Improving Robustness in Social Fabric-based Cultural Algorithms

By

Bahram Zaeri

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in the School of Computer Science

Improving Robustness in Social Fabric-based Cultural Algorithms

By

Bahram Zaeri

Supervisory Committee

Dr. Ziad Kobti, Supervisor

(School of Computer Science)

Dr. Mehdi Kargar, Internal Reader

(School of Computer Science)

Dr. Kemal Tepe, External Reader

(Department of Electrical and Computer Engineering)

ABSTRACT

In this thesis, we propose two new approaches which aim at improving robustness in social fabric-based cultural algorithms. Robustness is one of the most significant issues when designing evolutionary algorithms. These algorithms should be capable of adapting themselves to various search landscapes.

In the first proposed approach, we utilize the dynamics of social interactions in solving complex and multi-modal problems. In the literature of Cultural Algorithms, Social fabric has been suggested as a new method to use social phenomena to improve the search process of CAs. In this research, we introduce the Irregular Neighborhood Restructuring as a new adaptive method to allow individuals to rearrange their neighborhoods to avoid local optima or stagnation during the search process.

In the second approach, we apply the concept of Confidence Interval from Inferential Statistics to improve the performance of knowledge sources in the Belief Space. This approach aims at improving the robustness and accuracy of the normative knowledge source. It is supposed to be more stable against sudden changes in the values of incoming solutions.

The IEEE-CEC2015 benchmark optimization functions are used to evaluate our proposed methods against standard versions of CA and Social Fabric. IEEE-CEC2015 is a set of 15 multi-modal and hybrid functions which are used as a standard benchmark to evaluate optimization algorithms. We observed that both of the proposed approaches produce promising results on the majority of benchmark functions. Finally, we state that our proposed strategies enhance the robustness of the social fabric-based CAs against challenges such as multi-modality, copious local optima, and diverse landscapes.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

my cat, Star Trek, and the weather, for supporting me in the low moments.

Supervisor Main, for mentoring, support, encouragement, and patience.

Grant Organization Name, for funding me with a Scholarship.

I believe I know the only cure, which is to make one's centre of life inside of one's self, not selfishly or excludingly, but with a kind of unassailable serenity-to decorate one's inner house so richly that one is content there, glad to welcome any one who wants to come and stay, but happy all the same in the hours when one is inevitably alone.

Edith Wharton

DEDICATION

Just hoping this is useful!

Chapter 1

Introduction

1.1 Evolutionary Algorithms

Evolutionary algorithms take their roots from the evolution theory of Darwin. These algorithms try to mimic the collective behavior of natural systems. Natural processes such as natural selection, survival of the fittest, and reproduction have been subject to inspiration as the fundamental components of evolutionary problem-solving methods. The basic part of all these algorithms is that they start with a randomly generated set of solutions. Then, they try to evolve the solutions through applying a set evolutionary operators such as mutation and crossover.

Evolutionary algorithms are not limited to biological processes. There is another category of evolutionary algorithms known as Swarm Intelligence (SI). These algorithms take inspiration from social behaviors of living colonies such as ants, flocks, schools, and hives. Within these swarms, individuals have relatively simple structures, but their collective behavior usually looks very complex. The complex behavior of a swarm emerges as a result of the interactions between the individuals over time. This complex behavior can not be easily predicted by observing the simple behaviors of the agents separately.

There are some well-known examples categorized as evolutionary algorithms:

1. Genetic Algorithms
2. Cultural Algorithms
3. Prticle Swarm Optimization
4. Ant Colony Optimization
5. Honey Bee Colony

Both categories of evolutionary algorithms share the same idea of evolving some initially generated solutions. However, the difference is in the way that they manipulate and evolve individuals through applying evolutionary operators.

1.2 Robustness in Evolutionary Algorithms

The aim of this research work is to improve robustness in evolutionary algorithms. In this field, robustness means that an algorithm can be used to solve many kinds of problems, with a minimum number of adjustments to address particular problems with special qualities. Also, it can mean the capability of algorithms to deal acceptably with noisy or missing data.

As stated by No Free Lunch(NFL) theorem, there is no algorithm better than others over all cost functions. It means, there is no guarantee for an algorithm to work well for all functions if it shows promising results for a particular category of them. Therefore, robustness has been one of the most desired features which motivates researchers to invent new methods which are less dependent on the kind of a problem than others.

1.3 Research Motivation

abcd

1.4 Thesis Contribution

In the thesis, I am going to improve the robustness of Cultural Algorithms in both components of poplation and belief spaces. In the population space, a new neighborhood restructuring strategy is proposed which works based on a dynamic and irregular topology. In the belief space, a new normative knowledge source is proposed on Confidence Interval inspired form Inferential Statistics. Talk about IEEE-CEC2015 $\cdots$

1.5 Thesis Outline

abcd

Chapter 2

Related Works

Many variants of CAs have been proposed in a vast range of different applications such as single and multi-objective optimization, dynamic problems, social interactions simulation. Here, we are interested in studying socially motivated and multi-population variants as some modern approaches in solving optimization problems.

