University of Coimbra
Faculty of Science and Technology
Department of Computer Sciences
Final Report

Sexy Evolutionary Computation

José Carlos Clemente Neves
jcneves@student.dei.uc.pt

Advisor:
Fernando Jorge Penousal Martins Machado
Date: September 4, 2013

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Acknowledgements

I would like to acknowledge my immense gratitude to my great colleague and friend António Leitão for all his support, patience and research guidance. It was thanks to his contribution that I acquired the necessary knowledge to elaborate this thesis and for that I am truly indebted to him. I would also like to acknowledge my gratitude to my advisor Prof. Fernando Penousal Machado for his guidance and suggestions which helped improving of my work and knowledge on the field.

I would like to give a special thanks to two great friends, Leandro Braguês and Jorge Santos, for the care with which they reviewed this thesis. Their collaboration was truly important to me.

I also would like to acknowledge the Centre for Informatics and Systems of the University of Coimbra for providing me shelter and allowing me to use their resources.

Last but not least, I am truly grateful for the support that my beloved family has given throughout my studies, especially my parents, grandparents and partner for their relentless support, belief and comprehension, and my gratitude to all my friends who played an essential role during this period of my life.

José Neves
September, 2013

Abstract

The impact of Natural Selection overshadowed Darwin's Sexual Selection theory. However, over the past few decades it begun to draw the attention of researchers from several different fields and the amount of supporting evidence of its role in evolution rapidly increased. Today, although the dynamics aren't fully understood, the importance of sexual selection in evolution is undebatable. The enthusiasm around this field was not followed by evolutionary computation. On the one hand, canonical evolutionary computation systems were already well-established when sexual selection re-surfaced. On the other, so far, attempts to incorporate sexual selection approaches in evolutionary computation, particularly when applied to optimization problems, have encountered several difficulties and no generic tools and approaches applicable to a wide variety of problems exist.

This dissertation constitutes a step towards changing this situation. Based on an embracing survey of the state of the art and following a nature-inspired approach, a popular evolutionary computation framework is expanded through the incorporation of Mate Choice mechanisms – enabling the application of sexual selection models to a wide variety of problems with little effort. The approach is tested on symbolic regression benchmark problems. The analysis of such problems indicates that sexual selection is able to outperform conventional approaches in complex problem instances. Additional testing and analysis focused on understanding how sexual selection contributes to the evolution. The experimental results show that the evolved mate choice functions are able to select mating partners in meaningful ways, contributing to the evolutionary success of the descendants.

Contents

List of Figures

List of Tables

List of Algorithms

Listings

Chapter 1

Introduction

Since his journey on the *Beagle*, Darwin conceived two theories that explained the evolution of species: Natural Selection and Sexual Selection. The first theory is by far the most widely known and can be described as a process by which organisms possessing specific genotypic characteristics that make them better adjusted to the environment, tend to survive, reproduce and therefore transmit and perpetuate their genotypic qualities to succeeding generations. Since Darwin's publication *On the origin of species by means of Natural Selection* [21], his theory has found acceptance in the scientific community and was settled for decades as the absolute theory for the explanation of evolution.

Still there were some aspects observed by Darwin in nature that Natural Selection could not explain. The costly ornaments that some species possessed did not contribute to the survivability of its individuals and in some cases could even jeopardize it. But despite being costly, such ornaments still spread through future – which goes against Darwin's Natural Selection theory. Taking these facts into consideration, Darwin came up with the theory of Sexual Selection.

However, unlike the previous theory, the theory of Sexual Selection was not well received by the scientific community and was disregarded for over a century. It was Fisher's contributions [35] and soon after with Zahavi's, [83] that research on the theory re-emerged from its century-long rejection to take its place in the spotlight. It expanded to major research areas such as Anthropology, Biology and Psychology, through an increasing number of contributions and evidence supporting its importance and impact on evolution.

Although being widely accepted in other research fields, Sexual Selection is still far from being explored in Evolutionary Computation (EC). The conventional algorithms related to Natural Selection were introduced during the period in which the theory of Sexual Selection wasn't acknowledged and, when it finally re-emerged, they were already established in the scientific community. The lack of acknowledgement along with the complexity and unpredictability of results of Sexual Selection algorithms caused the withdrawal of researchers to adopt the mechanism for automated problem solving. However, the theory's success in the previously mentioned research fields makes it interesting to further explore the resulting effects it may have in the performance of Evolutionary Algorithms and that is a great motivation to its study.

1.1 Scope

Two main processes in Sexual Selection were introduced by Darwin: Male Competition and Female Mate Choice. The first process promotes the evolution of male traits valuable for them to fight others for access to females. The second process promotes the evolution of male ornaments used to attract females for reproduction. Darwin's research, as well as many other contemporary researchers since the theory's reappearance, focuses particularly on Female Mate Choice. Following this trend, this study will focus on the effects and implications of Female Mate Choice in Evolutionary Computation.

Also, this study focuses on genetic programming and on models that abide three nature inspired rules:

1. individuals choose their mates according to their perception of others and mating preferences;
2. the mental traits stated in the first rule are inheritable the same way

physical traits are;

3. mate selection introduces its own selection pressure but is submitted to selection pressure itself.

1.2 Research Objectives

Despite a number of publications in the past regarding Sexual Selection through Mate Choice in Evolutionary Computation, the subject has failed so far to attract the research community. Two of the reasons that have contributed for this scenario are the difficulty to model Mate Choice in Evolutionary Computation and the lack of knowledge regarding its behaviour and effects. Therefore we intend to contribute by meeting two different objectives:

1. expanding a well known framework to include Mate Choice mechanisms;
2. applying the developed framework on a set of benchmarking problems and assess its effects;

By meeting the first goal we expect to facilitate the application of Mate Choice to a wide set of problems with little effort; by meeting the second goal we contribute to existing knowledge regarding how Mate Choice performs and behaves on different contexts. Finally, we expect to deploy the developed framework for public use and disseminate the results on specialized conferences.

1.3 Outline

The rest of the document is organized as follows: Chapter 2 presents a short overview of Evolutionary Computation while focusing attention on Genetic Programming and Self-Adaptive Genetic Programming, since these techniques are the ones used for the study on Sexual Selection. Following the overview, is a review of the concepts on Natural Selection in Evolutionary Computation, an historical background on Sexual Selection through Mate Choice, its important breakthroughs, and applications in Evolutionary Algorithms. Previous works that have represented mating preferences using Genetic Programming and Self-Adaptive Genetic Programming are also highlighted. Chapter 3 includes a short explanation on the framework proposed as well as the reasons that led us to choose such framework, followed by its class diagram. Chapter 4 displays the experiments performed during the first term of research. In this chapter we find the configuration data defined for the experiments, followed by the obtained results and analysis over such results. Chapter 5 contains an explanation of the improvements made on the framework and a class diagram of the final framework. As for Chapter 6, we can find the experiments made during the second term, the results obtained and what can we interpret from those results. Finally, on Chapter 7, we draw conclusions about the proposed methods and obtained results.

Chapter 2

State of the Art

This chapter initially presents a brief overview of Evolutionary Computation, describing the field of study and its sub-fields as well as the mechanisms used, followed by a review of the studies and researches that are considered essential to the subject of Sexual Selection through Mate Choice in Evolutionary Computation. First of all, Section 2.1 describes Evolutionary Computation, and the relationship between this field of study and the theories that became the inspiration to its concept. In the Section 2.3 we will cover Natural Selection in Evolutionary Computation in a deeper context by addressing the arguably most significant implementations of selection methods. Most of these methods are relevant since they may be used as basis for comparison when analysing the results obtained from Sexual Selection through Mate Choice. In Section 2.4, related to Sexual Selection, we will present a review of its background, followed by Section 2.5 which will describe significant studies on the matter. This section is branched in three subsections. Subsection 2.5.1 and 2.5.2 describes the experiments of Mate Choice carried using Evolutionary Algorithms, highlighting performed experiments of Mate Choice using Genetic Programming on the latter. Since the process of Mate Choice approached by this report is based on self-adaptive mating preferences, Subsection 2.5.3 covers a review of related work on Self-Adaptive Algorithms applied on Sexual Selection through Mate Choice.

2.1 Short Overview of EC

Evolutionary Computation [6, 7, 29] is a research area within Computer Science. Its philosophy is based on the natural evolution of species.

In order to understand how Evolutionary Computation gets its inspiration from the natural evolution of species, we have to take into consideration how the evolution process works [21]. A given environment is populated with individuals of a certain species that will struggle for survival and chance to reproduce. Their fitness, which will be determined by the surrounding environment, is directly related to the ability to meet their main goals, namely, their chances for survival and dissemination of their genetic characteristics.

This natural evolution process can be linked to the context of problem solving through stochastic process of *trial-and-error*. In this process we have a set of candidate solutions and each of them is composed of a quality measure that will determine the chances of being stored and used as seed for new candidate solutions. Thus, Evolutionary Computation relates to the process of evolution by linking the environment with the problem, individual with candidate solution, and fitness with quality.

2.2 Evolutionary Algorithms

There are multiple approaches of Evolutionary Algorithms but the underlying idea behind those is based in the following common procedures: given a population of individuals, we apply selection pressure (survival of the fittest) thus causing an improvement of the average fitness of the population. With an existing fitness calculating function, we generate candidate solutions that are within the problem domain and apply such function as an abstract measure — when dealing with maximization problems the greater the individual's fitness, the better; with minimization problems the lower the fitness, the better. Finished the selection process, the recombination operator takes action, if defined, by choosing two individuals (it is possible to recombine more than two [27], although it is uncommon) from the mating pool, which will be the parents of the new generation of candidate solutions, commonly named offspring. After recombining the parents, the mutation operator is applied, also if defined, individually to each new candidate solutions. The resulting set of new individuals can compete, based on their fitness (and possibly age), with the existing ones for a position in the next generation or replace the entire population. The whole process is iterative until a satisfactory candidate solution is found or a user-defined limit is reached.

The variation operators such as recombination and mutation create the necessary diversity needed for the evolutionary flow and are responsible for the creation of new individuals. The selection mechanisms behave like a force promoting the better qualified individuals of the population. Joining variation operators and selection mechanisms in an iterative process will lead to the improvement of the average fitness of the population throughout the generations. During the selection process, fitter individuals have more chances to be selected as parents of the new set of individuals or as survivors for the next generation and those that are less fit will have very low chances of being chosen but even though the probability of selecting less fit individuals is insignificant, it is always possible. As for variation operators, decisions are made based on probability as well. During the recombination process, the components of each parent to be inherited by the offspring are chosen randomly and during the mutation process, the components of the candidate solution that undergo change are also chosen randomly.

The procedural flow of an Evolutionary Algorithm is shown according to the diagram in Figure 2.1 and in a pseudo-code shown in Algorithm 1.

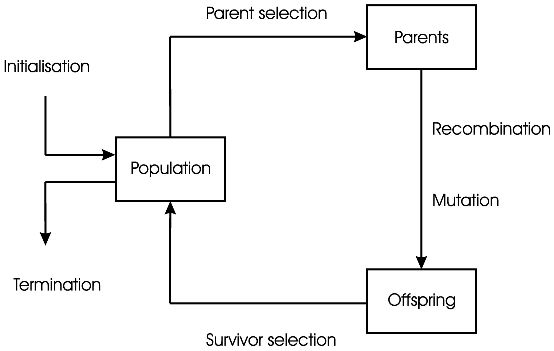


Figure 2.1: Example of an Evolutionary Algorithm Flow Chart [29]

Algorithm 1 Typical EA Algorithm [29]

```
procedure TypicalEA(parameters_list)
    population ← generateSolutions(population_size)
    population ← evaluateSolutions(population)
    repeat
        parents ← selectParents(population)
        offspring ← recombineParents(parents)
        offspring ← mutateOffspring(offspring)
        offspring ← evaluateOffspring(offspring)
        population ← selectNextGeneration(population, offspring)
    until termination_condition is true
end procedure
```

By analysing both figures, one can simply conclude that Evolutionary Algorithms fit in the category of “generate-and-test” algorithms. The evaluation function — or fitness function — represents an heuristic estimation of solution quality and the search process is led by selection, recombination and mutation operators. Evolutionary Algorithms possess some features that strengthen their position within the category of “generate-and-test” such as:

- Evolutionary Algorithms operate at population level. A collection of candidate solutions are processed simultaneously.
- They mostly use recombination to create new candidate solutions from existing ones.
- They perform actions according probabilistic information.

As mentioned above, there are multiple approaches that belong to the Evolutionary Computation domain. They all follow the general flow shown in the diagram above, differing only in some technical details as shown in the table 2.1 [29]. For instance, the representation of a candidate solution often distinguishes the approaches. Typically, the candidate solutions are represented by strings over a finite alphabet in Genetic Algorithms (GA) [25, 49, 50], real-valued vectors in Evolution Strategies (ES) [67, 71], finite state machines in classical Evolutionary Programming (EP) [36] , and binary trees in Genetic Programming (GP) [9, 53, 54, 63]. A given representation might be more adequate than others if it matches the problem better, i.e., if the encoding of candidate solutions represents an easier or more natural form of solution to the problem. It is important to emphasize that the variation operators working on candidate solutions must be chosen and altered, if necessary, to match the representations. Unlike variation operators, the selection mechanisms work independently from the candidates representation since it is based merely on the solution's quality.

Table 2.1: Typical components of Evolutionary Algorithms [29]

Component	GA	ES	EP	GP
Problems	Combinatorial optimization	Continuous optimization	Optimization	Modelling
Typical Representation	String over a finite alphabet	Strings (vectors) of real numbers	Application specific often as in ES	Trees
Role of Recombination	Primary variation operator	Important but secondary	Never applied	Primary/Only variation operator
Role of Mutation	Secondary Variation Operator	Important, sometimes only variation operator	The only variation operator	Secondary, sometimes not used at all
Parent Selection	Random, biased by fitness	Random, uniform	Each individual creates one child	Random, biased by fitness
Survivor Selection	Generational: n.a., all individuals replaced Steady-state: deterministic, biased by fitness	Deterministic: biased by fitness	Random: biased by fitness	Random: biased by fitness

2.2.1 Components of Evolutionary Algorithms

The purpose of this section will be to discuss Evolutionary Algorithms in a deeper detail. Evolutionary Algorithms are composed by a number of components, procedures, or operators that must be specified in order to define a certain Evolutionary Algorithm. The most important are:

- Representation (how the individuals will be defined)
- Evaluation — or fitness — function
- Population characteristics
- Parent selection mechanism
- Variation operators (recombination and/or mutation)
- Survivor selection mechanism (Replacement/Elitism)

Furthermore, to obtain a running algorithm we must specify an initialization procedure and a termination condition.

Representation

The first step in developing an Evolutionary Algorithm is to define the link between the problem context and the search space of the problem's solutions where evolution will occur. The objects that formulate candidate solutions within the problem's context are named phenotypes while their encoding, i.e., the individual belonging to a population of an Evolutionary Algorithm, is named genotype. The process of mapping a phenotype to a genotype is defined by representation. The phenotype's search space can be very different from the genotype's search space and the whole evolutionary search is made surrounding the individual's phenotype. In the original context of the problem, the candidate solution, phenotype, and individual represent positions in the search space named phenotype space. In the context of the Evolutionary Algorithm, the genotype, chromosome, and also individual, refer to positions in the search space where the evolutionary process will be undertaken.

The term “representation” can be used in two slightly different ways. In one way it stands for the mapping from the phenotype to the genotype space which in this sense is synonymous with encoding. The inverse process of mapping from genotype to phenotype space is usually called decoding and it implies that representations have invertible capability: to each genotype must exist at least one corresponding phenotype. Representation can also stand not for the mapping itself but for the data structure of the genotype space, for instance, when we refer to a binary string as a binary representation.

Evaluation Function

The fundamental role of the evaluation function is to form the basis for selection, thereby facilitating improvements. More specifically, it defines what improvement means. In an evolutionary context, it represents the task to be solved. Technically, the evaluation function is a procedure that assigns the quality measure to genotypes, which allows comparison between individuals in order to distinguish the fitter from the less fit, and is typically composed from a quality measure in the phenotype space and the corresponding inverse representation.

The evaluation function is commonly called the fitness function in Evolutionary Computation although this terminology may be counter-intuitive if we are dealing with a minimization problem since fitness is normally assigned to maximization problems. Generally, the original problem to be solved is an optimization problem, thus the term “evaluation function” is most commonly used in the problem context. As for the fitness function, it is commonly used in the Evolutionary Algorithm context but it may be identical or slightly different from the evaluation function.

Population

The main purpose of the population is to store a candidate solutions set. In this set it is possible to have multiple copies of the same element. The individuals of the population are static objects that do not change or adapt, it is the population that undergoes adaptation. Given a representation, the process of defining a population is done simply by specifying its size, i.e., how many individuals it will store. The selection mechanisms are the only of the above mentioned components that operate at population level, as opposed to the variation operators that work with one or more individuals but never the population as a whole. Generally, the selection mechanisms take the whole population into account and choices are made relative to what we have in it, for instance, the best individual of a given population is selected to survive and seed the next generation while the worst is chosen to be replaced by a newly created individual. Typically, the population size is constant and its size is maintained during the evolutionary process. However certain studies have proven that a dynamic population size can achieve better convergence as well as discover any gaps or missing trade-off regions at each generation [77, 58].

No single measure defines the population's diversity. Instead, the number of different solutions present in the population characterizes the diversity. Generally, we might define diversity by verifying the number of different fitness values present, the number of different phenotypes, or the number of different genotypes. Entropy may be also used as well as other statistical measures. It shall be noted that having one fitness value does not necessarily imply correspondence to one phenotypes and, in turn, one phenotype does not necessarily correspond to one genotype. However, the reverse correspondence is not true, one genotype always corresponds to one phenotype and one fitness value.

Parent Selection Mechanism

The fundamental role of parent selection mechanisms is to distinguish among individuals of a given population, based on their quality, so that better fitted individuals are chosen as parents of the next generation of individuals. Together with the survivor selection mechanism, it is the driving force of the average quality improvement of the population. In Evolutionary Computation, parent selection is normally stochastic, meaning that the best individuals have more chances of being selected than the worst. However, the worst individuals of the population still have a small but positive chance of being selected to undergo variation. This prevents the whole search to become too greedy and get stuck in a local optimum.

Variation Operators

Their function is to create new individuals using genotypes of previously selected individuals through parent selection mechanisms as seeds. In the phenotype search space, this corresponds to the creation of a new candidate solution using an existing one. Variation operators, from the "generate-and-test" perspective, are associated to the "generate" step. From the variation operators emerge two types of operator that can be distinguished based on the number of objects as their input:

Mutation is a variation operator that receives one object as its input. It is applied to one genotype and results in a slightly modified mutated genotype, the child or offspring. This is a stochastic operation since the resulting genotype depends on a series of random choices such as a pseudo-random number generator to generate a series of numbers according to a given probability distribution. Mutation has a very important theoretical role in Evolutionary Computation: it allows the operator to jump in the search space, for example, by changing one allele to any other allele with a non-zero probability [28].

Recombination is a variation operator that has two or more — if so implemented, although it is uncommon since it has no biological equivalent but it may have positive effects on evolution [27] — objects as its input. For the rest of the document, we will assume that recombination receives only two objects as input. It consists in merging information from the genotype of two parents that underwent parent selection in order to create new individuals. Like the mutation operator, the information merged is chosen stochastically. The principle seeks that different individuals with desirable features be merged to create an individual that possesses both parents combined features.

Evolutionary Algorithms create a number of offspring by random recombination and accept that some of them will have undesirable combination of features, some will be no better nor worse than their parents but it is hoped that some may have improved features. Despite the principles of Biology which recombination is the main process of reproduction, in Evolutionary Algorithms recombination is applied probabilistically, i.e., there is a slight chance that the operator will not be used and, instead, mutation and replication [29] (copy of the parent genotype) are used.

Survivor Selection Mechanism

Like the parent selection mechanism, its role is to distinguish among individuals based on their quality. But this mechanism, as opposed to the parent selection, is applied in the final stage of the evolutionary process, after the creation of the offspring of the selected parents. As it was mentioned before, the population size remains almost always unchanged during the evolutionary cycle which causes the algorithm to make choices on the individuals that will be allowed to remain for the next generation. The decision is typically based on their fitness values, promoting better fitted individuals although the concept of age can also be implemented. One other characteristic that distinguishes survivor selection from parent selection is that the former is deterministic. For instance, parents and newly created individuals are merged and then sorted by their fitness. The survivor selection mechanisms will select the segment containing the best individuals with size equals to that of the population.

Initialization Procedure

The initialization is normally built in order to be as simple as possible. The most common way to initialize an Evolutionary Algorithm is to generate random solutions that will fill the first generation of individuals. It is advised to maintain the solutions within the problem domain to prevent dissemination of invalid solutions.

Termination Condition

The termination condition can be defined using two different cases. If the optimal fitness value for a given problem is known, then reaching this value — and possibly accepting a certain margin for error — can be one of the cases for a termination condition. However, Evolutionary Algorithms are stochastic which means that such value is always reached. For those cases, it may be necessary to define a different condition that guarantees the end of the algorithm execution. The following termination conditions can be used as alternative to the previously mentioned:

- The maximum number of generations reaches a given value;
- The maximum CPU execution time is reached;
- The total number of evaluations in the algorithm is reached;
- The fitness improvement does not raise above a certain threshold value for a given period of time
- The population diversity drops below a given threshold

2.2.2 Genetic Programming

Since our Sexual Selection through Mate Choice model is implemented using Genetic Programming [9, 53, 54, 63], we will focus on this approach, describing the components, mechanisms and operators that composes a typical Genetic Programming algorithm. The aim of this particular technique is to automatically solve problems without knowledge or specification from the user in advance related to the form or structure to the solution. Abstractly, Genetic Programming is a systematic, domain-dependent method used in order to make computers automatically solve problems from a high-level statement of what needs to be done.

This is done by evolving a population of individuals containing computer programs. Generation by generation, Genetic Programming stochastically transforms individuals of a given population into new ones, hoping to obtain better solutions. The basic steps of a Genetic Programming algorithm follows the principle of Evolutionary Algorithms above mentioned. It aims to find out how well a program works by running it and then comparing its behaviour to some ideal solution. The comparison is later quantified to assign a numerical value to the candidate which will represent its fitness. Programs with better fitness are chosen to breed and produce new programs for the next generation. The primary variation operators used in this technique are recombination and mutation operators.

Representation

In Genetic Programming, programs are usually expressed as *syntax trees* rather than lines of code. An example of a syntax tree of the program $max(x + x, x + 3 * y)$ is shown in the Figure 2.2.

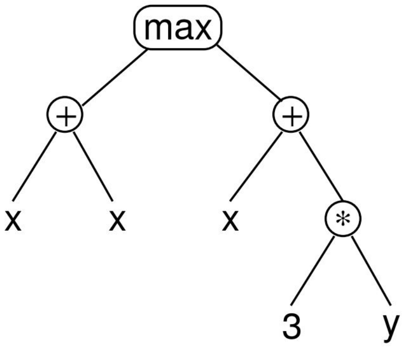


Figure 2.2: GP Syntax Tree representing $max(x + x, x + 3 * y)$ [63]

The variables and constants in the program are the leaves. In Genetic Programming they are named *terminals*, whilst arithmetic operations are the internal nodes named *functions*. The set of allowed functions and terminals together define the *primitive set* of a Genetic Programming system.

It is common to represent expressions of a candidate solution in a similar notation seen in Lisp or Scheme, e.g., the expression $max(x + x, x + 3 * y)$ is represented as $(max\ (+\ x\ x)\ (+\ x\ (*\ 3\ y)))$. With this, it is easier to detect the relationship between expressions and their corresponding trees.

Fitness Function

Without a measuring method, we do not know which elements of the search space include programs that solve or approximately solve the problem. This is the role of the fitness measure which is our primary, and often sole, mechanism for giving a high-level statement of the problem's requirements to the Genetic Programming system.

Fitness can be measured in many ways. For instance: the difference between its output and the desired output, also known as *error* [62]; the amount of *resources* required to bring a system to a desired *target state* [2]; the program's *accuracy* in recognising patterns of classifying objects [76, 57]; the *pay-off* that a game-playing program produces [5]; the *compliance* of a structure with user-defined design criteria [73].

Because the structures being evolved are computer programs, fitness evaluation usually requires executing all the programs in the population multiple times. The overhead of building a compiler to compile the Genetic Programming programs is typically substantial so it is much more common to use an interpreter for evaluation. Interpreting a program tree means executing the nodes in it in an order that guarantees that nodes are not executed before knowing the values of their arguments, if they exist. This is done by traversing the tree recursively from the root node, postponing the evaluation of each node until the values of its children are known. The Figure 2.3 and the Algorithm 2 represent the method for the calculation of the fitness function.

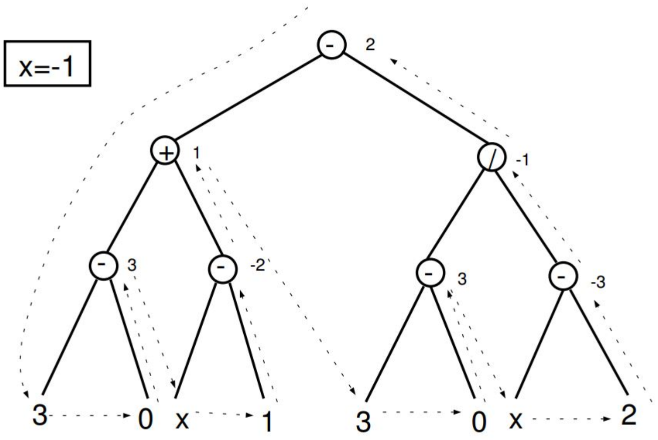


Figure 2.3: Example interpretation of a syntax tree with x as a variable and with value of -1. The number to the right of each node represents the result of evaluating the sub-tree root at that node [63]

Algorithm 2 Interpreter for GP [63]

```
procedure EVAL(expr)
    if expr is a list then
        proc ← expr(1)
        if proc is a function then
            value ← proc(EVAL(expr(2)), EVAL(expr(3)), . . . )
        else
            value ← proc(expr(2), expr(3)), . . . )
        end if
    else
        if expr is a variable or expr is a constant then
            value ← expr
        else
            value ← expr()
        end if
    end ifreturn value
end procedure
```

In some problems we are interested in the output produced by a program, namely the value returned after evaluation. In other problems we are interested in the actions performed by the program. But in either cases, the fitness depends on the results produced by its execution on many different inputs or under a number of different conditions. Another common feature of Genetic Programming fitness measure is that, for many practical problems, they are multi-objective [24], i.e., they combine two or more different elements that are often in competition with one another.

