INTRODUCTION TO EVOLUTIONARY COMPUTATION

- Evolution is an optimization process where the aim is to improve the ability of an organism (or system) to survive in dynamically changing and competitive environments.
- The Darwinian theory of evolution can be summarized as: In a world with limited resources and stable populations, each individual competes with others for survival. Those individuals with the "best" characteristics are more likely to survive and to reproduce, and those characteristics will be passed on to their offspring.
- *Evolutionary computation* (EC) refers to computer-based problem solving systems that use computational models of evolutionary processes, such as natural selection, survival of the fittest and reproduction, as the fundamental components of such computational systems.

Evolutionary Algorithm (EA)

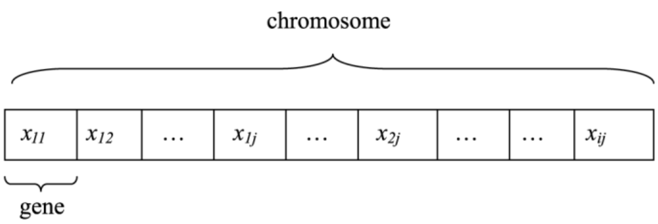
Evolution via natural selection of a randomly chosen population of individuals can be thought of as a *search* through the space of possible *chromosome* values.

Algorithm 8.1 Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
Create and initialize an n_x-dimensional population, $\mathcal{C}(0)$, to consist of n_s individuals;
while *stopping condition(s) not true* **do**
    Evaluate the fitness, $f(\mathbf{x}_i(t))$, of each individual, $\mathbf{x}_i(t)$;
    Perform reproduction to create offspring;
    Select the new population, $\mathcal{C}(t+1)$;
    Advance to the new generation, i.e. $t = t + 1$;
end

Representation – The Chromosome

- In the context of EC, each individual represents a candidate solution to an optimization problem.
- The characteristics of an individual are represented by a *chromosome*.
- These characteristics refer to the variables of the optimization problem.
- Each variable that needs to be optimized is referred to as a *gene*, the smallest unit of information.
- An assignment of a value from the allowed domain of the corresponding variable is referred to as an *allele*.



- Chromosomes are implemented as binary vectors of fixed length of n_x variables.

- If variables have *binary* values, the length of each chromosome is *nx*bits.
- In the case of *nominal*-valued variables, each nominal value can be encoded as an *nd*-dimensional bit vector.
- In case of *continuous*-valued variables, *mapping functions* are needed to convert the space $\{0, 1\}^{nb}$to the space R^{nx}. For such mapping, each continuous-valued variable is mapped to an *nd*-dimensional bit vector.
  - The domain of the continuous space needs to be restricted to a finite range, [**x***min,* **x***max*]. A standard binary encoding scheme can be used to transform the individual $\mathbf{x} = (x_1, \ldots, x_j, \ldots, x_{nx})$, with $x_j \in R$ to the binary-valued individual, $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_{nx})$.

Initial Population

- Evolutionary algorithms are stochastic, population-based search algorithms.
- Each EA therefore maintains a *population* of candidate solutions.
- The standard way of generating an initial population is to assign a *random* value from the allowed domain to each of the genes of each chromosome.
- Large numbers of individuals increase diversity, thereby improving the exploration abilities of the population.
  - However, the more the individuals, the higher the computational complexity per generation.
- A small population, on the other hand will represent a small part of the search space.
  - While the time complexity per generation is low, the EA may need more generations to converge than for a large population.

Fitness Function

- The *fitness function*, *f*, maps a chromosome representation into a scalar value:

$$f : \Gamma^{n_x} \rightarrow \mathbb{R}$$

- The fitness function represents the **objective function**, Ψ, which describes the optimization problem.

Unconstrained Optimization

The general unconstrained optimization problem is defined as

$$\begin{array}{lll} minimize & f(\mathbf{x}), & \mathbf{x} = (x_1, x_2, \ldots, x_{n_x}) \\ subject\ to & & x_j \in dom(x_j) \end{array}$$

where $\mathbf{x} \in \mathcal{F} = \mathcal{S}$, *and* $dom(x_j)$ *is the domain of variable* x_j.