2.0.1 Heterogeneous Multi-Population Cultural Algorithm

[] proposed a new architecture for cultural algorithms. In this pproach, the whole population is divided into a set of independent sub-populations which work in parallel without direct communication. They referred to the works of [] [] [] []. As their motivation they stated that most of proposed variants of evolutionary algorithms suffers from immature convergence. This occurs because these algorithms can not hold the diversity at a reasonable level. Based on existing research works, they hypothesized that multi-population streategies would be a better choice as they have the potential to perfrom an efficient search on complex landscapes. In their approach, the optimization parameters are divided among some heterogenous sub-populations. the sub-populations are called heterogenous because each sub-population is responsible for optimizing a different subset of parameters. Each sub-population represents

a partial solution instead of a complete solution. To evaluate a partial solution, it gets completed by its complement parameter values from the belief space. The complete solution is evaluated based on a numerical optimization function. Also, to make the convergence process faster a simple local search strategy is incorporated into the proppsed algorithm. The general architecture of their algorithm is presented in figure ???

In the experiments, they considered the whole population size to be 1000 individuals. It is divided into 30 sub-populations. So, the size of each sub-popualtion is 33. The algorithm runs for the maximum of 10000 generations and the local search strategy runs only for 10 iterations. They evaluated HMP-CA on a set of 8 complex optimization functions. It is able to find the minimum value for 7 functions out of 8. However, when the local search strategy is applied to the expriments, the proposed method outperforms all of the functions and it finds the optimum value very quickly. Ultimately, they claimed that their porpsed approach is efficient in both time and space complexity.

2.0.2 The Social Fabric Approach as an Approach to Knowledge Integration in Cultural Algorithms

[] begins with a brief introduction to socially motivated methods to problem solving. It compares qualities of Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Cultural Algorithms (CA) regarding the scale in which the interactions between agents occur. Figure (()) compares PSO, ACO, and CA in terms of the time and space continuum over which the social interactions occur. Individuals in ACO and PSO tend to interact in a reatively limited temporal and spatial scales. It is obvious because the agents in both ant and paricle swarm algorithms exchange information with only other agents in their local neighborhood. On the other hand, cultural algorithms let the individuals interact together using various types of sym-

bolic information emerged from complex cultural systems. In cultural algorithms the interactions among individuals occur indirectly through a shared belief space. So, cultural algorithms allow individuals interact in a global scale.

Then, they asked the essential question of what social structures might emerege alongisde the search process?. To answer such questions, they introduced a new influence function which utilizes the social fabric phenomena. The old influence function assumes no interactions between agents and works based on the simple roulete wheel method. On the other hand, in the new influence function, the individuals are connected through a social network (fabric). Multiple layers of such networks could be employed in a population. The interplay of these network connection forms a social fabric. At each iteration, an individual could specify its controller knowledge source. In this approach, the contoller knowledge source is chosen based on the majority of knowledge sources in the neighborhood of an individual. The neighborhood size of an individual is specified by the topology of the fabric. Inspired by Particle Swarm Optimization literature, different topologies could be taken into consideration to model the relationships among individuals. In their work, they only considered Ring and Square topologies. They stated that, the topology of the social fabric determines the extent to which the influence of knowledge sources could be spread thorugh the network.

To evaluate the social fabric approach, they implemented it in Repast frameork. Repast is a simuation tool for multi-agent systems. They created a cultual algorithms toolkit (CAT) to view the capabilities of cultural algorithms in solving various problems. They chose Cone World problem to evaluate and compare their approach with the standard cultural algorithm. The reason that they chose this problem is that by changing its parameters during the evolution process, it can show a dynamic behavior. So, Cone World problem provides an efficient way to test flexibility of search algorithms. They set the parameters of CAT as: 100 individuals, 100 cones and 1000

generations. They used ring and square topologies to from he social fabric. They stated that square topology works better than ring as it finds the solution after 250 iterations. While, the ring topology finds the best solution 450 iterations.

2.0.3 Robust Evolution Optimization at the Edge of Chaos: Commercialization of Culture Algorithms

The authors of [] aimed at commercializing Cultural Algorithm Toolkit (CAT) thourgh developing a robust variant of it. By robustness, they mean to develop a cultural algorithm which is capable of being applied across a vast range of complex problems. At first, they referred to [Peng model] as an standard model cultural algorithms which assumes no connection between individuals. Then, they referred to [Ali], which introduced the concept of social fabric to allow individuals interact together. The authors extended the work of Ali by allowing the social networks having a memory. In addition, they utilized a variety of different networks in order to deteremine the relationship between network and problem complexity.

They brought up the hypothesis that there might be multiple independent networks for different purposes such as kinship and economics. In such networks, there are always some individuals which are member of multiple networks and so, they can play the role of mediator between differet networks. To preserve the diversity, the authors utilized a variety of different topologies such as Lbest, Square, Hexagon, Octagon, Hexadecagon, and Gbest.

They stated that in the previous work of [Ali], he employed an un-weighted majority win strategy to determine the contoller knowledge source of an individual. It just relied on the count of each knowledge source in the neighborhood of each individual. The authors replaced it with a new strategy which use the average fitness of each KSs instead of each KSs count. So, the performance of individuals in a neighborhood influences which knowledge source to be chosen for the next iteration.

To evaluate the robustness of the algorithm, the authors used Cone World Generator [] as a dynamic problem. Reffered to the work by [Langton] ,they defined three classes of entropy in the connes world problem:

1. Fixed: There is a low entropy and the parameters of the environment do not tend to change at all.

2. Periodic: The environment parameters change in a regular period of time. So, the algorithm should be capable adapting itself regularly.

