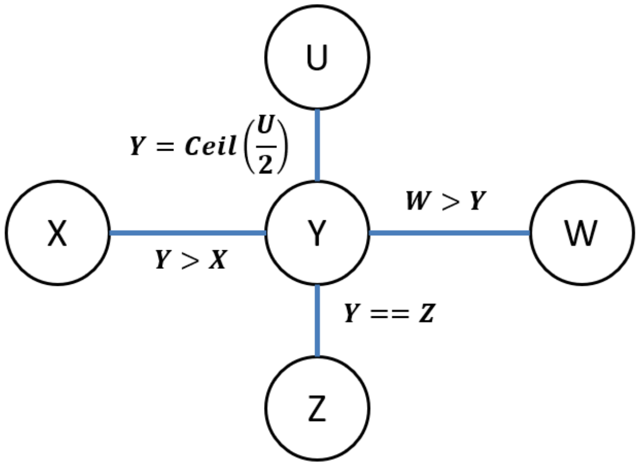
Group Members:

1. Mike Hennessy
2. Cameron Scolari
3. ______________________________

Problem 1 – Local Search

Consider a lazy variant of our numerical CSP from the previous classwork, with the constraint graph that follows (NB. the $Ceil$ operator rounds up its argument to the nearest integer, e.g., $Ceil(0) = 0, Ceil(0.5) = 1, Ceil(1) = 1, Ceil(1.5) = 2$):

- *Variables:* $V = \{U, W, X, Y, Z\}$
- *Domains:* $D_i = \{0, 1, 2\}\ \forall\ V_i \in V$
- *Constraints:* $C = \left\{W > Y,\ \ Y > X,\ \ Y == Z,\ \ Y = Ceil\left(\frac{U}{2}\right)\right\}$



1.1. Suppose now that we are solving the CSP above using Local Search and begin with the random initial state s that follows:

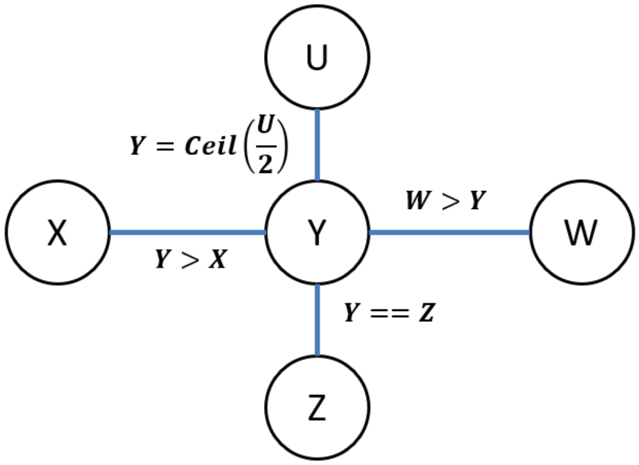
Current State s				
U	W	X	Y	Z
0	2	1	1	1

For each of the following states, and objective function $f(s) = numConstraintsSatisfied$ (i.e., count the number of constraints that are satisfied by the state), determine:

1. Whether or not the state is in the neighborhood of the initial guess s above.
2. ***If*** <u>it is in the neighborhood of s</u>, score it by the objective function and then determine if it is an Upward [UW], Sideways [SW], or Downward [DW] move compared to s (hint: the initial state satisfies $f(s) = 2$ constraints). If it is *not* in the neighborhood of s, place an "X" in the other columns.

Potential Next States s'							
U	W	X	Y	Z	In Neighborhood(s)?	Score $f(s')$ ($s' = \boldsymbol{row\ state}$)	UW/DW/SW Move?
0	2	1	1	0	Yes	1	DW
0	2	0	1	0	No	X	X
0	2	0	1	1	Yes	3	UW

Same problem! Just replicated here for your convenience (*Domains:* $D_i = \{0, 1, 2\}\ \forall\ V_i \in V$):



1.2. If we are using the min-conflict heuristic during Local Search in an attempt to reach a solution, suppose we find ourselves in each of the following states represented in each row of the table. For each state/row and randomly selected variable to reassign (from amongst the possible candidates), determine which value should be assigned by *min-conflict.*

U	W	X	Y	Z	Variable to Reassign	Value to Assign (by min-conflict)
0	0	0	1	0	**W**	2
2	2	0	0	2	**Y**	1