Initialization

Since Genetic Programming is an Evolutionary Algorithm, the initialization is made as previously stated, i.e., by generating a number of random individuals equal to the size of the population. The are a number of different approaches to generate this initial population. The two of the simplest and earliest are the "full" and "grow" methods. A combination of the two is also widely used and it is know as "ramped half-and-half" which we describe in more detail later in this subject.

Both full and grow methods generate individuals taking into account a given maximum depth (depth of a node is obtained by counting the number of traversed nodes starting from the tree's root, which is not included in the count, until given node is reached). Thus, the depth of a tree is the same as its deepest leaf. In the full method, elements from the function set are assigned to the tree nodes in a random fashion until the maximum tree depth is reached, assuring that all leaves have the same depth and only terminals are assigned to them (its called full method for generating full trees, i.e., all leaves have the same depth). Although trees generated by the full method have their leaves at the same depth level, it does not necessarily means that all initial trees will have the same number of nodes or the same shape. This only happens if every element of the function set contains functions with equal number of input objects.

The grow method, on the contrary, allows for the creation of trees of more varied sizes and shapes. When building a tree, both functions and terminals are selected from the function set until the depth limit is reached. Once it is reached, a random terminal is chosen (just as in full method) and the process is finished.

Because neither the grow or full method provide adequate variety of sizes or shapes, Koza [53] proposed a combination of both methods and called it "ramped half-and-half". This method consists in constructing half of the population using full and the other half using grow. For this, it is defined a range of depth limits (hence the term ramped) to help ensure variety of sizes and shapes in the initial population. Pseudo-code for a recursive implementation of both the full and grow methods is given in the Algorithm 3.

Algorithm 3 Pseudo-code for Ramped Half and Half method [63]

```
procedure GENERATERNDEXPR(fset, tset, maxd, method)
    if $maxd = 0$ or $\left(\text{method is grow and } rand \leq \frac{|tset|}{|tset|+|fset|}\right)$ then
        proc ← expr(1)
        if proc is a function then
            value ← proc(EVAL(expr(2)), EVAL(expr(3)), ...)
        else
            value ← proc(expr(2), expr(3)), ...)
        end if
    else
        if expr is a variable or expr is a constant then
            value ← expr
        else
            value ← expr()
        end if
    end if
    return value
end procedure
```

While these methods are easy to implement, they often make it difficult to control the statistical distribution of important properties such as sizes and shapes of the generated trees. For instance, the sizes and shapes of the generated trees using the grow method are highly sensitive to the proportion between the size of function and terminal sets. If there are significantly more terminals than functions, the grow method is more likely to generate very short trees regardless of the depth limit. But, if there are significantly more functions than terminals, the grow method will tend to behave quite similarly to the full method.

Selection Mechanisms

As with most Evolutionary Algorithms, operators in Genetic Programming are applied to individuals that have been probabilistically selected based on their fitness. The most common method for selecting individuals applied in Genetic Programming is *Tournament Selection* [40].

In it, a number of individuals are chosen at random from the population. They will, then, compete with each other and the individual with the best fitness is chosen to be one of the parents of the next generation. The selection mechanism only compares which program is better than another, it does not need to know how much better the program is. This maintains a constant selection pressure thus preventing a single extraordinarily good individual to completely dominate the next generation with its children. If that were to happen, we would lose diversity rapidly causing potentially disastrous consequences to the run.

While preferring the best, tournament selection does not guarantee that average quality individuals have a chance of producing children. Since this selection method is easy to implement and provides automatic fitness rescaling, it is commonly used in Genetic Programming.

Despite the fact that tournament selection is a common selection mechanism in Genetic Programming, many other mechanisms proposed in the Evolutionary Algorithms literature can be used as well. Goldberg [42] presents, for instance, *Fitness Proportionate Selection*, *Stochastic Universal Sampling* and several others.

Variation Operators

Genetic Programming is distinguished from other Evolutionary Algorithms in the implementation of recombination and mutation operators. The most common recombination method used is *sub-tree recombination.* Having two previously selected parents, this operator randomly and independently selects a crossover point — a tree node — in each parent. Then, it creates the offspring by copying the first parent and replacing, in the copied tree, the sub-tree rooted at the crossover point with the sub-tree rooted at the crossover point of the second parent. The process is repeated for the copy of the second parent, replacing its sub-tree rooted at the crossover point with the first parent's sub-tree rooted at the crossover point. Copies are used in order to prevent disruption of the parents' original content. The Figure 2.4 is a representation of the crossover procedure in Genetic Programming.

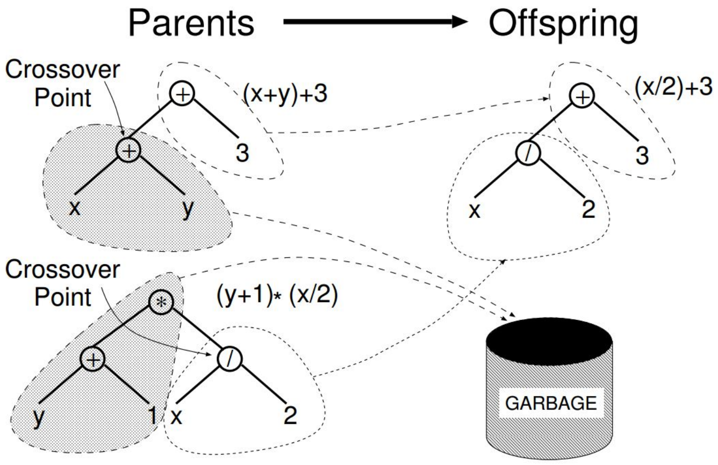


Figure 2.4: Example of a sub-tree crossover. The trees on the left are copies of the parents so that their genetic material can be used without altering the original trees [63]

Crossover points are usually selected in a non-uniform probability. Typical Genetic Programming primitive sets lead to trees with an average *branching factor* (number of children of each node) of at least two so the majority of nodes will be leaves. Consequently, the usage of uniform selection of crossover points will lead to crossover operations frequently exchanging only small content of genetic material. Many crossovers may in fact simply swap one leaf in each tree. To counter this problem, Koza [53] suggested to use an approach that chooses functions 90% of the time and leaves 10% of the time.

The most commonly used mutation operator in Genetic Programming is named *sub-tree mutation* and consists in randomly choosing a mutation point in a tree and substituting the sub-tree rooted in it with a randomly generated sub-tree as it is illustrated in Figure 2.5. Sub-tree mutation can also bee seen as a crossover between the sub-tree and the newly generated tree, which gave it the name of *headless chicken crossover* [4].

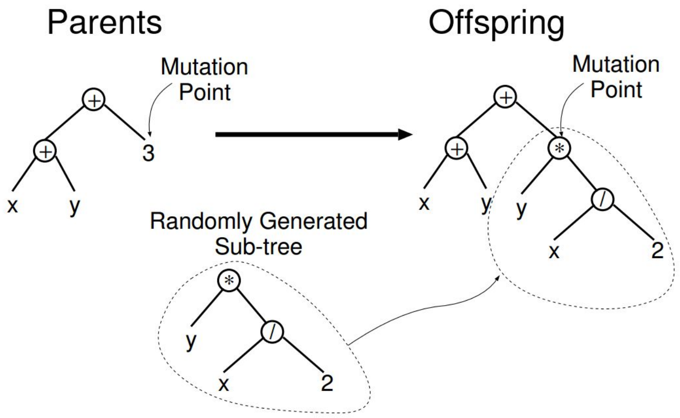


Figure 2.5: Example of a sub-tree mutation. As in crossover, the operation is processed on a copy of the parent tree [63]

Point mutation is also a common form of mutation that, in Genetic Programming, resembles the *bit-flip mutation* in Genetic Algorithms [41]. In point mutation, a random node is selected and the stored primitive is replaced with a different random one with the same number of input objects from the primitive set, if it exists. Otherwise, the selected node is not mutated but other nodes may still be.

Operators in Genetic Programming are usually mutually exclusive, unlike other Evolutionary Algorithms where offspring are sometimes obtained via a composition of operators, and their probability of application is called *operator rates*. Typically, crossover is applied with the highest probability, often being 90% or higher. On the contrary, the *mutation rate* is much smaller, typically about 1%. When the rates of crossover and mutation add up to a value p which is less than 100%, an operator called *reproduction operator* is also used with a rate of $1 - p$. Reproduction simply consists on selecting an individual based on its fitness and inserting a copy of it into the structure containing the next generation of individuals.

2.3 Natural Selection

The existing selection mechanisms applied in Evolutionary Algorithms are, in a way, a representation of the very familiar procedures observed in nature and that play a major role in the evolution of species, namely Natural Selection. Those mechanisms simulate a competition of individuals in a population struggling for natural resources, thus having a very determinant role on their relative survival rate. The fitter the individuals, the better the chances for them to reproduce and spread their genes through the next generations, resulting in a propensity of the individuals with higher fitness to dominate over those with lower fitness.

Usually in Evolutionary Algorithms, fitness is emulated resorting to a function that evaluates each individual on a determined environment, returning a value representing how fit that individual is. Selection mechanisms favour fitter individuals which give them better chances to produce more offspring and such results in a higher propagation of their genes causing a tendency in the population to move through the search space into the best manageable areas and include fitter and fitter individuals. One important feature of the selection mechanisms is that they allow control over the selection pressure which, in turn, affects the convergence speed of Evolutionary Algorithms. If such pressure is too low, the convergence of the population into the optimal area may be too slow. Otherwise, if the pressure is too high, the population may converge too early into a local optimal, to a state where it cannot improve further. These situations demand an adequate balance of the selection pressure for a good performance of Evolutionary Algorithms [42].

Since the appearance of Evolutionary Computation and its early stages, several selection mechanisms have been developed and applied to countless problems. The following subsections include a possible classification of arguably the most popular of those methods.

2.3.1 Proportionate Selection

The methods described below choose individuals for breeding based on their fitness value.

Roulette Wheel Selection

Developed by De Jong [23], it attributes a probability of an individual being picked, among the individuals of the entire population, directly proportionate to its fitness through Eq. 2.1 where $P_{sel}(i)$ is the probability of an individual with fitness $f(i)$ being selected over the sum of fitness from all the individuals, represented by $\sum_{j=1}^{n} f(j)$. This method strongly resembles a roulette wheel in which each space corresponds to an individual and its size varies according to the individual's selection probability.

$$P_{sel}(i) = \frac{f(i)}{\sum_{j=1}^{n} f(j)} \tag{2.1}$$

Deterministic sampling

This method was introduced by Brindle [14] and later explored by Goldberg [42]. This mechanism consists in calculating the number of expected copies of each individual to undergo crossover through Eq. 2.2 where $P_{sel}(i)$ corresponds to the result of Eq. 2.1 and n corresponds to the population size. With this mechanism, each individual is selected the same number of times as the integer part of $C(i)$. In case this method does not generate enough individuals to account for the needed mating pairs, they are sorted in descending order according to the fractional part of Eq. 2.1 and the missing slots are filled with the best ones.

$$C(i) = P_{sel}(i) \times n \tag{2.2}$$

Stochastic remainder sampling

Stochastic remainder sampling [14, 11] was introduced at the same time as deterministic sampling and it works in a very similar way. The difference resides in the case where not enough parents are selected based on the integer parts obtained by Eq. 2.2. If such happens, instead of sorting the population and fill the missing slots, this mechanism stochastically selects parents until all slots are filled. In case this is done with replacement, a roulette wheel method is applied where each individual will have a probability of being selected directly proportionate to the fractional part of $C(i)$. If it done without replacement, the fractional part of $C(i)$ is used as a bias for coin toss.

Stochastic universal selection

This selection procedure [8] follows the same methodology as the roulette wheel selection but, instead of running the method when selecting each parent, it uses a much simpler, less demanding approach. A number of markers equal to the number of individuals in the population are equally distributed on the outside of the wheel. After that, each individual is selected as many times as the number of markers placed in its corresponding wheel section.

2.3.2 Ranking Selection

The methods described in this section rely on how individuals are ranked in an ordered population for selection of parents in order to reproduce.

Linear ranking selection

This method was introduced by Baker [44, 82], who proposed a ranking mechanism on the individuals according to their fitness values and their selection based on a roulette wheel fashion where the probabilities of selecting each individual are linearly proportional to their rank. The individuals with higher fitness will have higher rank while the opposite happens to the individuals with lower fitness. On the one hand, high rated individuals are unlikely to be able to dominate the roulette wheel, giving better chances for others to be selected. On the other hand, very similar individuals that would likely have very close selection probabilities (considering a proportionate selection mechanism) may, in this case, have a large difference when selecting parents.

Exponential ranking selection

This mechanism [10] works like the previous one, with the difference residing on how the individuals are ranked. Instead of applying a linear rank, this mechanism defines the selection probability of parents exponentially proportionate to their rank. By doing so, better ranked individuals are more favoured when comparing to the linear ranking selection.

Truncation selection

Truncate selection allows only the best subset of individuals to be selected as parents for mating. An approach to genetic algorithms was first proposed by Muhlenbein [70] and it consists in defining a threshold n making only the best n individuals available for selection, each one with $1/n$ probability. This results in the assignment of the same probability to all the individuals in the subset, even though they have different performances but also results in the removal of all the individuals outside the *elitist* group from the selection process.

2.3.3 Tournament Selection

The next methods define selection as a set of randomly chosen individuals that compete between themselves for a chance to reproduce. Tournament selection implementations are very efficient given that is not necessary to sort the individuals within a population.

Binary tournament selection

As the name implies, the mechanism randomly selects two individuals from the population and simulates a competition between them based on their fitness as if they underwent a match, keeping the fittest as a parent [14]. Tournaments are carried out until all parents needed for reproduction have been chosen.

Larger tournament selection

Similar to the binary tournament selection with the exception of the number of participants. In this case, the mechanism selects n individuals instead of two as a generalization of the previously described mechanism. As n increases so does the selection pressure since each individual must compete against a larger set of candidates.

Boltzmann tournament selection

A more complex selection mechanism [43] than the previous two, it works by applying a tournament of three individuals but, instead of behaving like a standard size three tournament selection, this mechanism chooses its participants through three distinct methods: the first individual is, like the standard tournament selection, selected randomly; the next participant will be chosen based on a predefined threshold x which represents the difference between its fitness and the first individual. The third and last individual is selected by one of two ways in an alternate fashion. Either in a way that its fitness differs by x from both the first and the second participants or just from the first one.

2.3.4 Restricted Selection

The methods described below rely on sets of rules and conditions in order to restrain the selection operations.

Selection with chromosome differentiation

Chromosome differentiation selection is based in the ideas of male vigor and female choice and emulates these differences by applying different selection mechanisms for each of the two required parents [79]. As Wagner and Affenzeller suggest, this mechanism mimics male vigor by selecting the first individual randomly from the population and also mimics female choice by addressing a different method when choosing the second individual such as roulette wheel or linear ranking selection.

Restricted mating

The mechanism, introduced by Booker [12], consists in restricting each individual to mate with others similar to it thereby promoting niching. Consequently it prevents individuals from a particular niche to be selected along with others from a different niche which in turn becomes fitter to search landscapes that have a large number of local optima. Restricting interaction between niches implies a less likelihood for individuals to converge to one local optima thus promoting the exploration of various local optima.

Reserve selection

Introduced by Chen et al. [17], this selection mechanism's primary aim is to maintain population diversity relying on the segmentation of the population in two distinct parts: non-reserved area and reserved area. The non-reserved area consists of highly fit individuals that are subjected to evolution through the regular operators in order to exploit their high quality genes. The reserved area is particularly designed to maintain the less fit individuals so that diversity is maintained and the exploration of larger parts of the search space is promoted. Unlike in the non-reserved area, the reserved individuals evolve through specially built operators, allowing them to have an active role in the evolution process and possibly reaching solutions that can be considered of high fit, making the individual become a part of the non-reserved area.

2.4 Sexual Selection

It was during the expeditions made on board of the ship *Beagle*, between 1831 and 1836 that Charles Darwin furthered his studies and research leading to the development of one of the most revolutionary theories ever related to the evolution of species, the Theory of Natural Selection. In his book, published in 1859 and entitled *"On the Origin of Species by Means of Natural Selection"* [21], Darwin argues that the species we know nowadays went through an evolutionary process for millennia which allowed them to acquire a certain set of capabilities that assist them on their survival. In a general way his theory was very well received in the scientific community as most scientists and researchers recognized the strength of Darwin's arguments.

But despite such acceptance, there was one aspect in evolution that his theory couldn't explain which was the rich animal ornaments found in several researched species during the *Beagle* expedition. Those ornaments seemed to serve no survival purpose, were therefore useless and, more importantly, costly sometimes in a way that could possibly jeopardize the species rate of survivability. The evolution of such ornaments was an enigma for Darwin as they defied his idea that traits were shaped for survival purposes. With this in mind, Darwin developed his theory of Sexual Selection, aiming to overcome this gap that the theory of Natural Selection could not explain.

Even though Natural Selection was Darwin's main research from which his theory emerges, he began to look into animal ornamentation early, dedicating three paragraphs to the subject in his work on the origin of species [21]. Furthermore, in *"The Descent of Man and Selection in Relation to Sex"* [20] Darwin devoted seventy pages regarding Sexual Selection in humans and five hundred pages on the same subject regarding other species. While the publication's main topic was the origin of mankind, Sexual Selection was extensively introduced, a subject that was considered very touchy at the time.

Animal ornamentation was something that puzzled Darwin since Natural Selection was unable to explain the existence of largely, costly and complex traits that didn't seem to contribute to the individuals survival ability. According to his theory, only the traits that benefit the individuals rate of survivability would be favoured and, consequently, ornaments would become counter-productive. Taken these facts into consideration, Darwin envisioned that, besides Natural Selection, another trait-shaping selection feature must be part of the evolution process and that feature would cause traits that help individuals, when competing against others for a mating pair, to be spread through the species even if those traits can severely compromise the individual's rate of survivability.

The main purpose of Natural Selection is making species adapt to the surrounding environment while Sexual Selection's main purpose is making each sex adapt in relation to other through the struggle of individuals of one sex for the possession of individuals of the opposite sex. In a realistic environment and unlike Natural Selection, the outcome of failing this process is not death but rather few or possibly no offspring resulting in a very low or non-existent spread of the individual's genes. In an evolutionary perspective, failing to compete for mates will result on individuals perishing before being able to reproduce which has the same outcome as having very weak survival abilities. On the other hand, reproduction followed by an early death still serves its purpose anyway since the genes were already inherited by the offspring. Therefore, the previous process explains how differences in reproductive success leads to evolutionary changes.

As a way to explain Sexual Selection, Darwin resorts to Artificial Selection, an experiment also used earlier to explain his theory of Natural Selection. Darwin aimed to promote the theory by arguing that the same way human beings evaluate other species aesthetics and promote reproduction between those that fit their preferences, it is likely that the same can happen based on each species own preferences. He also argued that Sexual Selection is characteristic of species with highly developed perception systems, allowing courtship behaviour and active Mate Choice to be carried out.

Sexual Selection can be separated into two main processes: competition among males for access to females and female Mate Choice. The first process promotes the evolution of natural weaponry, body strength and other characteristics needed for males to quarrel with each other for access to females while the second one explains the existence of ornaments and courtship displays to attract females.

According to Darwin's ideas on Sexual Selection through Mate Choice, female preferences alone may lead mating candidates to develop traits to an high degree of decoration. However if trait evolution imposes a cost that threatens the survival of too many individuals, then Natural Selection will suppress it, risking extinction.

Darwin's theories are unprecedented and suggest that the process of evolution appears to result from the differences between reproduction success rather than survival alone and they also claim that evolution results not only from the adaptation of species to the environment but also from mating choices.

Despite the huge effort put by Darwin to introduce the theory of Sexual Selection and the mechanisms of Mate Choice, the theory was not well received in the scientific community and many researchers refuted Darwin's arguments. Alfred Wallace was possibly the most influential researcher to contradict this new theory. His explanation for the existence of ornaments in both sexes was that they had no weight in the evolutionary process but rather worked as a way for individuals to identify others [80]. In the end, Wallace's theory had been better accepted than Darwin's, sending the latter into a "limbo" for about a century with contributions of only a few other researchers such as August Weismann in his book, published in 1904 [81], and Ronald Fisher, who contributed the most to the theory of Sexual Selection through Mate Choice providing answers to many of Darwin's unanswered questions.

In 1915, Sexual Selection resurfaces with Ronald Fisher as the main supporter. That year, in a 1915 journal [35], Fisher presented the challenge of explaining the origin of mating preferences and, in 1930, he suggested that those preferences can be considered biological traits and thus are as subject to inheritance and evolution as the physical traits [34].

Fisher came up with two major concepts to the theory of Sexual Selection. The first, outlined in his 1915 publication, stated that ornaments evolve as biological indicators of fitness, health, energy, reproductive potential, etc. This idea described that if females have particular mating preferences and if these are complemented by specific male ornamentation with some being healthier than others, then females with those preferences will have reproductive advantage given that they select mating partners that seem to provide better genes to the offspring. Over time, preferences for particular ornaments will spread over the population as well as be inherited over generations, giving better chances of reproduction to the males possessing the preferred ornaments.

Regarding the second idea, stated in his 1930 publication, Fisher believed that both mating preferences and preferred features are able to evolve in a positive-feedback loop causing ornaments to suffer extreme evolution due to sexual competitiveness between individuals. This concept, known as the Runaway Sexual Selection, is more characteristic of populations of species where males are able to mate with multiple females and highly depends on the original existing traits and preferences as they evolve in unpredictable directions. For example, considering a population where preferences and traits can vary according to the inherited genes but haven't changed for a significant number of generations causing maintenance in both characteristics. Coincidentally, either by reproduction or mutation, a change in the mating preferences of a group of females occurs, triggering a preference in a specific trait that will give reproductive advantage to the individuals showing the preferred trait. With parents passing on their original preferences and traits, the number of offspring inheriting the runaway preferences and traits will also increase. The effect occurs showing a genetic correlation with increasing speed, providing momentum to the runaway process.

Fisher theorized that, in cases where individuals with most ornaments monopolize the reproductive advantage, the runaway process will strongly reinforce both ornaments and preferences to a point where the survival cost becomes unbearable for individuals to maintain and Natural Selection disrupts the runaway process. As happened to Darwin, Fisher's ideals were target of heavy criticism by various authors such as Julian Huxley in his 1938 and 1942 publications [52], which made them fall into oblivion until the 1960s.

Once again, Sexual Selection through Mate Choice was disregarded as an evolutionary theory. But, besides being subjected to heavy criticism, other existing factors were decisive to such contempt. First of all, it is a complex theory to model mathematically. Unlike Natural Selection which implies adaptation of a given population to an environment where effects of different genes are easier to predict, Sexual Selection implies that individuals contribute for the selection pressure while being subject to it themselves in the evolution process in addition to the existing one imposed by Natural Selection. This feedback loop between preferences and traits makes the theory very hard to model and analyse. Secondly, there was very few support from biologists since they considered Sexual Selection as a way for individuals to identify others of the same species as well as were focused in the study and explanation of speciation. Thirdly, animal behaviour including sexual behaviour was considered a result of stimulus instead of choice. Lastly, Sexual Selection would imply an active role of costly, threateningly ornaments in the evolutionary process and since Evolution was regarded as the adaptation of species as means to increase their rate of survivability, ornaments would be placed on a lower level of adaptation than others as they didn't contribute for the survival of the species.

In 1975, Amotz Zahavi expanded Fisher's theories by defending that ornaments could be considered indicators of fitness. Contesting the critics arguments, he argued that because ornaments were costly, weak and unhealthy individuals were unable to maintain them therefore those who are able to maintain such features must possess higher fitness than the former [83]. Zahavi named his principle as the Handicap Principle due to ornaments working as handicaps for survivability. Richard Dawkins promoted this idea as well in his 1976 publication [22].

Like most theories concerning Sexual Selection through Mate Choice, the Handicap Principle was target of heavy criticism. Even so, Zahavi's contribution got enough attention allowing Fisher's theory about fitness indicators to resurface and the discussion around both ideas led to Darwin's ideologies. The amount of publications regarding Sexual Selection increased so much that more research was performed on the following decade than during the past century. The aroused discussion attracted authors and researchers from various study areas such as biology, psychology and anthropology where many of the models developed to Sexual Selection in the past years were applied for the first time and confirmed many of the theories elaborated behind Mate

Choice. In the 1990s decade, research on Sexual Selection emerged from the century-long oblivion to finally gain acceptance in the scientific community. The history and ideas behind Sexual Selection through Mate Choice have been reviewed, for instance, by Helena Cronin [19] and Malte Andersson [3].

2.5 Related Work

In this section, we will describe in more detail the research made in the field of Mate Choice by several authors. We begin with the research experiments performed in Evolutionary Algorithms since its introduction in the scientific community. Next we will describe the research accomplished using Mate Choice along Genetic Programming and close the chapter with 2.5.3 where studies performed with Mate Choice models along with Self-adaptive are reviewed .

2.5.1 Mate Choice in Evolutionary Algorithms

Since 1980, Sexual Selection through Mate Choice has been accepted as one of the major driving forces behind evolution by Evolutionary Biology and many other science fields have also adapted to include Sexual Selection through Mate Choice models. Evolutionary Computation has its seed in the 1950s decade and by the 1990s its development allowed the identification of four main approaches. Evolutionary Programming was introduced by L. Fogel in 1960 [36], Genetic Algorithms were proposed by John Holland in the 1970s decade [49], Evolutionary Strategies were developed by Rechenberg and Schewefel during the 1960s and the 1970s decade [66] and Genetic Programming established itself during the 1990s decade following some of the same ideas of its ancestors but was initially idealized during the 1960s decade [55].