3. Chaotic: In this case, the parameters change without any order and no prediction could be made about them. In fact, it is more similar to real-world cases.

For the fixed category, the square topology successfully solved 84% of the problems and as the best topology. For the periodic case, the octagon topology showed a better performance than other toplogies. And, for the chaotic category, Gbest showed a better result mostly in terms of average time to find a solution and the standard deviation. In summary, as the complexity of problems grows, there is the need for more connections between individuals in a social fabric to keep the robustness of the algorithm.

2.0.4 Socio-Cultural Evolution via Neighborhood-Restructuring in Intricate Multi-Layered Networks

The authors of [] aimed at exploring the utilization of neighborhoods in the population level of cultural algorithms and see how it influences the knowledge swarming in the belief space. Their approach uses a dynamic neighborhood restructuring to preserve diversity efficiently. They referred to the reasearch works of [], [], []. Those papers proposed some adaptive methods which tried to make a trade-off between exploration

and eploitation properties of evolutionary algoithms. Among the referred papers, the work by [ALi] used the social fabric phenomena to from multi-layered hierarchical network structures in both homogenous and heterogeneous networks.
The authors defined the social fabric phenomena as follows:

> "The Social Fabric is a living informational skin created out of the engineered emergence of agents illustrating the tension between the individual and the community in a context of interaction between them"

The informational skin is created by the connectivity of individuals together. The social fabric is used to combine the behaviors at both indivials and society levels. Then, they proposed a strategy to determine the contoller knowledge source of each individual in each generation. In the strategy, the agents send the name of their contoller knowledge source to their neighbord through the social fabric. Each individual picks the mostly used knowledge source in its neighborhood as its controller for the next iteration. In case of a tie between two or more knowledge sources, three tie-breaking strategies are intorduced:

1. Most Frequently Used (MFU)
2. Random
3. Least Frequently Used (LFU)

They stated that their proposed approach is inspired from a previous work of Ali[22]. The new idea was called Cultual Algorithm with Restructuring Layered Social Fabric (CARLSOF). In this approach, the whole population is divided into two layers of rudimentary and advanced members. There are some independent tribes in the population space which their best performing individuals form the advanced layer and the others form the rudimantary level. In the previous works, the topology of social fabrics were supposed to be fixed and homogenous. They utilized a variety of regular

topologies to form the social fabric. The authors proposed a new strategy to enable the fabric to be restructured in the case of stagnation. Each tribe may decide to change its topology to a denser (with more conections) or sparser (with less connections). They referred to two different restructuring strategies as similar approaches. The first one was Layered Delaunay Triangulation which is based on voronoi diagram. The second approach was Random Rewiring Procedure which starts with a regular topology and then rewire each edge randomly with the probability p.

To evaluate the performance of CARLSOF, they used function set of IEEE-CEC2011 evolutiosry competition. The algorithm was implemented in JAVA. The authors reported that CARLSOF was successful to enhance all of the functions in the testbed interms of average, best, and worst obtained values. Also, they claimed that CARLSOF outperforms the best previously obtained results for European Space Agency (P12) and Casini (P13) problems results. They reported 2.983 km/sec compared to previous best 7.095. For p13, they reported 8.383091 km/sec compared to previous best 8.3832.

2.0.5 Tribe-PSO: A novel global optimization algorithm and its application in molecular docking

The authors proposed a new variant of Particle Swarm Optimization (PSO) called Tribe-PSO for the primary purpose of molecular docking in chemometrics. As some previous research work, they referred to [], [], [], []. Their approach is inspired from Hierarchical Fair Competition concept []. They divided the whole population into some tribes and the evolution process into three phases. The tribes are organized into two layers of basic and upper individuals.

The individuals in each tribe are completely isolated from other tribes. So, they form the basic layer. On the other hand, the best members of each tribe can see the other tribes and exchange information with their best members as well. The problem-

solving process is divided into three phases: isolated, communing, and united. In the first phase, the tribes are completely isolated and there is no exchange of information. In the second phase, the two-layered model is formed and the tribes begin to exchange information through their best performing individuals. In the third phase, all the tribes are merged into a single popualtion. Then, it operates as basic PSO model until meeting some stopping criteria. The autohrs claimed that their approach helps the individuals preserving diversity and avoiding local optima against multi-modal solution spaces.

The authors used two testbeds to evaluate and compare the performance of Tribe-PSO with standard PSO. The first was De Jong's function set and the second was a test set of 100 receptor-ligand X-ray structures selected for docking benchmark. In the De Jong's testbed, the basic PSO showed a better performance in the isolated phase. However, in the communing phase Tribe-PSO converged to a better value than basic PSO. In the unity phase, the basic PSO seemed to get stuck in a local optimum. However, Tribe-PSO continued to accelerate convergence and got better results than basic PSO. In the docking benchmark, Tribe-PSO was compared to the AutoDock library. Four parameters are calculated after 10 independent runs: Best, Run1, Average, and Standard Deviation of the results. The relative difference of the four benchmark factors between Tribe-PSO and AutoDock were calculated. The results for the four factors showed that Tribe-PSO leads to a better performance than AutoDock.

