Genetic Algorithm for Graph Layouts

Ryan Tanaka

Idea

- Given a graph (directed or undirected), try to plot it on a 2D plane such that the number of edge crossings are minimized
- **Search Space**: vector of coordinates
  - **Example**:
    - say we have a graph that looks like this o^A-----o^B-----o^C
    - and we have a 100x100 pixel canvas to draw this graph
    - then there are $(100^2)^3$ or a trillion different ways to plot this graph (points could end up on top of each other)
- **Objective Function:**

$$f(x) = \binom{|E|}{2} - \sum_{p \epsilon Pairs} \begin{Bmatrix} 1 & \text{if p has intersection} \\ 0 & \text{if p has no intersection} \end{Bmatrix}$$

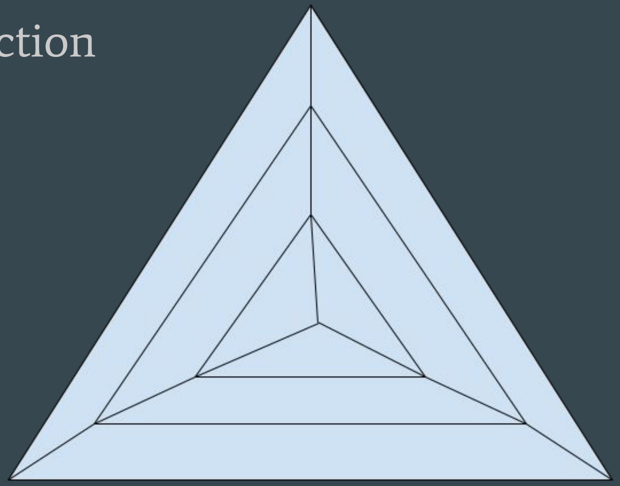
Variation Operators

- **Representation: <$(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ >**
- Used Identical methods presented in lecture 1 slides
  - **Recombination**: single point crossover
  - **Mutation**: randomly select (x,y) coordinate for random point if individual is to be mutated
  - **Selection**: fitness proportional selection

```python
def selection(population):

    population_sel = [None] * len(population)

    fitness_sum = sum([individual.fitness for individual in population])

    for index, individual in enumerate(population):
        random_stopping_point = random.randint(0, fitness_sum)
        partial_sum = 0

        for individual_x in population:
            partial_sum += individual_x.fitness

            if partial_sum >= random_stopping_point:
                population_sel[index] = deepcopy(individual_x)
                break

    return population_sel
```

Trial Run

- Population: 300
- Generations: 1500
- Target Fitness: at most 1 edge pair with intersection
- Graph: 10 nodes, 18 edges
- Canvas Size: 800x800



Results

Genetic Algorithm Performance | Population: 300 | Recombination Rate 0.5 | Mutation Rate 0.05
160
140
120
100
0
250
500
750
1000
1250
generation
max fitness
min fitness
average fitness
target fitness

Room For Improvement

- evaluate fitness based on different criteria, for example:
    - spacing between nodes or edges?
    - for each node, how many of its own edges intersect with other edges (if none then I shouldn't move that node)?
- scale fitness better, ie, 99% is much better than 96%
- combine high fitness individuals
- If a node is in an "already near optimal position", maybe mutation should have a smaller effect
    - currently a good node may get pushed all the way across the canvas, and if it has a high degree, there is a higher chance that now its edges will intersect with other edges

Example Graph Layouts