Evolutionary Computation emerged when Sexual Selection was considered of no true importance and by the time Sexual Selection became a solid theory, Evolutionary Algorithms were already settled based on the ideology of Natural Selection through its popular models from the 1950s to the 1970s decades. Although there is a number of studies on applications of Sexual Selection through Mate Choice in Evolutionary Computation, attracting the scientific community to further explore the theory has proven to be difficult leading to a failure in integrating it in the realm of Evolutionary Computation.

This indifference towards Sexual Selection through Mate Choice may occur on the account of the following probable reasons. First of all, this is a process that strongly depends on the individuals ornamentation, perception and mating preferences and these “new biological” traits suffer changes during the evolution process along with the existing physical traits. Given that such process is complex and unclear makes it very challenging to model. Secondly, the introduction of Sexual Selection through Mate Choice brings a new paradigm to Evolutionary Computation. Instead of solely have individuals adapting to a static environment as often happens with Natural Selection, in this new theory they not only suffer pressure from the environment through Natural Selection but also from other individuals in the population which are themselves under the same evolutionary pressure. With this new paradigm comes a new challenge in means to analyse the performance and behaviour of the Evolutionary Algorithm which may drive researchers away from such models. Thirdly, the results obtained on many performed experiments using traditional Evolutionary Algorithms are frequently good enough, which discourages researchers to explore the application of new methods. This has favoured resorting to Natural Selection and led researchers to spend their time in possible solutions that could help improve the experiments’ results on particular applications rather than exploring the use of Sexual Selection through Mate Choice. Ultimately, despite its development, Evolutionary Computation still has many unanswered questions and answering them have been more appealing to the interests of researchers.

In other study areas such as Biology, Anthropology and other Human Sciences, Sexual Selection through Mate Choice became a major research subject but in Evolutionary Computation, due to the reasons mentioned above, research took other directions, keeping it from having a more active role in the study area. Still, several studies worth reviewing have been published in the past couple of decades.

Of the existing models of Sexual Selection through Mate Choice, the two that are probably the most known are the Assortative Mating Selection [16] and Disassortative Mating Selection [32]. The first occurs when individuals with similar phenotypes or genotypes are more frequently paired for mating purposes than expected by chance. The second model consists in the inverse process of the former i.e. individuals with dissimilar phenotypes or genotypes tend to be more frequently paired for mating purposes. Individuals choose their partners based on Hamming distance, counting the number of different information between the choosing individual's phenotype and each individual's phenotype of the candidate set, or Euclidean distance by calculating the difference between the choosing individual's genotype information and each individual's genotype information of the candidate set [72].

Chien-feng Huang elaborated a study on assortative and dissortative mating selection [51] where he applies four different approaches using Hamming distance and analyses the obtained behaviour. The results of the approach showed that dissimilar selection promotes better and innovative solutions but at the population's mean fitness expense which decreases due to a higher probability of disrupting existing building blocks by mixing dissimilar individuals.

Ratford et al. [65] proposed a seduction function that combines the fitness of the mating candidates with the Hamming distance to the first parent. The measure benefits mating between individuals that are neither too similar nor too dissimilar. They also propose dynamically adapting the bias of the function at each generation so that mating between dissimilar individuals is favoured at the beginning of each run but gradually gives space to mating between similar individuals through the evolution process. The ability of the approach on finding multiple solutions on multi-modal problems was assessed and results show that the proposed strategy may be an important asset. The same research group has proposed a seduction function that doesn't rely on fitness but rather on either Hamming distance, Euclidean distance or common building block between the first parent and mating candidates [64]. They study the approach on a set of test problems and for the most of it, results are reported to be significantly better than a traditional approach. It is also discussed that either similarity measure performed robustly without the need to rely on the fitness of the individuals for mating purposes.

Hinterding and Michalewicz [47] apply Sexual Selection through Mate Choice to the constrained optimization of a Non-linear Programming Problem (NLP). They propose an alternative method where the fitness function, besides returning the individual's fitness (without penalizations), also returns information on the constraints. On their study, a individual is first selected using a tournament that favors feasible individuals which will then select a mating partner from a set of candidates. By preferring mates that, in conjunction with itself, violate the least number of constraints they aim at producing more feasible offspring. The approach was compared with standard constraint optimization methods achieving comparable results.

In [60], Matsui proposed two correlation based operators, the first one for Sexual Selection through Mate Choice, named *correlative tournament selection*, and the second one for survival. The approach selects mating partners from a set of candidates that are evaluated through a function that assesses both the fitness and the hamming distance to the first parent, promoting those with a higher correlation. The results obtained showed an increase on population diversity as well as on performance.

Sanchez-Velazco and Bullinaria proposed a gendered Sexual Selection through Mate Choice model where the gender is randomly attributed to individuals of the initial population and sequentially attributed to offspring [68]. In their approach males are selected first and choose a female mating partner. The authors also propose an application of a weighted mean of three distinct criteria as mating evaluation function. The first is the candidate's fitness, the second is the difference between the candidate's previous offspring's fitness and the male itself, and the third criteria is a scaling function of the female's age. This model was applied to the Travelling Salesman problem where improvements were consistently shown. Other functions were tested by the same authors where a similar approach was applied [69]. Besides presenting a certain level of problem dependency, the Sexual Selection model outperformed a standard approach.

A similarity-based approach for Mate Choice in Genetic Programming was proposed by Fry et al. [37]. The approach consists in the evaluation of candidates using a function that determines the similarity between the first parent and each candidate along with its fitness, favouring the mating of dissimilar pairs. As a way to measure such similarities, the authors suggest three methods: edit distance similarity, relative similarity and absolute similarity. They also propose that both mate selection and regular tournament selection should be used for the selection of the second parent along with a parameter that will define which of them will be applied and its value should adapt using either a population level dynamic probability or an individual level dynamic probability. The approach was tested on symbolic regression problems and the results obtained showed that mate selection stimulated species evolution. Both adaptive approaches allow the system to adapt its behaviour between exploration and exploitation, and the results of their implementation presented enhanced results. For instance, adaptation on the individual level resulted on good solutions showing up later on the evolution process but the probability of successful crossovers is consistently better.

In [39], Galán and Mengghoel propose a Sexual Selection through Mate Choice model where either survival fitness value or Euclidean distance are used as mating preferences, emphasizing three methods for Mate Choice: *Best-First* selection, *Best-Last* selection and *Self-Adaptive*. The first method consists in selecting the first individual from an ordered set of mating candidates according to the first parent's chosen preferences. The next method works the same way as the previous with the exception that the mating partner will be the last individual of the set. The last proposed approach consists of encoding a mating index in each individual, therefore, defining their own mating preferences. The index parameter is inherited from parents to offspring and is subject to a customized mutation operator. Obtained results show better performances on the *Best-First* approach on uni-modal functions while *Best-Last* performed better on multi-modal functions. The *Self-Adaptive* approach performed robustly in both functions and is considered a great option for cases where the multi-modal level is unclear.

In [78], Varnamkahsti and Lee apply Sexual Selection through Mate

Choice model to the multidimensional 0/1 knapsack problem. They propose splitting individuals into female and male populations in an alternate way, selecting females using tournament selection followed by the selection of a male partner from a candidate set by each triumphant female. Mating choices are based on the maximum Hamming distance but if a draw occurs the fittest male is selected. In case of a second draw, females choose the male with the most active genes. If, once again, a draw occurs, then the male is randomly selected. The authors compare this approach with traditional selection operators obtaining similar results.

In [45], Guntly and Tauritz propose a Learning Individual Mating Preferences approach (LIMP) with two variants: a centralized approach (C-LIMP) and a decentralized approach (D-LIMP). LIMP is adapted for binary representation problems using a floating-point vector with the same size as the genotype as representation of the individual's preferences (each position represents how much an individual desires that the same position of the genotype is set to one). In D-LIMP each individual possesses its own preference vector used to evaluate the genotype of a mating candidate. After choosing a mate, the offspring will be generated through crossover and they will inherit both preferences from both parents so that they correspond to the inherited genes from each parent. Parent's preferences as well as offspring's are updated according to the latter's fitness so that successful crossovers reinforce preferences while unsuccessful ones weaken them in order to favour others. The C-LIMP approach uses two common vectors to all individuals of the population. The first one corresponds to the genes set to zero while the second one corresponds to the genes set to one. When choosing its mate, the first parent relies on its own genes to choose between the two preference vectors and evaluates the candidates' genotypes based on the chosen vector. Both preference vectors adapt at each step as with D-LIMP. These two approaches were tested on DTRAP, MAXSAT and NK Landscape, either alone and along with a Restricted Tournament Replacement operator (RTR) that helped maintaining the population's diversity. When compared to a traditional Genetic Algorithm and to a Variable Dissortative Mating Genetic Algorithm (VDMGA), D-LIMP shows better performance than the others on the MAXSAT and DTRAP problems even though it requires more evaluations to converge. On the other hand, C-LIMP along with RTR had a better performance than D-LIMP on the NK Landscape. However VD-MGA outperformed both of them on that problem. More results on D-LIMP applications were published in [48].

Eshelman et al. [31] applied an approach that relies on a Hamming distance threshold, below which recombination is inhibited. An alternative approach has been proposed by Craighurst et al. [18] where similarity between individuals is measured by their genealogical trees. Individuals sharing ancestors to a certain degree are prevented from mating with each other. Fernandes and Rosa [32] have studied the two aforementioned strategies on a Genetic Algorithms (GA) model applied to the royal road function. Results show an increase in performance on both approaches, specially if coupled with populations with varying sizes. It is also discussed that approaches showing a greater diversity along the evolutionary process don't necessarily result in a better performance despite that being the goal of both the non-random mating strategies.

Burke et al. [15] have proposed a different approach where individuals are grouped according to their lineage. During selection, two groups are randomly selected and one individual is also randomly selected from each group. This approach entirely removes the influence of fitness from the selection process and focuses on lineage alone. Lineage selection, as labelled by the authors, reportedly changes the dynamics of evolution on various domains, aiming at the promotion of diversity. On regression of Binomial-3 problems, improving diversity resulted on worst results when compared to a standard approach. The authors discuss that combining parts of dissimilar solutions doesn't always result in solutions that make sense due to nodes losing their context. They argue that converged populations have an easier task when combining genetic material from different individuals without them losing context, suggesting that there should be a balance between selection pressure and diversity handling.

Gustafson et al. [46] experimented with mating between dissimilar individuals on regression of binomial-3 instances. In the discussed work, measuring the similarity between individuals relies on the edit-distance [15, 30]. Results suggest that the search process is equally influence by unfit solutions and solutions that are both fit and dissimilar. Ultimately, the experimented approach resulted on improved solution quality.

Fernandes and Rosa [33] applied both negative and positive assortative mating to a Vector Quantization problem using a similarity measure that accounts for phenotype information regarding individuals. The approach selects both a parent and a set of mating candidates through a roulette wheel operator and the parent mates with the most similar or dissimilar candidate. Results show an enhanced performance on negative assortative mating and it is argued that diversity handling is a key factor.

2.5.2 Genetic Programming and Mate Choice

Smorokdina and Tauritz proposed a self-adapting semi-autonomous parent selection model, an approach where mates are chosen according to the first parent's own mating preferences. The first parent encodes not only his potential solution but also an evolving mate selection function in form of an extra chromosome. Represented as a Genetic Programming tree that uses as its only possible terminal the entire population with the exception of itself, the function returns the selected mating partner. As non-terminals, a set of selection operators are used, e.g., tournament selection, biggest or more uniform hamming distance, etc. Constraints were used to make sure that rules for all operators are complied. The authors consider two approaches in the attribution of mate selection functions to new offspring, either by inheritance of the first parent's mating selection function or by a recombination of both parents' selection functions. In order to choose between both, they advise measuring the improvement of the produced offspring and, done that, inherit the first parent's mating selection function if the improvement is considerably significant. This approach was tested in a set of problems and the results showed that its performance is slightly lower than the traditional approach, but without statistically significant differences. While its generalization is an advantageous aspect, the approach suffers a trade off in the obtained results

[74].

In [59], Machado and Leitão propose the self-adaptation of mate selecting evaluation functions in the Circle Packing in Squares (CPS) problem. Their approach consists in individuals with an encoded Genetic Algorithm solution for the targeted problem and an extra chromosome containing the mating preferences of each individual. Two variants were tested: the encoding and evaluation of mating preferences using Genetic Algorithms, and full evaluation functions using Genetic Programming. Initially a parent is selected using tournament selection then a set of mating candidates are chosen randomly from the population. The first parent is able to perceive phenotype information from each candidate, specifically how they perform on simpler instances of the CPS problem. Candidates phenotypes and how they perform on simpler instances of the CPS problem contain information on their genotypes that can help determine if it can contribute to the breeding of healthy offspring. The Genetic Algorithm evaluation function is modelled as a weighted function of the fitness values obtained by each candidate on smaller instances of the problem, where only the weights are evolved. Regarding Genetic Programming, the approach evolves the entire evaluation function using the fitness values on smaller instances of the problem as terminals and a set of arithmetic operators as non terminals. Obtained results show that mate selection based on Genetic Algorithm had worst performance than the traditional approach while mate selection based on Genetic Programming surpassed both conventional and Genetic Algorithm approaches, frequently with statistically significant difference even though it searches a wider search space.

2.5.3 Self-Adaptive Algorithms

Self-Adaptive dynamic adaptation relies on the theory of evolution to modify Evolutionary Algorithms parameters or operators through their evolution. In this case, the target control parameters subject to adaptation are found encoded in each individual of the population and experience recombination and mutation processes as part of the individual's genotype without influencing its fitness since they are not taken into account during evaluation processes. Considering this approach, individuals containing “better” parameter values will expectedly have evolutionary advantage over those not containing such parameters thus leading to “better” individuals that will, consequently, have more chances of survival and reproduction therefore helping the propagation of the included parameters/operators through the population. This kind of representation and evolution of parameters/operators defines the main difference between adaptive and self-adaptive dynamic adaptation. In the second approach, the methods of credit assignment and update are not explicitly design but are rather implicit in the Evolutionary Algorithm itself. The subjects of self-adaptation can go from parameter values that control the execution of Evolutionary Algorithms to operators managing selection, recombination and mutation processes, or the probability of applying single or multiple alternative methods.

The method of Sexual Selection through Mate Choice studied and developed relies on self-adaptation of each individual selected for mating. The parent selection mechanism is, then, altered in order to select the second parent biased by the first parent’s mating preferences. These mating preferences will evolve in parallel with the traditional evolutionary process imposed by the used Evolutionary Algorithm.

In [26], Eiben, Schut and de Wilde studied self-adaptation of the tournament selection size. They proceeded to include an extra parameter in each individual’s chromosome encoding a tournament size. For each selection step, a voting system is used to decide which will be the size of the tournament at that point. The mechanism was tested and the obtained results showed that improvements occurred in the evolution processes.

Spears proposes in his publication [75] a method that selects between two-point and uniform crossover in the recombination process. He suggests the encoding of an extra gene in the individual’s genotype which represents one of the crossover types and analyses possible population-level implementations and individual-level implementations.

In [13] Braught assesses the self-adaptation of mutation rates by including an extra gene in the representation that determines the mutation probability of its offspring. The offspring inherit its parents mutation rates that undergoes evolution as part of the genotype.

Chapter 3

Model/Framework Proposal

Two options stood out for potential use: OpenBeagle[38], a C++ Evolutionary Computation framework following strong principles of object oriented programming, and ECJ[1], a Java-based Evolutionary Computation Research System framework that follow the same principles as OpenBeagle but since it is written in Java, it has poorer execution speed and bigger memory footprints. Both are full featured Object Oriented Systems and are quite similar in philosophy and features.

Despite the fact that ECJ is inferior in performance when compared to OpenBeagle, it has better documentation and support as well as third-party content available such as packages for Gene Expression Programming that may reveal themselves useful. Since we are implementing a new approach that will become an extension of the original framework and want to develop it in order to be as generic as possible, we considered that a well documented framework has priority over one with better performance. For these reasons, ECJ was the chosen framework to aid on the study of Sexual Selection through Mate Choice. The framework's diagram can be found attached to the report's appendix.

When modelling Sexual Selection through Mate Choice in Evolutionary Computation, one should keep in mind certain concepts based on species' natural behaviour. Taking into account the approach tackled by Machado and Leitão[59], the most relevant concept is that, as opposed to traditional models where they are randomly paired, individuals must choose a mating partner. The choice is made according to a relation between mental preferences from the one finding its partner and sexual attractiveness from a number of mating candidates and such mental traits are heritable the same way physical traits are. Finally, mate selection is a process that induces its own selection pressure but is subject to selection pressure itself. Through generations, individuals of a population are able to evolve both their ornaments and mating preferences in a feedback loop that allows them to perceive what are good mating preferences and adapt according to such preferences.

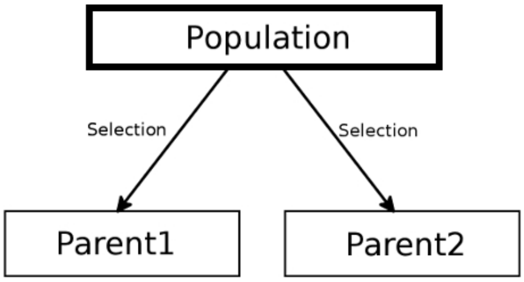


Figure 3.1: Traditional selection mechanism

In figure 3.1 the process of traditional selection mechanisms is described. To put it simply, two individuals are picked from the population through traditional selection methods and then paired with each other for mating purposes.

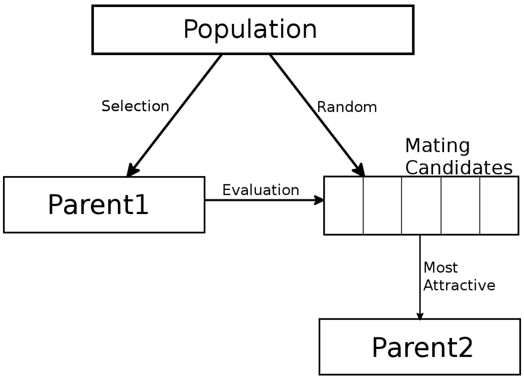


Figure 3.2: Sexual Selection mating mechanism

The mating mechanism for Sexual Selection through Mate Choice is represented by figure 3.2. Unlike the traditional mechanisms, a set of individuals from the population undergo selection via tournament and the winner is selected to be the first parent along with a set of mating candidates. Then the selected individual evaluates all the individuals of the candidate set classifying their sexual attractiveness and chooses the most attractive to pair with for mating purposes.

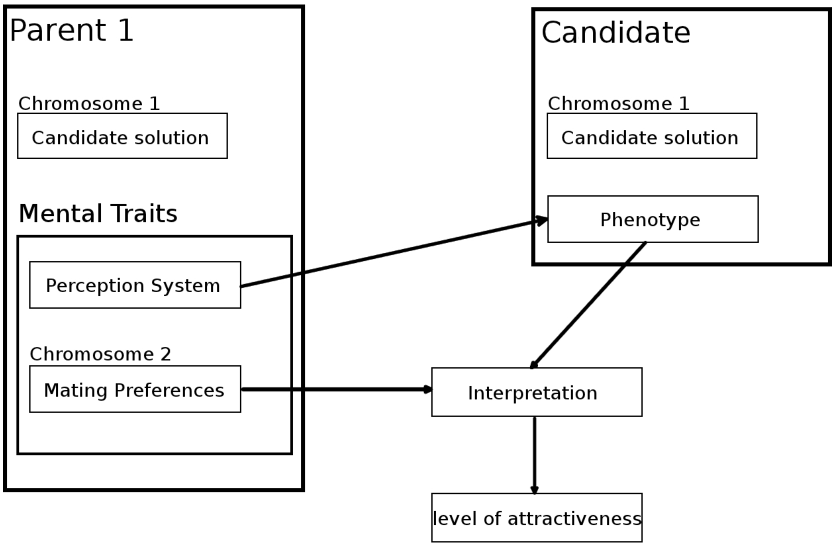


Figure 3.3: Evaluation process of mating candidates

Figure 3.3 explains how evaluation performed by the first parent works. The mechanism consists in the tournament winning individual resorting to its perception system to assess each candidate's phenotype. After that it uses an evaluation function, represented by the mating preferences in the figure, to interpret and assign the candidate's level of attractiveness.

Algorithm 4 shows a simple pseudo-code representation of the selection operator in Sexual Selection through Mate Choice.

Algorithm 4 Sexual Selection through Mate Choice

```
procedure MATECHOICE(population)
    parent_1 ← SELECTINDIVIDUAL(population)
    candidates ← SELECTCANDIDATESET(population)
    EVALUATECANDIDATES(parent_1, candidates)
    parent_2 ← SELECTBESTOF(candidates)
end procedure
```

As seen in Algorithm 4, the first parent is selected from the population by means of a chosen selection mechanism as well as a set of n candidate solutions for mating with the first parent. These are, after being selected, subjected to evaluation according to the first parent's mating preferences. This is obtained by comparing the first parent's representation of its ideal mating partner with the phenotype of each individual of the candidate set. The result will determine the attractiveness a mating candidate causes on the first parent. The second parent is selected for mating by the first which will choose the most attractive candidate of the set.

Chapter 4

First Term Experimentation

The performed work is split into two distinct phases. First the framework for Evolutionary Computation was chosen, as already stated in Chapter 3. Then the target problem was defined and Genetic Programming approaches were implemented in order to gain experience both on the framework as well as on the target problem. Afterwards, the framework was expanded to include mechanisms of Mate Choice for the target problem. Finally, several experiences were carried out on two different models, Natural Selection and Sexual Selection through Mate Choice. The evolution process and the resulting solutions were analysed for a better understanding of each model and to assess the particularities behind Mate Choice.

In order to compare the developed mechanisms for Sexual Selection through Mate Choice against the conventional existing mechanism, namely Natural Selection, both were applied in several symbolic regression functions using Genetic Programming. When benchmarking Genetic Programming problems, one should take into account several criteria to have a good benchmark producing reliable results: it should be adjustable in order to generate more difficult instances as required; it should be fast enough to allow large number of runs to be executed for meaningful comparison between approaches; its results should be easy to interpret and compare, and measures like average best fitness are statistically better than, for instance, the best fitness found. Along with easy implementation, these criteria make symbolic regression a widely used resource for benchmarking Evolutionary Algorithms [61]. The chosen method was applied for both Natural Selection and Sexual Selection through Mate Choice mechanisms.

4.1 Experimental Set-up

In order to get familiarized with the ECJ framework and Genetic Programming, several Natural Selection models were initially developed for symbolic regression on a set of 7 functions. The functions were selected based on a study regarding benchmarking of Genetic Programming by McDermott et al. [61]. The functions tackled have been compiled in table 4.1 as well as the training intervals. Each training interval states the minimum and maximum values of x as well as either the number of steps in intervals, labelled U, or the distance between steps in intervals, labelled E.

Function	**Objective Function**	**Training Set**
Keijzer-1	$0.3xsin(2\pi x)$	E[-1, 1, 0.1]
Keijzer-2	$0.3xsin(2\pi x)$	E[-2, 2, 0.1]
Keijzer-3	$0.3xsin(2\pi x)$	E[-3, 3, 0.1]
Keijzer-4	$x^3e^{-x}cos(x)sin(x)(sin^2(x)cos(x) - 1$	E[0, 10, 0.05]
Koza-1	$x^4 + x^3 + x^2 + x$	U[-1, 1, 20]
Nguyen-5	$sin(x^2)cos(x) - -1$	U[-1, 1, 20]

Table 4.1: Functions for the application of symbolic regression

Each symbolic regression function is associated with a particular function set as well as a particular terminal set used to build the GP trees. The function sets can be found in table 4.2. The terminal set is composed by the ERCs in the table and the variable x.

Function	**Function Set**	**Constants (ERC)**
Keijzer	$+ \ - \ \frac{1}{n} \ -n \ \sqrt{n}$	Random value from $(\mu = 0, \ \delta = 5)$
Koza	$+ \ - \ \times \ \div$ sin cos e^n $ln(\|n\|)$	None
Nguyen	$+ \ - \ \times \ \div$ sin cos e^n $ln(\|n\|)$	None

Table 4.2: Function and terminal set of each of the used functions for symbolic regression

4.1.1 Natural Selection Models

Having established the function set and the terminal set, the next step resides in defining the Genetic Programming parameters used on each target function. Characteristics such as the population size, the total number of generations, the minimum and maximum depth of individuals genotype defined by Grow, the maximum tree depth resulting from crossover operations and the value range of each problem's variable as well as the incremental step of that value vary from one problem to another. The parameters that remain unchanged will be the total number of runs of each problem, the probability of each node being a terminal or a non-terminal during initialization, the crossover, reproduction and mutation probabilities and the elitist proportion of the fittest individuals. Thus, the parameter set definition for each addressed problem is described in table 4.3

	Koza	**Nguyen**	**Keijzer**
Total number of runs	50	50	50
Population size	4000	500	100
Number of generations	100	100	100
Tournament size	5	5	5
Crossover probability	90.0%	90.0%	90.0%
Mutation probably	0.0%	0.0%	0.0%
Reproduction probability	10.0%	10.0%	10.0%
Elitist proportion	1	1	1
Crossover maximum depth	17	15	10
Grow minimum depth	5	6	5
Grow maximum depth	5	6	5
Non-terminal probability	90.0%	90.0%	90.0%
Terminal probability	10.0%	10.0%	10.0%

Table 4.3: Parameter set for each problem using the Natural Selection approach

4.1.2 A Mate Choice approach to Symbolic Regression

As described in chapter 3, on the model of Sexual Selection through Mate Choice, a new chromosome is implemented on each individual. This chromosome contains the individual's mating preferences and will allow it to evaluate mating candidates in order to determine their “sexual attractiveness” according it its own preferences and is represented by a Genetic Programming tree and encodes the ideal mating partner of the individual. During the process of parent selection, the first parent is selected via Tournament Selection from the population along with a random set of mating candidates. The first individual will, then, evaluate each candidate based on its mating preferences defined by the second chromosome. This evaluation is performed by calculating the quadratic error between the functions encoded in the individual’s second chromosome and each candidate’s first chromosome, selecting the one with the lowest distance error as the mating partner. Selecting the candidate with the lowest distance error means that its phenotypic information is closer to the one of an ideal mating partner for the first parent, therefore, satisfying the most of its mating preferences. The mechanisms for recombination, reproduction and mutation remain the same as the mechanisms applied for Natural Selection with the exception that they apply independently to both chromosomes.