2.0.6 Heterogeneous Particle Swarm Optimizers

The authors peresented a survey on heterogeneous variants of Paritcle Swarm Optimization (PSO). They claim that the homogenous models have been attractive because of their simplicity in conceptual and application levels. However, heterogeneous models are ubiquitous in nature. Here, heterogeneity means the individuals

may differ from each other regarding their parameters and search behavior.

At first, the authors presented a breif description of three well-known variants of PSO: Accelerated PSO[ref], Fully-informed PSO[ref], and Bare Bones PSO[ref]. Then, they authors categirzed the heterogeneous variants of PSO into four categories:

1. Neighborhood
2. Model of Influence
3. Update Rule
4. Parameters

Neighborhood heterogeneity refers to the cases that the topology of the swarm is not regular. This type of heterogeneity occurs when the neighborhood size of each individual is different. They claimed that the neighborhood heterogeneity allows some population to be more influential than otehrs. Individuals with higher number of connections have the potential to attract more individuals through the search process. [Kennedy ref]

Model of Influence heterogeneity refers to the situations that the individuals employ different strategies to specify their informers. The word Informer refers to an individual which is going to influence another individual. []

Update-rule heterogeneity means the individuals utilize different strategies to update their position in the search space. This kind of heteogeneity makes it possible for the individuals to explore the search space in several ways. Also, the particles can play different but complementary roles. For example, some of them may tend to explore unseen parts of the solution space and some others only follow those scout individuals. The second type are the exploiters.

Parameter heterogeneity occurs when some individuals in a group which follow the same update rule use different update rule's parameter settings. Having different

search parameters, even in a group of similar particles, leads to various search behaviors which improves the level diversity. The authors referred to [], [] ,[] which utilize different initial values for the parameters such as acceleration coefficient, maximum velocities and inertia weight.

To compare the mentioned categories, the authors careted two test cases. The first one compared two PSO variants with different update rules: Velocity-based and Bare bones swarm. The evaluation results confirmed that velocity-based outperformed the other one. The second test case compared two PSO variants with different models of influence: best-of-neighborhood and fully-informed swarm. The evaluation results showed that the fully-informed algorithm outperfromed the best-of-neighborhood.

2.0.7 Leveraged Neighborhood Restructuring in Cultural Algorithms for Solving Real-World Numerical Optimization Problems

In the paper, the authors made an overall review of related works on Cultural Algorithms and Social Fabric [], [], []. They introduced the concept of social fabric as an infrastructure which facilitates the propagation of the influence of the knowledge sources thorugh the population space. The whole population is divided into multiple small sized tribes. Unlike other PSO-based multi-swarm variants, the tribal subswarms change their topology during the evolution process to keep diversity against stagnations.

The tribes are organized in a two-layer structure of advanced and rudimentary classes. The advanced calss is composed of best perfoming individuals of each tribe. From each tribe, two examplars are chosen. The first one is the best of explorer knowledge sources (Normative and Topopgraphic) and the second one is the best of exploiter knowlegde sources (Situational). The authors utilized only three kinds of knowledge

sources: Normative, Topographic, and Situational. They claimed that, other thpes of knowledge sources are suitable for dynamic problems. However, they used a set of static problems to evaluate their approach. The authors divided the evolution process of CAs into three stages: the secluion stage , the rapport stage , and the cohesive stage. In the seclusion stage, the tribes evolve independetly and without any exchange of information. Then, in the rapport stage, the tribes start to exchange knowledge via the advenced class. Also, the neighborhood restructuring strategy is lunched in each tribe. So, the tribes have the capability to restructure their topology to keep diversity and avoid local optima. The first two stages ensure that diversity has been kept in the initial iterations of the search process. Finally, in the third stage, all the tribes are merged together and continue the search process as a single CA model.

Based on the categorization proposed in [], the authors claimed that their apparoach is heterpgeneous regarding update-rule heterogeneity and neighborhood heteogeneity. Since the topology of each tribe is changing during the evolution process, the tribes might use different toplogies at the same time. So, it could be considered as heterogenous neuighborhood restructuring. Also, their approach utilizes update-rule heteogeneity because, there are individuals with different roles (explorer and exploiter).

In the exprimental results section, the authors used IEEE-CEC2011 competition on testing evolutionary algorithms on real-world numerical optimization problems. They presented an extended comparison of their approach with other propsed methods. At first, the authors tested the approach (T-SCANeR) with different values for these parameters: tie-breaking rule, M_{thresh}, and number of elites. There is no best value for tie-breaking rule as it is problem-dependent. For the M_{thresh} parameter, the results show that the best value is 50. The best value for n_{tElite} is 2. The second expriment was on varying the window size within which the social fabricis applied and aggregated (W_{size}). The best reported value for W_{size} is 20.

In the two first exprients the best values for the parameters were determined. Based on the obtained values, the authors compared T-SCANeR with five other evolutionary algorithms on the IEEE-CEC2011 which is a set of 18 functions. T-SCANeR managed to outperform for most of the functions except for T_3, T_4, T_6, and $T_{11.5}$. Problems T_{12} (full messenger problem) and T_{13} (cassini problem) are associated with the European Space Agency (ESA). The authors claimed that T-SCANeR outperformed other methods by 2.193724 km/s for the messenger problem and 8.383090 for the cassini problem.

Chapter 3

Evolutionary Algorithms

This chapter presents a detailed description of related evolutionary algorithms such as cultural algorithms, particle swarm optimization, and their social variations. In this section, we try to give a brief description of those evolutionary algorithms as a background to our proposed approaches.

