1 Set-based search

1.1 Model

We define our model $A : (S, T)$ for use with a set-based search paradigm. Given the set of *TA*'s $t_i \in TA$ and *labs* $l_j \in LABS$, a fact is defined as a set of unique 2-tuples where the first element is a TA and the second element is a lab. We further constrain the definition of a fact such that no hard-constraints are violated. That is,

1. No TA has more than *MAX_LABS* labs: for any fact f, $\forall t_i |(t_i, l) \in f| \leq MAX\_LABS$.

2. If a TA has a lab, that TA has at least *MIN_LABS* labs: for any fact f, $\forall t_i |(t_i, l) \in f| = 0 \vee |(t_i, l) \in f| \geq MIN\_LABS$

3. No lab has more than one TA and all labs have a TA: for any fact f, $\forall l_i |(t, l_i) \in f| = 1$

4. No TA has a time conflict: for any fact f, $\forall (t_i, l_i) \in f, time\,(l_i) \notin \{time\,(c_k) \wedge c_k \in courses\,(t_i)\} \wedge time(l_i) \notin \{time(l_j) \wedge (t_i, l_j) \in fx\}$

The type F is a set of facts and state S is a superset of facts ($S = 2^F$). A transition $T : S \times S = \{(s, s')|\exists A \rightarrow B \in ExtA \subseteq s \vee s' = (s - A) \cup B\}$

Define $Ext : \{A \rightarrow B | A, B \subseteq F\}$ where $B = A \cup C$ where C is generated by specifying an allowable time and calling *Generate* then *Combine* until that time is exceeded or until $|C| = |A|$ so that $|B| = 2|A|$. The operations used are defined as:

1. *Generate* - Do a random walk through the defined in **??**. The random walk does not compare leafs, it only tries paths at random until a solution is found. If no solution is found within the allowed time, this operation fails.

2. *Combine* - First, map each element in a fact f from a 2-tuple to a 3-tuple $(t, l) \rightarrow (t, l, b = time(l))$. Then, for each b such that $\exists (t', l', b) \in f$, match each instance of t' and l' once at random. This does not change the times that any TA teaches, it only changes which labs a TA is teaching. If the result violates any hard constraints, this operation fails. For the implementation, we will consider lazy evaluation of hard constraints.

1.2 Process

We define our process $P : (A, Env, K)$ for the set based search. The model A has already been defined. It is assumed that the environment Env is unchanging so $K : S \times Env \rightarrow S$ is just $K : S \rightarrow S$. The control K is ??? from rubric ???.

f_{wert} is defined as $-\sum_i penalty_i(f)$ where $penalty_i$ is defined in Table **??** as a function of a fact which is either the penalty value from the table or zero if the penalty does not apply.

f_{select} is defined as a tournament. The number of facts is "culled" down to a specified number N. This is done by repeating the following operation $|A| - N$ times. At random, two facts in $f_1, f_2 \in F$ are selected. A random number $0 < r < 1$ is generated. If $r < \frac{f_{1wert}}{f_{1wert}+f_{2wert}}$, f_1 is removed from A, otherwise f_2 is removed from A. For the implementation, f_{wert} may not need to be calculated for all $f \in F$.