POLITÉCNICO
DE LEIRIA

Worksheet – Genetic Algorithms

Artificial Intelligence Applied to Games

Department of Computer Science - Games and Multimedia

Introduction

Genetic algorithms are parallel and stochastic search algorithms, inspired by Darwin's theory of natural selection and by molecular biology. The search is said to be parallel because at any given moment a set of n possible solutions is maintained, not just one, as it happens, for example, with the Hill-Climbing algorithm. In genetic algorithms terminology, this set of possible solutions is designated by **population**. The **general structure of a genetic algorithm** is shown below.

Listing 1 Genetic Algorithm Structure

```
t = 0
create_initial_population(P(t))
evaluate(P(t))

while stopping_condition_not_met:
    P_selected(t) = select(P(t))
    P_offspring(t) = apply_genetic_operators(P_selected(t))
    P(t + 1) = create_next_population(P(t), P_offspring(t))
    evaluate(P(t + 1))
    t = t + 1

return best_individual_found
```

In general, the initial population is generated randomly, however, knowledge of the domain may be used.

Evaluation and Fitness

Each individual of the population is evaluated using an evaluation function also known as **Fitness function**. The value returned by this function should reflect the quality of the individual as a possible solution to the problem.

Generations and Selection

The search progresses over a number of iterations, these iterations are designated as **Generations**. In each generation, the first step is to select n individuals. The selection method should follow the following principles:

1. The best individuals are more likely to be selected
2. The selection is performed with replacement

In this exercise sheet we will use two methods of selection: the **roulette wheel** and the **tournament**. By applying these selection methods, it results in an intermediate population *P_selected(t)*, where *t* represents the generation.

Genetic Operators

The next step is to apply genetic operators to the individuals of *P_selected(t).* These operators are applied in order to create different individuals, eventually better than the existing individuals, allowing other regions of the search space to be explored. The most common operators are the **crossover operator** and the **mutation operator**.

- The **crossover operator** consists in creating new individuals by "mixing" genetic material from two existing individuals.
- The **mutation operator** consists in modifying a single individual.

Typically, the mutation operator is applied after the crossover operator. The application of both operators is stochastic. From the application of the genetic operators, results the population *P_offspring(t).*

Next Generation

The population $P(t+1)$ can be constructed from the population $P(t)$ and from the population *P_offspring(t).* However, the most common method, denominated the **generation strategy**, is to make *P(t+1) = P_offspring(t).* The last step in each generation is to evaluate the new population.

At the end of the evolutionary process, the algorithm returns the best individual found, hoping that it is a good solution to the problem.

Exercise 1

The project provided corresponds to the training of a neural network to control a very simple racing car.

1. In this assignment you must implement a genetic algorithm that can be used to find a solution to this problem. To do this, you must complete the project that was provided with the exercise sheet, by implementing the following functions/events:

   1. **Tick** in the Blueprint CarController

   2. **CheckpointPassed** in the Blueprint CarController

   3. **ComputeFitness()** in the Blueprint CarController

2. Additionally, add new types of mutation and crossover (Blueprint "Individual").

Exercise 2

Perform tests that allow you to study the following aspects:

- The performance of the algorithm when the **roulette selection** method is used
- The performance of the algorithm when the **tournament selection** method is used
- The performance of the algorithm when the **single-point crossover** is used
- The performance of the algorithm when the **two-point crossover** is used
- The performance of the algorithm when the **uniform crossover** is used
- The effect of variation of the **tournament size**
- The effect of variation of the **population size**
- The effect of variation of the **probabilities of the genetic operators** used

Statistical Significance

In order for the obtained results to be statistically significant, perform at least **30 independent experiments** with each algorithm configuration.

Crossover Operators

The most common crossover operators are the single-point crossover, the two-point crossover and the uniform crossover. The following image illustrates the behavior of these operators.

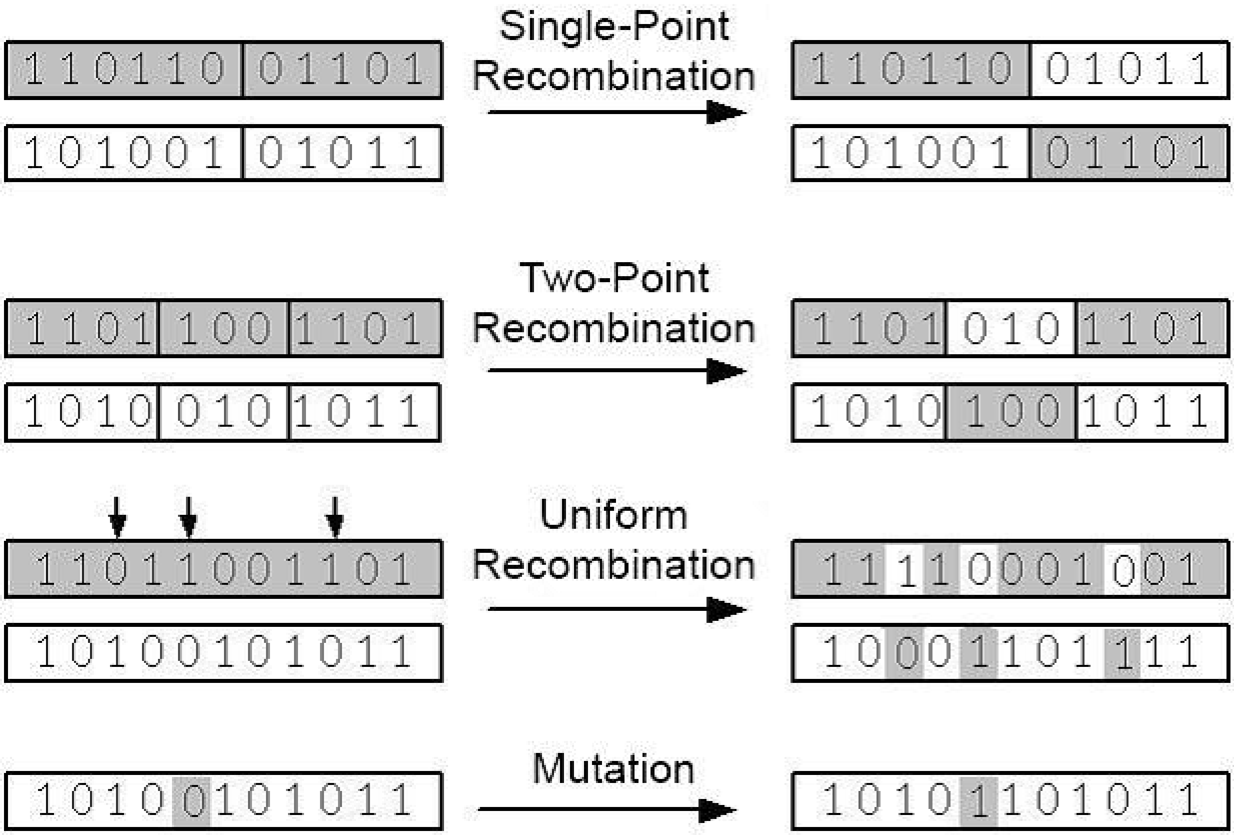

Figure 1: Crossover operators illustration