3.1 Evolutionary Algorithms

Evolutionary computing is a research area within computer science. As the name suggests, it is a particular flavor of computing, which draws inspiration from the process of natural evolution. In fact, they are computer programs which try to solve and optimize complex problems by simulating the behavior of natural systems. They utilize evolutionary operators such as crossover and mutation to improve the quality of a population of solutions.

Evolutionary algorithms start with a population of randomly generated solutions for a particular problem. Then, the algorithm modifies the solutions in an iteration-based process of some finite number of generations. The modification occurs through applying the so-called evolutionary operators such as crossover and mutation. The evolutionary operators might be binary or unary. For example, the crossover is a

binary operator because it combines two solutions (individuals) to generate a new one. On the other hand, mutation is a unary operator because it makes random modifications on a single solution to improving its performance. The performance of each is evaluated using a fitness function. The fitness function represents the problem that needs to be solved or optimized. Usually, the fitness function is chosen from NP problems which are hard to solve by traditional problem-solving methods. In each iteration of an evolutionary algorithm, the evolutionary operators are applied to the individuals and the best performing offsprings will be transferred to the next generation. This process continues until meeting some stopping criteria which are already defined by a human user. Some examples of evolutionary lgorithms are:

1. Genetic Algorithms
2. Evolutionary Porgramming
3. Differential Evolution

However, the evolutionary algorithms are not restricted to biological processes. Some researchers have drawn inspiration from social systems to find solutions to complex problems. The complex and coordinated behavior of swarms not only fascinates biologists but also is an inspiration to computer scientists. Ant colonies and birds flocking are remarkable examples of coordinated collective behavior that emerges without centralized control. Swarm intelligence is a field of computer science that invents computational methods for solving problems in a way that is inspired by the behavior of real swarms and colonies. Principles of self-organization and communication (local and indirect) are essential to understanding the complex collective behavior. There are different types of swarm-based algorithms such as:

1. Particle Swarm Optimization (PSO)
2. Ant Colony Optimization (ACO)

3. Cultural Algorithms (CA)

3.2 Cultural Algorithms

Cultural Algorithms (CA) were introduced by Reynolds as a type of population-based problem-solving approaches. CA combines biological evolution with socio-cognitive concepts to yield an optimization approach based on a dual inheritance theory. As defined by [Durham], culture is a "system of symbolically encoded conceptual phenomena that are socially and historically transmitted within and between populations".From the definition, it can be stated that in cultural systems, evolution occurs at two levels: Macro-evolutionary level and micro-evolutionary level. Cultural algorithms define the evolution process through the cooperation of three distinct components:

1. Belief Space (Macro-evolutionary Level)
2. Population Space (Micro-evolutionary Level)
3. Communication Protocol

3.2.1 Belief Space

Belief space keeps different kinds of knowledge obtained from the individuals' experience during the evolution process. It extracts the knowledge from the population and stores the knowledge in various formats called knowledge sources (KS). Each knowledge source extracts and keeps knowledge about a particular aspect of the search space.

After each iteration, the best performing individuals are chosen and sent to the belief space to be utilized by the knowledge sources. The stored knowledge is used to bias

the search process in the population space. Five types of knowledge sources have been identified in the belief space:

1. Situational Knowledge: Successful exemplars of individuals
2. Normative Knowledge: The best range of value for each parameter
3. Topographic Knowledge: The best areas of the search space
4. Domain knowledge: The domain ranges for all the parameters and the best exemplars
5. Temporal Knowledge: Knowledge about the past changes in a dynamic environment

3.2.2 Population Space

In the population space, any population-based algorithm could be used. In the earlier variants of cultural algorithms, only GA was used. However, researchers utilized other population-based algorithms such as PSO and EP in the population component. In each generation, the best individuals are sent to the belief space to update the knowledge sources. Then, the belief space influences the population through the search process. Figure [] ....

3.2.3 Communication Protocol

In cultural algorithms, the belief space and the population communicate together through a two-way protocol. The population space sends the best individuals to the belief space using the acceptance function to update the knowledge sources. Then, the belief space bias the search direction of the population space through the influence function.

3.3 Social Fabric-based Cultural Algorithms

asd

———————-

A Latex document is composed of two parts: the Preamble, and the Document Body. The *Preamble* is the site for inclusion of all document set up commands: definition of new commands, inclusion of prebuilt packages, template declaration, etc. The *Body* is where the document content is placed.

3.3.1 Preamble

The Preamble refers to the input which precedes the documents contents. It is the area where the author determines the general template for the document using the **documentclass**[*options*]{*doc style*} command. For example, \documentclass[11*pt*]{*article*} declares that a document will follow the *article* document class, and have 11pt font.

If the document requires support of any library packages they must be included in the preamble using the **usepackage**{*package name*} command. For example, \usepackage{graphicx} is the command needed to include the graphicx package.

3.3.2 Document Body

The document body is the area which follows the Preamble. It is defined by the **begin**{*document*} and **end**{*document*} commands. The content of a Latex document is declared in the document body. Input which appears after the \end{document} command is ignored.

3.4 How to Number Pages

To number the pages of a document use the **pagenumbering**{*style*} command. Numbering is defined in the documents preamble. There are several different *styles* to choose from.

