TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Faculty of Electronics, Telecommunications and Information Technology
Communications Department
Adaptive Systems Laboratory

Marcel Cremene

Complexity in Adaptive Systems

Habilitation Thesis

2014

"...engineers tend to ignore the social concerns of their work, and social scientists, on the other hand, do not know very much about technology and are reluctant to consider the artificial reality of technical objects..."

Günter Ropohl

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

Abstract

Faculty of Electronics, Telecommunications and Information Technology
Communications Department

Habilitation

Complexity in Adaptive Systems

by Marcel Cremene

The author's research activity, following the PhD defense in 2005, is presented from a unified interdisciplinary perspective of complex adaptive systems. The Thesis subject lies at the crossroads of *Software Engineering*, *Optimization Theory*, *Meta-heuristics*, *Evolutionary Computing* and *Game Theory*. Service adaptation - a main topic of this Thesis - is a central concept in *Software Engineering*. Automatic complexity management represents an important challenge in fields such as *Autonomic Computing* and *Self-Organizing Networks*. Meta-heuristic based approaches offer efficient solutions for adaptation problems modeled as search/optimization problems. Multi-agent complex systems have interesting properties that provide powerful solutions to adaptation problems when a decentralized approach is adopted.

Four main directions are investigated: a) designing a general model for the service adaptation problem based on a centralized control approach, b) proposing models and algorithms for complex decentralized distributed adaptive systems, c) analyzing different perspectives in decomposing/recomposing an adaptive system in/from several modules, and d) including psycho-social elements in the framework of technical systems.

Complexity is considered here an orthogonal aspect related to all four directions stated before. The term *complexity* is used with two-fold meaning: the first one is the *computational complexity* as it is used in *Theoretical Computer Science*; the second meaning of *complexity* is related to distributed decentralized *complex systems* composed of multiple autonomous entities. Complexity definitions, specific problems, properties, measures, and issues are discussed.

The most significant author's contributions presented in the Thesis are:

- An interdisciplinary vision about complex adaptive systems of services, which also includes psycho-social aspects.
- An approach for modeling service adaptation as an optimization problem. The proposed model is based on the notion of *service-context* distance, which represents the function(s) to be minimized/maximized by an optimization/search algorithm.
- A comparative analysis of different evolutionary optimization algorithms applied on the QoS-aware optimization problem.
- Comparative experiments that illustrate the advantages of multi-objective optimization approach versus single-objective optimization.
- An adaptation algorithm for automatically tuning the parameters of other evolutionary algorithms.
- A solution for medical service optimization that combines QoS attributes and transport costs.
- Two solutions for natural language based web service composition.
- A general architecture for distributed systems of services based on agents and services.
- New models for Cognitive Radio based on Game Theory. Scenarios for TV white spaces and small-cell open access environments are investigated.
- A framework providing an adaptive security protocol for *Mobile-Cloud* applications.
- A model for a dynamic market of Cloud services based on Cournot and Bertrand games.
- Adaptive algorithms for *Affective Computing* applications based on ontologies and neural networks.
- A computational model called 'Cascade Computing' that solves the NP-complete 'sub-set sum' problem in linear time using a digital hardware circuit. The model was patented.
- An adaptive 'combiner-equalizer' module for radio receivers. This module illustrates the idea of solving in one step two signal processing operations solved usually in two distinct steps. This idea was also patented.

- An evolutionary game model called 'Social Honesty' is proposed for analyzing strategic social interactions in socio-technical systems.

An *ecological* and *holistic* approach is the base of the proposed vision for guiding the future development of technical systems. Future research will continue to explore the usage of *Computational/Artificial Intelligence* in general and *Game Theory* in particular for solving *Software Engineering* and *Telecommunications* problems. Regarding complex adaptive systems, there are still many challenges and unsolved issues.

Future lines of inquiry are:

- Investigating a hybrid approach for an adaptation control mechanism that will be both function and rule based.
- Analyzing and experimenting complex adaptation scenarios, where a larger number of S-C distances are involved. Studying the search/optimization algorithms scalability.
- Applying the service-context model to decentralized distributed scenarios. Implementing the proposed architecture model based on agents and services.
- Analyzing and comparing different distributed algorithms for complex adaptive systems.
- Further developing the proposed adaptive security protocol framework for Mobile-Cloud applications.
- Implementing the dynamic service's market model and evaluating the impact of different price models.
- Proposing a general design method for decentralized algorithms based on different game equilibria.
- Conducting new experiments with the *SmartRadio* CR devices.
- Performing additional experiments with the proposed 'Social Honesty' game.
- Finishing a mobile application for psychological therapy that is currently in progress.
- Further analyzing socio-technical interactions and proposing new game-based models.

A list of co-directed PhD theses is included in *Appendix A*. A research laboratory called ASL (Adaptive Systems Laboratory) founded by the author is mentioned. The author development plan, based on critical and creative thinking for a personalized motivation-oriented learning, is presented in *Appendix B*.

Acknowledgements

A unique formulation that could resume my acknowledgments is: *all the good ideas came to me by interacting with other people.* Therefore, I want to thank all those whom I interacted with on scientific themes. The first place is reserved to all my co-authors.

All these people are important yet there are some who had a special influence on my evolution:

Professor Dan Dumitrescu offered me the great opportunity to collaborate with him and his research team. Evolutionary optimization and Game Theory tools came to me this way. Besides the numerous fruitful discussions, I also learned how important it is to stimulate and encourage people.

My wife, Ligia, is also a great scientific collaborator. The most important aspect for which I should thank her is the ability to make unexpected connections between very different concepts from different fields and to find interesting analogies. She was at the origin of the interest in Cognitive Radio and socio-technical systems.

Thanks are due to professor Costin Miron for the very useful interactions that continued after my Ph.D. studies. The most important thing learned from these interactions is the use critical thinking and a systematic intellectual approach. This is how I discovered the great importance of asking questions, looking from different perspectives, and managing time in small intervals.

I also thank professor Michel Riveill for the collaboration after my PhD. An important number of results have been published in collaboration with his team. These interactions exposed me to high quality research. I received a very important support for the research as ATER and invited professor at University of Nice Sophia-Antipolis. I express my thanks to all my collaborators from University of Nice.

Contents

List of Figures

Abbreviations

AC	**A**ffective **C**omputing
AI	**A**rtificial **I**ntelligence
ANR	**A**utomatic **N**eighbor **R**elation
AoA	**A**spect of **A**ssemblies
AOP	**A**spect **O**riented **P**rogramming
AWGN	**A**dditive **W**hite **G**aussian **N**oise
BER	**B**it **E**rror **R**ate
CAS	**C**omplex **A**daptive **S**ystem
CASS	**C**omplex **A**daptive **S**ystem of **S**ervices
CC	**C**ascade **C**omputing
CCT	**C**omputational **C**omplexity **T**heory
CE	**C**ombiner-**E**qualizer
COP	**C**omponent-**o**riented **P**rogramming
CR	**C**ognitive **R**adio
CS	**C**omputer **S**cience
DE	**D**ifferential **E**volution
ECA	**E**vent-**C**ondition-**A**ction
EMOA	**E**volutionary **M**ulti-objective **O**ptimization **A**lgorithm
GA	**G**enetic **A**lgorithm
GT	**G**ame **T**heory
HV	**H**yper**v**olume
IGD	**I**nverted **G**enerational **D**istance
IT	**I**nformation **T**echnology
LMS	**L**east **M**ean **S**quares
LTE	**L**ong **T**erm **E**volution

MOA	**M**ulti-objective **O**ptimization **A**lgorithm
MVC	**M**odel-**V**iew-**C**ontroller
NGMN	**N**ext **G**eneration **M**obile **N**etworks
PD	**P**risoner's **D**ilemma (game)
QoS	**Q**uality **o**f **S**ervice
SAT	**Sat**isfiability (problem)
SBSE	**S**earch **B**ased **S**oftware **E**ngineering
S-C	**S**ervice-**C**ontext
SH	**S**ocial **H**onesty (game)
SLA	**S**ervice **L**evel **A**greement
SMS	**S**hort **M**essage **S**ervice
SNR	**S**ignal to **N**oise **R**atio
SOA	**S**ervice-**O**riented **A**rchitecture
SOC	**S**ervice-**O**riented **C**omputing
SON	**S**elf-**O**rganized **N**etworks
STS	**S**ocio-**t**echnical **S**ystems
TM	**T**uring **M**achine

Chapter 1

Introduction

This Thesis presents the author main research results and contributions since November 2005 (after the PhD defense). Future lines of inquiry and a personal development strategy are also proposed. The unifying vision and contextualization of the presented solutions are highlighted. Theoretical sections illustrate the connections existing between the various scientific contributions.

The subject lies at the crossroad of *Software Engineering*, *Mobile Communications*, and *Computation Intelligence*. *Complexity Theory*, *Optimization Theory* and *Meta-heuristic/Evolutionary* algorithms are particularly of concern.

Four main directions are investigated:

- *Designing a general model for the web service adaptation problem. A centralized adaptation control model is considered.*
- *Proposing models and algorithms for distributed adaptive systems. A decentralized multi-agent approach is considered.*
- *Analyzing different possibilities of decomposing and recomposing an adaptive system.*
- *Including the psycho-social aspects in the framework of adaptive systems.*

An orthogonal aspect related to all four directions stated previously is *complexity*, which is systematically analyzed for each type of analyzed system.

It should be mentioned that, the type of *adaptation control model* should not be mistaken for the type of system architecture. For instance, a system having a distributed architecture may still have a centralized control. In this case a *super-entity* has a global vision of the entire system and may control it. A distributed control mechanism involves multiple independent autonomous control entities (possibly belonging to different owners/providers). In this case an important aspect is how to achieve an optimal global behavior.

Another consideration regards the optimality. In many cases, finding the optimal solution is not always possible and, when it is possible, it requires a significant computation time. Therefore, sub-optimal solutions may be more useful in practice. This fact motivates the interest for heuristic-based approaches.

