1 Program flow for TPR

Testing will proceed in two phases: in the first, the crowd can only 'like' robots; in the second, they teach the robots language.

1.1 Phase One: Likability

In this phase there is a single population of robots. Robots are optimized against two objectives: age and likability. Likability is simply the sum of all 'likes' that the robot has received from the crowd.

In the case of a tie, the robot with the higher ID (i.e., the older one) dominates the one with lower ID.

During each generation, each robot in the population is shown to the crowd once (including those already shown to the crowd). Once all robots have been shown, the pareto front is calculated and the dominated solutions are discarded. The empty slots are filled with mutants created from the survivors, and the final empty slot is filled with a new random robot with an age of zero.

Mutants inherit the age of their parent and its number of likes. This ensures that if a mutant gets one more like than its parent (which was also shown to the crowd that generation), it dominates its parent.

In the case of a tie, the robot with the higher ID (i.e., the older one) dominates the one with lower ID.

We can test phase 1 in the following manner. A bot stands in for human users. At each generation g, the bot awards a like to each robot that generates more movement that the mean amount of movement among all robots during generation $g-1$. Should see robots with more movement evolving over time.

1.2 Phase Two: Learnability

In the second phase, the population is divided into two subpopulations: those that have received one or more likes or positive reinforcement signals from the crowd, and those that have not. These two populations will be referred to as the seen and unseen populations, respectively.

The first population proceeds exactly as the population in phase one, but the second objective is now simply all of the likes it (and its ancestors) received, plus all of the positive reinforcement signals (and its ancestors)

received. If a mutant in this first population does not receive a like or a positive reinforcement, it is dominated by its parent and dies.

The second population is formed by creating populations of k mutants from each of the s robots from the first population. This thus leads to s populations. Each of these populations are optimized using two objectives: obeisance (how well they obey all commands issued by the crowd so far) and predictive power (how well they can predict crowd response to all seen robots so far).