Numbering Style	**Output**
\pagenumbering{arabic}	1, 2, 3, ...
\pagenumbering{roman}	i, ii, iii, ...
\pagenumbering{alph}	a, b, c, ...
\pagenumbering{Roman}	I, II, III, ...
\pagenumbering{Alph}	A, B, C, ...

Table 3.1: Page Numbering Styles

The numbering of pages for a thesis is, however, much more complex than for an article and, in fact, the *book* class has been adopted. Make changes to those settings only if you are really familiar with LaTeX.

3.5 How to Create a Title Page

A title page can be either on a separate page or integrated directly into the first page of the document. It is defined by three declarations, followed by the **maketitle** command as illustrated below.

```
\title{Title of Paper}
\author{Author(s) of Paper}
\date{Publication Date}
\maketitle
```

The article document class defaults on an integrated title page. To make a separate title page, use the **titlepage** option with the \documentclass[*titlepage*]{*doc style*} command.

For this thesis style the title page has been completely formatted for you. Just insert the various names of people in the supervisory committee, the title, your name and so on in the location where the *dummy* entries exist right now and you will be done. I would suggest to avoid doing any other changes unless you are absolutely sure!

3.6 How to Create an Abstract

To create an abstract, place contents of abstract between the **begin**{*abstract*} and **end**{*abstract*} commands.

3.7 How to Create a Table of Contents

The **tableofcontents** command automatically generates a table of contents from all section headers. The default behavior for the article document class is to produce an integrated table of contents. However, the document can be altered to generate the table of contents on a separate page using the **newpage** command (see section Formatting Extras).

For this thesis template a special command has been added, namely the **textTOCadd**. You can find it in the file *macros/style.tex*. It has to be explicitly called for an insertion into the Table of Contents and it is already in place appropriately for the existing sections and subsections.

3.8 How to Create Sections

Creating sections, subsections, and subsubsections is completed using the \section{Section Name}, \subsection{Subsection Name} and \subsubsection{Subsubsection Name} commands, respectively. Each sectional division is numerically labeled with respect

to it's placement in the section hierarchy. For example, this section was defined with the code:

```
\section{How to Create Sections}
Creating sections, subsections, and ...
```

It is useful to give a label using the **label** command to a section or subsection if a reference to it is made, so that the reference will be automatically updated should the structure of the document change.

3.9 How to Create a List

Lists can be either enumerated, non enumerated, or descriptive. Each element of a list is termed an 'item'.

1. enter the list environment with the \begin{*list style*} command.
2. define each item with the \item command for non\enumerated lists, or \item[*label*] for descriptive lists.
3. terminate list environment with the \end{*list style*} command.

3.10 How to Insert Tables, Figures, Captions, and Footnotes

The table and figure environments contain input blocks which cannot be split across pages. Rather than divide the input of either of these environments, the contents are relocated, or floated, to a location in the document which optimizes page layout with the surrounding document content.

loc	Purpose
l	left justified column
r	right justified column
c	centered column
|	vertical rule

Table 3.2: Table Example

3.10.1 Tables

Tables are created in the tabular environment. A single parameter is used to define the number of columns and item justification pertaining to each column. The single parameter is a combination of the following ones shown in Table 3.2.

\\ and & are used to define rows and columns, respectively. A table can either have the contents of its rows and columns lined or not. Each line used to construct the table must be individually specified, using | and \hline for vertical and horizontal lines, respectively.

Table 3.2 was generated with the following input:

```
\begin{center}
    \begin{tabular}{|l|l|} \hline
    l & left justified column   \\ \hline
    r & right justified column  \\ \hline
    c & centered column         \\ \hline
    $|$ & vertical rule         \\ \hline
 \end{tabular}
 \end{center}
```

You will want to include your table in the "List Of Tables" section at the beginning of your thesis. To do this you enclose the above table inside a table environment like so:

```
\begin{table}
```

```
\begin{center}
...
\end{center}
\caption{Sentence describing table.}
\label{unique:label}
\end{table}
```

The caption is the text that appears underneath the table. It should be short and precise. The label is a unique label that you can use to refer to the table within your document. You can use the `\ref{label}` to insert the table number into your text as in Table 3.2. In the example above you would use as in:

```
I am referring to Table \ref{unique:label}.
```

3.10.2 Figures

The first step to including an externally prepared image into a document, is to declare the graphixs package into the documents preamble. Integrating the image can be done using the figure environment. Enter and exit the figure environment with the **begin{figure}**[*loc*] and **end{figure}** commands, respectively. The *loc* dictates the placement of the included image, and can be any of the following:

h here: location in text where the environment appears

t top: top of the page

b bottom: bottom of the page

p page of floats: on a separate page with no text

For organizational purposes, it is best to have keep all figures in a folder together. I usually label the folder as "*Figures*" (with great creativity) and I placed it in the same directory as the topmost main *.tex* file. Include the image into the document with

the **includegraphics**[*dim*]{path to image} command. *dim* dictates the magnitude of the `height` or `width`. The image is scaled proportionally. An example and its resulting output follow below.

```
\begin{figure}[h]
\centering
\includegraphics[height = 1in]{LinuxPenguin.eps}
\caption{The Linux Penguin}
\end{figure}
```

Why is the output for the figure not shown? Because inserting figures into LaTeXis not that simple and it is highly dependent on the system you are using together with the type of figure. This is not the place to dwell upon the inconsistencies which can make your life difficult. Suffice it to say that the original LaTeXand its tools was geared to accept *.eps* files for figures and it still maintains that expectation if one compiles using a *Latex to dvi to (pdf or ps)* series of commands. On the other hand, if one uses the *Latex to pdf* direct path, then files of other types are perfectly fine (e.g. *pdf, jpg, gif, etc.*).