1.1 Key concepts

For a proper introduction to the subject, the basic concepts should be firstly explained. The most important terms and concepts are defined bellow:

Service. Two main meanings are given to the term 'service' in the field of Information Technology (IT). In *Telecommunications*, 'service' usually indicates a telephony service, a data transport service, or a Short Message Service (SMS). In *Computer Science* (CS) a 'service' is defined as a reusable software component providing application functionality to other applications. A large number of reusable web services, designed for different purposes, exists today: travel reservation, maps, data type converters, etc. In this Thesis the CS definition is used.

One of the most interesting feature of service-orientated design is *composability*: new services (named *composite services*), providing new functionality, are created by assembling (or composing) existing services. This is possible with the help of another important service feature: *reusability*.

Adaptive Service. The most common analogy for an adaptive entity comes from biology: an organism that adapts to its environment. Fields like *Context-awareness*, *Autonomic*, *Pervasive* and *Ubiquitous Computing* use this analogy to define applications and services capable to adapt themselves to the *context* [Abowd et al., 1999]. The context is considered here from the service point of view and includes: user requirements, available resources, physical location, social proximity, and other elements that may interact with the service. A counter-example of adaptive system is, for instance, a washing machine which blindly executes a predefined program - an *open loop* control system. An adaptive entity is usually a *closed-loop system*, which is able to sense the environment, analyze, plan and act based on some internal rules and/or goals [Kephart and Chess, 2003].

System. A '*system*' refers to a software or hardware-software application/device which is made of several components (modules, services, agents). Web services, mobile applications, and programmable wireless devices are illustrations of considered systems. A special attention is placed on user-system interactions.

System of Services. A system of services includes a large number of services from different domains: human resources, transportation, provisioning, etc. This concept was proposed by IBM (Jim Spohrer, keynote conference ICSOC 2010) [Spohrer, 2010]). A store, an enterprise, a university, a city, all involve the co-existence and co-interaction of different types of services. It is like an artificial ecosystem of services managed by different providers. In practice, such a system cannot have a unique centralized control entity. This is why self-organization/management as a mechanism and efficiency as a property are very important features in such systems.

Adaptive System of Services. An adaptive system of services is a system that includes adaptive services (at least one entity/part of the system is an adaptive service). Such a system is usually decentralized and includes a large number of autonomous entities.

Complexity. The term 'complexity' is used in the Thesis with two-fold meaning. The first one is the *computational complexity* as it is used in *Theoretical Computer Science.* Most of the Thesis's contributions are related to the field of *Search Based Software Engineering* (SBSE). In this case, the complexity is given mainly by the size and the shape of the search space.

The second meaning of 'complexity' is related to distributed decentralized *complex systems.* A complex system is composed by multiple autonomous entities. The psycho-social aspect may also be included in such a system. Complexity in this second meaning is reflected on the most recent author's contributions and connects to domains like *Multi-Agent Systems* (MAS).

Complex Adaptive Systems of Services. A Complex Adaptive Systems of Services (CASS) is a particular type of CAS, where the entities composing the system are adaptive services managed by intelligent agents.

1.2 Topic actuality and importance

This section underlines the topic's actuality and importance. *Web Services* and *Wireless Networks* are especially of concern. The necessity of an interdisciplinary approach is discussed.

1.2.1 Services sector

In the context of global economy, the Service sector is one of the most important and dynamic. In the service sector revenues largely exceeds other sectors such as agriculture and goods [Spohrer, 2010]. A significant part of the services sector is represented by software (web) services. The software services impact is expected to grow even more in the future. Service-Oriented Computing (SOC) and Service-Oriented Architecture (SOA) occupy a central place in *Software Engineering.* Automatic service composition and adaptation represent key issues in SOC/SOA.

1.2.2 Self-Organized Networks

An increasing interest for self-management may also be noticed in *Mobile/Wireless Communications.* The fourth of mobile phones generation, 4G, already includes the concept of *Self-Organized Networks* (SON). Self-configuration, self-optimization and self-healing represent highly desirable network features [Yilmaz et al., 2011].

Standardization bodies such as '3rd Generation Partnership Project' (3GPP) and 'Next Generation Mobile Networks' (NGMN) started to release standards for SON [Feng and Seidel, 2008]. 'Long Term Evolution' (LTE), the most important 4G standard, already includes SON features such as 'Automatic Neighbor Relation' (ANR) detection [Feng and Seidel, 2008] (starting with Release 8).

Nature inspired algorithms (i.e. swarm intelligence) have been recently proposed for addressing SON challenges [Zhang et al., 2014]. In fact, SON may be seen as the application of *Autonomic Computing* principles to networks.

1.2.3 Autonomic/Autonomous Computing

Two terms having slightly different meanings appear in literature: '*autonomic computing*' and '*autonomous agents*'. In the context of this Thesis, the differences between 'autonomic' and 'autonomous' are not very relevant. Both terms will be further used.

As noticed by researchers from *IBM* - the promoters of the *Autonomic Computing* metaphor [Kephart and Chess, 2003] - *complexity* of software systems is continuously increasing. Very high costs are related to manual systems maintenance and upgrade. How to automatically manage the complexity and its continuous evolution is a difficult yet important challenge.

The most common analogy for autonomic/autonomous adaptive systems is the central nervous system: a set of sensors collects data about the service and its context, an intelligent system analyzes the data, and some actions are triggered. Even if the general architecture of an autonomic/autonomous system is well established, the difficult problems reside in concrete details: which data to collect, how to represent them, how to define the system 'intelligence', how to discover which action to take. Numerous solutions for self-healing, self-configuring, self-optimizing, and self-managing systems have been proposed, yet they are still far from solving all the existing problems.

The term 'adaptation' include a large set of meanings. To find a general solution to the adaptation problem remains an important challenge. The adaptation control mechanism is a key part of an adaptive system. This mechanism may be centralized or distributed.

1.2.4 Centralized vs. distributed control models

A centralized approach is usually preferred because the control model is more simple. However, this is not always possible because different services may be provided by different owners. A central control point is also a single point of failure. In the 2011 Fukushima disaster 3G networks failed but the Internet was still able to offer limited functionality yet enough to communicate with isolated people (as declared in a keynote to CROWNCOM 2011 conference). This is another proof that distributed self-organized systems may be more robust in unpredictable situations. In such situations *cognitive radio*, self-adaptive, devices may play an important role in maintaining a reliable communication.

A decentralized distributed system is composed by several autonomous entities. In biology there are numerous examples of insect colonies and other animal societies where simple 'agents' are able to manifest very complex global behavior (*swarm intelligence*). Such an approach may lead to solutions to managing highly complex systems. However, the challenge in such systems is how to design the agents in order to assure an evolution towards a desired state, an optimal one if possible.

This where *Game Theory* [Osborne, 2004] comes into place and may help us to make some predictions about the evolution of such distributed systems towards certain equilibria. Game Theory is already widely used not only in its main field, Economy, but also in radio engineering (*Cognitive Radio*) [Wang et al., 2010]. Our intention is to use it also in the *Software Engineering* domain, where it is less present for the moment. GT is a mathematical framework for studying strategic interactions: one player's payoff depends on other player's actions.

1.2.5 Interdisciplinary, transdisciplinary and holistic approaches

In the recent years, scientists discuss more and more about the necessity of *interdisciplinary*, *transdisciplinary* [Nicolescu, 2002], and *holistic* approaches. The psycho-social aspect needs also to be included in the proposed vision of adaptive systems. Communication is first of all human-related. In practice, psychological and social aspects may be even more important than technical ones. In his article 'Philosophy of socio-technical systems', G. Ropohl observes that: '*...engineers tend to ignore the social concerns of their work, and social scientists, on the other hand, do not know very much about technology and are reluctant to consider the artificial reality of technical objects...*' [Ropohl, 1999]. This fact is a natural consequence of labor division and a direct result of the common disciplinary approach. However, one needs to be aware that the domains' boundaries are purely artificial. Since IT solutions are used by humans it is only natural to be interested in human-related aspects of technical systems.

1.3 Application fields

The contributions presented in this Thesis are mainly related (without being limited) to the following application fields: *Web Services in general and medical services in particular*, *Wireless Communications*, *Complexity Theory*, *Cloud Computing*, *Mobile Computing (including Mobile-Cloud applications)*, *Security*, *Affective Computing* [Picard, 1997], *Cognitive Radio*, *Multi-Agent Systems* and *Socio-technical Systems* [Ropohl, 1999].

1.4 Thesis structure and main contributions

The Thesis is structured in seven chapters, two appendices and a reference list. All seven chapters describe the most author's significant scientific contributions. The appendices briefly present the academic contributions and the personal development strategy.

The most important scientific contributions are:

- A general horizontal contribution of this Thesis is the interdisciplinary vision about adaptive systems of services that includes also the psycho-social aspect. Complexity is central aspect to this vision.
- **Chapter 2** explains the two meanings of complexity discussed here: computational complexity and complex adaptive systems. The most important aspects related to complexity are discussed: definitions, properties, measures, specific problems and issues. A computational model called 'Cascade Computing' is presented. The proposed model solves the NP-complete 'sub-set problem' in deterministic polynomial time using a digital hardware circuit. This idea was patented.
- **Chapter 3** is dedicated to the idea of modeling service adaptation as an optimization problem. The proposed model is based on the notion of *service-context distance*, which represents the function(s) to be minimized/maximized. Different evolutionary optimization algorithms are tested on the QoS-aware optimization problem. New evolutionary algorithms are proposed. The advantages of multi-objective optimization approach are investigated and comparative tests are performed. An adaptation algorithm for tuning the parameters of an evolutionary

algorithm is proposed. A solution for medical services optimization, which includes also the transport costs is presented. Two solutions for natural language based web service composition are briefly described.

- **Chapter 4** presents aspects related to distributed adaptation mechanisms and multi-agent models. *Cognitive Radio* and *Cloud Computing* fields are particularly targeted. A general architecture for distributed systems of services is proposed. Scenarios for small-cell open access environments are investigated from a *Game Theory* perspective. Models for *Cognitive Radio* based on Cournot, Bertrand, and Stackelberg games are presented. New types of game equilibria are investigated. A framework providing an adaptive security protocol for Mobile-Cloud applications is described. A game-based model for a dynamic market of Cloud services is presented.

