



**UNIVERSITY OF INFORMATICS OF MADRID - UPM**  
***ARTIFICIAL INTELLIGENCE DEPARMENT***

INTELLIGENT SERCHING BASED ON METHAHEURISTICS  
MASTER IN ARTIFICIAL INTELLIGENCE

**Solving the magic square problem with the NSGA II  
genetic algorithm**

**Alumni:** Petra Franjic, Gustavo Puig  
**Professor:** D. Alfonso Mateos Caballero  
**Date:** 2th November 2015



## Summary:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer luctus varius augue vel imperdiet. Maecenas neque turpis, accumsan ut eleifend vel, eleifend gravida est. Duis mattis pulvinar porta. Nunc consequat, felis ut malesuada egestas, ipsum urna sodales enim, eu lobortis mauris tellus in augue. In tristique tortor id nisi placerat et rutrum tortor auctor. Aenean erat elit, faucibus non egestas semper, volutpat sit amet ante. Nam nulla justo, suscipit nec sagittis sed, varius id magna. Duis a lorem id felis eleifend aliquet eget vel arcu. Etiam ultricies diam a magna facilisis ornare. Integer tempor massa quis sem gravida bibendum. Sed massa felis, condimentum sit amet bibendum in, tristique sit amet mauris. Nullam tellus ante, ornare eget mollis ac, gravida id nunc. Donec elit neque, aliquam eget auctor sed, gravida in nibh. Vestibulum sit amet diam a magna tempor bibendum quis a ante. Integer augue sapien, facilisis ac malesuada sed, consectetur id turpis. Quisque a leo dapibus felis aliquet vestibulum vitae vitae urna. Nunc ut felis ut metus gravida ultricies. Cras tincidunt nisi ante. Nunc eu quam interdum dui cursus lacinia. Praesent adipiscing ultrices nulla at posuere.

Aliquam gravida facilisis eleifend. Nunc id lorem nibh. Nulla facilisi. Nunc suscipit lectus quis tellus adipiscing adipiscing. Proin ipsum justo, egestas dictum iaculis et, pretium non diam. Aenean vehicula hendrerit lectus, vel bibendum nisi volutpat quis. Fusce molestie condimentum libero, nec scelerisque nisi mollis et. Aliquam placerat accumsan congue. In scelerisque nulla sit amet mauris consectetur consectetur. Suspendisse mattis mi in sapien egestas sit amet accumsan urna sagittis. Mauris metus massa, porta at rutrum eget, sollicitudin eu risus. Donec non diam erat. Nam sagittis, nibh non fermentum ornare, ipsum est mollis felis, eget vulputate tellus ante at erat. Praesent porta tempus velit at faucibus. Donec rhoncus varius nunc, eget imperdiet ligula blandit eget. Morbi pulvinar, nunc ac porta interdum, mauris ligula sollicitudin sem, et sollicitudin turpis magna eu lectus. Pellentesque nulla dui, feugiat quis pharetra eu, accumsan imperdiet quam. Aenean lacinia porttitor leo vitae hendrerit. Pellentesque dictum ultricies dui, et rhoncus nulla sagittis ut. Suspendisse suscipit, risus sed posuere rhoncus, neque metus scelerisque diam, eu gravida quam orci eu dui.

Nulla sit amet sem massa, aliquet auctor magna. Vivamus quis magna sollicitudin felis bibendum mollis. Etiam elementum lacus id quam tincidunt pulvinar. Ut vehicula aliquam ante, vel venenatis libero bibendum eget. Sed gravida lobortis velit, et ornare nunc ullamcorper et. Morbi neque sapien, mattis id semper in, varius at arcu. Duis aliquam ornare sem, eu sagittis libero facilisis non. Nunc quis ultricies augue. Pellentesque ac nulla libero, ut suscipit dui. Morbi aliquet quam non metus iaculis quis accumsan ipsum consectetur.

Mauris mauris odio, varius quis accumsan bibendum, vulputate vel tellus. Sed non consectetur eros. Aenean et mi ut est tincidunt commodo. Donec nunc metus, facilisis a consequat vitae, dignissim quis leo. Cras condimentum, felis a molestie viverra, massa eros consequat nisl, at porta nibh sapien condimentum ante. Duis convallis velit quis metus pulvinar convallis. Suspendisse sed purus non purus dictum ultricies in vitae nisl. Cras convallis erat eu ipsum molestie fermentum. Mauris turpis nunc, condimentum non aliquam ac, ornare at lectus. Donec scelerisque fermentum malesuada. Praesent urna nisl, iaculis sed pharetra eu, tincidunt et lectus. Nulla posuere eros at elit vehicula laoreet. Sed ut erat orci, ac pellentesque nibh. Nulla at sapien sed justo luctus laoreet.

Sed convallis massa laoreet mi venenatis eget dapibus orci varius. Pellentesque purus nisi, placerat nec pretium ac, pretium vitae augue. Vivamus eleifend vestibulum mi, ac dignissim urna vulputate vel. Integer volutpat neque purus. Nulla hendrerit tempus est nec aliquet. Ut facilisis bibendum nisl eget porttitor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Fusce consectetur sem eget ante rhoncus nec mollis lorem volutpat. Fusce in libero lectus, ut gravida lectus. Nam nulla nunc, dignissim sed sollicitudin a, commodo vel felis. Donec lacinia justo quis neque tempor interdum ut cursus mi. Phasellus venenatis porttitor mi consectetur egestas. Pellentesque quis elit nibh. Pellentesque molestie turpis ac arcu gravida sed ultricies felis pulvinar.