If you are interested, look at the actual file for this section namely "sec_latexhelp.tex" and consider the set of lines commented out just above this paragraph. There are two examples of insertion of figures, the first with the *.eps* version and the second with the *.pdf* version of the same picture (of a penguin). Delete the comments from one of the two sets and use the appropriate tools.

To refer to a figure, the same approach used for tables should be used, namely a `\ref{label}` command which includes the unique identifier label for that figure, as in:

```
I am referring to Figure \ref{unique:label}.
```

3.10.3 Captions

Captions for tables and figures are created using the **caption**{caption goes here}. Captions are automatically numbered with separate counters for tables and figures. **caption**{caption contents} can only be used in the Figure or Table environment.

3.10.4 Footnotes

Footnotes are inserted with the **footnote**{footnote contents} command. This footnote[1] is generated as follows:

```
...This footnote\footnote{this is a footnote} is generated...
```

3.11 How to Alter Font

3.11.1 Type Style

Roman Family is the default type style. The types style can be modified using the following commands.

Command	**Output**
\textit{Italic Characters}	*Italic Characters*
\textsl{Slanted Chartacters}	*Slanted Characters*
\textsc{Small Cap Characters}	Small Cap Characters
\textbf{Boldface characters}	**Boldface characters**
\textsf{Sans Serif Characters}	Sans Serif Characters
\texttt{Typewriter Characters}	`Typewriter Characters`

[1] this is a footnote

3.11.2 Type Size

The font size can be modified using the following commands.

Command	Output
\tiny{tiny font}	tiny font
\scriptsize{scriptsize font}	scriptsize font
\small{small font}	small font
\normalsize{normalsize font}	normalsize font
\large{large font}	large font
\Large{Large font}	Large font
\huge{huge font}	huge font
\Huge{Huge font}	Huge font

3.12 Math Mode

To incorperate mathematical content into a document, Latex provides three different environments: Displaymath, Math, and Equation. Brief descriptions for each environment, and environment short cuts are displayed in the table below.

Environment	Function	Shortcut
math	displays an in-text formula	\ (...\ )
displaymath	displays an unnumbered formula	\ [...\ ]
equation	displays a numbered formula	N/A

The following examples, using Einstein's famous $e \doteq mc^2$ equation, illustrate how to include a formula into a document.

```
...Einstein's famous \( e \doteq mc^{2} \) equation, illustrate...
```

```
\[e \doteq mc^{2}\]
```

$$e \doteq mc^2$$

```
\begin{equation}
   \doteq mc^{2}
\end{equation}
```

$$e \doteq mc^2 \tag{3.1}$$

3.13 Formatting Extras

The following table illustrates some formatting tips for perfecting the layout of a Latex document.

Command	**Purpose**
\hspace{*len*}	insert a horizontal space of length *len*
\vspace{*len*}	insert a vertical space of length *len*
\mbox{*text*}	ensure that *text* is not split over multiple lines
\\	new line
\newpage	start new page
\pagebreak	insert a page break
%	precedes comments

Chapter 4

Experiments

Assuming you have some experimental results to support your claims this is where all the data is reported. There are a few issues you should consider before dumping a lot of stuff here, or it will lose its effectiveness.

First of all you must describe precisely the experimental setup and the benchmarks you used. In any scientific discipline an experimental result is only good if it is reproducible. To be reproducible then somebody else must have sufficient details of the setup to be able to obtain the same data. Thus the first section in this chapter is a super precise history of the decisions made towards experimentation, including mentions of the paths which became infeasible. The setup must be valid and thus your description of it must prove that it is indeed sound. At times, terrifying times, when writing this section, both supervisor and student realize belatedly that something is missing and more work needs to be done!

The second portion of this chapter is dedicated the the actual results. At least two issues arise here:

1. Should all the data be reported here or should some be placed in the Appendix?

2. Should this be an exposition of the raw facts and data or should it include its analysis and evaluation?

There are no definite answers here, but I follow a few rules.

Should all the data be reported here or should some be placed in the Appendix?

- If there is a large number of tables of data, it might be better to present here only a handful of the most significant ("best") results, leaving all the rest of the data in the Appendix with proper linkages, as it would make the chapter so much more easily readable (not to mention limiting the struggle with a word processor for the proper placement of tables and text).

- Use an example throughout, call it a "case study" to make it sound better, so that all the data and results are somehow linked in their logic, and even better if this is one of the examples you used in Chapter 2 to describe the original problem.

- Highlight in some manner the important new data, for example the column of your execution speed where all the numbers are much smaller. Make the results highly easy to read!

- It is normally expected that data should be presented only in one form and not duplicated, that is, you are not supposed to include both a table of raw numbers and also its graphical representation from some wonderful Excel wizard. I tend to disagree. I would not wish to see every results repeated in this manner, but some crucial ones need to be seen in different manners, even with the same information content, in order to show their impact. One good trick is to place the more boring tables in the Appendix and use wonderful graphs in this chapter.