We expect the mate choice model to be able to evolve mating evaluation functions that choose mating partners in an appropriate, meaningful way. While we hope that such an approach leads to improved results, our main concern is to assess and discuss the behaviour. Also, the test set has been selected to include only functions with one variable so that the resulting functions can easily be plotted and analysed.

The parameter set used for each addressed problem is described in table 4.4.

	Koza	**Nguyen**	**Keijzer**
Total number of runs	50	50	50
Population size	4000	500	500
Number of generations	100	100	100
First Parent Selection	Tournament	Tournament	Tournament
Tournament Size	5	5	5
Candidate Selection	Random	Random	Random
Candidate Set Size	5	5	5
Number of Chromosomes	2	2	2
Evaluating Chromosome	1	1	1
Candidate Chromosome	0	0	0
Crossover probability	90.0%	90.0%	90.0%
Reproduction probability	10.0%	10.0%	10.0%
Mutation probably	0.0%	0.0%	0.0%
Elitist proportion	1	1	1
Crossover max depth	17	15	10
Grow minimum depth	5	6	5
Grow maximum depth	5	6	5
Terminal probability	10.0%	10.0%	10.0%
Non-terminal probability	90.0%	90.0%	90.0%

Table 4.4: Parameter set for each problem using the Mate Choice approach

4.2 Experimental Results

Table 4.5 shows a comparison of the results obtained by both Natural and Sexual Selection approaches on the test set along 50 runs. The Column labelled “Standard” shows the mean best fitness value obtained using the traditional selection mechanism inspired by Darwin’s first theory, the column labelled “Mate Choice” shows the mean best fitness value of the best individuals obtained by the Sexual Selection through Mate Choice framework mentioned on Chapter 3 and the column labelled “Random” shows the mean best fitness value obtained using random parent selection mechanism on the candidate set, not regarding their fitness. The Random approach uses the same parameter values as mentioned on table 4.3 .

Function	Standard	Mate Choice	Random
Keijzer-1	0.008005462	**0.0059473756**	0.0072442644
Keijzer-2	0.0063776454	**0.0052139161**	0.0062104645
Keijzer-3	0.0071500245	**0.0056003145**	0.0067438776
Keijzer-4	0.0890397335	**0.0833904122**	0.0840754187
Koza-1	**0.0000121344**	0.0007854719	0.0000917544
Nguyen-5	0.0014892713	0.0004783439	0.0025763115

Table 4.5: Mean best fitness obtained by the Standard, mate choice and random approaches on each function over 50 runs. Since they are minimization problems, lower values are better than higher values

4.3 Analysis of the Experimental Results

A Wilcoxon-Mann-Whitney test with a significance level of 0.01 was conducted, comparing each approach with the remaining two. The instances where the proposed approach performed significantly better than the standard approach or the other way around where emphasized in table 4.5. Regarding the random approach, it performed significantly better than the mate choice approach on the Koza-1 instance but worst on Keijzer-1, Keijzer-2 and Keijzer-3. On the Nguyen instance no significant differences between the three approaches were found.

Figure 4.2 shows that the mean best fitness rapidly evolves for the standard approach for the first 5 or so generations, probably taking advantage of the initial population diversity. The Koza-1 instance is regarded as a particularly easy instance for symbolic regression[61] which may explain the results obtained, specially if the large size of the population is considered. The results suggest that the overhead created by the mate choice model does not pay off. The extra effort put on the evolution process to adapt mating preferences slows down convergence, giving the standard approach the observed advantage.

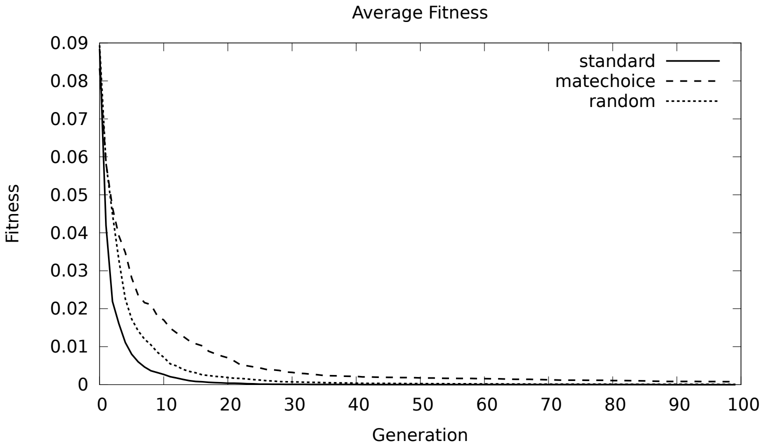


Figure 4.1: Mean best fitness obtained along 50 runs for the `Koza-1` function

Regarding the Nguyen-5 function, the three approaches performed considerably better than on the Keijzer instances. While the instance is not regarded as particularly simple for symbolic regression, the use of a population with 500 individuals may explain why the approaches were able to achieve comparable mean best fitness results. Still, the Mate Choice approach performed slightly better than the remaining two.

On the Keijzer instances, however, the Mate Choice approach achieved a significantly better mean best fitness value than the standard approach, suggesting that the proposed strategy is able to contribute to enhanced results. Despite the generated overhead on the evolution process, the approach seems capable of evolving mating preferences that favour mating partners in a way that is beneficial to the evolution process. Overall, the differences in mean best fitness values between the proposed approach and the random approach are quite noticeable, emphasizing that the behaviour produced by the Mate Choice strategy is not similar to randomly selecting mating partners, but a behaviour that produces larger benefits, therefore backing up our assumption that Mate Choice evolves mating preferences that help selecting mating partners in meaningful ways. It's also noticeable that while no significant differences where found between the standard and the random approaches, the later performs slightly but consistently better on the harder instances, suggesting that the reduced selection pressure inherent may be beneficial, allowing for a better exploration rather than exploitation of the fittest individuals.

Further analysis will focus on a single function. However, the discussed behaviours generalize to the other instances with the exception of the Koza-1, where the standard approach performs better. We have chosen to focus on the Keijzer-3 function. Figure 4.2 shows how the mean best fitness evolves along the 100 generations. The mate choice approach has a slower mean best fitness evolution on the first generations, surpassing however the standard approach at around the 10th generation.

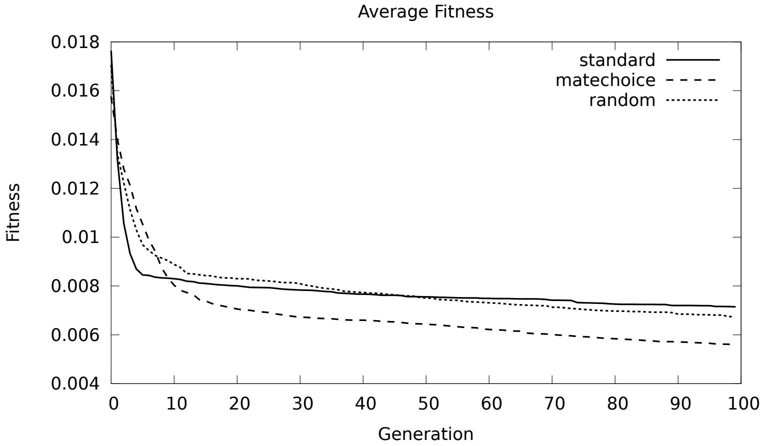


Figure 4.2: Mean best fitness obtained along 50 runs for the `Keijzer-3` function

During our preliminary analysis we found several examples that fall into one of the following three categories: individuals possess mating evaluation functions that closely reproduce the expected function; individuals possess mating evaluation functions that benefit mating partners that are likely to correct their own flaws; individuals possess mating evaluation functions that, while not being close reproductions of the expected function, are simplifications or share some of its characteristics.

4.3.1 Replication

Figures 4.3 and 4.4 show examples of individuals that hold mating evaluation functions that have evolved through the generations to closely resemble the expected function, therefore promoting mating partners that have high genetic quality.

Mating selection functions have no connection to the expected function and evolve without any knowledge of it. Individuals adapt to each other and their mating preferences adapt to other individuals in order to select mating partners containing candidate solutions that resembles the target function to be optimized through a feedback loop that hasn't, in any way, knowledge of the target function. They evolve without any knowledge while the first chromosome is evaluated on how well it correlates with the expected function through the fitness function and therefore adapts to the environment. Mating evaluation function adapts to other individuals through a feedback loop where good mating preferences are more likely to select fitter individuals and therefore produce better offspring that contribute to the spreading of good mating preferences and physical characteristics.

The fact that mating evaluation functions are able to evolve toward resembling the target function suggests that mating preferences are able to perceive what characteristics are beneficial and correspond to good survival abilities and evolve towards promoting good fitness indicators and corresponding mating candidates.

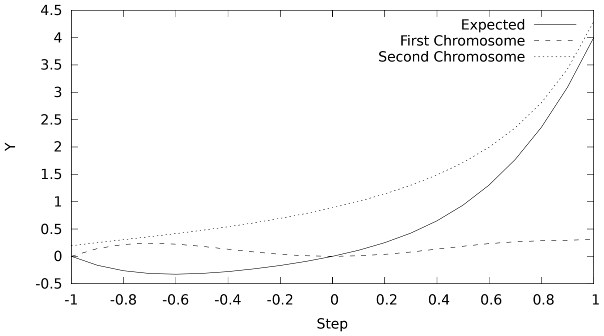


Figure 4.3: Representation of an evolved individual for the `Koza-1` function

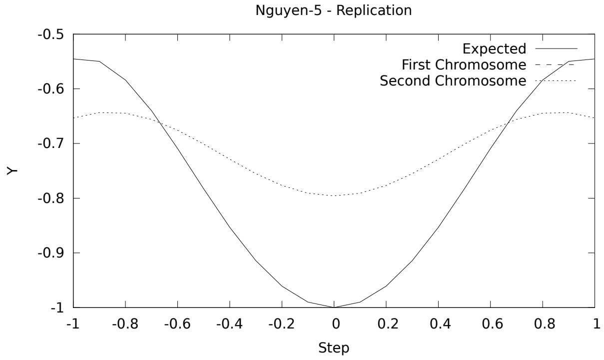


Figure 4.4: Representation of an evolved individual for the `Nguyen-5` function

4.3.2 Correction

Figures 4.5 through 4.9 show individuals that own mating evaluation functions that while not reproducing the target function, make an attempt to compensate for the flaws shown in their candidate solutions. By promoting the selection of mating partners that are similar to the mating evaluation function, individuals actively make an effort to mate with individuals whose genetic material may contribute to healthier offspring that help the perpetuation of the individuals' genes. Unlike the previous examples, these mating evaluation functions don't correspond directly to an adaptation towards particularly good fitness indicators but rather toward the needs of the individuals. Through the feedback loop, mating preferences show an aptitude that promote individuals that, while lacking mating evaluation functions that help mating with particularly good partners, promote mating with individuals that show promise to correct their own flaws and therefore contribute to their own reproductive success.

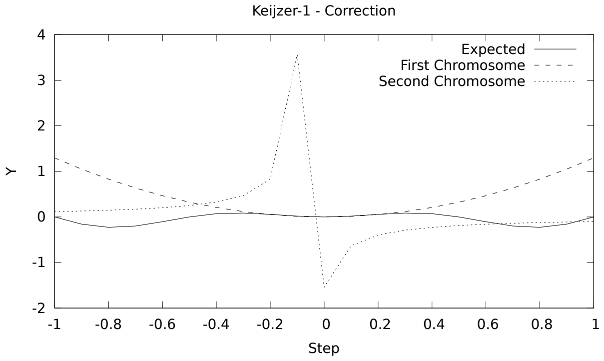


Figure 4.5: Representation of an evolved individual for the `Keijzer-1` function

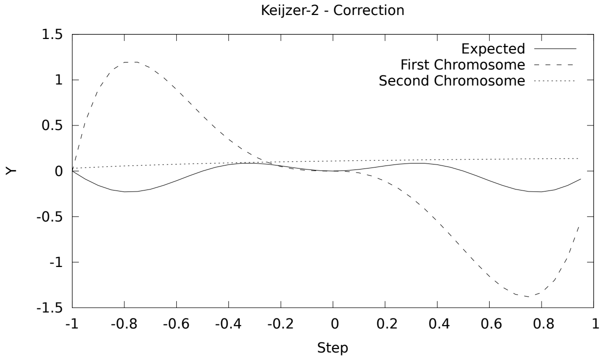


Figure 4.6: Representation of an evolved individual for the `Keijzer-2` function

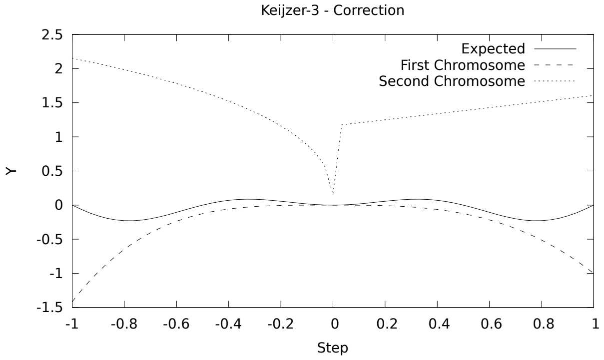


Figure 4.7: Representation of an evolved individual for the `Keijzer-3` function

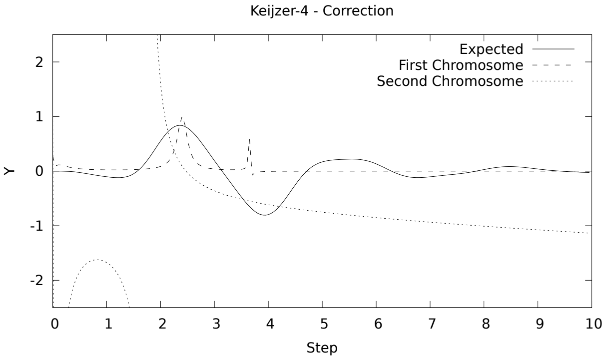


Figure 4.8: Representation of an evolved individual for the `Keijzer-4` function

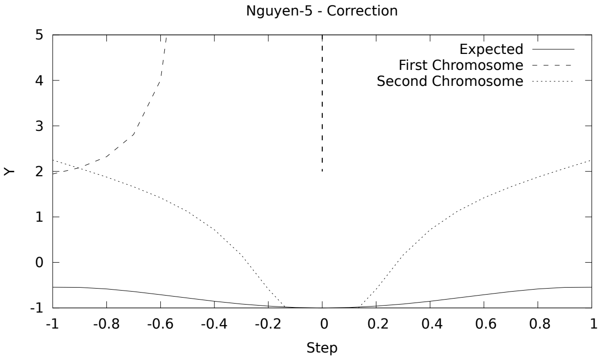


Figure 4.9: Representation of an evolved individual for the `Nguyen-5` function

4.3.3 Simplification

Finally, figures 4.10 through 4.12 show individuals that own mating evaluation functions that, while unable to reproduce the expected function, show an approximation and share some of its characteristics. On the one hand, these mating evaluation functions may be seen as an earlier stage of the effect seen in Section 4.3.1. On the other hand, these functions may show that a close reproduction of the expected result is unnecessary and that a simplification may serve the same purpose.

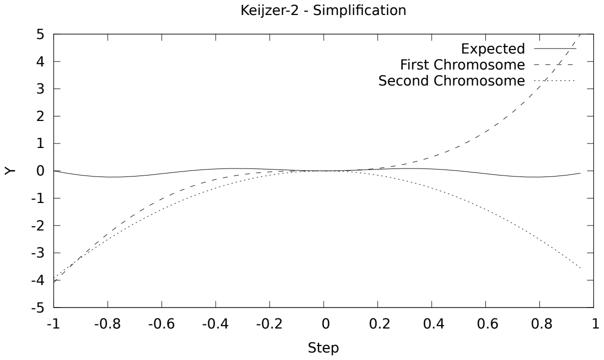


Figure 4.10: Representation of an evolved individual for the `Keijzer-2` function

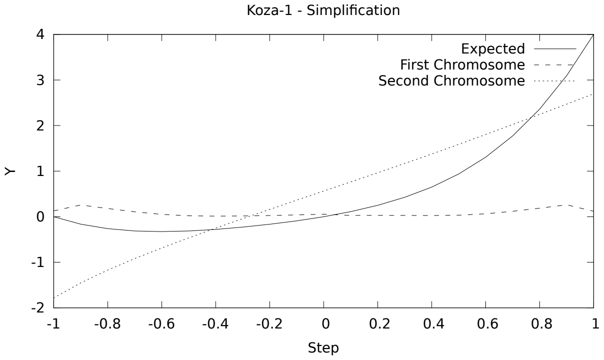


Figure 4.11: Representation of an evolved individual for the `Koza-1` function

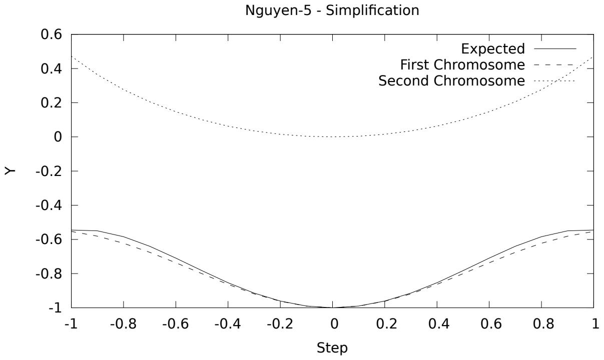


Figure 4.12: Representation of an evolved individual for `Nguyen-5` function

Chapter 5

Improvements over Original Proposal

The ECJ framework as it doesn't support tool to easily implement and experiment with Mate Choice. In order to integrate the proposed model, described in Chapter 3, in the existing framework, several adaptations were made. The purpose of the this chapter is to list and describe the limitations encountered during this process and the methods developed to overcome those limitations. In Section 5.2 as well as present the resulting class diagram of the model implementation and the source of most relevant classes.

5.1 Workings of ECJ on Natural Selection and Genetic Programming

To better understand the Mate Choice framework implemented, we will briefly describe how ECJ works when running a standard evolutionary process. Before the evolutionary process starts, an object of the *Population* class is created which, in turn, has an array of *Subpopulation* objects. Each of these will contain a list of individuals that compose the population of the evolutionary algorithm. In the experiments conducted only one subpopulation was needed.

The *Initializer* class initializes a population which, in turn, initializes a subpopulation. A length and a *Species* class object is assigned to the subpopulation. The object contains an *Individual* prototype – which contains an *Individual* prototype. The sub-population is filled by calling the `newIndividual()` method from *Species* which returns individual clones(), differing in the genetic information they contain. Since we are dealing with Genetic Programming, the classes for that purpose are *GPInitializer*, *GPSpecies* and *GPIndividual*.

It is after the initialization that the evolutionary process begins. From this point on, the population will evolve throughout generations according to a set of mechanisms and variations that will allow the creation of new individuals, the *offspring*, that will partially or totally replace those in the original population.

At the beginning of each *generation*, individuals from the population undergo selection to become part of the *mating pool*. This process is performed using the *SelectionMethod* class, an abstract class that provides a base instantiation of selection methods such as *TournamentSelection*. Selection subclasses contain the implemented methods that perform selection of individuals from the population and their addition to the mating pool.

The procedures that follow selection consist in applying variation to the selected parents. This is achieved by a number of subclasses of the abstract class *BreedingPipeline*: *CrossoverPipeline*, *MutationPipeline* and *ReproductionPipeline*. They are applied in exclusion with a given probability. The application of such operators will result in a set of newly created individuals based on the genetic information of their parents. Note that every time a variation operator is applied, a clone of each individual undergoing variation is made in order to maintain the original individuals unaltered. The operator will, then, act directly on the created clone. *Elitism* is also supported and is applied previous to selection and variation and simply clones n individuals to the new generation

The final step of the evolutionary process for each iteration is to determine the individuals that will pass on to the next generation. If the Evolutionary Algorithm adopts a generational model, the offspring will replace the entire population in the next generation (except for individuals passed by elitism) and the parents are discarded. If a steady-state model is adopted by the algorithm, a survival selection mechanism is applied and the offspring will replace part of the individuals in the population. This is typically a deterministic process biased by fitness. The parents and the offspring are sorted from best to worst individual and a fraction of the parents is discarded. The survivor selection process ends with the best offspring of the set being chosen fill the empty positions of the sub-population until it reaches its original size.

5.2 Limitations and Their Overcoming

One of the main concern of implementing Sexual Selection through Mate Choice was doing so while keeping the original ECJ classes and class hierarchy unaltered. Doing so represented one of the biggest challenges but would allow the framework to keep all of its features such as multiple representations (Genetic Algorithms, Genetic Programming, etc.), Island Models, Steady State and Generational evolution and many others. The Mate Choice model was designed so that it could be coupled with any of these pre-existing features. The solution found to overcome this limitation was to create a new set of classes and build a package that will connect to the ECJ framework by relations described along the chapter. During the next subsections we will refer specifically to the GP approach to symbolic regression but the methods support other representations

5.2.1 Package `ec.ss.gp`

First of all, the model needed an object that could store not only the genetic information of a Genetic Programming individual but also its mating preferences. To this end, an extension to the *Individual* class was created, named *MultiSpeciesIndividual*, which works the same way but contains the appropriate data structures to accommodate multiple chromosomes rather than one. This type of individual belongs to a specific species created as an extension to the *Species* class, called *MultiSpeciesSpecies*, responsible for the initialization of the initial individual.

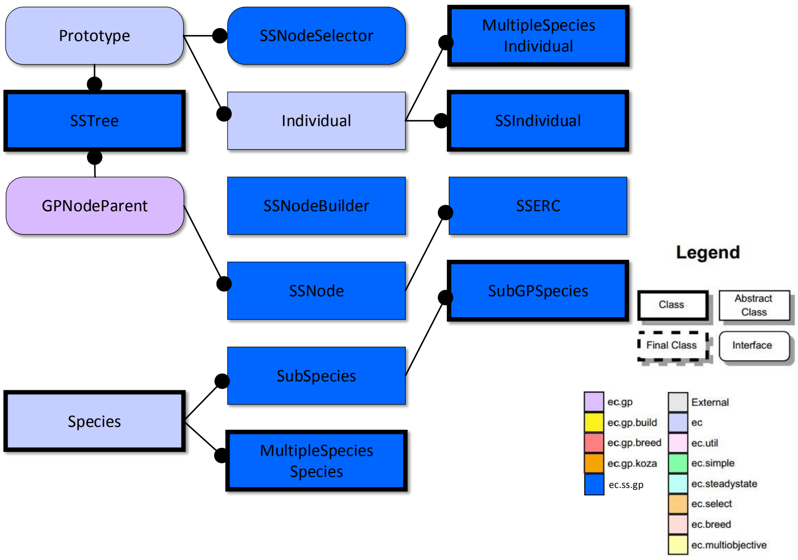


Figure 5.1: Class diagram for the `ec.ss.gp` package

Listing 5.1 contains the Java source code with information on the most relevant variables and methods from *MultiSpeciesIndividual*. The size of `chromosomes` is user-defined and can store the candidate solution and mating preferences on independent chromosomes. Despite the fact that `chromosomes` may have length greater than two, only two elements of the array are used throughout the evolutionary process.

```
public class MultiSpeciesIndividual extends Individual {
  public Individual[] chromosomes = null;
                         ⋮
}
```

Listing 5.1: Relevant variables of MultiSpeciesIndividual class

The process of initialization of a *MultiSpeciesIndividual* object consists, itself, in initializing the data structures that compose such objects both from *Individual* class along with the specific data structures of the initialized object. The `clone()` method of the same class, represented in Listing 5.2, simply creates a clone object of the *Individual* class with the whole content of a *MultiSpeciesIndividual* object. This method is called either while filling the population at the initial stage of the evolutionary process or while applying variation operators.

```
public class MultiSpeciesIndividual extends Individual {
                        ⋮
  public Object clone() {
    MultiSpeciesIndividual myobj = (MultiSpeciesIndividual)
        super.clone();
    if (chromosomes != null) {
      myobj.chromosomes = new Individual[chromosomes.length];
      for (int i = 0; i < chromosomes.length; ++i) {
        if (chromosomes[i] != null) {
          myobj.chromosomes[i] = (Individual) chromosomes[i].clone();
        }
      }
    }
    return myobj;
  }
                        ⋮
}
```

Listing 5.2: Method clone() of MultiSpeciesIndividual class

The initialization of a *MultiSpeciesIndividual* through *MultiSpeciesSpecies* class is shown in Listings 5.3, 5.4 and 5.5. Since a *MultiSpeciesIndividual* object is composed of n sub-*Individual* objects (identified as chromosomes), they need to be initialized as well. For that purpose, two array objects were defined in *MultiSpeciesSpecies* class: `species` and `initializers`. For each chromosome object on *MultiSpeciesIndividual*, a corresponding species and initializer are assigned. This will allow the *MultiSpeciesIndividual* to accommodate chromosomes from different species or with different specifications. The algorithm then proceeds to the creation of each chromosome that composes a *MultiSpeciesIndividual* object by changing the active initializer to the initializer of the corresponding chromosome to be initialized. Before this process begins, the `mainInitializer` object stores the algorithm initializer responsible for the creation of *MultiSpeciesIndividual* objects so that it can be restored when all chromosomes have been initialized. After it is complete, the algorithm initializer is restored to its original state.

```java
public class MultiSpeciesSpecies extends Species {
  int numChromos;
  public SubSpecies[] species;
  public Initializer[] initializers;
  public Initializer mainInitializer;
                    ⋮
}
```