- **Chapter 5** briefly discusses the principle of decomposing and recomposing an adaptive system. A given system may be always decomposed in different manners. Sometimes, instead of decomposing it is preferable to unify. A proposed adaptive *combiner-equalizer* for radio receivers illustrates the idea of solving in one step two types of signal processing operations. This idea was patented.

- **Chapter 6** underlines the importance of introducing psycho-social aspects in the framework of technical systems. Adaptive mechanisms for *Affective Computing* applications based on ontologies and neural networks are presented. The 'socio-technical system' metaphor is briefly discussed. The role of trust in highly connected information-based networks is analyzed. An evolutionary game model for analyzing strategic social interactions in technical systems is presented.

- **Chapter 7** concludes the Thesis with brief discussions and presents the future research directions.

- **Appendix A** enumerates some relevant academic contributions of the author: several co-advised PhD theses and a recently founded research entity.

- **Appendix B** briefly presents the author's personal development strategy concerning future education and research activities.

Chapter 2

Aspects of Complexity

This chapter discusses the most significant aspects related to *complexity*: types, definitions, specific issues, typical problems, examples, and measures of complexity. Computational complexity and complex adaptive systems are analyzed.

2.1 Computational Complexity

Computational Complexity Theory (CCT) has a key place in *Theoretical Computer Science*. CCT main subject is the classification of computational problems according to their complexity or difficulty. This section is composed by two parts: the first one presents the most important aspects related to computational complexity and the second part presents an author contribution related this domain.

2.1.1 Computational complexity - definitions and classes of problems

In CCT's context, the meaning of 'complexity' is related to the number of steps (time complexity) or memory space (space complexity) necessary for an algorithm in order to solve a computational problem.

Time complexity. The time complexity is a measure that describes how the number of steps required by an algorithm increases with the size of the problem. A standard notation 'O' (called *Landau's symbol*) is used to express this measure. For instance, $O(n^2)$ means that the time complexity increases four times when the problem size (n)

doubles. If the time (or number of steps) T required to solve a problem is, for instance: $T(n) = 3n^2 + 17n + 25$ then the complexity is still O(n^2) - the constants and the lower terms are less important.

Space complexity. The space complexity is a measure of memory space required by an algorithm for solving a problem. The same notation 'O' is used to express spatial complexity. For instance, O(N^2) means that the memory space required increases four times when the problem size (N) doubles. In most cases the time complexity and the space complexity cannot be simultaneously minimized, therefore a trade-off is necessary.

A commonly used abstract model in CCT is the *Turing Machine* (TM). A Turing Machine is an abstract computing machine that manipulates symbols recorded on a tape. The Church-Turing thesis [Church, 1936], [Davis, 2006] is that, if an algorithm for solving a computational problem exist, then the problem can be solved by a TM. The TM model is used because it is very general and easy to be analyzed from a mathematical point of view. Deterministic and non-deterministic TM models exist. The non-deterministic model is obtained from the deterministic model by adding some stochastic rules (some problems may be solved more efficiently when using probabilistic decisions).

Several classes of problems are defined function of the following criteria: a) the problem type (decision problem, function problem, counting problem, optimization problem, etc.), b) the problem complexity/difficulty, and c) the model type (deterministic or non-deterministic).

For instance, if we consider the time complexity, the problems of class 'P' are solvable by a deterministic TM in a polynomial time ($T(n) = poly(n)$ - a polynomial function of n). The problems of class 'NP' are solvable by a non-deterministic TM in a polynomial time. The problems of class 'EXPTIME' are solvable by a deterministic TM in an exponential time $2^{poly(n)}$. 'NEXPTIME' is the correspondent of 'EXPTIME' for a non-deterministic TM.

If we consider the space complexity, the class 'L' includes the problems solvable by a deterministic TM in a logarithmic time, O($log(n)$). 'NL' is the correspondent of 'L' for non-deterministic TM. 'EXPSPACE' is the space correspondent for 'EXPTIME'.

The relations between the classes of complexity described before is depicted in Fig. 2.1.

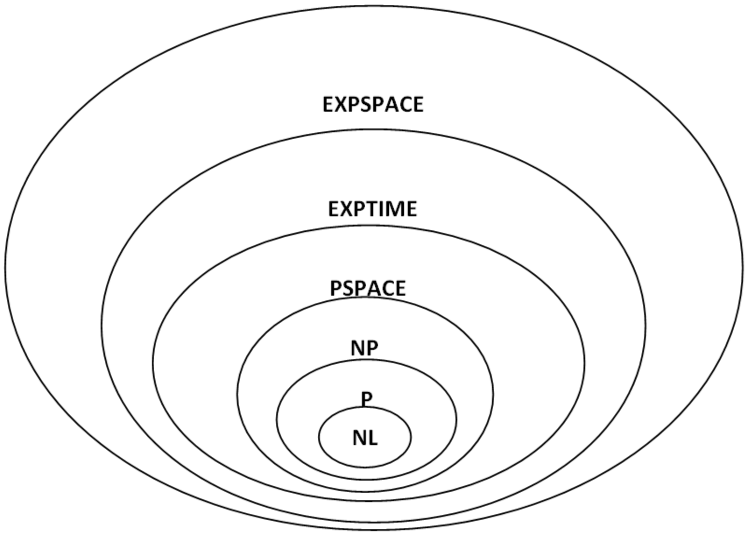


Figure 2.1: Relation between classes of computational complexity.

A central preoccupation of the CCT domain is to decide if the inclusion relations between the classes of complexity are strict or not (i.e. L =? NL, NL =? P, P =? NP, etc).

A problem H is considered to be '*hard*' for a class C if every problem in C can be reduced to H. This means that no problem from C is more complex than H, and an algorithm for H can also solve any problem from C. If the problem H belongs to C then H is called to be '*complete*', in other words H is one of the most complex problems from the class C. For instance, 'NP-complete' represent the class of the most difficult problems from the NP class.

2.1.2 A computational machine that solves the NP-complete 'sub-set sum' problem in deterministic polynomial time

A computation model called '*Cascade Computing*' (CC), proposed for solving the classical '*subset-sum*' problem, is presented in the national patent [Dumitrescu et al., 2013]. Let Γ be a finite set of integers. The 'subset-sum' problem is to decide if a non-empty subset β of the set Γ exists with the property that the sum of β elements is zero (or equal with a given number *sum*). The proposed CC machine has practical applications in cryptography and optimization.

This problem is known to be *NP-complete* and therefore cannot be solved by a deterministic TM in a polynomial time. In other words, there is no efficient solution to solve such a problem because the search space size increases exponentially with the cardinality of set Γ. The possibility solve a 'NP' problem in 'P' is an essential problem in CCT. Subset-sum may be seen as special case of the 'knapsack problem' [Martello and Toth, 1990] - another very important problem.

2.1.2.1 Brief state-of-the-art

The classical method for solving the 'sub-set sum' problem is to use a *dynamic programming* algorithm [Garey and Johnson, 1979] that requires a pseudo-polynomial time and has a spatial complexity equal to *sum* multiplied by the cardinal of the set Γ. However, this algorithm is difficult to be parallelized and no efficient hardware implementation is known.

Another approach is to use *spiking neural networks* [Leporati and Gutiérrez-Naranjo, 2008]. This method has the advantage to be easily parallelizable and therefore faster than the previous sequential method. However, the number of components (space complexity) increases exponentially with the cardinal of Γ.

A method based on *optical computing* [Oltean and Muntean, 2009] - a system based on optical signals. This approach is very fast and the space complexity is proportional to the cardinal of Γ. However, such optical systems are costly and there are an important number of technical issues to be solved in order to implement them.

2.1.2.2 Cascade Computing machine description

The Cascade Computing general architecture is depicted in Fig. 2.2. The CC machine is composed by: a source node S, a series of modules M_i, and a test node T (Fig. 2.2).

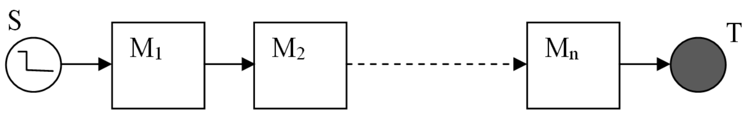


FIGURE 2.2: General architecture of the Cascade Computing machine.

Each module M_i of the CC machine architecture depicted in Fig. 2.2 has an internal architecture as depicted in Fig. 2.3. A module M_i is composed by k cells $a_1 \ldots a_k$ and a logical OR gate. All a_i cell are identical. Each cell introduces a one clock delay of the binary signal.

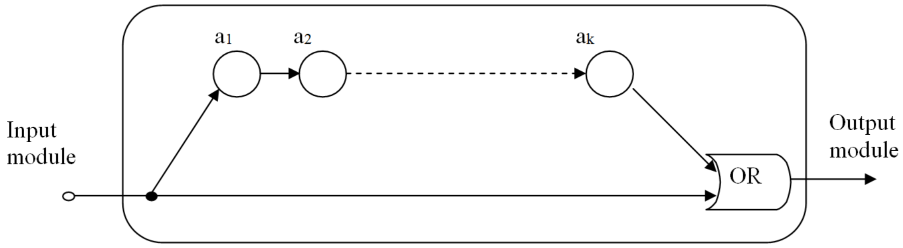


FIGURE 2.3: Internal architecture of a module M_i used in the Cascade Computing machine architecture.

The CC machine uses discrete binary electric signals. The source S generates a step impulse. This impulse is transmitted to the input of the first module $M1$. For each integer of the given set, a module M_i is included in the architecture. Each of these modules includes a number of delay cells equal to the corresponding integer value. The modules M_i compute the partial sums.

A test node T placed at the system output checks if a solution was found. If the logic state at the output of the last module M_i is '1' at a discrete time moment equal to the given value '*sum*' it means that a solution was found. The integers are supposed to be all positive (if they are not, a translation is applied to each value in order to have only positive integers - the problem is equivalent).

The time complexity of the proposed system is deterministic and linear with the value *sum*.

The space complexity is also deterministic and it is mainly given by the number of delay cells, which is equal to the sum of all elements from the given set Γ, therefore it is also linear.

A hardware implementation of the CC machine is required in order to benefit from all its speed advantages.