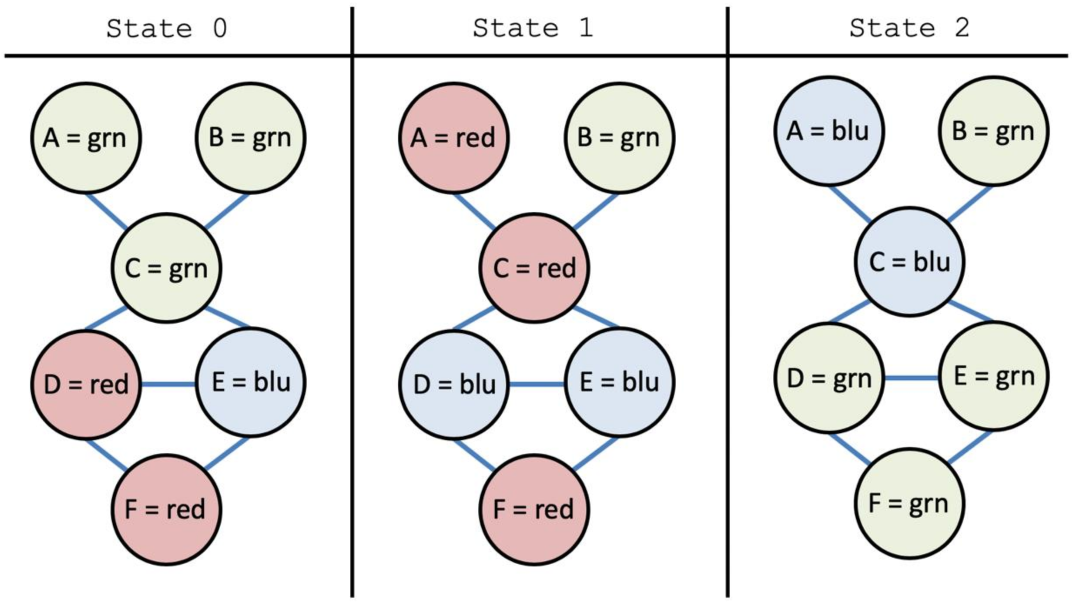
1.3. If we add simulated annealing to our local search with current Temperature $T = 2$, difference in objective function f values $\Delta E = f(s') - f(s)$, and probability of taking a **downward** move $P = e^{\Delta E/T}$, determine the likelihood of moving to each of the table's next states s' if currently in the state s (Hint: these are the same states from problem 1.1., and note that simulated annealing will *never* move to a state that is not its previous state's neighbor).

Current State s				
U	W	X	Y	Z
0	2	1	1	1

Randomly Selected Next States s'					
U	**W**	**X**	**Y**	**Z**	**Likelihood of moving from $s \to s'$**
0	2	1	1	0	.61
0	2	0	1	0	0
0	2	0	1	1	1

Problem 2 – Artificial Evolution

Whew, that's enough numbers! Let's get back to good, old-fashioned map coloring… Suppose we have the following population size $P = 3$ states (ignoring for now that population sizes are usually an even number, but meh) created during a genetic algorithm solving the map coloring problem, and the same objective function $f(s)$ counting the number of constraints satisfied in each state s.



2.1. For each of the states above, determine the number of satisfied constraints (Hint: count the number of edges that have their constraint satisfied).

$State\ 0 =$ 4 $\qquad State\ 1 =$ 5 $\qquad State\ 2 =$ 3

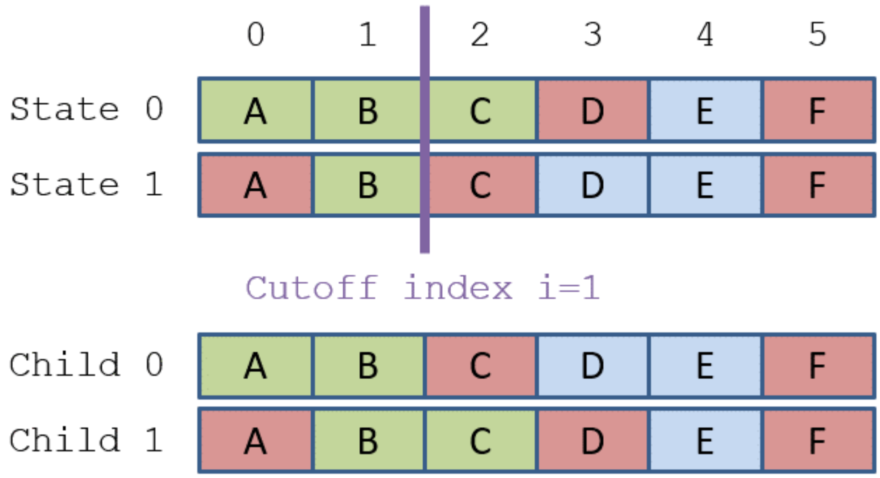
2.2. For each of the states above, determine its likelihood of *selection / mating* for producing states in the next generation (to get a likelihood, remember to divide by the total number of satisfied constraints from 2.1. -- OK if these are left as fractions):

$State\ 0 = \frac{1}{3}$ $\qquad State\ 1 = \frac{5}{12}$ $\qquad State\ 2 = \frac{1}{4}$

Now time for the fun stuff – mating! In order to produce the next generation of states in a genetic algorithm, remember that we select two parents P_0, P_1 who then produce pairs of offspring that are equivalent to their recombined "genes," where the genes are a sequential-representation of the problem state. In the map coloring problem depicted above, suppose we turn these states into an array representation with each variable indexed in ascending alphabetic order, e.g., with State 0 represented as:

Variable	A	B	C	D	E	F
Index	0	1	2	3	4	5
Value	grn	grn	grn	red	blu	red

A recombination method `State recombo(State p0, State p1, int cutoff)` returns a single child State from two parent States `p0, p1` given a *cutoff index,* with the following definition: The assignment in the returned child State comes from indexes $[0, \text{cutoff}]$ (inclusive) from parent P_0, then from indexes $[\text{cutoff} + 1, n - 1]$ from parent P_1 (for n variables). For example, see the results of $recombo(0,1,1)$ and $recombo(1,0,1)$ in Child 0 and Child 1 below, respectively:



2.3. With these recombination semantics, what would the resulting child State be from calling $recombo(2,1,3)$ (i.e., State 2 as parent `p0`, State 1 as parent `p1`, and cutoff index 3)?

Variable	A	B	C	D	E	F
Index	0	1	2	3	4	5
Value	B	G	B	G	B	R

2.4. Given the child state you found in 2.3., which variable, if reassigned by the *mutation* step of genetic solvers, would yield a solution? What value would you reassign to it? An answer like $A = red$ is all you need here.

C = Red