Listing 5.3: Relevant variables of MultiSpeciesSpecies class

Listing 5.4 presents the Java source code responsible for the set-up of a *MultiSpeciesIndividual* object. The most relevant aspect of the `setup()` method is the length definition of each array object in the class by assigning a value to the variable `numChromos` through a user-defined parameter. That done, the *Initializer* array and *SubSpecies* instantiation is carried to start the initialization of each chromosome. The instantiation of the *MultiSpeciesIndividual* is performed only after the instantiation of each array objects in *MultiSpeciesSpecies*.

```java
public class MultiSpeciesSpecies extends Species {
                   ⋮
  public void setup(final EvolutionState state, final Parameter base) {
    mainInitializer = state.initializer;
    numChromos = state.parameters.getInt(base.push("num-chromosomes"),
        def.push("num-chromosomes"));
    species = new SubSpecies[numChromos];
    for(int i = 0; i < numChromos; ++i){
      species[i] = (SubSpecies)
          state.parameters.getInstanceForParameter(base.push("" +
          i).push("species"), def.push("" + i).push("species"),
          SubSpecies.class);
    }
    initializers = new Initializer[numChromos];
    for (int i = 0; i < numChromos; ++i) {
      initializers[i] = (Initializer)
          state.parameters.getInstanceForParameter(base.push("" +
          i).push("species").push("init"),def.push("" +
          i).push("species").push("init"), Initializer.class);
      state.initializer = initializers[i];
      initializers[i].setup(state, base.push(i + ".species.init"));
      species[i].setup(state, base.push(i + ".species"));
    }
    state.initializer = mainInitializer;
    super.setup(state, base);
  }
                   ⋮
}
```

Listing 5.4: Method setup() of MultiSpeciesSpecies class

The method `newIndividual()` from the *MultispeciesSpecies* class, presented in Listing 5.5, deals with the creation of *MultiSpeciesIndividual* objects by initializing each *Individual* object in the `chromosomes` array through configuration of the corresponding *Initializer* object.

```java
public class MultiSpeciesSpecies extends Species {
                    ⋮
  public Individual newIndividual(final EvolutionState state, int
      thread){
    MultiSpeciesIndividual newind =
        (MultiSpeciesIndividual)(super.newIndividual(state, thread));
    newind.chromosomes = new Individual[numChromos];
    for(int i = 0; i<numChromos; ++i){
      state.initializer = initializers[i];
      newind.chromosomes[i] = species[i].newIndividual(state, thread);
    }
    state.initializer = mainInitializer;
    return newind;
  }
                    ⋮
}
```

Listing 5.5: Creation of a MultiSpeciesIndividual object on MultiSpeciesSpecies class

Listing 5.6 shows the Java source code of the method `newIndividual()` called by *MultiSpeciesSpecies* for the creation of *SSIndividual* objects stored in the `chromosome` array of *MultiSpeciesIndividual*. The instantiation of new *SSIndividual* objects is performed based on the corresponding *SSInitializer* which extends from the *SimpleInitializer* class which, in turn, extends from the *Initializer* class. *SSInitializer* contains the function and terminal set nodes, and the tree and node constraints configuration, used to build the genetic tree stored in the *SSIndividual* object. These configurations are specific for the problem at hand, meaning that both the *SSindividual* and the *SSInitializer* are specific for symbolic regression and given the Mate Choice model used, are similar for both the first and the second `chromosome` of each *MultiSpeciesIndividual*.

```
public class SubGPSpecies extends SubSpecies{
                        ⋮
  public Individual newIndividual (EvolutionState state, int thread){
    SSIndividual newind = ((SSIndividual) (i_prototype)).lightClone();
    for (int x = 0;x < newind.trees.length;x++)
      newind.trees[x].buildTree(state, thread);
    newind.fitness = (Fitness) (f_prototype.clone());
    newind.evaluated = false;
    newind.species = this;
    return newind;
  }
                        ⋮
}
```

Listing 5.6: Creation of a SSIndividual object on SubGPSpecies class

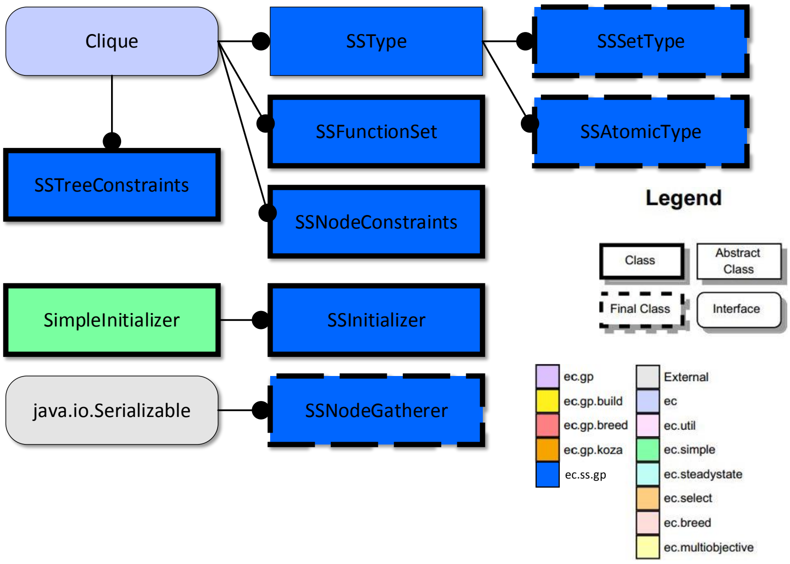


Figure 5.2: Class diagram for the `ec.ss.gp` package

5.2.2 Package `ec.ss.gp.koza`

The classes of this package have the same functionalities as the original classes with minor alterations in order to support Sexual Selection. They could almost be seen as a copy of the original Genetic Programming specific classes but solve some faults that didn't allow the use of Genetic Programming representation on Mate Choice models. This way, the original `koza` package can be used for non Mate Choice models while the new ones add support for Mate Choice.

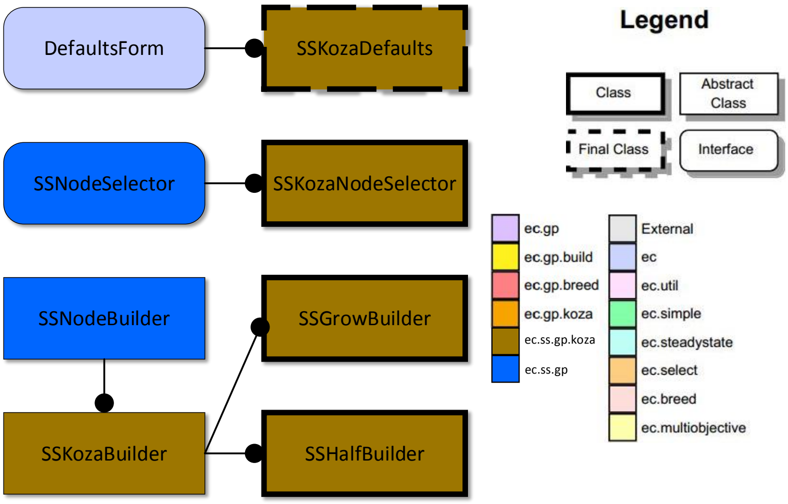


Figure 5.3: Class diagram for the `ec.ss.gp.koza` package

5.2.3 Package `ec.ss.breed`

After both the individuals and their species are defined, it was necessary to develop the Sexual Selection through Mate Choice model for the framework. This model is implemented in the *SexualSelectionPipeline*, which will deal with the selection of the second parent for mating based on the mating preferences of the first parent. The *ReproductionPipeline* and *SexualSelectionBreeder* classes are variant classes of *ReproductionPipeline* from package `ec.breed` and *SimpleBreeder* from package `ec.simple` respectively, adapted to our Mate Choice model.

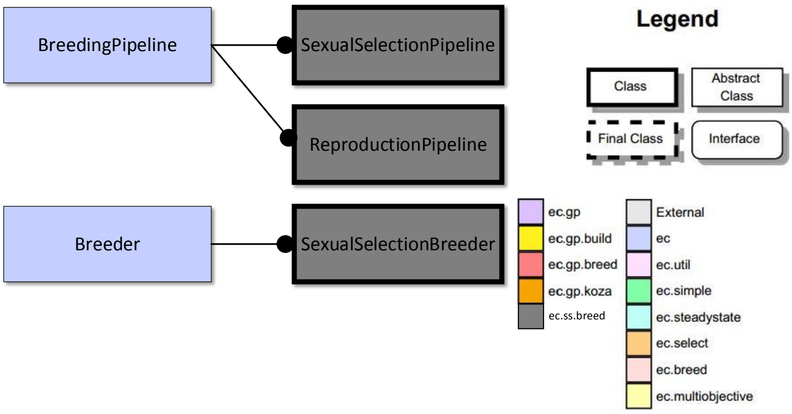


Figure 5.4: Class diagram for the `ec.ss.breed` package

Listings 5.7 and 5.8 present the most relevant elements of our Sexual Selection through Mate Choice model. Listing 5.7 contains the variables needed for individuals to perform mate choice. `evalchromo` corresponds to the index of the *MultiSpeciesIndividual* chromosome array that will be used as mating preferences to evaluate the candidate partners stored in `candidates` array. In our specific case, the second `chromosome` represents mating preferences through an ideal mating partner.

```
public class SexualSelectionPipeline extends BreedingPipeline {
  public int evalChromo;
  public int candChromo;
  Individual candidates[];
                    ⋮
}
```

Listing 5.7: Most relevant variables of SexualSelectionPipeline class

Listing 5.8 shows the `evaluate()` method call that will determine the level of attractiveness that a mating candidate induces on the individual assessing it. The next step is simply to verify if the assessed candidate arouses a better sexual attraction than the previous candidates, placing it as the best mating partner found until that point.

```java
public class SexualSelectionPipeline extends BreedingPipeline {
                   ⋮
  public int produce(final int min, final int max, final int start,
      final int subpopulation, final Individual[] inds, final
      EvolutionState state, final int thread) {
                     ⋮
    Individual bestIndividual = null;
    double bestScore = 0;
    int bestIndex = -1;
    for (int i = 0; i < candidates.length; i++) {
      ((FitnessFunctionKeijzer1)
          state.evaluator.p_problem).evaluate(state,
          (MultiSpeciesIndividual) parent, (MultiSpeciesIndividual)
          candidates[i], thread, evalChromo, candChromo);
                       ⋮
      if (bestIndividual == null) {
        bestIndividual = candidates[i];
        bestIndex = i;
        bestScore = ((MultiSpeciesIndividual)
            parent).chromosomes[evalChromo].fitness.fitness();
      } else if (pickWorst && ((MultiSpeciesIndividual)
          parent).chromosomes[evalChromo].fitness.fitness() <
          bestScore) {
        bestIndividual = candidates[i];
        bestIndex = i;
        bestScore = ((MultiSpeciesIndividual)
            parent).chromosomes[evalChromo].fitness.fitness();
      } else if (!pickWorst && ((MultiSpeciesIndividual)
          parent).chromosomes[evalChromo].fitness.fitness() >
          bestScore) {
        bestIndividual = candidates[i];
        bestIndex = i;
        bestScore = ((MultiSpeciesIndividual)
            parent).chromosomes[evalChromo].fitness.fitness();
      }
    }
    inds[q] = bestIndividual;
    q++;
    if (q < n + start) {
      inds[q] = parent;
      q++;
    }
  }
  return n;
                   ⋮
}
```

Listing 5.8: Method produce() of SexualSelectionPipeline

5.2.4 Package `ec.ss.gp.breed`

This package contains the classes corresponding to the variation operators, *GPCrossoverPipeline* and *GPMutationPipeline*. The *ReproductionPipeline* does not suffer any changes. Both are variant classes of *CrossoverPipeline* and *MutationPipeline* respectively, from package the `ec.gp.koza` and adapted in order to execute the variation operators on *MultiSpeciesIndividual* objects, maintaining the original classes for Natural Selection models.

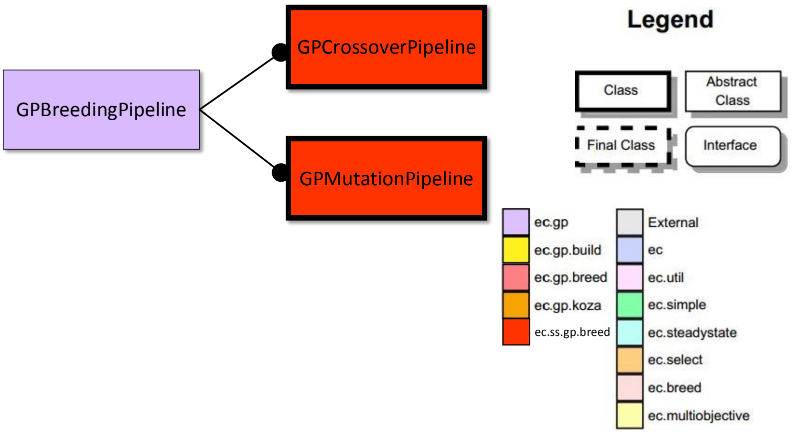


Figure 5.5: Class diagram for the `ec.ss.gp.breed` package

5.2.5 Package `ec.ss.select`

This package contains the selection method used to select the first parent via tournament. This selection mechanism works similarly to the existing tournament selection mechanism of ECJ, *TournamentSelection* from package `ec.select`, but altered in order to support *MultiSpeciesIndividual*.

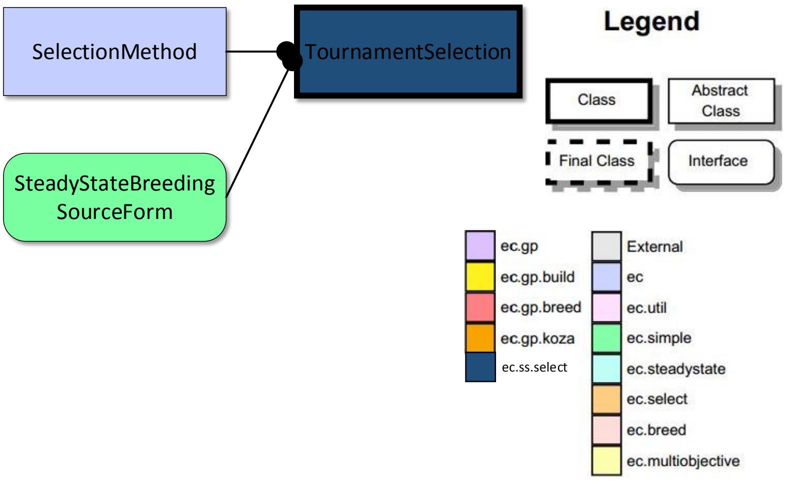


Figure 5.6: Class diagram for the `ec.ss.select` package

5.2.6 Package `ec.ss`

This package contains the class *SexualShortStatistics*, responsible for the storage of the population information. This class is a variation of the class *SimpleShortStatistics*, from the package `ec.simple`, and adapted to our Mate Choice Model. There were new functionalities implemented to store the information of the population, allowing a better observation of the evolutionary process. One of the functionalities implemented was the storage of all the chromosomes of each individual in order to better analyse the evolution of each individual mating preferences. The other implemented functionality consists in storing information of the crossover types – destructive, neutral and constructive – occurred during each generation as well as the sum of each type along generations.

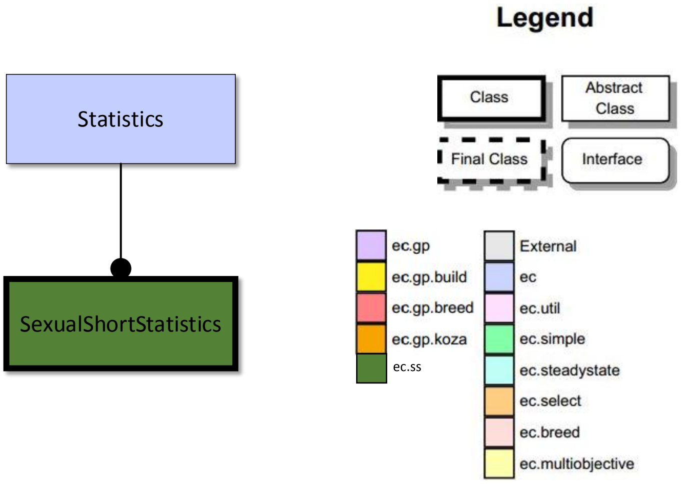


Figure 5.7: Class diagram for the `ec.ss` package

5.3 Parameters Configuration

ECJ is configured according to a set of parameters that will define the evolutionary process, for instance, individuals representation, selection and variation methos, and survival mechanisms. Each method has its own configuration, thus, a corresponding set of parameters. To implement the Sexual Selection through Mate Choice model, a particular set of parameters was added in order to set up the algorithm execution. This subsection will describe the most relevant parameters that will allow the implementation and execution of the model. During the description of the parameters set, we will assign pre-defined values for reasons of simplification and they will be mentioned as parameters are described.

5.3.1 Parameters related to the Individual

To implement Mate Choice according to the proposed model, the corresponding individual and its characteristics must be explicitly defined by parameters. Assuming that there is only one population in the evolutionary process and individuals can possess two chromosomes, we will describe the set of parameters used for the configuration of the model implemented below. Further information about standard ECJ framework configuration can be found in [1].

```
pop.subpop.0.species = ec.ss.gp.MultiSpeciesSpecies
pop.subpop.0.species.ind = ec.ss.gp.MultiSpeciesIndividual
pop.subpop.0.species.num-chromosomes = 2
```

By doing so, we are now allowed to define each chromosome's characteristics independently, in particular, its class and species. The next parameter set will describe the first and second chromosome's definition:

```
pop.subpop.0.species.0.species = ec.ss.gp.SubGPSpecies
pop.subpop.0.species.0.species.ind = ec.ss.gp.SSIndividual
pop.subpop.0.species.1.species = ec.ss.gp.SubGPSpecies
pop.subpop.0.species.1.species.ind = ec.ss.gp.SSIndividual
```

The individual characteristics such as number of trees, tree class, tree constraints and fitness are defined the same way as for a *GPIndividual.* Initializing each chromosome is made simply by defining the *Initializer* to the corresponding chromosome:

```
pop.subpop.0.species.0.species.init = ec.ss.gp.SSInitializer
pop.subpop.0.species.1.species.init = ec.ss.gp.SSInitializer
```

SSInitializer properties are also defined in a way similar to the original class of ECJ framework , *GPInitializer.*

5.3.2 Parameters related to Selection

The next set of parameters define the configuration of the mating selection process.

```
sexual-selection.evaluating-chromosome = 1
sexual-selection.candidate-chromosome = 0
sexual-selection.num-sources = 2
```

```
sexual-selection.source.0 = ec.ss.selection.TournamentSelection
sexual-selection.source.0.chromosome = 0
sexual-selection.source.0.pick-worst = true
sexual-selection.source.0.size = 5
sexual-selection.source.1 = ec.select.RandomSelection
sexual-selection.source.1.n-candidates = 5
```

In this set, we define the chromosome of the *MultiSpeciesIndividual* responsible for the evaluation of mating candidates with `evaluating-chromosome` and each candidate chromosome subjected to evaluation with `candidade-chromosome`. Next, we designate the number of sources used for parent selection with `num-sources`. The first source corresponds to the selection of the first parent and the second source to the selection of a set of mating candidates that will undergo evaluation from the previously selected parent.

5.3.3 Parameters related to Variation

The parametrization of the variation operators became rather extensive with the implementation of the Mate Choice model. Figure 5.8 represents a diagram with the connections between variation operators classes for the proposed model.

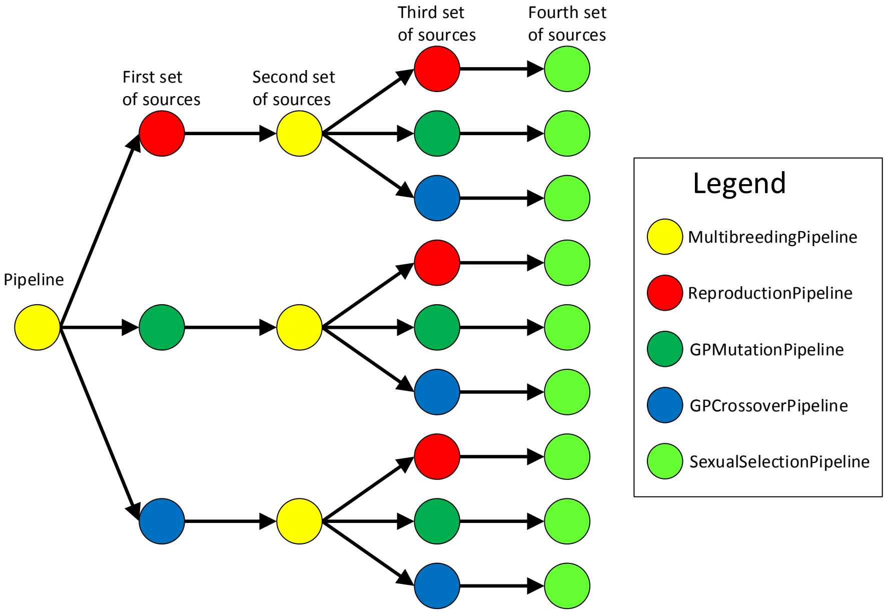


Figure 5.8: Pipeline organization and connection for variation operators

In order to set up the variation operators, we need to specify the complete path to the source we are configuring. Given that, if we want to configure an element of one of the sources we would have the prefix `pop.subpop.K.species.pipe` related to the population, the index – in our case, 0 – of the sub-population, the species and the pipeline added to every configuration parameter of the source. The resulting parameter would be the prefix plus `source.X.source.Y.source.Z` where `X` corresponds to operators of the first set, `Y` to one of the second, `W` to one of the third and `Z` to one of the forth. If, for instance, we wanted to configure the properties of an operator from the fourth set, the resulting parameter without the class properties would be, `pop.subpop.K.species.pipe.source.X.source.Y.source.W.source.Z`.

To overcome the extension of such parameters, a new parameter model was designed for the purpose of shortening them. Taken into account that they followed the pattern mentioned above, we opted to omit every word of the parameter and keep the numbers only. We, then, obtained the string `K.X.Y.W.Z` where `K` corresponds to the index of the sub-population, `X` to the index of the first set of sources, `Y` to the index of the second set of sources, `W` to the index of the third set of sources and `Z` to the index of the fourth set of sources. The configuration of the variation operator is handled after the last digit. With this parameter model, the configuration set to set up the variation operators became clearly shorter than before.

However, while the parameter set became shorter and easier to handle it also became easier to cause confusion. With this in mind, we developed a way to identify which variation operator is being configured. For the configuration parameters related to the crossover operator, we added the prefix `gp.sexualselection.xover`, and for the configuration parameters related to the mutation operator, we added the prefix `gp.sexualselection.mutate`. The resulting parameters are `gp.sexualselection.xover.K.X.Y.W` and `gp.sexualselection.mutate.K.X.Y.Z` for crossover and mutation, respectively. Note that the source corresponding to the `Z` element is not on the resulting parameters due to the fact that the variation operators are only on the first and third set, so the parameter configuration for them goes as far as the third digit. For instance, if we want to configure the chromosome used for crossover operations of the third set, the resulting parameter will be `gp.sexualselection.xover.0.0.0.1.chromosome = 1`.

Chapter 6

Second Term Experimentation

After performing the improvements on the proposed model mentioned on Chapter 5, a new set of symbolic regression functions were configured, executed and their results were obtained for further analysis. We start the chapter by presenting the functions used for those tests in Section 6.1 as well as their corresponding configuration. Section 6.2 will contain the results obtained for each function regarding the performance and crossover behaviour. We end the chapter with Section 6.3 where we will debate about the obtained results for both performances.

6.1 Experimental Setup

In addition to the functions mentioned in the Table 4.1 from Section 4.1, we choose six functions based, as well, on the study regarding benchmarking of Genetic Programming by McDermott et al.[61]. Like mentioned in the section related to the first semester experiments, the training set states the minimum and maximum values of x as well as the number of steps to be performed for U labeled intervals or the distance between steps for E labeled intervals.

Function	**Objective Function**	**Training Set**
Keijzer-8	$\sqrt{x}$	E[0, 100, 1]
Keijzer-9	$\text{arcsinh}(x)$	E[0, 100, 1]
Korns-4	$-2.3 + 0.13 * \sin(x)$	U[-50, 50, 10000]
Korns-5	$3 + 2.13 * \log(x)$	U[-50, 50, 10000]
Nguyen-6	$\sin(x) + \sin(x + x^2)$	U[-1, 1, 20]
Nguyen-7	$\ln(x + 1) + \ln(x^2 + 1)$	U[-1, 1, 20]

Table 6.1: New functions for the application of symbolic regression

Each objective functions is associated with a set of mathematical functions and a set of terminal variables and constants. Both define the nodes that will compose the trees. The function and terminal sets for the *Keijzer* and *Nguyen* functions are already defined in the Table 4.2 on Chapter 4. For the remaining objective function, the *Korns*, the function set and terminal set used to build the tree is as follows:

Function	**Function Set**	**Constants (ERC)**
Korns	$+\ -\ \times\ \div\ \sin\ \cos\ e^n$ $\ln(\lvert n\rvert)\ n^2\ n^3\ \sqrt{n}\ \tan\ \tanh$	Random finite 64-bit IEEE double

Table 6.2: Function and terminal set for the *Korns* objective function

The next step consists in defining the parameters for the execution of the Evolutionary Algorithm used on each function. Tables 6.3 and 6.4 show the configuration of the algorithms for the Natural Selection and the Mate Choice approaches, respectively.

	Keijzer	**Korns**	**Nguyen**
Number of runs	100	100	100
Population size	500	100	500
Number of generations	100	100	100
First parent selection	Tournament	Tournament	Tournament
Tournament size	5	5	5
Second parent selection	Tournament	Tournament	Tournament
Tournament size	5	5	5
Crossover probability	90%	90%	90%
Mutation probability	0%	0%	0%
Reproduction probability	10%	10%	10%
Elitist proportion	1	1	1
Crossover maximum depth	10	17	15
Ramped half-and-half probability	50%	50%	50%
Grow minimum depth	5	5	6
Grow maximum depth	5	5	6
Non-terminal probability	90%	90%	90%
Terminal probability	10%	10%	10%

Table 6.3: Configuration set for each objective function using Natural Selection

	Keijzer	**Korns**	**Nguyen**
Number of runs	100	100	100
Population size	100	100	100
Number of generations	100	100	100
First parent selection	Tournament	Tournament	Tournament
Tournament size	5	5	5
Candidate selection	Random	Random	Random
Candidate set size	5	5	5
Number of chromosomes	2	2	2
Evaluating chromosome	1	1	1
Candidate chromosome	0	0	0
Crossover probability	90%	90%	90%
Mutation probability	0%	0%	0%
Reproduction probability	10%	10%	10%
Elitist proportion	1	1	1
Crossover maximum depth	10	17	15
Ramped half-and-half probability	50%	50%	50%
Grow minimum depth	5	5	6
Grow maximum depth	5	5	6
Non-terminal probability	90%	90%	90%
Terminal probability	10%	10%	10%

Table 6.4: Configuration set for each objective function using Mate Choice

Description of the parameter set for the Natural Selection model can be found in Subsection 4.1.1 as well as description of the parameter set for the Mate Choice model in Subsection 4.1.2.