An electric scheme of the device implementing the CC machine is presented in the patent [Cremene et al., 2011a] (pending). This scheme is illustrated in Fig. 2.4 and it is based on standard digital components (gates, counters, shift registers, etc.). A simplified example for a set composed by three integers is considered here.

The circuit depicted in Fig. 2.4 works as follows: a clock generator generates a uniform rectangular signal. The clock frequency gives the calculation speed: after a number of clock periods equal to *sum* the decision problem is solved. Connected to the clock

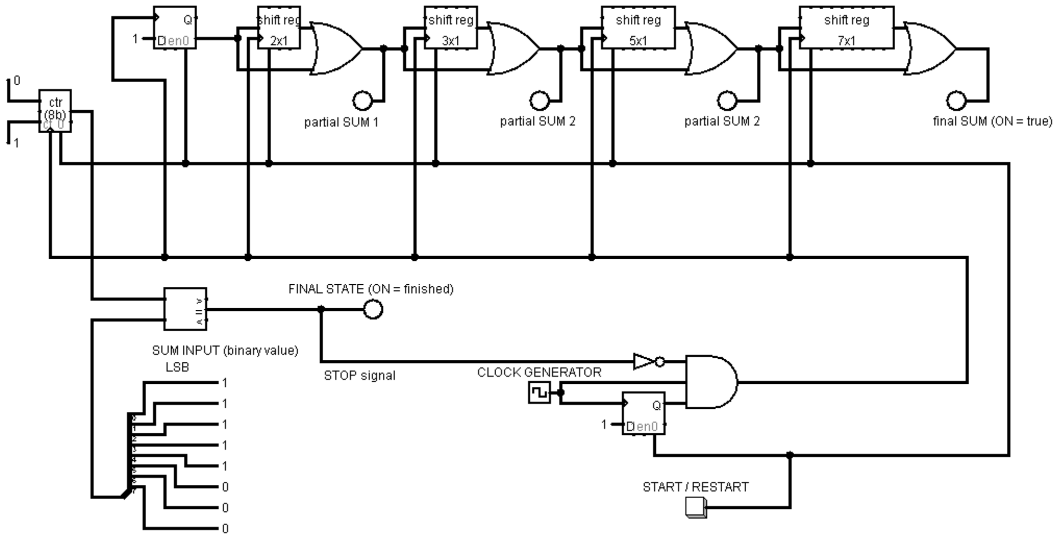


FIGURE 2.4: Electric scheme of the device implementing the Cascade Computing machine.

generator, a module composed by a flip-flop D, a NOT gate and an AND gate is used for stopping and restarting the computation machine.

The START/RESTART button produces an impulse that starts or stops the machine by reseting the counter and the registers. The binary counter receives the clock signal and has the 'Load' inputs set to '0'. The number of bits n for the counter respects the relation $sum < 2^n$ (in this example n = 3).

The source node (S) of the system (see Fig. 2.2) generates a '1' for the first clock period and '0' for the rest. S is implemented using a flip-flop D with the input set on '1'. The negative output of the flip-flop represents the source signal.

Each M_i module (see Fig. 2.3) is composed by a shift register and an OR gate.

The test node (T) verifies if at a given moment the last M_i module output is on '1'. T is implemented as follows: the value sum is specified as a binary number. A comparison bloc generates an impulse of '1' when the counter equals to sum. When this happens the clock will be interrupted and the CC machine stops.

The outputs are signaled by several LEDs. If the LED corresponding to the last module M_i is 'ON' it means the sum may be computed and the decision problem is solved. If the state is 'OFF' it means that the sum cannot be computed.

An improved solution of the CC machine, having the additional capability of identifying the parts of the subset-sum, is presented within the patent [Bartha et al., 2011] (pending).

An electric scheme of the improved CC machine is depicted in Fig. 2.5. This implementation is also based on standard digital components (gates, counters, shift registers, etc.). An example for a set composed by five elements is considered here.

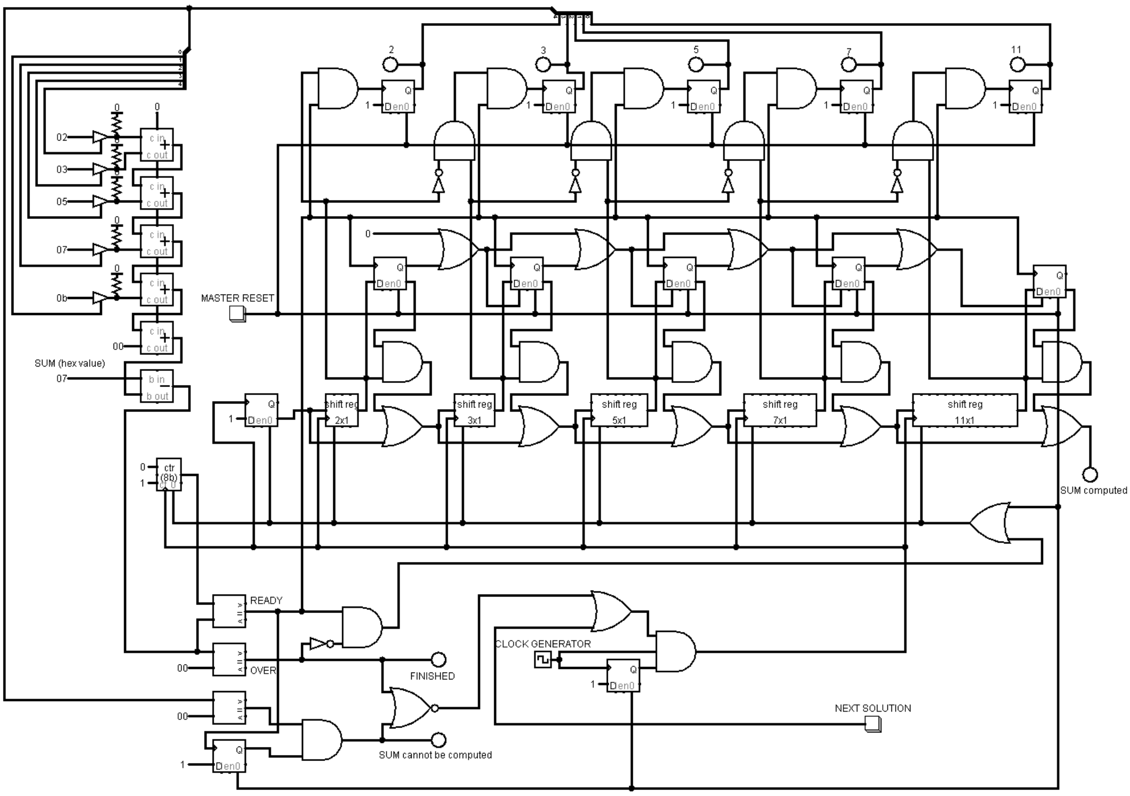


FIGURE 2.5: Electric scheme of the device implementing the Cascade Computing machine with sum parts identification. A five element set example is considered.

The circuit details are not included in this Thesis, they are presented on the patent request [Bartha et al., 2011].

Comparing to the state-of-the-art solutions, the proposed CC model is comparable to the optical-based approach but is is much easier and cheaper to implement, being based on general purpose digital electronic components.

2.2 Complex Adaptive Systems

A different meaning of 'complexity' is related to *complex systems* composed by multiple autonomous entities. This section presents a brief analyze of distributed complex systems.

2.2.1 Definition and examples

A complex system is composed by multiple autonomous entities. *Complex Adaptive System* (CAS) is a term introduced by John H. Holland from Santa Fe Institute [Holland, 1995]. A CAS represents a system composed by autonomous entities which interact, aggregate, learn and adapt their behavior. Typical examples of such systems are: a city (with all its sub-systems/public services), the brain, the biological immune system, a colony of social insects.

Self-organization, non-linear dynamics, and emergence are the most interesting properties of complex systems.

The *intelligent agent* paradigm [Axelrod, 1997], [Niazi and Hussain, 2011] seems to be the best suited as CAS base entity. Beside artificial agents, a CAS may include resources, services, and even human agents.

2.2.2 Properties and mechanisms

Several common *CAS properties* have been identified [Holland, 1995]:

1. *Aggregation or composition.* There are two meaning for aggregation here. The first imply to decide the characteristics of an object that are essential and the characteristics that may be ignored (abstraction). Similar things should be put in same classes (pattern recognition). The second sense of aggregation concerns how a CAS acts and it is related to the emergence of a complex global behavior from simple local actions. Several basic agents may aggregate and form meta-agents [Holland, 1995].

2. *Nonlinear dynamics.* Most CASs cannot be described by linear functions. For instance, in the classical example of a CAS composed by predators and preys

the population dynamics is defined by nonlinear (differential) equations (Lotka-Volterra model [Brauer and Castillo-Chavez, 2001]).

3. *The existence of flows.* Informational, resources, goods, cash or other types of flows exist usually inside a CAS. Flows are produced, processed/transformed and consumed by CAS entities. Effects as multiplication and recycling may appear in these flows [Holland, 1995].

4. *Diversity.* Nature is the most known source of diversity and non-uniformity. A free biological niche is always occupied, a new resource will always attract different consumers, a 'hole' will be always filled (phenomenon called '*convergence*' in biology [Holland, 1995]). What happens with an agent depends on the agent's context (elements external to the agent that may influence it), including other agents. Sometimes symbiotic relations may appear and new types of meta-agents may emerge.

Besides the properties stated before, several CAS *mechanisms* have also been identified [Holland, 1995]:

1. *Tagging.* Tagging is describing an agent (type, inputs, outputs, function, etc.). Tags make possible to check the compatibility between different entities. The agents recognize each other by tags. Tags make possible the aggregation.

2. *Internal models.* An agent uses an internal representation of the external world but also an internal decision model. A common approach to build an internal model is to use *rules*. Heuristic-based algorithms are also frequently used as decision models. Usually the internal model is problem-specific.

3. *Building blocks.* Building blocks are strongly related to the idea of aggregation. How an agent can recognize an infinite number of patterns? How an agent responds properly to an infinite number of situations? The idea is that a finite set of blocks may lead to an infinite number of combinations of these blocks. Decomposition combined with pattern recognition may be used for perception. Composition may be used to generate a large number of complex actions. For instance, a large number of actions may be triggered using a finite set of rules that may be activated simultaneously [Holland, 1995].