For example:

Colville:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 \\ &\quad + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) \\ &\quad + 19.8(x_2 - 1)(x_4 - 1) \end{aligned}$$

with $x_1, x_2, x_3, x_4 \in [-10, 10]$ and $f^*(x_1, x_2, x_3, x_4) = 0.0$.

Multi-objective optimization problems (MOP)

- Many real-world problems require the simultaneous optimization of a number of objective functions.

- Some of these objectives may be in conflict with one another.
- For example, consider finding optimal routes in data communications networks, where the objectives may include minimizing routing cost, to minimize route length, to minimize congestion, and to maximize utilization of physical infrastructure.
- MOP can be solved by using a weighted aggregation approach, where the fitness function is a weighted sum of all the sub-objectives.

$$\text{minimize} \quad \sum_{k=1}^{n_k} \omega_k f_k(\mathbf{x})$$

Where,

$$\sum_{k=1}^{n_k} \omega_k = 1.$$

Selection

The main objective of selection operators is to emphasize better solutions. This is achieved in two of the main steps of an EA:

1- **Selection of the new population**: A new population of candidate solutions is selected at the end of each generation to serve as the population of the next generation. The new population can be selected from only the offspring, or from both the parents and the offspring.

2- **Reproduction**: Offspring are created through the application of crossover and/or mutation operators.

- In terms of crossover, "superior" individuals should be selected.
- In the case of mutation, selection mechanisms should focus on "weak" individuals.

There are several selection operators:

1- **Random Selection**: Random selection is the simplest selection operator, where each individual has the same probability of $\frac{1}{ns}$ (where *ns* is the population size) to be selected.

2- **Proportional Selection**: Proportional selection biases selection towards the most fit individuals. A *probability* distribution proportional to the fitness is created, and individuals are selected by sampling the distribution,

$$\varphi_s(\mathbf{x}_i(t)) = \frac{f_\Upsilon(\mathbf{x}_i(t))}{\sum_{l=1}^{n_s} f_\Upsilon(\mathbf{x}_l(t))}$$

where *ns* is the total number of individuals in the population, and $\varphi_s(x_i)$ is the probability that $\mathbf{x}_i$ will be selected. $f\gamma(x_i)$ is the scaled fitness of $\mathbf{x}_i$.

- For *minimization* problems:

$$f\gamma(x_i(t)) = \frac{1}{1 + f(x_i(t)) - f_{\min}(t)},$$

Where $f_{\min}(t)$ is the minimum fitness up to time step *t*. Here $f\gamma(x_i(t)) \in (0,1]$

- For *maximization* problems:

$$f\gamma(x_i(t)) = \frac{1}{1 + f_{\max}(t) - f(x_i(t))}$$

Where $f_{\max}(t)$ is the maximum fitness up to time step *t*.

Example: Suppose you have four individuals of three inputs as follows:

x1	x2	x3	fitness	scaled fitness	probabilty
1	0	2	3	0.5	0.176470609
2	1	1	4	0.333333333	0.117647073
1	4	5	2	1	**0.352941218**
2	3	3	4	0.333333333	0.117647073
			Min	**sum**	
			2	2.166666667	

The fitness function is $f(x1, x2, x3) = x1^2 - x2 + 3$, for minimization problem, the third individual will be selected.

H.w) Select the individual for the maximization problem.

Roulette wheel sampling

Assuming maximization, and normalized fitness values, roulette wheel selection is summarized in Algorithm 8.2.

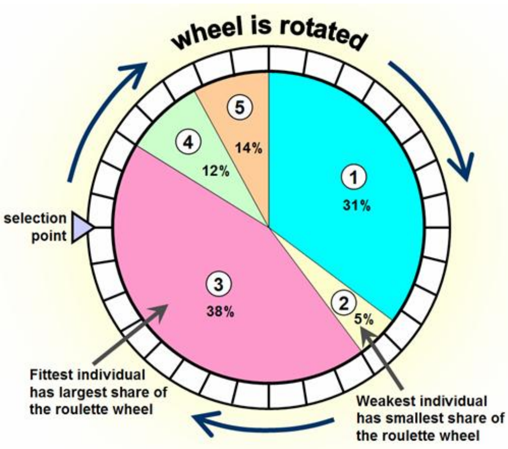
Algorithm 8.2 Roulette Wheel Selection

Let $i = 1$, where i denotes the chromosome index;
Calculate $\varphi_s(\mathbf{x}_i)$ using equation (8.9);
$sum = \varphi_s(\mathbf{x}_i)$;
Choose $r \sim U(0, 1)$;
while $sum < r$ **do**
    $i = i + 1$, i.e. advance to the next chromosome;
    $sum = sum + \varphi_s(\mathbf{x}_i)$;
end
Return $\mathbf{x}_i$ as the selected individual;