6.2 Experimental Results

In this section, we can find the relevant results obtained during the execution of each objective function using the standard Natural Selection approach, the proposed Sexual Selection through Mate Choice approach and the Random Selection approach which consist in selecting, for crossover, the first parent via tournament selection and the second parent via random selection.

Function	Standard	Mate Choice	Random
Keijzer-8	$1.40129846432e^{-45}$	**$1.4e^{-45}$**	$1.40129846432e^{-45}$
Keijzer-9	0.2256588004	**0.0305635135**	0.0597596787
Korns-4	0.0315580546	**0.0239037031**	0.0310657165
Korns-5	4.7735830338	4.3397756109	**2.7512731444**
Nguyen-6	0.0181183084	0.0047443458	0.0061174161
Nguyen-7	0.0195324862	0.0110878289	**0.0066977619**

Table 6.5: Mean best fitness values along 50 runs

Tables 6.5 contains the mean best fitness values of 50 runs for each function. Note that the columns labelled "Standard" contain the mean best fitness values obtained by the Natural Selection model. The columns labelled "Mate Choice" correspond to the mean fitness values using the Sexual Selection through Mate Choice model. Lastly the columns labelled "Random" relate to the mean fitness values using Random Selection.

Function	Standard	Mate Choice	Random
Keijzer-8	61.74	**694.4**	74.92
Keijzer-9	602.72	**749.26**	765.1
Korns-4	662.52	768.28	705.08
Korns-5	952.94	879.34	1014.84
Nguyen-6	905.68	836.7	948.8
Nguyen-7	727.58	814.88	792.04

Table 6.6: Mean cumulative constructive crossover values along 50 runs

As for Table 6.6, it contains the mean constructive crossover obtained from the 50 runs executed for each new objective function. The mean cumulative constructive crossover of previous experiments is not presented as that analysis was not performed at that time. We consider a constructive crossover to be the resulting offspring of a crossover operation possessing a better fitness value than the first parent selected for genetic recombination.

Function	**Standard**	**Mate Choice**	**Random**
Keijzer-1	3124.6	**1760.94**	2524.54
Keijzer-2	3278.84	**1708.12**	2060.58
Keijzer-3	2852.96	**1741.4**	2187.86
Keijzer-4	2997.26	**1528.56**	2169.32
Koza-1	8666.42	8016.44	7397.52
Nguyen-5	17749.86	**10583.78**	15592.58

Table 6.7: Mean cumulative neutral crossover values along 50 runs for functions on Table 4.1

Function	**Standard**	**Mate Choice**	**Random**
Keijzer-8	7358.86	**1494.42**	6369.06
Keijzer-9	3309.06	**1390.84**	1692.68
Korns-4	3118.86	**1495.54**	2200.08
Korns-5	1955.82	**1220.28**	1015
Nguyen-6	1690.6	**1218.58**	1157.06
Nguyen-7	2554.12	**1304.96**	1607.08

Table 6.8: Mean cumulative neutral crossover values along 50 runs

In Tables 6.7 and 6.8, we show the mean cumulative neutral crossover performed along 50 runs for the previous optimization of the objective functions shown in the Table 4.1 from Chapter 4 and the mean cumulative neutral crossover performed on the new test suite along 50 runs. We consider a neutral crossover to be the resulting offspring of a crossover operation possessing the same fitness value as the first parent selected for genetic recombination.

Function	**Standard**	**Mate Choice**	**Random**
Keijzer-1	5847.6	**7102.22**	6336.2
Keijzer-2	5793.92	**7039.2**	6370.88
Keijzer-3	5738.58	**6830.3**	6375.36
Keijzer-4	5726.4	**6615.66**	6191.24
Koza-1	57737.565	**68469.805**	62787.615
Nguyen-5	28539.18	**33254.02**	26258.88

Table 6.9: Mean cumulative destructive crossover values along 50 runs for functions on Table 4.1

Function	**Standard**	**Mate Choice**	**Random**
Keijzer-8	1861.66	**6636.62**	2846.08
Keijzer-9	5462.38	**6764.44**	6923.7
Korns-4	5596.4	**6630.38**	6468.5
Korns-5	6471.32	**6802.38**	7347.42
Nguyen-6	6775.28	**6856.78**	7277.88
Nguyen-7	6091.98	**6781.14**	6981.46

Table 6.10: Mean cumulative destructive crossover values along 50 runs

Lastly, Tables 6.9 and 6.10 contain the mean destructive crossover of the 50 runs executed for each objective function mentioned in Tables 4.1 and 6.1, respectively. For destructive crossover we reckon it to be the resulting offspring of a crossover operation possessing a worst fitness value than the first parent selected for genetic recombination.

6.3 Analysis of the Experimental Results

We compared the obtained mean fitness results shown in Table 6.5 by pergorming a Wilcoxon Mann Whitney test with significance level of 0.01. The instances where the proposed model performed significantly better than the Standard approach or vice-versa were emphasized. Regarding the performance of the approaches, we note that Mate Choice performed better for the Keijzer-8, Keijzer-9 and Korns-4. As for Nguyen-6, while neither the Mate Choice nor the Random approaches performed significantly better than the standard approach, there is a statistically significant difference between them. This emphasizes that the Mate Choice approach doesn't behave similarly to randomly selecting the second parent but rather is able to evolve mating preferences in ways that are beneficial to the evolutionary process.

Despite the differences between approaches, none of them presents significant differences with the Standard approach which means that its mean best fitness is possibly being impaired by some runs where the best solution found has very low fitness value. This aspect may suggest that the Standard approach becomes stuck in an local optima in some of the runs. By introducing stochasticity, the Mate Choice and the Random approaches provide a better search with lower probability of becoming stuck in a local optima.

The Wilcoxon-Mann-Whitney test was also conducted on the reported mean cumulative values of each crossover type, presented in tables 6.6 through 6.10, also with a significance level of 0.01. Instances where the Mate Choice approach achieved a significantly higher mean cumulative number of destructive crossovers or a significantly smaller mean cumulative number of neutral crossovers than the Standard approach are depicted in bold. In Table 6.6 we can observe that, for the objective functions Keijzer-8 and Keijzer-9, Mate Choice performs a significantly higher number of constructive crossovers than the other two approaches. Since Keijzer-8 function is very easy to optimize, the Standard approach rapidly converges and stops performing constructive or destructive crossovers as the population diversity becomes very low due to highly fit individuals taking over the population. As a result, the mean cumulative number of neutral crossovers is very high. The Random approach also eventually converges, even though it performs a slightly greater number of constructive and destructive crossovers than the Standard approach, mainly caused by its greater stochasticity. Still, the approach also results on a high number of neutral crossovers that result from pairing similar individuals. Mate Choice behaves distinctly different from the other two approaches by maintaining a lower number of mean cumulative neutral crossovers and greater number of mean cumulative constructive and destructive crossover. This proves the approach capability in maintaining diverse, non-converged population, allowing a wider exploration of the search space, which ultimately results in a significantly better mean best fitness.

On a more general analysis, there is clearly an advantage in maintaining a low amount of neutral crossovers. Mate Choice always produces better mean best fitness results than the Standard approach and consistently maintains the number of mean cumulative neutral crossovers significantly lower than the Standard, even when the mean best fitness is not significantly lower. In functions where the Random approach obtained a significantly better mean best fitness than the Standard approach, the number of mean cumulative neutral crossover also proved to be significantly lower than the number obtained by the other. For Nguyen-6 results, there are no significant differences.

As the result of a low number of cumulative neutral crossover values, the mate choice approach performs a high number of cumulative destructive crossovers which allows a better exploration due to the possibility of low fitted individuals disseminating their genetic information to the next generation instead allowing the genes of highly fit individuals to take over. In the three instances where Mate Choice obtained significantly better results, the number of mean cumulative destructive crossovers was also significantly greater. Also, in instances with functions where the Random approach produced significantly better results, the number of mean cumulative destructive crossovers, for Mate Choice, was also significantly higher.

Tables 6.7 and 6.9 show cumulative values for neutral and destructive crossovers on the functions assessed in Chapter 4. They suggest a similar behaviour as that discussed above. A brief perusal of Tables 6.7 and 6.9 reveals that, on this test suite, Mate Choice consistently results in a higher mean cumulative number of destructive crossovers as well. Conversely, it attains lower mean cumulative number of neutral crossovers on all instances where it outperformed the other approaches. On the Koza-1 instance, probably due both to its large population and simplicity, the approach that performed the smallest cumulative number of destructive crossovers was able to outperform the others. In this case, candidate solutions are likely to evolve faster and further towards the target function by means of successful crossovers while diversity is assured by the large number of individuals in the population which keeps the process from drasticly increasing the number of neutral crossovers.

Although correlation does no imply causality, these results suggest that the explanation for the success of the Mate Choice approach may rest on the reduction of neutral crossover. Individuals that result from neutral crossovers make no contribution to fitness enhancements and may ultimately stall evolution and decrease diversity. The higher mean cumulative number of destructive crossovers obtained by Mate Choice may be explained by the stochasticity it introduces in the mate selection process. Additionally, Mate Choice is outperforming other approaches and, as such, the results are inherently more difficult to improve.

Further analysis will focus on a single function. However, the discussed behaviours generalize to the other instances with the exception of the Koza-1, where the Standard approach performs better. On the Korns-5, Nguyen-6 and Nguyen-7, the behaviour of the random approach may not share all the behaviours described as they achieve better mean best fitness values and higher mean cumulative values of destructive crossover. We have chosen to focus on the Keijzer-3 function. Figure 2 shows how the mean best fitness evolves along the 100 generations while Figures 6.1 and 6.2 show the mean of destructive crossovers and the mean of neutral crossovers at each generation and how they evolve along the runs.

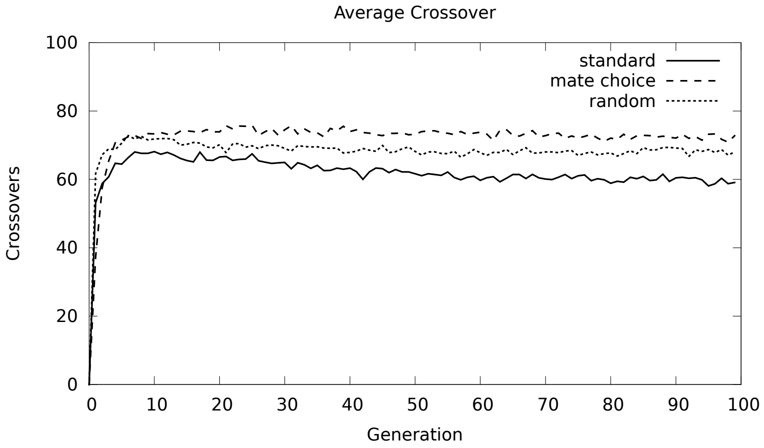


Figure 6.1: Mean of destructive crossovers obtained along 50 runs for the `Keijzer-3` function

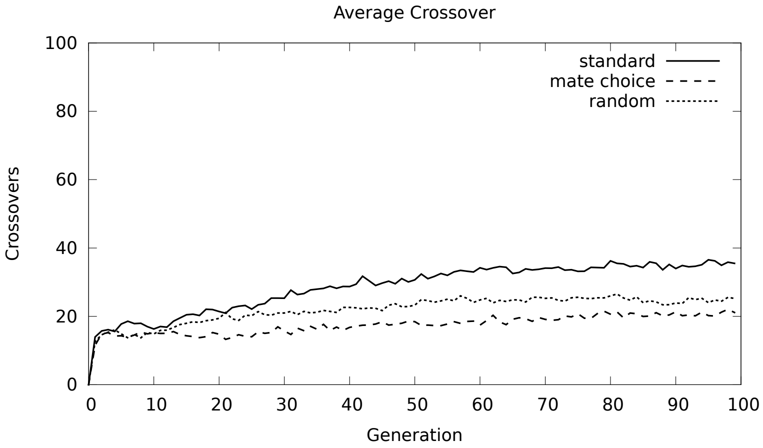


Figure 6.2: Mean of neutral crossovers obtained along 50 runs for the `Keijzer-3` function

Figure 4.2 from Chapter 4 shows that the mean best fitness rapidly evolves for the Standard approach for the first 5 or so generations, probably taking advantage of the initial population diversity. Figure 6.1 shows that for this period, the mean of destructive crossovers also raises rapidly, stabilizes and then gradually descends for the remaining of the generations. The behaviour suggests that, from this point on, the Standard approach promotes crossover between fit solutions. Individuals have a greater chance of promoting their genetic material by mating with fit partners, those who are unable to do so are gradually discarded and no longer contribute to evolution. This behaviour is supported by Figure 6.2 which shows that the Standard approach produces a gradually higher mean of neutral crossovers. This is consistent with a decreasingly diverse population where a small number of individuals takes over. As a result, while fitter individuals gradually mate with other fit individuals, their impact on the evolution process is reduced and advances are made slowly as individuals exploit a smaller set of genes.

The Mate Choice approach, as shown by Figure 4.2 and Table 4.5, from Chapter 4, and by Table 6.9, a slower mean best fitness evolution on the first generations, surpassing however the Standard approach at around the 10th generation. Figure 6.1 shows that by generation 10, the Mate Choice approach is producing more destructive crossovers than the Standard approach.

While mating between fit and similar solutions is less likely to cause destructive crossovers, mating with unfit or dissimilar individuals is more likely to produce destructive crossovers as genes mix either with worst genes or with genes from different contexts. The Mate Choice approach seems to promote this behaviour along the runs, causing the mean of destructive crossovers to remain steady for the remainder of the generations. However, such a behaviour has a positive impact on the mean best fitness, likely the result of a better exploration through the contribution of genes from less fit or dissimilar individuals. Figure 6.2 supports this behaviour as a smaller mean of neutral crossovers is observed on the Mate Choice approach throughout the entire run. Thus, in the considered experimental settings, the Mate Choice strategy promotes risktaking resulting in higher mean cumulative destructive crossovers and lower mean cumulative neutral crossovers and taking risks is beneficial in the most complex problem instances.

When comparing the Mate Choice approach with the Random one, both regarding the mean of destructive crossovers and the mean of neutral crossovers, it is noticeable that the produced behaviours happen with a different frequency, indicating that the evolution of mate preferences has an impact on the behaviour of the algorithm. The results reported in both Tables 4.5 and 6.9, and the analysis of Figure 4.2 show that such an impact is positive for the performance of the evolutionary process.

Chapter 7

Conclusion

The field of Evolutionary Computation is a relatively recent research field where, just as in the Evolution Theory, new groundbreaking theories and ideas are constantly devised and experimented. While in many research fields, the theory of Sexual Selection has been acknowledged, its practical effects on Evolutionary Computation are yet to be fully perceived. During the performed study, we explore Mate Choice, a nature inspired selection mechanism biased by mating preferences, and analyse the phenomenon and its effects on Evolutionary Algorithms. An introduction to Evolutionary Computation is presented with emphasis on Genetic Programming, the adopted approach for this study. A review of the background of Sexual Selection and particularly Mate Choice is also included, followed by the studies and applications of the approach on Evolutionary Computation.

A self-adaptive Mate Choice approach is proposed based on three nature inspired rules: individuals choose who they mate with based on their perception of others and their own mating preferences; mating preferences are inherited similarly to physical traits; mate selection introduces its own mating selection pressure but is subject to selection pressure itself.

An extra chromosome is introduced, allowing the encoding of the mating preferences of each individual. The way it is used to assess mating candidates before selecting the most desirable are presented. The mechanisms and operators by which mating preferences are inherited and undergo evolution are also discussed. A test suite is introduced using a functional prototype of the proposed model along with experimentation results. Three approaches are compared: Standard, Mate Choice and Random.

The obtained results show that the Mate Choice approach is able to significantly outperform the other two on complex instances of Symbolic Regression. Analysis on the evolved mating preferences also show interesting behaviours that may allow selected partner to complement the imperfections of the selecting parent.

The improvements on the prototype model to a fully fledged version are presented using a set of diagrams representing the relations between classes followed by implementation details and description.

The second test suite with a new set of objective functions is performed and its results are shown revealing a good performance of the proposed approach in most of the objective functions of the test suite. These results are now accompanied by a discussion on the behaviour of the proposed model compared to the Standard and Random approaches, showing that the Mate Choice approach is capable of evolving mating preferences that choose mating partners in meaningful ways, allowing them to actively contribute to an enhanced performance by the Evolutionary Algorithm on the target function.

The results described in section 4.3 of Chapter 4 and in Table 6.7 and 6.9 from Chapter 6 resulted in the paper "A self-adaptive Mate Choice model for Symbolic Regression" [56] published and presented at the IEEE Congress on Evolutionary Computation 2013. Due to time constraints and to the incompatibility between the delivery date of the dissertation and the submission deadlines of the conferences we consider strategic, it was impossible, so far, to publish, and therefore, disseminate the near results. We foresee the submission of a paper communicating these results to the european Conference on Genetic Programming 2014. Additionally the writing of an encompassing article for the Evolutionary Computation Journal is already underway.

Future work may include applying the proposed model to a larger function set, implementation of multiple Evolutionary Algorithms representations, and the study of genealogical trees with emphasis on the interpretation of the evolution of mating preferences and its effects on the evolution of candidate solutions.

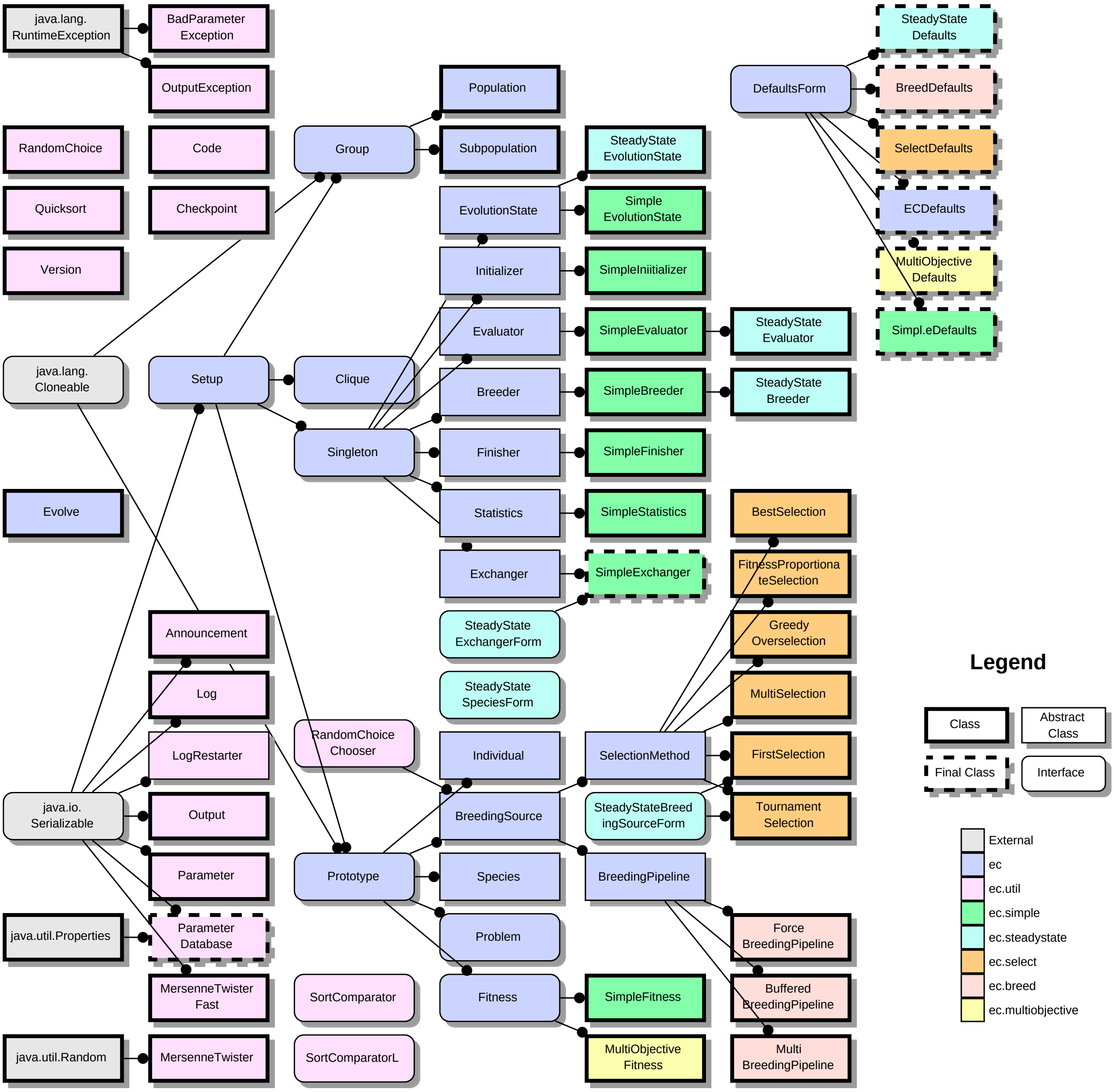
Appendix A

ECJ Framework

ECJ Evolutionary Computation System

http://www.cs.umd.edu/users/seanl/ec/ecj/

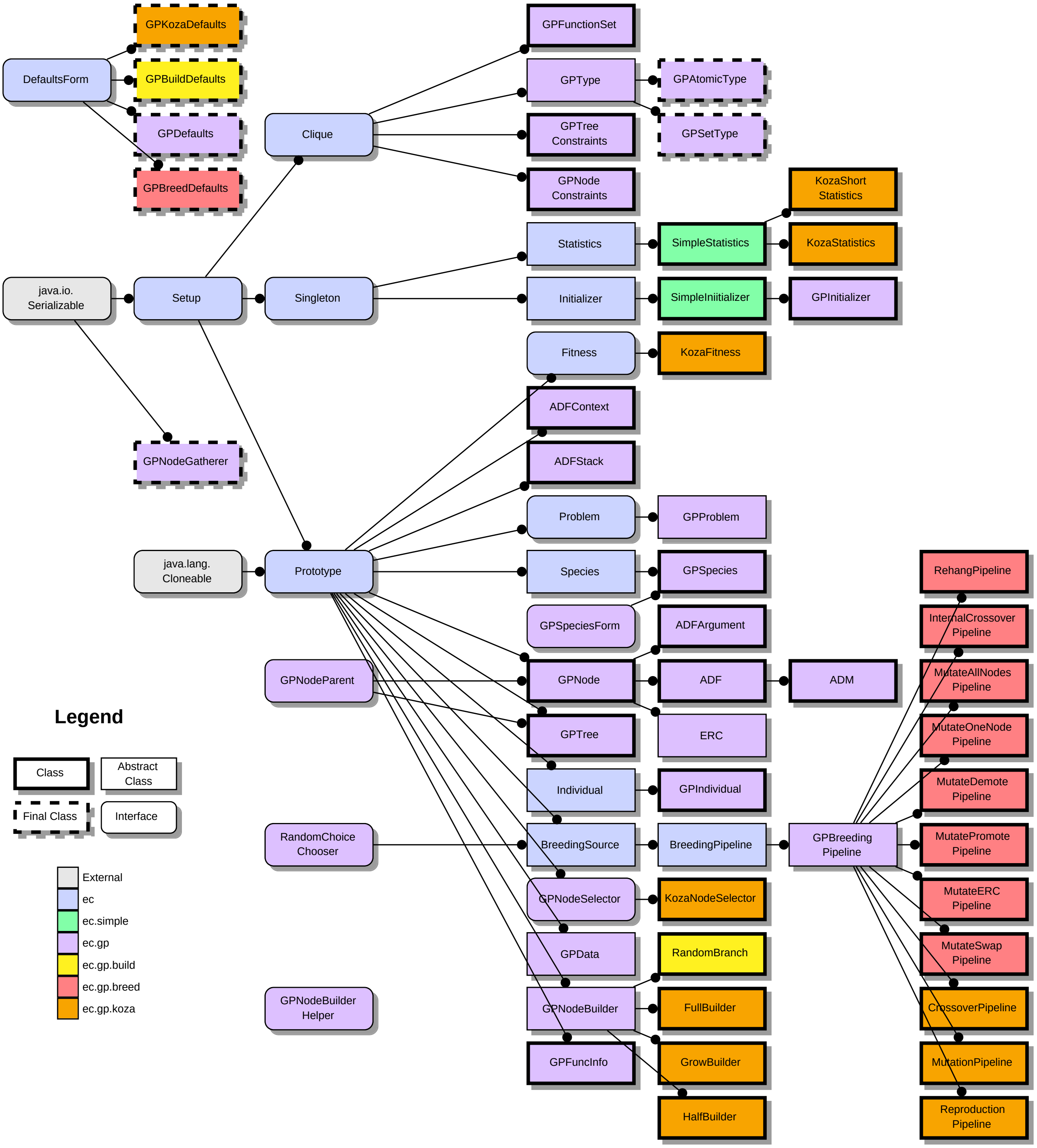
Page 1: all packages except for the ec.gp and ec.app package hierarchies
Packages: ec, ec.util, ec.simple, ec.steadystate, ec.breed, ec.multiobjective



ECJ Evolutionary Computation System

http://www.cs.umd.edu/users/seanl/ec/ecj/

Page 2: The ec.gp package hierarchy and parent classes
Packages: ec.gp, ec.gp.breed, ec.gp.build, ec.gp.koza



Appendix B

“A Self-adaptive Mate Choice Model for Symbolic Regression”

A Self-adaptive Mate Choice Model for Symbolic Regression

António Leitão
CISUC
Department of Informatics Engineering
University of Coimbra
3030-290 Coimbra, Portugal
Email: apleitao@dei.uc.pt

José Carlos Neves
CISUC
Department of Informatics Engineering
University of Coimbra
3030-199 Coimbra, Portugal
Email: jcneves@student.dei.uc.pt

Penousal Machado
CISUC
Department of Informatics Engineering
University of Coimbra
3030-290 Coimbra, Portugal
Email: machado@dei.uc.pt

Abstract—**Sexual Selection through Mate Choice has for the past few decades attracted the attention of many researchers from different fields. Numerous contributions and supporting evidence for the role and impact of Sexual Selection through Mate Choice in Evolution have emerged since then. Just like Evolutionary Theory has had to adapt its models to account for Sexual Selection through Mate Choice and its effects, it is relevant to study and analyse the impact that Mate Choice may have on Evolutionary Algorithms.**

In this study we describe a nature inspired self-adaptive Mate Choice approach designed to tackle Symbolic Regression problems. Results on a set of test functions are presented and compared to a standard approach, showing that Mate Choice is able to contribute to enhanced results on complex instances of Symbolic Regression. Also, the resulting behaviours are contrasted and discussed, suggesting that Mate Choice is able to evolve Mating evaluation functions that are able to select partners in meaningful and valuable ways.