4. *Adaptability.* Agents may use a mechanism to evolve behavior rules. A reinforcement-based learning algorithm is proposed in [Holland, 1995]. This algorithm evaluates how efficient is a rule and maintain a score for each rule. The rules may be evolved using a Genetic Algorithm. New rules are created from existing rules using crossover and mutation operators. The rules (behaviors) with the best response are promoted. Such an evolutionary mechanism is capable to generate new valid rules as the system operates.

Emergence is a central aspect of complex systems. Emergence is the property of complex systems to generate sophisticate structural or behavioral patterns from multiple simple actions. Distinct levels may be identified. For instance the chemistry level emerges from the elementary particle physics. The biological level may be considered as emergent from the chemical level. The psychological level emerges from the biological level, etc.

In physics, the laws are not the same at microscopic level and macroscopic level. An insect colony present a complex emergent behavior based on the coordinated action of very simple individuals. This may be explained by the fact that a complex system is more than a sum of entities: it includes also the relations between these entities. For instance, in the human brain the number of neurons is about 100 billion. But they are interconnected via trillions of synapses. It may be observed that it is the number of interconnection which gives the complexity.

In artificial systems, emergence may be a very useful property when we want to create a complex behavior using only simple rules.

2.2.3 Dynamics of complex systems

This section briefly presents some phenomena specific to complex systems such as: *phase transition, critical points, percolation, self-synchronization, chaos, stability, equilibria,* and *convergence.*

2.2.3.1 Phase transitions and critical points

In Thermodynamics (or statistical mechanics), a *phase transition* is a transformation of a system from one state of matter to another state (e.g. from solid to liquid, from liquid to gas, from solid to gas) [Blundell and Blundell, 2006]. In such a system, a *critical point* is a special combination of pressure and temperature with the property that a transition between liquid and gas phases is continuous (a mixed state exists near the critical point) and involves a modification of the internal symmetry ('second order' transition[Altland and Simons, 2006]).

Biological and social systems present similar behaviors. In social systems critical points have been identified regarding the spreading of one particular behavior/strategy [Gladwell, 2000].

Phase transitions in complex systems. Similarly to natural systems, artificial complex systems may also have states, transitions, and critical points. A phase may be assimilated to a pattern in a system structure and/or dynamics [Sole, 2011]. When a certain parameter of a system is tuned and some threshold is passed this phase may change. When a very narrow transition domain separates two distinct phases we have a *critical point* [Sole, 2011].

One of the most known model used to study phase transitions is the *Ising model* [Brush, 1967],[Cipra, 1987], [Sole, 2011]. The model is inspired from the ferromagnetic properties of a material (iron). When such a material is heated to a high temperature the magnetic attraction property disappears. This fact is due to the random ordering (*spin*) of atoms inside the material. At low temperature the atoms influence each other and tend to be ordered. The temperature changes the equilibrium state between the order induced by inter-atoms influences and the disorder introduced by thermal noise [Sole, 2011].

The *Ising* model may be implemented on a computer using a square lattice $N \times N$. Each lattice element represents an atom. Each atom may have two possible states: spin up (+1) or spin down (-1). Each possible state of the lattice represents a system configuration.

The total magnetization is given by the sum of all spins divided by N^2. The magnetization is maximum when all spins are the same. It was demonstrated that an energy function associated to the material has a minimum for this ordered configuration.

The inter-atom influence is implemented using a simple rule: an atom surrounded by other atoms having a same spin will imitate that spin [Sole, 2011].

The temperature effect is introduced by an exponential probabilistic function based on Boltzmann distribution. The probability of having the system in a specific configuration σ is:

$$P(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z},$$

where:

$\beta = (k_B T)^{-1}$, (k_B is the Boltzmann constant, T is the temperature),

$H(\sigma)$ is called the Hamiltonian function and represents the energy of the configuration σ (a formula for H may be found in [Sole, 2011]), and

$Z = \sum_\sigma e^{-\beta H(\sigma)}$ is the partition function and expresses statistical properties of the system having the configuration σ.

The simulation starts from a random configuration. Monte-Carlo method and Metropolis algorithm [Landau and Binder, 2005] are used to iteratively update each atom (lattice cell).

A randomly chosen spin is flipped with a probability Q given by the temperature T and the difference ΔE of energies corresponding to the state with and without the spin flip:

$$Q = e^{-\frac{\Delta E}{k_B T}}$$

Numerical simulations of the *Ising* model indicate that, for specific initial conditions and parameter values, cluster composed by atoms having the same spin emerges. If the temperature is slowly decreased in time the magnetization phenomenon is put on evidence.

A narrow transition interval between zero magnetization and full magnetization is identified on the temperature axis. High magnetization fluctuations in time may be observed in this critical interval [Sole, 2011].

A phase transition phenomenon based on a slightly different model is studied in [Néda et al., 2006]. The analyzed model is an infinite-range '*Potts glass*' structure with unrestricted number of states [Binder and Kob, 2011]. The model can be applied in optimal

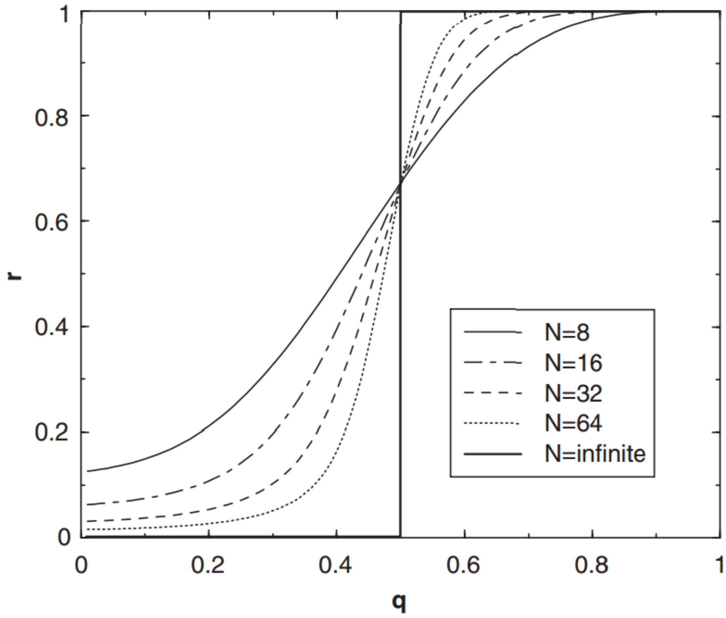


FIGURE 2.6: Phase transitions in a Potts-glass structure (q is the probability to have a specific type of bonds ($+J$) in the 'Potts glass' structure and r is the normalized size of the largest cluster). Source: [Néda et al., 2006].

clusterization problems (i.e. coalition formation in social systems). Phase transitions of this model are illustrated in Fig. 2.6 [Néda et al., 2006].

It may be observed that the normalized size of the largest cluster, r, depends on probability q, where q is the probability to have a specific type of bonds ($+J$) in the 'Potts glass' structure. Transition intervals for q may be observed. When the number N of system states tends to infinity, a percolation-like phase transition is found for the value $q = 0.5$.

The paper [Biroli et al., 2002] analyzes a phase transition phenomenon related to the *satisfiability* (SAT) problem. To solve a SAT problem means to find a solution that satisfies a set of clauses/constraints. A model inspired from Statistical Physics is proposed in [Biroli et al., 2002].

For a 3-SAT problem, when the number N of variables grows, a phase transition appears function of the value of $alpha$, where $alpha$ represents the number of clauses per variable. A large number of solutions can be found if $alpha$ is inferior to 4 and almost no solution if $alpha$ is superior to 5. When N tends to infinity, the transition interval becomes a point: the critical value $alpha = 4.3$ separates the satisfiable problems from the non-satisfiable ones.

Percolation and self-synchronization. *Percolation* [Grimmett, 2010] is a key phenomenon studied in complex systems. An example is the formation of a big cluster in a system where usually only small clusters exist. Such phenomenon also involves a phase transition and may appear for specific values of the system parameters.

Self-synchronizations is another phenomenon widely studied in complex systems composed by independent coupled oscillators [Rosenblum et al., 1997], [Néda et al., 2000]. In particular conditions independent oscillators arrive to be in-phase for a certain period of time (i.e. claps synchronization [Néda et al., 2000]).

Sometimes small perturbations are necessary in order to obtain synchronized clusters [Bartha and Dumitrescu, 2011]. Synchronization is also a fundamental problem in Telecommunications domain, where such models may find an application.

2.2.3.2 Chaos and stability

A chaotic system is a system that is: *deterministic* (does not necessary mean predictable), *nonlinear*, and *unstable* [Smith, 2007]. Unstable means sensitive to initial conditions (presents the 'butterfly effect'). Weather is a typical example of chaotic system.

How can we measure how chaotic is a system? The degree of system sensitivity to initial conditions is characterized by the *Lyapunov exponent* - λ. In non-chaotic systems, a small difference in the initial conditions may lead in time to similar trajectories. In a chaotic system an exponential difference between these trajectories appears.

Let $\delta\mathbf{Z}_0$ be the initial conditions separation/difference. The system sensitivity to initial conditions may be approximated by an exponential function:

$$|\delta\mathbf{Z}(t)| \approx e^{\lambda t}|\delta\mathbf{Z}_0|$$

If λ is positive it means that the trajectories are exponentially divergent and the system is chaotic.

For a dynamic system it is important to define its *phase space*. A phase space is a space (in the mathematical sense) that includes all possible states of a system. For a mechanical system, for instance, this means all possible values of position and momentum.

Each state is represented in the phase space by one unique point. There are as many Lyapunov exponents as dimensions of the *phase space*. The larger exponent is the most significant - as it indicates the higher divergence.

Stability and convergence are usually important desired properties of a complex system. A general principle of stability for feedback control systems says that '*a stable system is a dynamic system with a bounded response to a bounded input*'.