# Contents

|  |            |
|--|------------|
| <b>Introduccion</b>                                      | <b>iii</b> |
| <b>1 Problem design</b>                                  | <b>1</b>   |
| <b>2 NSGA-II algorithm</b>                               | <b>3</b>   |
| 2.1 Outline of genetic algorithms . . . . .              | 4          |
| 2.2 Search components of the NSGA-II algorithm . . . . . | 5          |
| 2.2.1 Dominance based ranking . . . . .                  | 6          |
| 2.2.2 Crowding distance . . . . .                        | 7          |
| 2.2.3 Elitism . . . . .                                  | 8          |
| 2.3 Outline of the NSGA-II algorithm . . . . .           | 9          |
| 2.4 Codification . . . . .                               | 10         |
| 2.5 Operators . . . . .                                  | 11         |
| 2.5.1 The crossover operator . . . . .                   | 12         |
| 2.5.2 The mutation operator . . . . .                    | 13         |
| <b>3 Implementation</b>                                  | <b>15</b>  |
| <b>4 Conclusion</b>                                      | <b>17</b>  |



# Introduccion

Genetic algorithms (GAs) belong to a family of stochastic search methods which is covered by the generic term Evolutionary Computation (EC). The main characteristic of EC techniques is that they computationally simulate the natural evolutionary process. The evolutionary process is modeled by the neo-Darwinian paradigm, which combines Darwinian evolution with Mendelian genetics. Other techniques covered by EC are evolution strategies (ESs) and evolutionary programming (EP). The EC family of optimization techniques finds its place among other stochastic search and optimization approaches such as Simulated Annealing, Monte Carlo methods and Tabu search. These stochastic approaches were developed as an alternative to solving high-dimensional, discontinuous and multimodal problems, where traditional deterministic search techniques often proved ineffective. The disadvantage of these methods is that they don't guarantee to find the absolute best solution, but they do promise to find an acceptable solution in a reasonable amount of time. A chromosome or an individual is an encoded candidate solution to a problem. It corresponds to a biological genotype. Chromosomes are typically represented as strings of some predefined alphabet  $A$ . The alphabet can be binary or non-binary. Typical examples of non-binary alphabets are real and integer numbers. The phenotype of an individual is the point in the solution space that this individual genotype maps to. A population is a set of individuals. Genes encode a particular element of the candidate solution. The values which a gene can take are called alleles. \*\*\*Figure search space solution space\*\*\* All evolutionary algorithms share a number of common properties. Firstly, all of them utilize the collective learning process of a population of individuals. Secondly, descendants of individuals are generated by randomized processes intended to model mutation and recombination occurring in nature. Lastly, by means of evaluating individuals in their environment, all individuals are assigned a measure of quality or fitness. This way, it is possible to make a comparison of individual fitness. This is the basis of the process of selection, which favors better individuals to reproduce more often than those relatively worse. There are several basic differences in the utilization of these principles which make genetic algorithms different from the rest of the techniques of EC. Genetic algorithms



emphasize recombination (crossover) as the most important search operator and apply mutation with a very small probability. They also use a probabilistic selection operator and often rely on the binary representation of individuals. The general principle by which the genetic algorithm works is as follows. There exists a population of individuals where each individual represents one possible solution to the given problem. Each individual can be assigned a fitness measure by calculating the value of the function that is being optimized. The selection operator is used to choose those individuals in the population which are going to become the parents of the next generation. The parents produce children by means of the crossover operator, which emulates genetic recombination. Next the mutation operator is applied on the children. Lastly, the reinsertion operator is used to include children in the population of solutions, which closes the cycle of an algorithm.

### Algorithm

**General algorithm design** There are two typical genetic algorithm designs, the steady-state genetic algorithm and the generational genetic algorithm. The steady-state genetic algorithm or incremental algorithm can be thought of as a simpler version of the generational genetic algorithm. At every cycle of the algorithm, two parents are chosen, the crossover operator is applied and a child is formed. The child is subsequently mutated and inserted in the population. Since the population size is maintained constant, the child replaces an individual in the old generation. The individual to be replaced is typically the worst one in the population. Contrary to the steady-state genetic algorithm, the generational genetic algorithm makes a clear distinction between parents and children. At each algorithm cycle, a new generation, composed exclusively of children is formed and the old generation dies out. In this practical assignment we have chosen to implement the steady-state algorithm with replacement of the worst individual in the population. The main reason against the generational approach is that absence of elitism would have made the convergence process slower. An additional reason for choosing the steady-state algorithm is the much lower computational cost needed to generate each generation of individuals. The scheme of the algorithm is shown in Figure ? and the pseudocode is shown in figure ?. \*\*\* The scheme of the algorithm \*\*\* \*\*\* The pseudocode of the algorithm \*\*\*

I'll have to add some modifications to the implementation so that the new generation is formed by choosing best individuals in the set of parents + children

Implementation Class diagrams + explanations + code

Bibliography: Redes de neuronas artificiales y computacion evolutiva EAMOP handbook of CE Cupic(home university book)

Resumen captulo

## Chapter 1

# Problem design



## Chapter 2

# NSGA-II algorithm

## **2.1 Outline of genetic algorithms**

## **2.2 Search components of the NSGA-II algorithm**

### 2.2.1 Dominance based ranking

### **2.2.2 Crowding distance**



### **2.2.3 Elitism**

## **2.3 Outline of the NSGA-II algorithm**

---

## 2.4 Codification

---

## 2.5 Operators

---

### 2.5.1 The crossover operator

---

### 2.5.2 The mutation operator



## Chapter 3

# Implementation





## Chapter 4

## Conclusion



# Bibliography

- [1] Joseph L. Jones; Anita M. Flynn y Bruce A. Seiger. *Mobile Robots: Inspiration to implementation (2nd edition)*. ISBN 1-56881-097-0. AK Peters, 1998.