I. INTRODUCTION

Darwin's theory of Natural Selection has for long been widely accepted by the scientific community. Described by Darwin as the result of competition within or between species affecting their individuals relative rates of survival [1], Natural Selection has found its way into many research fields and is utterly relevant in Evolutionary Computation (EC). Given a population where genetic variance occurs, individuals with favorable characteristics have a higher rate of survival and are more likely to spread their genes through future generations by means of heritability while unfavored characteristics and corresponding genes are gradually discarded.

In the years after his masterpiece on Natural Selection, Darwin put much effort on developing his theory of Sexual Selection, a force capable of shaping complex traits and behaviours across the species. He described it as the result of competition within species affecting its individuals relative rate of reproduction [2], but unlike Natural Selection, this theory found little acceptance at the time. Interest rose however in the 1970s mostly due to Zahavis works [3] and previous contributions by Fisher [4], [5]. With the emergence of new fields of research such as Evolutionary Anthropology [6] and Evolutionary Psychology [7], aided by modern knowledge and technology, new ideas and supporting evidence have contributed to a much wider acceptance of Sexual Selection as playing a major role in evolution.

Darwin has proposed two main processes composing Sexual Selection: Male Competition and Female Mate Choice. The latter is the psychological process by which individuals choose their mating partners based on their perception of others and mating preferences and is the scope of this study. Just like Evolutionary Theory, EC is an ongoing research field, where new ideas are constantly introduced and experimented with. The same way that Evolutionary Theory has had to adapt its evolutionary models to account for Sexual Selection through Mate choice, it is relevant to study the impact and inherent effects that it may have on Evolutionary Algorithms (EA).

Studying Sexual Selection through Mate Choice in EC presents a number of challenges, the most prominent being the difficulty of modeling nature-inspired Mate Choice mechanisms as well as analysing its effects on EAs. The process depends on individuals' ornamentation, perception of others and mating preferences that evolve similarly to physical traits [4], which is not straightforward to model. Also, unlike Natural Selection based models, individuals adapt not only to an environment but also in relation to each other through the intrinsic relation between mating preferences and displayed ornaments or characteristics [5]. This presents a new paradigm that is challenging to analyse.

In this research we propose a self-adaptive [8] Mate Choice approach to Symbolic Regression. The study aims at analysing the ability of the nature-inspired model to evolve mating evaluation functions that help choose mating partners in meaningful and beneficial ways. The performance of the proposed strategy is discussed and the impact on the algorithm's behaviour is addressed with a particular interest on destructive crossover.

The following section introduces Sexual Selection through Mate Choice, including a review of the theory's background and of related work and implementations on Evolutionary Algorithms. A Mate Choice approach for symbolic regression is then introduced and implementation details addressed. Section III describes the experimental setup and results are analysed and discussed. Conclusions are drawn in Sec. IV and future work presented.

II. SEXUAL SELECTION THROUGH MATE CHOICE

The subject of Sexual Selection through Mate Choice is introduced on the following subsections. Firstly, sec. II-A covers theoretical background on Sexual Selection through Mate Choice. Secondly, previous research on Mate Choice in EC is presented in sec. II-B. Finally, sec. II-C discusses our approach to symbolic regression.

A. Background

Ever since publishing his theory of Natural Selection, Darwin was interested in the origin of animal ornamentation and courtship behaviours, briefly addressing the subject in his masterpiece *On the Origin of Species* [1] and later extensively discussing the subject in *The Descent of Man, and Selection in Relation to Sex* [2]. Darwin's intrigue was that Natural Selection could not explain the emergence of these costly and complex traits that don't seem to contribute to the individuals survival ability. To some extent they seemed counterproductive to Natural Selection. Nonetheless, Darwin knew that in order for ornaments and courtship behaviours to emerge and spread they needed to bring some kind of evolutionary advantage to individuals. In order to account for such traits, Darwin envisioned the theory of Sexual Selection, a trait-shaping selection feature capable of evolving complex traits that bring reproductive rather than survival advantages.

While Natural Selection adapts species to their environment [1], Sexual Selection adapts individuals in relation to others in a struggle of individuals of one sex for access to individuals of the other [2]. In this scenario, failure means that the individuals will produce few or no offspring. While such individuals may have strong survival abilities, from an evolutionary perspective, individuals that have a small reproductive success are akin to individuals with weak survival abilities as their genes are less likely to spread through future generations. Darwin's ideas on Sexual Selection showed that adaption occurs not only due to differences in survival rates but also from differences in reproductive success. They also show how psychological traits can shape physical traits through Mate Choice.

The theory of Sexual Selection imposes that individual's reproductive success is determined by how attractive they are to others. Such a paradigm includes cases where individuals with poor survival abilities may attain a high reproductive success because they display phenotypic characteristics that are favored by mating preferences and the other way around. The role of mating preferences is therefore of great relevance. Fisher [4], [5] has made important contributions to better understand the relation between mating preferences and evolved traits. He suggested that ornaments have evolved as indicators of fitness and that individual whose mating preferences favour ornaments associated to highly fit individuals will have an evolutionary advantage as they will select fitter mating partners, helping produce fitter offspring and contributing to the spread of their genetic material [4]. Fisher also proposed, among other contributions, that mating preferences are heritable as part of the genotype and are therefore subject to evolution in a similar way as physical traits [5].

The aforementioned characteristics suggest an intrinsic and deep dependence between both mating preferences and evolved physical traits. Fisher described the relation as a positive-feedback loop [5], an arms race where both mating preferences evolve in relation to exhibited ornaments and ornaments adapt according to existing mating preferences. In a nutshell, features that result on a higher survival rate will bring evolutionary advantage to individuals whose mating preferences favour them, which results on the reinforcement and spreading of both ornaments and mating preferences. The opposite may also occur, with ornaments adapting to popular mating preferences in order to bring reproductive success to individuals, resulting on the reinforcement of both ornaments and mating preferences [5].

The feedback loop described above makes Sexual Selection through Mate Choice an extremely difficult subject to analyse. Moreover, Zahavi introduced the *handicap principle* [3] which suggests that not all ornaments act as honest indicators of fitness. Examples such as the peacock's tail or the Irish elk's antlers show that ornaments may work as handicaps. Only individuals with a high fitness are able to maintain such costly ornaments, indicating good genes. A variety of behaviours can result from Sexual Selection through Mate Choice, with sometimes unexpected results that are challenging to analyse and interpret.

B. Related Work

The increasing interest of researchers on Sexual Selection through Mate Choice has spread its impact to various research fields. While on EC the number of publications on the subject is still reduced, there are a number of contributions regarding the design and implementation of Sexual Selection through Mate Choice that are worth reviewing. We are especially interested in models that follow three nature-inspired rules:

1) individuals must choose who they mate with based on their perception of others and on their own mating preferences;
2) mating preferences are heritable the same way as physical traits;
3) mate selection introduces its own selection pressure but is subject to selection pressure itself [9];

Several publications fall into this scope but others have given important contributions as well. The remainder of this section covers relevant contributions to the present work.

Eshelman et al. [10] applied an approach that relies on a Hamming distance threshold, below which recombination is inhibited. An alternative approach has been proposed by Craighurst et al. [11] where similarity between individuals is measured by their genealogical trees. Individuals sharing ancestors to a certain degree are prevented from mating with each other. Fernandes and Rosa [12] have studied the two aforementioned strategies on a Genetic Algorithms (GA) model applied to the royal road function. Results show an increase in performance on both approaches, specially if coupled with populations with varying sizes. It is also discussed that approaches showing a greater diversity along the evolutionary process don't necessarily result in a better performance despite that being the goal of both the non-random mating strategies.

Burke et al. [13] have proposed a different approach where individuals are grouped according to their lineage. During selection, two groups are randomly selected and one individual is also randomly selected from each group. This approach entirely removes the influence of fitness from the selection process and focuses on lineage alone. Lineage selection, as labeled by the authors, reportedly changes the dynamics of evolution on various domains, aiming at the promotion of diversity. On regression of Binomial-3 problems, improving diversity resulted on worst results when compared to a standard approach. The authors discuss that combining parts of dissimilar solutions doesn't always result in solutions that make sense due to nodes losing their context. They argue that converged populations have an easier task when combining genetic material from different individuals without them losing context, suggesting that there should be a balance between selection pressure and diversity handling.

Vrajitoru [14] proposed a scheme where individuals are grouped into four genders: self-fertilizing, hermaphrodite, female or male. Each gender is associated with its own mating preferences, i.e. what groups they may mate with. Two approaches are compared, either with or without fitness-proportionate selection. It is discussed that natural selection plays a role on determining each individual's reproduction mode as dominant individuals will promote their gender through future generations.

Gustafson et al. [15] experimented with mating between dissimilar individuals on regression of binomial-3 instances. In the discussed work, measuring the similarity between individuals relies on the edit-distance [13], [16]. Results suggest that the search process is equally influenced by unfit solutions and solutions that are both fit and dissimilar. Ultimately, the experimented approach resulted on improved solution quality.

Fernandes and Rosa [17] applied both negative and positive assortative mating to a Vector Quantization problem using a similarity measure that accounts for phenotype information regarding individuals. The approach selects both a parent and a set of mating candidates through a roulette wheel operator and the parent mates with the most similar or dissimilar candidate. Results show an enhanced performance on negative assortative mating and it is argued that diversity handling is a key factor.

Ratford et al. [18] proposed a seduction function that combines the fitness of the mating candidates with the Hamming distance to the first parent. The measure benefits mating between individuals that are neither too similar nor too dissimilar. They also propose dynamically adapting the bias of the function at each generation so that mating between dissimilar individuals is favoured at the beginning of each run but gradually gives space to mating between similar individuals through the evolution process. The ability of the approach on finding multiple solutions on multimodal problems was assessed and results show that the proposed strategy may be an important asset. The same research group has proposed a seduction function that doesn't rely on fitness but rather on either Hamming distance, Euclidean distance or common building block between the first parent and mating candidates [19]. They study the approach on a set of test problems and for the most of it, results are reported to be significantly better. It is also discussed that either similarity measure performed robustly without the need to rely on the fitness of the individuals for mating purposes.

Booker [20], [21] proposed an approach where classifier systems are allowed to mate only if they match the same message. If no full matches are possible, partial matching individuals are allowed to crossover. The approach was labeled *Restricted Mating*. Booker [20] and Goldberg [22] have also explored models where a tag is added to each individual's chromosome and mating occurs when a number of bit-positions between a tag and other individuals are matched. Variations such as one-way, two-way and partial matching have been proposed as well as matching tags with templates rather than individuals. Tags and templates are allowed to evolve as part of the genotype.

Fry et al. [9] experimented in GP with a negative assortative mating scheme that self-adapts along the run. They propose that individuals choose their partners based on a function combining fitness and dissimilarity so that mating between dissimilar pairs is promoted. Similarity is assessed based either on relative or absolute edit distance between individuals. Finally, they propose applying this operator, as an alternative to tournament selection, with a given probability. This value adapts along the run either on a population or on an individual level according to how successful crossover is. Results show that self-adapting the probability of choosing between the most fit or the most dissimilar mating partners provides a valuable balance between exploration and exploitation that enhances performance.

Hinterding and Michalewicz [23] tackle the constrained optimization of a nonlinear programming problem. They propose using a Mate Choice approach that promotes the feasibility of individuals rather than using traditional methods for constrained optimization. On their study, a individual is first selected using a tournament that favors feasible individuals which will then select a mating partner from a set of candidates. By preferring mates that, in conjunction with itself, violate the least number of constraints they aim at producing more feasible offspring. The approach was compared with standard constraint optimization methods achieving comparable results.

Smorokdina and Tauritz have proposed a self-adaptive approach where each individual encodes its own Mate selection function in addition to a candidate solution for the working problem [24]. The Mate selection function is represented using Genetic Programming (GP). The terminal nodes of the trees are, exclusively, the remaining individuals in the population. The non-terminals are a set of selection operators such as *tournament selection*, *biggest hamming distance*, etc. Constraints are enforced to make sure the operators are applied correctly. Mate selection functions are inherited from parents to offspring following one of two proposed rules: given that the new offspring show improvements they inherit the function that was actively used; otherwise, the function attributed to new offspring is the result of the recombination of those from both parents. Results on a set of test problems were slightly worse than a traditional approach.

Guntly and Tauritz [25] propose two variants to a Learning Individual Mating Preferences (LIMP) approach, either using a centralized (C-LIMP) or a decentralized approach (D-LIMP). LIMP has been designed for binary representations, using a real-valued preferences vector that encodes how desirable it is that each gene in the genotype is set to one. The D-LIMP approach attributes a preferences vector to each individual that is used to select mating partners. Offspring inherit the vectors from their parents so that preferences match the genes inherited from each one. Also, the vectors are adapted according to the success of the offspring. The C-LIMP approach relies on two centralized preferences vectors, one relative to genes set to 0 and the other relative to genes set to 1. These vectors are accessed by individuals when evaluating others and adapt to match the success of the offspring in the same fashion as in D-LIMP. Both approaches were compared with a traditional GA and a variable dissortative mating GA, achieving better results in part of the test set.

Machado and Leitão [26] describe a model with self-adapting mate evaluation functions for the Circle Packing in Squares (CPS) problem. They rely on the fact that candidate solutions to this problem also encode candidate solutions to smaller instances, which when assessed can be used as indicators of good genetic quality. Apart from a candidate solution to the problem, each individual also encodes an extra chromosome that represents its own mating preferences. Two approaches were tested either relying on GA or GP representations. The GA mate evaluation function was designed as a weighted sum matching the weights on the first individual with the fitness values obtained by each candidate on smaller instances of the problem. In this case only the weights are evolved. The GP approach evolves whole evaluation functions that are built using the fitness values from each candidate on smaller instances as the terminal set and a set of arithmetic operators as the function set. Results achieved by the GA approach were poor when compared to a standard approach. On the other hand, the GP approach was able to improve upon the results obtained with a classical approach. The same authors have studied a Mate Choice approach to the optimization of Morse Clusters [27]. The approach achieved a slightly more robust behaviour on a global level than a standard selection scheme. The Mate Choice mechanism is coupled with a steady state model and it is discussed that by adapting to the replacement strategy, it is able to produce a higher rate of valuable offspring which have a smaller yet longer impact on the population's structural diversity.

C. A Self-adaptive Mate Choice Approach to Symbolic Regression

The approach implemented for this study encodes in each individual its own mating preferences, which it uses to assess other individuals for mating purposes. In order to do so, each individual includes two chromosomes, the first representing a candidate solution to the target problem, as in a standard approach, and the second representing the mating preferences. Therefore, the approach follows the first rule set in sec. II-B since each individual will rely on its own preferences to assess and evaluate potential partners. The second rule is enforced by the heritability inherent in the evolution process. The application of genetic operators on each chromosome, independently, ensures that genetic material is passed both regarding candidate solutions and mating preferences. Finally, the spectrum of available mating preferences impacts the reproductive success of individuals by favoring certain characteristics over others. The selection pressure caused by mating preferences will ultimately impact both chromosomes by causing the adaption of individuals to each other, therefore adhering to the third rule.

Both chromosomes rely on a GP representation. The first tree is mapped to a candidate solution to the target function, once mapped this is the phenotype of the individual and hence visible to potential mating partners. The second chromosome encodes a tree that is mapped to a function representing the ideal mating partner according to the preferences of the individual. When assessing potential mating partners, each individual compares the function that represents its ideal mate with the phenotypes of each mating partner choosing the most similar one. In a nutshell, evaluation of mating candidates relies on the same mechanisms as when evaluating the first chromosome to determine its fitness. However, instead of comparing the function represented by the first chromosome to the target one, it is compared to the function representing the ideal mating partner. The differences between the ideal mating partner and each of the candidates are measured in a similar way as in a standard approach and the one that most resembles the ideal mating partner is selected for mating. The similarity measure therefore accounts for differences between phenotypes. Traditional choices such as edit distance rely on genotype characteristics, however, we feel that assessing others based on their phenotypes shows a closer resemblance to natural processes.

Algorithm 1 Parent selection using Sexual Selection through Mate Choice

```
mate selection {
    parent1 = parent_selection(Pop)
    candidates = mating_candidates_selection(Pop)
    evaluate_mating_candidates(parent1,candidates)
    parent2 = select_best(candidates)
}
```

Algorithm 1 succinctly describes the selection process applied. Firstly, *Parent1* is selected by means of Tournament Selection and a number of mating candidates is randomly sampled from the population. Secondly, *Parent1* assesses the displayed functions and chooses for mating the individual that

TABLE I

Symbolic Regression Functions. $U[a, b, c]$ represents c uniform random samples drawn from the interval $[a, b]$. $E[a, b, c]$ is a set of points equally spaced with an interval of c, in the interval $[a, b]$.

Name	Objective Function	Set
Keijzer-1 [29]	$0.3xsin(2\pi x)$	E[-1,1,0.1]
Keijzer-2 [29]	$0.3xsin(2\pi x)$	E[-2,2,0.1]
Keijzer-3 [29]	$0.3xsin(2\pi x)$	E[-3,3,0.1]
Keijzer-4 [29]	$x^3e^{-x}cos(x)sin(x)(sin^2(x)cos(x) - 1)$	E[0,10,0.1]
Koza-1 [30]	$x^4 + x^3 + x^2 + x$	U[-1,1,20]
Nguyen-5 [31]	$sin(x^2)cos(x) - 1$	U[-1,1,20]

TABLE II

Mean Best Fitness obtained by the Standard, Mate Choice and Random approaches on each function over 50 runs.

Function	Standard	Mate Choice	Random
Keijzer-1	0.008005462	**0.0059473756**	0.0072442644
Keijzer-2	0.0063776454	**0.0052139161**	0.0062104645
Keijzer-3	0.0071500245	**0.0056003145**	0.0067438776
Keijzer-4	0.0890397335	**0.0833904122**	0.0840754187
Koza-1	**0.0006384168**	0.0014386396	0.0006481816
Nguyen-5	0.0014892713	**0.0004783439**	0.0025763115

best matches its ideal partner. Afterwards, the pair generate offspring by means of GP operators applied independently to each chromosome and the new individuals are introduced into the population of the new generation. The process is repeated until the new population is complete.

This process illustrates how the attractiveness of each individual influences its reproductive success. Genes that are coupled with good mating preferences have a better chance of being mixed with good quality genes during reproduction, therefore increasing their chance of spreading through future generations. On the other hand, mating preferences coupled with attractive genetic material are also more prone to successfully spread. The resulting feedback-loop causes individuals in the population to not only adapt to the environment but also to each other in a struggle for reproduction. The success of the approach therefore relies on its ability to evolve mating preferences that help choose partners in ways that benefit the evolution process.

III. RESULTS

The following subsection details the experimental setup applied to study the effects of Sexual Selection through Mate Choice and the proposed approach on Symbolic Regression. The results are then analysed and discussed.

A. Experimental Setup

Experiments were conducted on symbolic regression targeting a set of functions mixing the six functions displayed in Table I. The decision to tackle this particular subset relies mostly on their inclusion and discussion on the publication regarding GP benchmarking by McDermott et al. [28].

Three approaches were considered: Standard approach where both individuals are selected using tournament selection and mating preferences take no part; a Random approach where the first parent is selected by means of tournament selection but its pair is selected randomly; the proposed approach where the first parent is selected through tournament selection and chooses from a pool a candidates the one that best resembles its ideal mating partner.

At each run, the Keijzer functions evolve a population of 100 individuals while for both the Koza and the Nguyen functions, populations of 500 individuals are evolved. The populations evolve for 100 generations. Tournament size has been set to 5 individuals and the number of mating candidates has also been set to 5. Crossover is applied on the selected parents, independently on the first and the second chromosomes, with a probability of 90% and is 90% biased to function nodes and 10% biased to terminal nodes as crossover points [30]. Elitism is also imposed so that the best individual of each generation is included in the next one. Information regarding the terminal and non-terminal sets can be found in McDermott's publication [28] and implementation details can be found in each of the articles describing the functions [29]–[31].

A total of 50 runs are executed for each approach and data regarding the fitness and the number of individuals resulting from destructive crossover are recorded for analysis. In this study, an individual is considered to be the result of destructive crossover if its fitness is worst than the fitness of the parent that actively chose a mate. An individual is considered the result of a neutral crossover if its fitness is equivalent to the fitness of the parent that actively chose a mate.

B. Analysis and Discussion

Table II shows a comparison of the Mean Best Fitness (MBF) obtained by each studied approach on each function along 50 runs. A Wilcoxon Mann Whitney test with a significance level of 0.01 was conducted, comparing each approach with the remaining two. The instances where the proposed approach performed significantly better than the Standard approach or the other way around are presented in bold in Table II. Regarding the Random approach, it performed significantly better than the Mate Choice approach on the Koza-1 instance but worst on Keijzer-1, Keijzer-2 and Keijzer-3 as well as on the Nguyen-5.

The Koza-1 instance is regarded as a particularly easy instance for symbolic regression [28] which may explain the results obtained, specially if the large size of the population is considered. The results suggest that the overhead created by the Mate Choice model does not payoff. The extra effort put on the evolution process to adapt mating preferences, as seen in Figure 1, slows down convergence, giving the Standard approach the observed advantage. Regarding the Nguyen-5 function, the three approaches performed considerably better than on the Keijzer instances. While the instance is not regarded as particularly simple for symbolic regression, the use of a population with 500 individuals may explain why the approaches were able to achieve better MBFs. In this case, the Standard approach seems able to benefit from a larger, and likely more diverse, population and achieve a lower MBF than the Random approach. Still, Mate Choice was able to

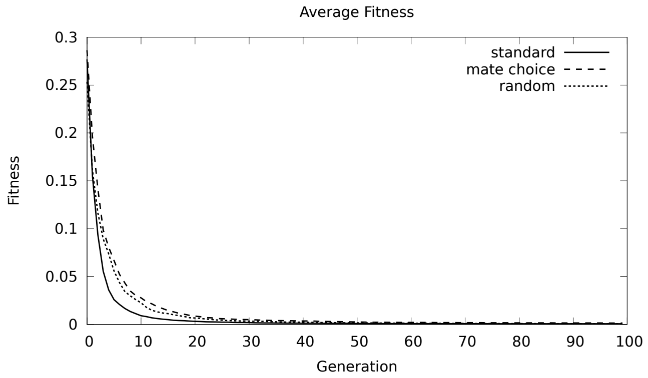


Fig. 1. Mean Best Fitness obtained along 50 runs for the Koza-1 function

TABLE III
MEAN CUMULATIVE DESTRUCTIVE CROSSOVERS OBTAINED BY THE STANDARD, MATE CHOICE AND RANDOM APPROACHES ON EACH FUNCTION OVER 50 RUNS.

Function	Standard	Mate Choice	Random
Keijzer-1	6013.14	**7062.88**	6484.58
Keijzer-2	5892.56	**7149.06**	6929.12
Keijzer-3	6173.56	**7198.2**	6808.86
Keijzer-4	5968.18	**7239.4**	6805.04
Koza-1	34889.32	**36399.72**	36797.82
Nguyen-5	27597.62	**34656.84**	30013.76

outperform the Standard approach with a significant MBF difference.

The Mate Choice approach achieved a significantly better MBF value than the Standard approach on the Keijzer instances as well, suggesting that the proposed strategy is able to contribute to enhancing results. Despite the generated overhead on the evolution process, the approach seems capable of evolving mating preferences that favour mating partners in a way that is beneficial to the evolution process. Overall, the differences in MBFs between the proposed approach and the Random approach are quite noticeable, emphasizing that the behaviour produced by the Mate Choice strategy is not similar to randomly selecting mating partners, but a behaviour that produces larger benefits, therefore backing up our assumption that Mate Choice evolves mating preferences that help selecting mating partners in meaningful ways. It's also noticeable that while no significant differences were found between the Standard and the Random approaches, the later performs slightly but consistently better on the harder instances (Keijzer-1 to Keijzer-4), suggesting that the reduced selection pressure inherent may be beneficial, allowing for a better exploration rather than exploitation of the fittest individuals. Different studies explore selection pressure, one where different schemes are coupled with the ideas of Sexual Selection and compared has been conducted by Wagner and Affenzeller [32].

Tables III and IV further highlight behavioural differences between the studied approaches. Table III shows the mean cumulative number of destructive crossovers (MCDC) obtained along the 50 runs by each approach while Table IV shows the mean cumulative number of neutral crossovers (MCNC). A Wilcoxon Mann Whitney test with a significance level of 0.01 was conducted, comparing each pair of approaches. Instances where the Mate Choice approach achieved a significantly higher MCDC or a significantly smaller MCNC than the Standard approach are depicted in bold.

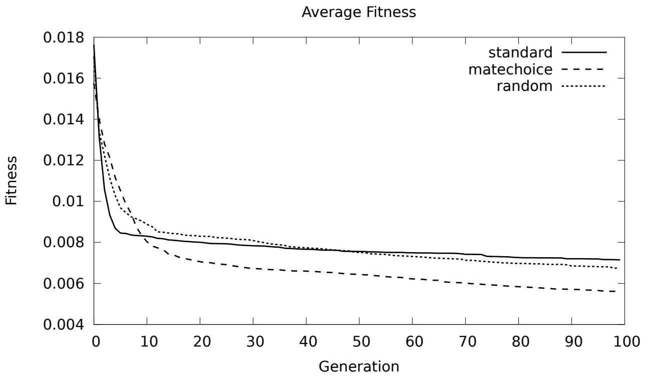


Fig. 2. Mean Best Fitness obtained along 50 runs for the Keijzer-3 function

TABLE IV
MEAN CUMULATIVE NEUTRAL CROSSOVERS OBTAINED BY THE STANDARD, MATE CHOICE AND RANDOM APPROACHES ON EACH FUNCTION OVER 50 RUNS.