Linear systems are simpler and have been widely studied from the stability point of view. For non-linear systems [Merkin, 1997] there is a stability theory proposed by Lyapunov. According to this theorem, a system is *stable* if the system state $x(0)$ starts near (at a distance δ) an equilibrium point x_e and remains forever (for any time moment $t > 0$) near (at a distance ϵ) to x_e. If the system state converges in time to the equilibrium point we have an *asymptotic stability*. If the system converges to the equilibrium point in time faster than a known rate $\alpha\|x(0) - x_e\|e^{-\beta t}$ we have an *exponential stability*.

The concept of '*equlibrium*' is very important in analyzing a system stability. In the next subsection we analyze the equilibrium notion in the framework of Game Theory.

2.2.4 Game Theory and complex systems

Game Theory (GT) [von Neumann and Morgenstern, Second Edition, 1947], [Binmore, 2007], [Osborne, 2004], [Ross, 2010] is a mathematical discipline that studies how autonomous agents take decisions. Developed for coping with economical problems, GT has been extended to other fields. Recently an important number of contributions based on a GT approach appeared in the Cognitive Radio field [Wang et al., 2010], [Zhang and Guizani, 2011].

A situation may be modeled as a game if the outcome of one agent depends, not only on his/her actions, but also on the actions taken by other agents (strategic interactions). Numerous situations where limited resources are used in common by different competing agents may be modeled as games.

Equilibrium is the GT key concept. An equilibrium state indicate a system stability point. It is important if a game has stable states. In artificial agent-based systems it is possible to avoid undesirable equilibria states by encoding within agents the right rules of behavior and payoff functions. For instance, *Potential games* [Monderer and Shapley, 1996] have been widely used for *Cognitive Radio* applications.

The field who studies how to design rules in order to obtain a desired behavior is called '*Mechanism Design*' (reverse Game Theory) [Reiter and Hurwicz, 2006]. Such an approach may be useful for designing rules for artificial agents.

2.2.4.1 Game definition

A game can be formally described as a system $G = ((N, S_i, u_i), i = 1, ..., n)$, where:

N represents a set of players, n is the number of these players;

for each player $i \in N$, S_i is the set of actions available for player i.

$$S = S_1 \times S_2 \times ... \times S_n$$

is the set of all possible situations of the game. $s \in S$ is named a game '*strategy*' or a '*strategy profile*'.

for each player $i \in N$, $u_i : S \to R$ represents the payoff function of the player i.

The most known equilibrium concept is the Nash equilibrium (NE) [Nash, 1951], [Osborne, 2004]. This concept is formally described in the following section.

2.2.4.2 Nash equilibrium

A game is in a Nash equilibrium state if no player can improve his/her payoff by unilaterally changing his/her action. Therefore, no player has any incentive to deviate from a NE.

Let us denote by (s_i, s^*_{-i}) the strategy profile obtained from s^* we say that by replacing the strategy of player i, s^*_i, with s_i :

$$(s_i, s^*_{-i}) = (s^*_1, ..., s_i, ..., s^*_n).$$

A strategy profile $s^* \in S$ *is a Nash equilibrium if the inequality*

$$u_i(s_i, s^*_{-i}) \leq u_i(s^*),$$

holds $\forall i = 1, ..., n, \forall s_i \in S_i, s_i \neq s^*_i$.

A strict inequality leads to a *strict Nash equilibrium.* If the inequality is not strict we have a *weak Nash equilibrium.*

Nash proven that, if a game has finite number of players and if each player has a finite set of pure strategies to chose from, then at least one mixed Nash equilibrium exists. An example of mixed strategy is to chose one pure strategy with a probability p and another pure strategy with a probability $(1 - p)$.

The Nash equilibrium is important in CAS because it represents a stable state of the game/system. Since no player can increase its payoff by unilateral deviation, no selfish player will have an incentive to deviate from that state. Players are expected to be selfish in the sense that each one will try to maximize his/her payoff. The payoff is usually expressed by a difference between a good-put/utility and a cost.

If a game has a Nash equilibrium it is possible to design a *distributed algorithm* that will lead the system towards that Nash equilibrium. Such a general algorithm is proposed, for instance, in [Pradelski and Peyton, 2012]. This means that a decentralized system where agents use such an algorithm converges to a stable state.

But is Nash equilibrium always an optimum state for the system? Unfortunately, in many games this is not the case.

One of the most famous game is the Prisoner's Dilemma (PD) [Dawes, 1991]. In this game the Nash equilibrium is far from being optimal (the reason to call this game a 'dilemma'). Another class of games where Nash equilibria are inefficient are those included in the 'tragedy of the commons' class [Ostrom, 1990].

Besides Nash equilibrium, other types of game equilibria have been defined. Some examples are presented in the next sections.

2.2.4.3 Pareto equilibrium

Let consider two strategies s and $s^* \in S$. A strategy s *Pareto dominates* another strategy s^* if the payoff of each player i using strategy s is greater than or equal to the payoff of the same player using strategy s^*, and at least one payoff is strictly greater [Cremene et al., 2013].

A strategy profile $s^* \in S$ is Pareto efficient, when it does not exist a strategy $s \in S$, such that

$$u_i(s) \geq u_i(s^*), i \in N,$$

with at least one strict inequality.

A Pareto efficient strategy profile has the property that no other strategy can increase one player's payoff without decreasing any other player's payoff [Fudenberg and Tirole, 1991].

2.2.4.4 Lorenz equilibrium

Lorenz dominance has been introduced in [Kostreva and Wodzimierz, 1999] as a particular case of Pareto dominance. Therefore, Lorenz solutions are a subset of Pareto solutions set: the most equitable and balanced Pareto solutions.

Let $s \in S$ be a strategy profile vector.

Let us consider $u_{(1)}(s) \leq u_{(2)}(s) \leq ... \leq u_{(n)}(s)$ as the components of $u = (u_1(s), u_2(s), ..., u_n(s))$ sorted by increasing order.

The generalized Lorenz vector associated to s is:

$$L(s) = (l_1, ..., l_n),$$

where:

$$l_1 = u_{(1)}(s),$$

$$l_2 = u_{(1)}(s) + u_{(2)}(s),$$

...

$l_n = \sum_{i=1}^{n} u_{(i)}(s)$.

For $s', s'' \in S$ Lorenz dominates if:

$$s' \succ_L s'' \Leftrightarrow L(s') \succ_P L(s''),$$

where $\succ_P$ denotes the Pareto-domination.

2.2.4.5 Joint Nash-Pareto equilibrium

Nash-Pareto and Pareto-Nash joint equilibrium concepts [Dumitrescu et al., 2009] define game situations where players are biased towards different types of rationality: Nash or Pareto. The two types of rationality may be associated to different agent behaviors: either oriented towards maintaining a certain payoff, not deviating unilaterally from the current strategy (Nash-biased) or oriented towards getting maximum payoff (Pareto-biased) [Cremene et al., 2013].

Chapter 3

Adaptation through optimization

The service adaptation problem is discussed in this chapter. A centralized control mechanism is considered (the distributed control case is discussed in Chapter 4). We propose an approach for transforming the adaptation problem into an optimization problem. Thus, it becomes possible to design a general control mechanism for service adaptation. Complexity issues related to search space and number of optimization objectives are analyzed.

3.1 What is 'service adaptation'?

Service adaptation was informally defined in Chapter 1 using an analogy with a biological organism, which adapts to its environment, having as goal to survive and reproduce itself.

Another definition of the term 'adaptation' is given by a general law inspired from Psychology [Martin H. et al., 2009]: '*Every adaptive system converges to a state in which all kind of stimulation ceases*'.

A large number of artificial adaptive services have been proposed in literature. Examples of such systems have been investigated in [Cremene et al., 2007]. Usually, by service adaptation we understand a modification in the service behavior and/or structure when the service's context changes. In general, the adaptation goal is to efficiently use available resources and/or to obtain a higher user satisfaction.

A common approach for service adaptation is to use rules. The most known model is *Event-Condition-Action* (ECA): an *event* is generated from the context, a *condition* is verified and a specific *action* is triggered. Let consider a simple service for sending messages installed on a mobile phone. An adaptation rule may be: *when the WiFi connection drops (the event), if a GSM connection is available (the condition), then the message will be sent by one or more SMS messages (the action).*

According to this approach, the problem is reduced to the selection of the adequate behavior from several possibilities. A rule verifies the context and service state and triggers a specific action that is meant to put the service a new state with a higher benefit for the user. Strategy design pattern or more sophisticated paradigms like '*multifaceted component*' [Rarau et al., 2007] may be used for specifying the predefined behavior.

The most important inconvenient of the ECA approach is the fact that the events, conditions and actions are service-specific [Cremene et al., 2007], [Cremene et al., 2012c]. This means that such a mechanism is not reusable from one service to another. More important, a human expert should anticipate in advance all the rules. This is why we call it 'adaptation by anticipation'.

Unanticipated adaptation may be achieved if the adaptation control mechanism is service-independent. But is it possible to design general, service-independent adaptation rules? In [Cremene et al., 2007] we prove that this is possible if the services are semantically described by meta-data called '*profiles*' (a similarity with the CAS *tags* discussed in Chapter 2 exists).

Our research about unanticipated adaptation conducted us to the idea that the adaptation problem may be modeled as an optimization problem. Optimization is a term with a simple and precise mathematical definition, which is not the case of the term 'adaptation'.

3.2 Solving service adaptation as an optimization problem

Service adaptation may be modeled as an optimization problem if the adaptation goal may be expressed in terms of functions maximization or minimization [Cremene et al., 2011b]. In this section we describe our own solution to this problem.

In [Cremene et al., 2012c] we present a generic model of the service and its context. We call it the *Service-Context (S-C) model* (Fig. 3.1). The model is represented by a directed graph describing: the service, the context and the interactions between them. The graph elements are:

a. *Nodes* corresponding to services (S) or context (C) entities. A composite service may be described by a graph of basic services. The context is seen from the service point of view. Three basic context entities are defined: users (U), infrastructure elements (I) (i.e. devices, networks) and environment (E) (i.e. physical and social environment elements).

b. *Edges* corresponding to interactions existent between the nodes. We define three types of interactions:

  - *Informational flows* interactions. Services are communicating between them by exchanging information. Users also exchange information with services.
  - *Resource utilization* interactions. Services consume resources provided by the hardware infrastructure (memory, CPU, bandwidth, etc).
  - *Environment* interactions. The environment may influence the user needs (e.g. location effect on user preferences) or the infrastructure (e.g. the spatial coverage or rain effect on a radio network).