Example

No.	Chromosome	$Value_{10}$	X	Fitness f(x)	% of Total
1	0001101011	107	1.05	6.82	31
2	1111011000	984	9.62	1.11	5
3	0100000101	261	2.55	8.48	38
4	1110100000	928	9.07	2.57	12
5	1110001011	907	8.87	3.08	14
Totals				22.05	100

Example population of 5 for: $f(x) = -\frac{1}{4}x^2 + 2x + 5$



3- Tournament Selection

Tournament selection selects a group of n_{ts}individuals randomly from the population, where $n_{ts} < n_s$($n_s$is the total number of individuals in the population). The performance of the selected n_{ts}individuals is compared and the *best* individual *from this group* is selected and returned by the operator.

- For crossover with two parents, tournament selection is done *twice*, once for the selection of each parent.

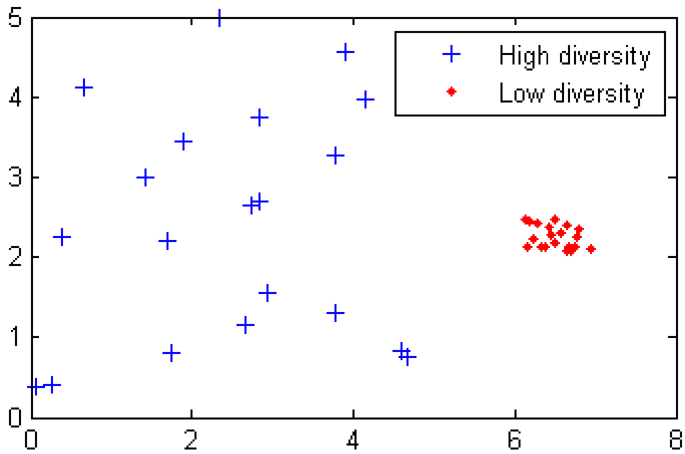
- If n_{ts}= n_s, the best individual will always be selected, resulting in a very high selective pressure.
- On the other hand, if n_{ts}= 1, random selection is obtained.

4- Elitism

Elitism refers to the process of ensuring that the best individuals of the current population survive to the next generation. The best individuals are copied to the new population without being mutated.

Reproduction Operators

- Reproduction is the process of producing offspring from selected parents by applying *crossover*and/or*mutation* operators.
- *Crossover* is the process of creating one or more new individuals through the combination of genetic material randomly selected from two or more parents.
- *Mutation* is the process of randomly changing the values of genes in a chromosome.
    - The main objective of mutation is to introduce new genetic material into the population, thereby increasing genetic *diversity*.
- Diversity refers to the average distance between individuals in a population.
- A population has high diversity if the average distance is large; otherwise ithas low diversity.



- Mutation should be applied with care not to distort the good genetic material in highly fit individuals.
    - For this reason, mutation is usually applied at a low probability.
- Reproduction can be applied with *replacement*, in which case newly generated individuals replace parent individuals only if the fitness of the new offspring is better than that of the corresponding parents.

Stopping Conditions

- The evolutionary operators are iteratively applied in an EA until a stopping condition is satisfied.
- There are several stopping conditions:

1- A limit is placed on the number of generations.
2- A limit is placed on the number of fitness function evaluations.
3- Terminate when no improvement is observed over a number of consecutive generations.

Assignment

1- Explain the genetic algorithm in terms of: encoding, fitness function, selection, crossover, mutation.

2- Consider the problem of finding the shortest route through several cities,such that each city is visited only once and in the end return to the startingcity (the Travelling Salesman problem). Suppose that in order to solve thisproblem we use a genetic algorithm, in which genes represent links betweenpairs of cities. For example, a link between London and Paris is representedby a single gene 'LP '. Let also assume that the direction in which we travelis not important, so that LP = P L.

a) How many genes will be used in a chromosome of each individual ifthe number of cities is 10?

b) How many genes will be in the alphabet of the algorithm?