Function	Standard	Mate Choice	Random
Keijzer-1	3124.6	**1760.94**	2524.54
Keijzer-2	3278.84	**1708.12**	2060.58
Keijzer-3	2852.96	**1741.4**	2187.86
Keijzer-4	2997.26	**1528.56**	2169.32
Koza-1	8666.42	8016.44	7397.52
Nguyen-5	17749.86	**10583.78**	15592.58

A brief perusal of Tables III and IV reveals that Mate choice consistently results in a higher MCDC than the standard approach. Conversely, it attains lower MCNC on all instances where it outperformed the other approaches. On the Koza-1 instance, probably due to its simplicity, the approach that performed the smallest MCDC was able to outperform the others as candidate solutions are likely to evolve faster and further towards the target function by means of successful crossovers. Although correlation does no imply causality, these results suggest that the explanation for the success of the Mate Choice approach may rest on the reduction of neutral crossover. Individuals that result from neutral crossovers make no contribution to fitness enhancements and may ultimately stall evolution and decrease diversity. The higher MCDC obtained by Mate Choice may be explained by the stochasticity it introduces in the mate selection process. Additionally, Mate Choice is outperforming other approaches and, as such, the results are inherently more difficult to improve.

Due to lack of space, further analysis will focus on a single function. However, the discussed behaviours generalize to the other instances with the exception of the Koza-1, where the Standard approach performs better. We have chosen to focus on the Keijzer-3 function. Figure 2 shows how the MBF evolves along the 100 generations while Figures 3 and 4 show

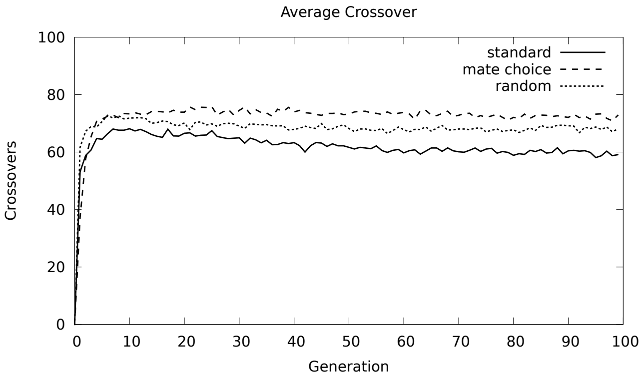


Fig. 3. Mean of Destructive Crossovers obtained along 50 runs for the Keijzer-3 function

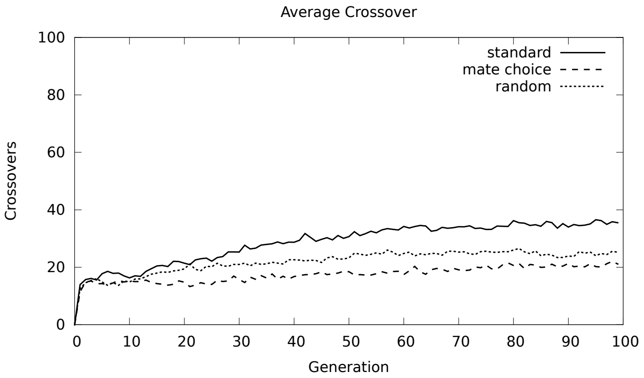


Fig. 4. Mean of Neutral Crossovers obtained along 50 runs for the Keijzer-3 function

the mean of destructive crossovers and the mean of neutral crossovers at each generation and how they evolve along the runs.

Figure 2 shows that the MBF rapidly evolves for the Standard approach for the first 5 or so generations, probably taking advantage of the initial population diversity. Figure 3 shows that for this period, the mean of destructive crossovers also raises rapidly, stabilizes and then gradually descends for the remaining of the generations. The behaviour suggests that, from this point on, the Standard approach promotes crossover between fit solutions. Individuals have a greater chance of promoting their genetic material by mating with fit partners, those who are unable to do so are gradually discarded and no longer contribute to evolution. This behaviour is supported by Figure 4 which shows that the Standard approach produces a gradually higher mean of neutral crossovers. This is consistent with a decreasingly diverse population where a small number of individuals takes over. As a result, while fitter individuals gradually mate with other fit individuals, their impact on the evolution process is reduced and advances are made slowly as individuals exploit a smaller set of genes.

The Mate Choice approach, as shown by Figure 2, has a slower MBF evolution on the first generations, surpassing however the Standard approach at around the 10th generation. Figure 3 shows that by generation 10, the Mate Choice approach is producing more destructive crossovers than the Standard approach. While mating between fit and similar solutions is less likely to cause destructive crossovers, mating with unfit or dissimilar individuals is more likely to produce destructive crossovers as genes mix either with worst genes or with genes from different contexts. The Mate Choice approach seems to promote this behaviour along the runs, causing the mean of destructive crossovers to remain steady for the remainder of the generations. However, as seen by Figure 2 and by Table II, such a behaviour has a positive impact on the MBF, likely the result of a better exploration through the contribution of genes from less fit or dissimilar individuals. Figure 4 supports this behaviour as a smaller mean of neutral crossovers is observed on the Mate Choice approach throughout the entire run. Thus, in the considered experimental settings, the Mate Choice strategy promotes risk-taking resulting in higher MCDC and lower MCNC and taking risks is beneficial in the most complex problem instances.

When comparing the Mate Choice approach with the Random one, both regarding the mean of destructive crossovers and the mean of neutral crossovers, it is noticeable that the produced behaviours happen with a different frequency, indicating that the evolution of mate preferences has an impact on the behaviour of the algorithm. The results reported in Table II and the analysis of Figure 2 show that such an impact is positive for the performance of the evolutionary process.

IV. CONCLUSIONS

Evolutionary Computation is an ongoing research field where, just like in Evolutionary Theory, new ideas are constantly introduced and experimented. While many research fields have come to study and embrace the theory of Sexual Selection, its impact on Evolutionary Computation and inherent effects are yet not fully understood. During this study we explore a nature-inspired Mate choice mechanism and analyse its effects on Evolutionary Algorithms. A comprehensive review of the background of Sexual Selection and more specifically Mate Choice is included as well as an extended review of related work and applications on Evolutionary Computation.

A self-adaptive Mate Choice approach to Symbolic Regression is proposed, following three nature inspired rules: individuals must choose who they mate with based on their perception of others and their own mating preferences; mating preferences are inherited the same way as physical traits; mate selection introduces its own selection pressure but is subject to selection pressure itself.

A representation of mating preferences as an extra chromosome on each individual is introduced and the means by which they are used to assess mating candidates and select the most desirable mating partners are presented. The mechanisms and operators by which mating preferences are inherited and undergo evolution are also discussed. A test suite is introduced and implementation details are presented. Three approaches are experimented and compared: Standard, Mate Choice, and Random.

The obtained results show that the Mate Choice approach is able to significantly outperform the other two on complex instances of Symbolic Regression. The behaviour of the approach is discussed and contrasted with the behaviour obtained by the Random and the Standard approaches, showing that the Mate Choice approach is able to evolve mating preferences that choose mating partners in meaningful ways, actively contributing to an enhanced performance by the Evolutionary Algorithm on the target problem.

Future work may include applying the presented model on a larger function set as well as studying the genealogical trees of the individuals, how mating preferences evolve and affect the evolution of candidate solutions.

ACKNOWLEDGMENTS

This work was partially funded by the Portuguese Foundation for Science and Technology (FCT) under project INO164-PTDC/EIA-EIA/102212/2008, High-Performance Computing over the Large-Scale Internet.

REFERENCES

[1] C. Darwin, “The origin of species,” 1859.

[2] ——, *The descent of man and selection in relation to sex*. John Murray London, 1906.

[3] A. Zahavi, “Mate selection–a selection for a handicap,” *Journal of theoretical Biology*, vol. 53, no. 1, pp. 205–214, 1975.

[4] R. Fisher, “The evolution of sexual preference,” *The Eugenics Review*, vol. 7, no. 3, p. 184, 1915.

[5] ——, “The genetical theory of natural selection.” 1930.

[6] J. Cartwright, *Evolution and human behavior: Darwinian perspectives on human nature*. The MIT Press, 2000.

[7] D. Buss, *Evolutionary psychology.* Washington, DC, US: American Psychological Association; New York, NY, US: Oxford University Press, 2000.

[8] P. J. Angeline, “Adaptive and self-adaptive evolutionary computations,” in *Computational intelligence: a dynamic systems perspective*. IEEE Press, 1995.

[9] R. Fry, S. Smith, and A. Tyrrell, “A self-adaptive mate selection model for genetic programming,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3. IEEE, 2005, pp. 2707–2714.

[10] L. J. Eshelman and J. D. Schaffer, “Preventing premature convergence in genetic algorithms by preventing incest,” in *Proceedings of the Fourth International Conference on Genetic Algorithms*, vol. 115, 1991, p. 122.

[11] R. Craighurst and W. N. Martin, “Enhancing ga performance through crossover prohibitions based on ancestry,” in *Proceedings of the 6th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1995, pp. 130–135.

[12] C. Fernandes and A. Rosa, “A study on non-random mating and varying population size in genetic algorithms using a royal road function,” in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1. IEEE, 2001, pp. 60–66.

[13] E. K. Burke, S. Gustafson, and G. Kendall, “Diversity in genetic programming: An analysis of measures and correlation with fitness,” *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 1, pp. 47–62, 2004.

[14] D. Vrajitoru, “Natural selection and mating constraints with genetic algorithms,” *International Journal of Modelling and Simulation*, vol. 28, no. 2, p. 188, 2008.

[15] S. Gustafson, E. K. Burke, and N. Krasnogor, “On improving genetic programming for symbolic regression,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1. IEEE, 2005, pp. 912–919.

[16] A. Ekárt and S. Németh, “A metric for genetic programs and fitness sharing,” in *Genetic Programming*, ser. Lecture Notes in Computer Science, R. Poli, W. Banzhaf, W. Langdon, J. Miller, P. Nordin, and T. Fogarty, Eds. Springer Berlin Heidelberg, 2000, vol. 1802, pp. 259–270.

[17] C. Fernandes, R. Tavares, C. Munteanu, and A. Rosa, “Using assortative mating in genetic algorithms for vector quantization problems,” in *Proceedings of the 2001 ACM symposium on Applied computing*. ACM, 2001, pp. 361–365.

[18] M. Ratford, A. Tuson, and H. Thompson, “An investigation of sexual selection as a mechanism for obtaining multiple distinct solutions,” *Emerg. Technol*, 1997.

[19] ——, “The single chromosome’s guide to dating,” in *Artificial Neural Nets and Genetic Algorithms*. Springer, 1998, pp. 171–174.

[20] L. Booker, “Intelligent Behavior as a Adaptation to the Task Environment,” PhD Thesis, University of Michigan, 1982.

[21] L. B. Booker, “Improving the performance of genetic algorithms in classifier systems,” in *Proceedings of the 1st international conference on genetic algorithms*. L. Erlbaum Associates Inc., 1985, pp. 80–92.

[22] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[23] R. Hinterding and Z. Michalewicz, “Your brains and my beauty: parent matching for constrained optimisation,” in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE, 1998, pp. 810–815.

[24] E. Smorodkina and D. Tauritz, “Toward automating ea configuration: the parent selection stage,” in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 63–70.

[25] L. Guntly and D. Tauritz, “Learning individual mating preferences,” in *Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*. ACM, 2011, pp. 1069–1076.

[26] P. Machado and A. Leitão, “Evolving fitness functions for mating selection,” in *Genetic Programming - 14th European Conference, EuroGP 2011, Torino, Italy, April 27-29, 2011. Proceedings*, ser. Lecture Notes in Computer Science, S. Silva, J. A. Foster, M. Nicolau, P. Machado, and M. Giacobini, Eds., vol. 6621. Springer, 2011, pp. 227–238.

[27] A. Leitão and P. Machado, “Self-adaptive mate choice for cluster geometry optimization,” in *Proceedings of the fifteenth international conference on Genetic and Evolutionary Computation Conference*, ser. GECCO’13. ACM, (to appear) July 2013.

[28] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. De Jong, and U.-M. O’Reilly, “Genetic programming needs better benchmarks,” in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, ser. GECCO ’12. New York, NY, USA: ACM, 2012, pp. 791–798.

[29] M. Keijzer, “Improving symbolic regression with interval arithmetic and linear scaling,” in *Genetic Programming*, ser. Lecture Notes in Computer Science, C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, Eds. Springer Berlin Heidelberg, 2003, vol. 2610, pp. 70–82.

[30] J. Koza, “Genetic programming as a means for programming computers by natural selection,” *Statistics and Computing*, vol. 4, no. 2, pp. 87–112, 1994.

[31] N. Q. Uy, N. X. Hoai, M. O’Neill, R. McKay, and E. Galván-López, “Semantically-based crossover in genetic programming: application to real-valued symbolic regression,” *Genetic Programming and Evolvable Machines*, vol. 12, no. 2, pp. 91–119, 2011.

[32] S. Wagner and M. Affenzeller, “Sexualga: Gender-specific selection for genetic algorithms,” in *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*, vol. 4. Citeseer, 2005, pp. 76–81.

Bibliography

[1] *Ecj - a java based evolutionary computation research system.* `http://cs.gmu.edu/~eclab/projects/ecj/`.

[2] Agapitos, Alexandros, Julian Togelius, and Simon M Lucas: *Multiobjective techniques for the use of state in genetic programming applied to simulated car racing.* In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 1562–1569. IEEE, 2007.

[3] Andersson, M.: *Sexual selection.* Princeton University Press, 1994.

[4] Angeline, Peter J: *Subtree crossover: Building block engine or macromutation.* Genetic Programming, 97:9–17, 1997.

[5] Azaria, Yaniv and Moshe Sipper: *Gp-gammon: Genetically programming backgammon players.* Genetic Programming and Evolvable Machines, 6(3):283–300, 2005.

[6] Bäck, T, DB Fogel, and T Michalewicz: *Basic algorithms and operators, volume 1 of evolutionary computation.* Institute of Physics Publishing, Bristol and Philidelphia, 11:12–13, 1999.

[7] Bäck, Thomas, David B Fogel, and Zbigniew Michalewicz: *Evolutionary computation 2: advanced algorithms and operators*, volume 2. CRC Press, 2000.

[8] Baker, James E.: *Reducing bias and inefficiency in the selection algorithm.* In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*,

pages 14–21, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc., ISBN 0-8058-0158-8. `http://dl.acm.org/citation.cfm?id=42512.42515`.

[9] Banzhaf, Wolfgang, Peter Nordin, Robert E Keller, and Frank D Francone: *Genetic programming: An introduction: On the automatic evolution of computer programs and its applications (the morgan kaufmann series in artificial intelligence).* 1997.

[10] Blickle, Tobias and Lothar Thiele: *A comparison of selection schemes used in evolutionary algorithms.* Evolutionary Computation, 4(4):361–394, 1996.

[11] Booker, Lashon Bernard: *Intelligent behavior as an adaptation to the task environment, university of michigan.* Ann Arbor, MI, 1982.

[12] Booker, L.B.: *Improving the performance of genetic algorithms in classifier systems.* In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 80–92. L. Erlbaum Associates Inc., 1985.

[13] Braught, G.: *Evolving evolvability: Evolving both representations and operators.* Adaptive and Natural Computing Algorithms, pages 185–188, 2005.

[14] Brindle, A.: *Genetic algorithms for function optimization.* 1981.

[15] Burke, Edmund K., Steven Gustafson, and Graham Kendall: *Diversity in genetic programming: An analysis of measures and correlation with fitness.* Evolutionary Computation, IEEE Transactions on, 8(1):47–62, 2004.

[16] Buss, David M and Michael Barnes: *Preferences in human mate selection.* Journal of Personality and Social Psychology, 50(3):559, 1986.

[17] Chen, Y., J. Hu, K. Hirasawa, and S. Yu: *Gars: an improved genetic algorithm with reserve selection for global optimization.* In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1173–1178. ACM, 2007.

[18] Craighurst, Rob and Worthy N Martin: *Enhancing ga performance through crossover prohibitions based on ancestry.* In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 130–135. Morgan Kaufmann Publishers Inc., 1995.

[19] Cronin, H.: *The ant and the peacock: Altruism and sexual selection from Darwin to today.* Cambridge University Press, 1993.

[20] Darwin, C.: *The descent of man and selection in relation to sex.* John Murray London, 1906.

[21] Darwin, Charles: *On the origins of species by means of natural selection.* London: Murray, 1859.

[22] Dawkins, R.: *1976the selfish gene.*

[23] De Jong, Kenneth Alan: *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, Ann Arbor, MI, USA, 1975.

[24] Deb, Kalyanmoy: *Multi-objective optimization.* Multi-objective optimization using evolutionary algorithms, pages 13–46, 2001.

[25] DEJONG, K: *An analysis of the behavior of a class of genetic adaptive systems.* Ph. D Thesis, university of Michigan, 1975.

[26] Eiben, AE, MC Schut, and AR De Wilde: *Boosting genetic algorithms with self-adaptive selection.* In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 477–482. IEEE, 2006.

[27] Eiben, Agoston E: *Multi-parent recombination.* Evolutionary computation, 1:289–307, 1997.

[28] Eiben, Agoston E, Emile HL Aarts, and Kees M Van Hee: *Global convergence of genetic algorithms: A markov chain analysis.* In *Parallel problem solving from nature*, pages 3–12. Springer, 1991.

[29] Eiben, Agoston E and JE Smith: *Introduction to Evolutionary Computing.* Springer Publishing Company, Incorporated, 2010.

[30] Ekárt, Anikó and S Németh: *A metric for genetic programs and fitness sharing.* Genetic Programming, pages 259–270, 2000.

[31] Eshelman, Larry J and J David Schaffer: *Preventing premature convergence in genetic algorithms by preventing incest.* In *Proceedings of the Fourth International Conference on Genetic Algorithms*, volume 115, page 122, 1991.

[32] Fernandes, Carlos and Agostinho Rosa: *A study on non-random mating and varying population size in genetic algorithms using a royal road function.* In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 60–66. IEEE, 2001.

[33] Fernandes, Carlos, Rui Tavares, Cristian Munteanu, and Agostinho Rosa: *Using assortative mating in genetic algorithms for vector quantization problems.* In *Proceedings of the 2001 ACM symposium on Applied computing*, pages 361–365. ACM, 2001.

[34] Fisher, RA: *The genetical theory of natural selection.* pages 1,12,14, 1930.

[35] Fisher, Ronald A: *The evolution of sexual preference.* The Eugenics Review, 7(3):184, 1915.

[36] Fogel, L.J., A.J. Owens, and M.J. Walsh: *Artificial intelligence through simulated evolution.* 1966.

[37] Fry, R., S.L. Smith, and A.M. Tyrrell: *A self-adaptive mate selection model for genetic programming.* In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2707–2714. IEEE, 2005.

[38] Gagné, Christian and Marc Parizeau: *Ecj - a java based evolutionary computation research system.* month, 2005.

[39] Galán, SF and OJ Mengshoel: *A novel mating approach for genetic algorithms.* Evolutionary Computation, 17(1):55–88, 2009.

[40] Goldberg, David E and Kalyanmoy Deb: *A comparative analysis of selection schemes used in genetic algorithms.* Urbana, 51:61801–2996, 1991.

[41] Goldberg, David E and John H Holland: *Genetic algorithms and machine learning.* Machine learning, 3(2):95–99, 1988.

[42] Goldberg, D.E.: *Genetic algorithms in search, optimization, and machine learning.* 1989.

[43] Goldberg, D.E.: *A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing.* Complex Systems, 4(4):445–460, 1990.

[44] Grefenstette, J.J. and J.E. Baker: *How genetic algorithms work: A critical look at implicit parallelism.* In *Proceedings of the third international conference on Genetic algorithms*, pages 20–27. Morgan Kaufmann Publishers Inc., 1989.

[45] Guntly, L.M. and D.R. Tauritz: *Learning individual mating preferences.* In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1069–1076. ACM, 2011.

[46] Gustafson, Steven, Edmund K Burke, and Natalio Krasnogor: *On improving genetic programming for symbolic regression.* In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 912–919. IEEE, 2005.

[47] Hinterding, R. and Z. Michalewicz: *Your brains and my beauty: parent matching for constrained optimisation.* In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 810–815. IEEE, 1998.

[48] Holdener, E.A. and D.R. Tauritz: *Learning offspring optimizing mate selection.* In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1109–1110. ACM, 2008.

[49] Holland, J.H.: *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artific.* 1992.

[50] Holland, John H: *Genetic algorithms and the optimal allocation of trials.* SIAM Journal on Computing, 2(2):88–105, 1973.

[51] Huang, Chien feng: *A study of mate selection schemes in genetic algorithms - part i*, 2003.

[52] Huxley, J. *et al.*: *Evolution. the modern synthesis.* Evolution. The Modern Synthesis., 1942.

[53] Koza, John R: *Genetic Programming: vol. 1, On the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[54] Koza, John R: *Genetic programming II: automatic discovery of reusable programs.* MIT press, 1994.

[55] Koza, JR and R. Poli: *Search methodologies: Genetic programming.* In *Search methodologies: introductory tutorials in optimization and decision support techniques*, pages 147–164, 2005.

[56] Leitao, António, José Carlos Neves, and Penousal Machado: *A self-adaptive mate choice model for symbolic regression.*

[57] Lin, Yingqiang and Krzysztof Krawiec: *Evolutionary synthesis of pattern recognition systems.* Springer, 2005.

[58] Lu, Haiming and Gary G Yen: *Dynamic population size in multiobjective evolutionary algorithms.* In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1648–1653. IEEE, 2002.

[59] Machado, P. and A. Leitao: *Evolving fitness functions for mating selection.* In *Genetic Programming: 14th European Conference, EuroGP 2011, Torino, Italy, April 27-29, 2011, Proceedings*, volume 6621, page 227. Springer, 2011.

[60] Matsui, K.: *New selection method to improve the population diversity in genetic algorithms.* In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 1, pages 625–630. IEEE, 1999.

[61] McDermott, J., D.R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. De Jong, *et al.*: *Genetic programming needs better benchmarks.* In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, pages 791–798. ACM, 2012.

[62] Miller, Julian F and Peter Thomson: *Cartesian genetic programming.* In *Genetic Programming*, pages 121–132. Springer, 2000.

[63] Poli, Riccardo, W William B Langdon, Nicholas F McPhee, and John R Koza: *A field guide to genetic programming.* Lulu. com, 2008.

[64] Ratford, M., A. Tuson, and H. Thompson: *The single chromosomes guide to dating.* DAI RESEARCH PAPER, 1997.

[65] Ratford, Michael, AL Tuson, and Henry Thompson: *An investigation of sexual selection as a mechanism for obtaining multiple distinct solutions.* Emerg. Technol, 1997.

[66] Rechenberg, I.: *Evolution strategy.* Computational Intelligence: Imitating Life, 1, 1994.

[67] Rechenberg, Ingo: *Evolutionsstrategie–optimierung technisher systeme nach prinzipien der biologischen evolution.* 1973.

[68] Sanchez-Velazco, J. and J. Bullinaria: *Sexual selection with competitive/co-operative operators for genetic algorithms.* In *IASTED International Conference on Neural Networks and Computational Intelligence*, pages 191–196, 2003.

[69] Sánchez-Velazco, J. and J.A. Bullinaria: *Gendered selection strategies in genetic algorithms for optimization.* survival, 8(6):11.

[70] Schlierkamp-Voosen, D.: *Predictive models for the breeder genetic algorithm.* Evolutionary Computation, 1(1):25–49, 1993.

[71] Schwefel, HP: *Evolution and optimum seeking.* Sixth-generation computer technology series, 1995.

[72] Sendhoff, Bernhard, Martin Kreutz, and Werner Von Seelen: *A condition for the genotype-phenotype mapping: Causality.* arXiv preprint adap-org/9711001, 1997.

[73] Silva, AP Alves da and Pedro José Abrão: *Applications of evolutionary computation in electric power systems.* In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1057–1062. IEEE, 2002.

[74] Smorodkina, E. and D. Tauritz: *Toward automating ea configuration: the parent selection stage.* In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 63–70. IEEE, 2007.

[75] Spears, W.M.: *Adapting crossover in evolutionary algorithms.* In *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 367–384, 1995.

[76] Tackett, Walter Alden: *Genetic programming for feature discovery and image discrimination.* In *ICGA*, pages 303–311, 1993.

[77] Tan, Kay Chen, Tong Heng Lee, and Eik Fun Khor: *Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization.* Evolutionary Computation, IEEE Transactions on, 5(6):565–588, 2001.

[78] Varnamkhasti, M.J. and L.S. Lee: *A genetic algorithm based on sexual selection for the multidimensional 0/1 knapsack problems.* In *International Journal of Modern Physics: Conference Series*, volume 9, pages 422–431. World Scientific, 2012.

[79] Wagner, S. and M. Affenzeller: *Sexualga: Gender-specific selection for genetic algorithms.* In *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*, volume 4, pages 76–81. Citeseer, 2005.

[80] Wallace, A.R.: *Darwinism, an exponent of the thory of natural selection, with some of its applications.* Macmillan and Co., 1889. `http://books.google.pt/books?id=e_kZAAAAYAAJ`.

[81] Weismann, A., J.A. Thomson, and M.R. Thomson: *The Evolution Theory V2 (1904).* Kessinger Publishing, 2010, ISBN 9781161729740. `http://books.google.pt/books?id=3x1ccgAACAAJ`.

[82] Whitley, D. *et al.*: *The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best.* In *Proceedings of the third international conference on Genetic algorithms*, volume 1, pages 116–121, 1989.

[83] Zahavi, Amotz: *Mate selection—a selection for a handicap.* Journal of theoretical Biology, 53(1):205–214, 1975.