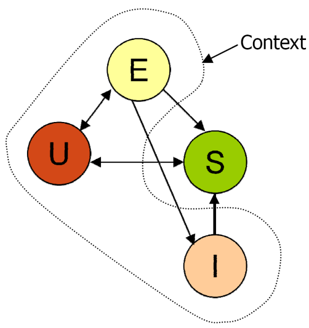


FIGURE 3.1: The Service-Context model represents the services, the context entities and their interactions as a directed graph. Source: [Cremene et al., 2012c]

Fig. 3.2 depicts an examples of the S-C model for a service with four components (inner services) S_1, S_2, S_3, S_i. The first three services are provided by P_1 and S_i is provided by the provider P_2. S_1 is a client component and it directly interacts with a user U, which is influenced by it's environment E (location, social environment, etc.). The infrastructure element I is the client terminal (PC, mobile phone, etc.).

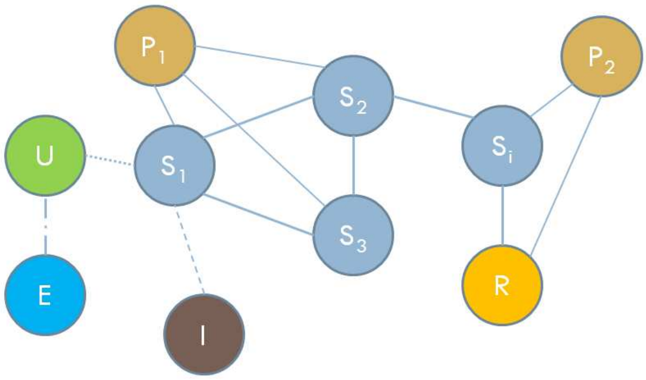


FIGURE 3.2: An example of the Service-Context model.

The *S-C model* offers the possibility to define the *Service-Context adequacy*. S-C adequacy is an objective measure that indicates how well suited is a service S for a given context C.

The *S-C adequacy* is a vector that includes several metrics defined for each *Service-Context attribute*. An example is the '*memory*' attribute: the adequacy means that the service should use less memory than the memory provided by the infrastructure.

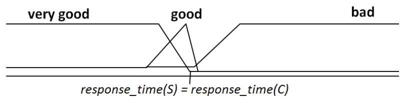
Another attribute may be the '*language*': a service is adequate from the language point of view if the user and the service use the same language. Another example may be '*location*': a location-related service is adequate if it provides information specific to user location.

For each attribute, which is common to context and service, an *adequacy measure* called *S-C distance* is defined. Examples of *S-C attributes* with corresponding distances are described in Fig. 3.3.

The S-C adequacy may be seen as a generalization of the *Quality of Service* (QoS) concept. Common QoS attributes such as: response time, reputation, availability, throughput, may be used for defining S-C adequacy measures related to user-service interaction.

In order to compute the S-C adequacy a problem needs to be solved in a first place: S-C *attribute aggregation*. Attribute aggregation means to automatically compute the attributes for a composite service when the attributes of the inner services are known.

For each S-C attribute some aggregation operators are defined. Examples of such operators are given in [Cremene et al., 2007] and [Cremene et al., 2012c]. For the QoS

Information flow distance evaluation		
Attribute name	**Domain**	**Distance**
response_time	*0..inf*	
security_level	*{none, low, medium, high, very high}*	*d = 0 if security_level(S) >= security_level(U) and ∞ else*
language	*{RO, FR, DE, ES……}*	*d = 0 if language(S) == language(U) and ∞ else*

Resource utilization distance evaluation		
Attribute name	**Domain**	**Distance**
memory	*0…100%*	*d = memory(S) / memory(I) * 100*
cpu_time	*0..100%*	*d = cpu_time(S) / cpu_time(I) * 100*
network_capacity	*0..100%*	*d = network_capacity(S) / network_capacity(I) * 100*

FIGURE 3.3: Service-context adequacy expressions for different attributes described as S-C distances. Source: [Cremene et al., 2012c]

attributes aggregation operators examples are presented in [Pop et al., 2011b] and [Suciu et al., 2012]. An example about how the physical distance may be included also as S-C attribute is given in [Pop et al., 2011a].

A special case of S-C adequacy is based on the distance between a request expressed by the user in natural language and the function provided by the service. In order to compute this distance, the service needs to be semantically described. A dictionary is used for measuring the similarity between the words used by the user and the semantic concepts attached to the services. This problem was addressed in the papers [Cremene et al., 2009], [Pop et al., 2009] and [Pop et al., 2010a].

In this section we have explained how the adaptation problem may be modeled as an optimization problem by introducing the notion of service-context adequacy or distance. According to this approach, an adaptation problem is reduced to a multi-objective optimization problem [Cremene et al., 2011b] where several S-C distance functions need to be minimized or maximized. Since these functions may have any form, an heuristic search algorithm may be used in this respect.

3.3 Computational complexity aspects

Complexity issues arise in the case of optimization/search from two directions: the search space and the number of optimization objectives. In this section we examine in more detail these issues.

3.3.1 Search space issues

Using search/optimization algorithms makes sense only if a direct way to find the solution cannot be found. If the search space is very small an exhaustive algorithm may solve the problem. But such an approach cannot be used for medium and large search spaces. We analyze further the situations where the solution needs to be searched in a large space.

A composite service is described as a process (workflow) that involves the execution of several activities. Executing an activity means invoking a service. For each activity, which is assimilated to an *abstract service* several *concrete services* exist. Each concrete service has different QoS properties.

Given m abstract services and n concrete services for each abstract service, there are n^m possibilities for the same composite service, each having a different QoS. The search space is discrete. For each abstract service one concrete service must be selected. Any combination is possible - we have a combinatorial optimization problem. Finding the solution with the optimal QoS is an *NP-hard* problem. This generally means that we need heuristic algorithms for solving such a problem in reasonable time.

The QoS optimization problem may be seen as a particular instance case of service adaptation. The same optimization principle may be applied for any S-C attribute. An important aspect is to be aware about the search space characteristics. For instance, in the case of QoS optimization problem we start from a given abstract workflow supposed to be functionally valid. This means that any combination of concrete services - any solution from the search space - is valid. The only difference between different solutions is given by the QoS properties (non-functional properties).

If we try to automatically generate the abstract workflow starting from a user request we have a completely different problem. This problem have been addressed using different techniques such as *AI planning* [Rao and Su, 2005]. We proposed a method for generating composite services starting from a user request expressed in natural language [Cremene et al., 2009], [Pop et al., 2009], [Pop et al., 2010a]. The AI planning and the QoS optimization may be combined in a more complete solution for service composition and adaptation/optimization [Pop et al., 2011a].

The initial hypothesis about the problem input may change significantly the type of approach. Sometimes the problem is to find valid composite services by combining other existing basic services [Pop et al., 2010b]. In this case the search space includes also invalid solutions. This approach requires measures for the solution validity. Syntactic matching measures [Pop et al., 2010b] or model checking tools may be used in this purpose [Todica et al., 2012].

Is it always possible to model an adaptation problems as an optimization one? Usually the adaptation involves a goal. Sometimes this is not explicitly expressed by the human expert. For instance, in the example with the messaging service presented before, the adaptation goal is to keep the service functional. The functionality will be clearly interrupted if the service tries to send the message by e-mail without having a data connection. *Utility* may be a useful concept here. The *utility* of such a service is lower than the utility of a service who uses the SMS when a WiFi connection is not present. In general, any rule can be described as a utility, efficiency or cost function.

A problem may be the form of the objective/evaluation function. During the search process it is useful to have a precise measure of how close/far is the process from a solution.

Different solutions from the search space should have different evaluation values. The error surface (*fitness landscape*) may significantly influence the optimization algorithm performances. The form of the objective functions gives the error surface.

All techniques discussed previously share a common idea: there is a search space and in this space an algorithm searches for a solution. This solution has specific characteristics measured by the various distances included in the S-C adequacy concept. By transforming an adaptation problem into an optimization one, domains like *Service-oriented*

Computing, *Context-aware*, *Ubiquitous* and *Pervasive Computing* are connected to domains like *Search Based Software Engineering* [Harman et al., 2012] and *Computational Intelligence*.

3.3.2 Multi-objective optimization issues

A second issue related to the computational complexity in optimization problems is generated by the number of optimization objectives. This number is given by the number of S-C attributes. In order to reduce the number of objectives, a solution is to aggregate all objectives into a unique global score.

Unfortunately, there is no precise theory about designing aggregation functions. Several types of functions may be used. The most common is a linear combination (weighted sum). When a weighted sum aggregation function is used, the optimization algorithm finds one solution from the Pareto set [Miettinen, 1999; Kathrin and Tind, 2007]. However, there is no direct correspondence between weighting coefficients and the relative importance of the objective functions [Miettinen, 1999]. Additionally, non-convex parts of the Pareto set cannot be reached by minimizing convex aggregation functions [Miettinen, 1999].

In a multi-objective optimization algorithm the number of objectives may be a problem. Most algorithms are efficient for 3-4 objectives. If more objectives exist, a *many-objective* algorithm may be used [Zhou et al., 2011].

Another problem is that, multi-objective algorithms provide, not just one solution but a set of solutions named the *Pareto-optimal* set. If only one solution is needed a decision making algorithm is also necessary. Comparing to single-objective algorithms, the multi-objective ones are usually 3-4 times slower but, combined with a decision making algorithm, they offer a more precise control of the solutions.

3.4 The most relevant contributions related to Search-based Software Engineering

This section presents the most relevant contributions of the author based on the use of search/optimization algorithms for service composition/adaptation. *Search-based Software Engineering* is the field where all these contribution may be included.

3.4.1 QoS-based service optimization using Differential Evolution

Web service composition aims to provide new functionalities by combining existing ones. A certain functionality may be offered by different web services, each one having different quality of service (QoS) attributes. The goal is to find the composite service with the best QoS. This problem is defined in details in the next section.

