

# Roulette Wheel Selection and Tournament Selection

November 28, 2024

```
[1]: import numpy as np

population = ["01101", "11000", "10110", "00111", "10101", "00010"]
fitness = [5, 2, 1, 10, 3, 100]
total_fitness = sum(fitness)
best_string = population[np.argmax(fitness)]
best_fitness = max(fitness)
population_size = len(population)

[2]: def roulette_wheel_selection(fitness, population_size):
    probabilities = [f / total_fitness for f in fitness]
    best_prob = max(probabilities)
    expected_copies = best_prob * population_size
    return expected_copies

[3]: def tournament_selection(fitness, population_size, k=2):
    best_prob = (max(fitness) / total_fitness) ** k
    expected_copies = best_prob * population_size
    return expected_copies

[4]: def generations_to_dominate(selection_prob, population_size):
    return int(np.ceil(np.log(population_size) / np.log(1 / (1 -
↪selection_prob)))))

[5]: # 1. Roulette Wheel Selection
roulette_expected_copies = roulette_wheel_selection(fitness, population_size)
roulette_selection_prob = max(fitness) / total_fitness
roulette_generations = generations_to_dominate(roulette_selection_prob,
↪population_size)

[6]: # 2. Tournament Selection (with k = 2)
tournament_expected_copies = tournament_selection(fitness, population_size, k=2)
tournament_selection_prob = (max(fitness) / total_fitness) ** 2
tournament_generations = generations_to_dominate(tournament_selection_prob,
↪population_size)

[7]: print("Roulette Wheel Selection:")
print(f"Expected copies of best string: {roulette_expected_copies:.2f}")
```

```
print(f"Generations to dominate: {roulette_generations}")
```

Roulette Wheel Selection:  
Expected copies of best string: 4.96  
Generations to dominate: 2

```
[8]: print("\nTournament Selection:")  
      print(f"Expected copies of best string: {tournament_expected_copies:.2f}")  
      print(f"Generations to dominate: {tournament_generations}")
```

Tournament Selection:  
Expected copies of best string: 4.10  
Generations to dominate: 2