- This is the one chapter where I would splurge and use colour printing where necessary, as it makes an *enourmous* difference.

Should this be an exposition of the raw facts and data or should it include its analysis and evaluation?

- Is the evaluation of the data really obvious? For example you have 10 tables to show that your chemical process is faster in development and gives purer material - you may simply need to highlight one column in each table and state the obvious.

- Most results are not that obvious even if they appear so. Moreover this is where you are comparing your *new* results to data from other people. I usually describe other people's work at this point and make comparisons. That is why I prefer to talk about the analysis and evaluation of the results in a separate chapter.

- There is absolutely no clear structure here which is best.

Chapter 5

Evaluation, Analysis and Comparisons

For a Master's research this chapter represents the critical part where **you** are truly evaluated to determine whether you should be given your degree. Even more so for a PhD. Consider carefully what the University calendar states regarding the expectations for a master's thesis, paraphrased here.

1. *A Masters thesis is an original lengthy essay.* The main implication here is that the essay is original, that is, it is completely newly written by you and does not contain any writings from others unless precisely quoted. Any paraphrased items must be cited.

2. *It must demonstrate that:*

    - students understand research methods;
    - students are capable to employ research methods;
    - students demonstrate command of the subject.

3. *The work may be based on:*

- original data;
- original exercise from scholarly literature;
- data by others.

4. *The work must show that:*
   - appropriate research methods have been used;
   - appropriate methods of critical analysis supplied.
5. *The work must contain:*
   - evidence of some new contribution;
   - evidence of a new perspective on existing knowledge.

Only the last point uses the attribute *new* and it refers almost entirely to giving a new perspective and analysis, even if based on data from others. This truly implies that this current chapter on evaluation and analysis of results is the most important and must be written with care. You are demonstrating here that, even if given data and methods from others, your skills of critical judgment and analysis are now at the level that you can give professional evaluations.

Things are slightly different for a PhD. According to the Graduate Calendar:
a doctoral dissertation must embody original work and constitute a significant contribution to knowledge in the candidate's field of study. It should contain evidence of broad knowledge of the relevant literature, and should demonstrate a critical understanding of the works of scholars closely related to the subject of the dissertation. Material embodied in the dissertation should, in the opinion of scholars in the field, merit publication.

The general form and style of dissertations may differ from department to department, but all dissertations shall be presented in a form which constitutes an integrated

submission. The dissertation may include materials already published by the candidate, whether alone or in conjunction with others. Previously published materials must be integrated into the dissertation while at the same time distinguishing the student's own work from the work of other researchers. At the final oral examination, the doctoral candidate is responsible for the entire content of the dissertation. This includes those portions of co-authored papers which comprise part of the dissertation.

The second paragraph makes it clear that one must emphasize what is new and different from others, without arrogance, yet without being too subtle either. The first paragraph implies that for a PhD it is required that one approached an important open problem and gave a new solution altogether, making chapters 3, 4, 5 all part of the body of research being evaluated. In fact at times even the problem may be entirely new, thus including chapter 2 in the examination. This is in contrast to a Master's degree where the minimum requirement is for chapter 5 to be original.

Chapter 6

Conclusions

My first rule for this chapter is to avoid finishing it with a section talking about future work. It may seem logical, yet it also appears to give a list of all items which remain undone! It is not the best way psychologically.

This chapter should contain a mirror of the introduction, where a summary of the *extraordinary* new results and their wonderful attributes should be stated first, followed by an executive summary of how this new solution was arrived at. Consider the practical fact that this chapter will be read quickly at the beginning of a review (thus it needs to provide a strong impact) and then again in depth at the very end, perhaps a few days after the details of the previous 3 chapters have been somehow forgotten. Reinforcement of the positive is the key strategy here, without of course blowing hot air.

One other consideration is that some people like to join the chapter containing the analysis with the only with conclusions. This can indeed work very well in certain topics.

Finally, the conclusions do not appear only in this chapter. This sample mini thesis lacks a feature which I regard as absolutely necessary, namely a short paragraph at the end of each chapter giving a brief summary of what was presented together with

a one sentence preview as to what might expect the connection to be with the next chapter(s). You are writing a story, the *story of your wonderful research work*. A story needs a line connecting all its parts and you are responsible for these linkages.

Appendix A

Additional Information

This is a good place to put tables, lots of results, perhaps all the data compiled in the experiments. By avoiding putting all the results inside the chapters themselves, the whole thing may become much more readable and the various tables can be linked to appropriately.

The main purpose of an Appendix however should be to take care of the future readers and researchers. This implies listing all the housekeeping facts needed to continue the research. For example: where is the raw data stored? where is the software used? which version of which operating system or library or experimental equipment was used and where can it be accessed again?

Ask yourself: if you were given this thesis to read with the goal that you will be expanding the research presented here, what would you like to have as housekeeping information and what do you need? Be kind to the future graduate students and to your supervisor who will be the one stuck in the middle trying to find where all the stuff was left!

Bibliography

[1] Cliff Atkinson. *Beyond Bullet Points: Using Microsoft Office PowerPoint 2007 to Create Presentations That Inform.* Microsoft Press, 2008.

[2] Wayne Booth. *The Craft of Research.* University of Chicago Press, 2003.