3.4.1.1 Problem statement

A composite service is defined by a business workflow that includes a set of abstract services. A set of concrete service implementations exist for each abstract service. The goal is to find the optimal combination of concrete services. A process example for a *flight booking process* is illustrated in Fig. 3.4 [Pop et al., 2011b].

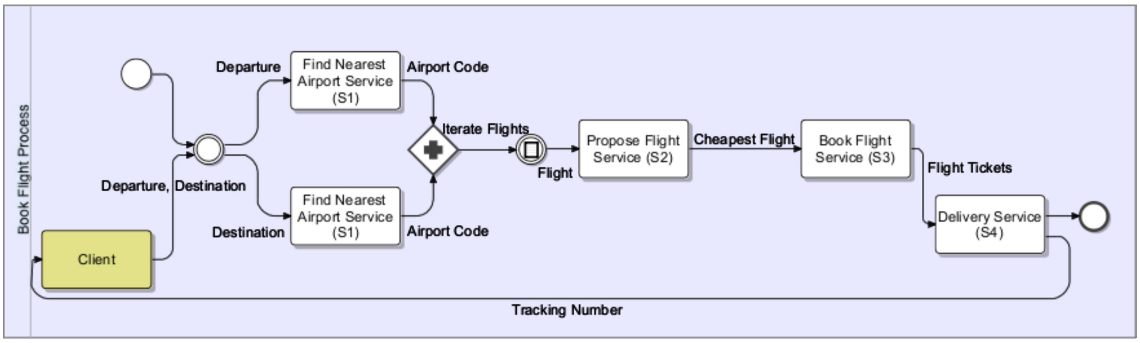


FIGURE 3.4: Example of a flight booking abstract process. Source: [Pop et al., 2011b]

The process illustrated in Fig. 3.4 includes the following abstract activities: *Find Nearest Airport* for identifying the airport that is closest to the desired departure or destination location, *Propose Flight* for finding all flights that match a certain criteria, *Book Flight* for the actual purchase of the flight tickets and *Delivery* for mailing the tickets and the receipt.

Executing an activity means invoking a web service. For each *abstract service* S1, S2, ... (see Fig. 3.4), several *concrete services* having different QoS properties exist.

The QoS parameter set includes the following parameters: *response time, reliability (r), availability (a)* and *cost (c)*. The *response time (t)* or *latency* is a measure for the performance of a service. It represents the round-trip time between sending a request and receiving the response. *Reliability (r)* represents the capacity to ensure reliable message delivery for services. The probability of a service to be ready for immediate use is called *availability* (a). The *cost* (c) is the price to pay for each service request.

The QoS of a composite service is obtained by aggregating the QoS attributes of its inner services. The aggregation operations depend on the composite service specific architecture.

Fig. 3.5 [Pop et al., 2011b] depicts the QoS aggregation formulas for each architectural pattern.

QoS Property	Flow	Sequence	Switch	While
Response Time (T)	$\max_{i \in 1..m} \{t_i\}$	$\sum_{i=1}^{m} t_i$	$\sum_{i=1}^{m} p_i \cdot t_i$	$k \cdot t$
Reliability (R)	$\prod_{i=1}^{m} r_i$	$\prod_{i=1}^{m} r_i$	$\sum_{i=1}^{m} p_i \cdot r_i$	r^k
Availability (A)	$\prod_{i=1}^{m} a_i$	$\prod_{i=1}^{m} a_i$	$\sum_{i=1}^{m} p_i \cdot a_i$	a^k
Cost (C)	$\sum_{i=1}^{m} c_i$	$\sum_{i=1}^{m} c_i$	$\sum_{i=1}^{m} p_i \cdot c_i$	$k \cdot c$

FIGURE 3.5: QoS aggregation formulas. Source: [Pop et al., 2011b]

Since '*flow*' means executing several activities in parallel, the total response time (T) is given by the maximum response time of all executed activities. The '*sequence*' involves to execute a service one after another therefore the total response time is a sum.

For a '*switch*' construct, the process needs to be monitored at runtime during multiple executions to determine the probabilities p_i associated to each case branch, $\sum_{i=1}^{m} p_i = 1$, where p_i represents the probability to select case branch i. In case of the '*while*' loop the average number of iterations k is the time multiplication factor.

To evaluate the quality of each potential solution (composite service) we consider an Aggregate Objective Function (AOF) having the following form:

$$F(y) = w_1 \cdot R + w_2 \cdot A + \frac{w_3}{T} + \frac{w_4}{C} \tag{3.1}$$

where w_i are the weights that correspond to the significance of each QoS attribute for the user and $R,\ A,\ T,\ C$ are the aggregate QoS values for the composite service.

3.4.1.2 Proposed approach

A Differential Evolution (DE) based solution to the QoS optimization problem is proposed in [Pop et al., 2011b]. This NP-hard problem is well known in the SOC field. Most of recent contributions indicate the Genetic Algorithms (GA) as the best approach for complex workflows. We propose a new approach based on Differential Evolution (DE) and a new encoding scheme.

Three DE variants are explored: *TruncDE* [Lampinen and Zelinka, 1999], *XueDE* [Xue et al., 2003], and a new method called *LongDE* [Pop et al., 2011b]. The LongGA/DE version is based on a genotype that facilitates discretization [Pop et al., 2011b]. We compare these algorithms with the genetic algorithm proposed by Canfora et al. in [Canfora et al., 2005] named *IntGA*. We also test a modified version of *IntGA* that uses the long genotype - *LongGA*.

3.4.1.3 Genotype encoding (solution representation)

Let S_A=$\{S_{A_1},\ S_{A_2},\ ..,\ S_{A_m}\}$ be the set of abstract services, S_{C_i}=$\{S_{C_{i,1}},\ S_{C_{i,2}},\ ..,\ S_{C_{i,n}}\}$ the set of concrete services, and $Q_{i,j}$=$(t,\ r,\ a,\ c)$ the vector of QoS properties (*response time* - t, *reliability* - r, *availability* - a and *cost* - c) for $S_{C_{i,j}}$.

The genome used for *TruncDE* and *XueDE* algorithms is depicted in figure 3.6. This type of genome encoding is the same as proposed in [Canfora et al., 2005]. An array of integer values having the length equal to the number of abstract services in S_A contains in each cell (gene) the index of the concrete service.

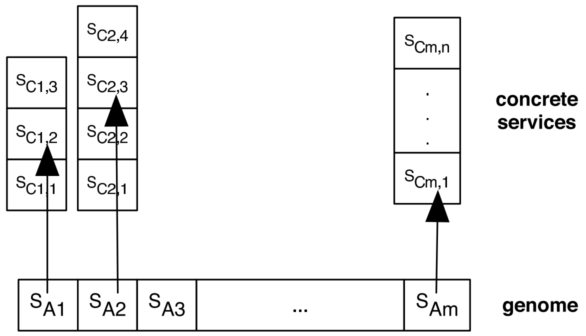


FIGURE 3.6: Genome encoding for *TruncDE* and *XueDE* algorithms. Solution proposed in [Canfora et al., 2005].

Most of DE discretization approaches involve either a bi-directional transformation from a discrete domain to a continuous domain $\mathbb{R}$ (e.g. *TruncDE*) or attempt to modify the canonical form of DE algorithm (e.g. *XueDE*). We propose a new genome encoding that facilitates discretization without changing the original DE algorithm or requiring any data transformation.

Figure 3.7.*a* illustrates the proposed genome encoding scheme. A gene encodes each concrete service in S_{C_i} that can realize the abstract service S_{A_i}. The value stored in the gene represents the preference for choosing the corresponding concrete service.

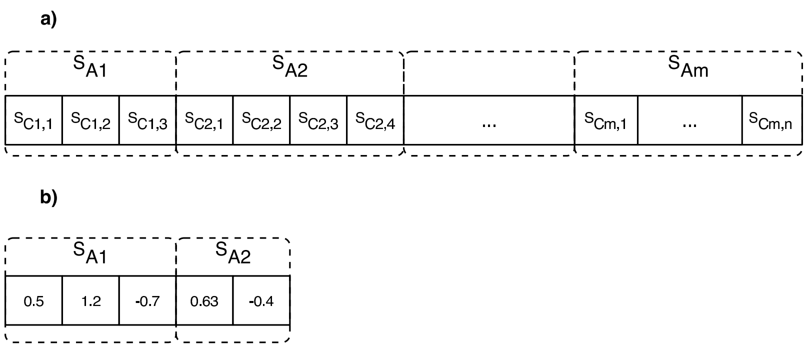


FIGURE 3.7: Genome encoding scheme for *LongDE* algorithm - a) proposed genome, b) an example for two abstract services. Source: [Pop et al., 2011b]

For instance, if the process consists in 2 abstract services S_{A_1} and S_{A_2}, and for S_{A_1} there are 3 alternatives ($S_{C_{1,1}}$, $S_{C_{1,2}}$ and $S_{C_{1,3}}$) and for S_{A_2} there are 2 alternatives ($S_{C_{2,1}}$ and $S_{C_{2,2}}$), then the genome would be similar to the one depicted in Fig. 3.7.*b*. The greater the value of the allele, the most likely the corresponding concrete service, will be selected. In our example, the preferred alternative for S_{A_1} is $S_{C_{1,2}}$, while for S_{A_2} is $S_{C_{2,1}}$. The preferences values are initialized randomly between 0 and 1. On each generation, these preferences are updated according to the DE algorithm.

3.4.1.4 Experimental results

Experiments were conducted for 25 scenarios that include all combinations of $m \in \{10, 20, 30, 40, 50\}$ abstract services and $n \in \{10, 20, 30, 40, 50\}$ concrete services. The data for the considered scenarios is generated based on a normal distribution. Each scenario is executed 100 times and the results are averaged. Statistical experiments are conducted using the data from the first 500 generations of the 25 considered scenarios in order to test the significance of the results. For each generation, the best and the mean individual fitness is considered.

Fig. 3.8, 3.9, and 3.10 depict the evolution of the best fitness for three test scenarios.

A first scenario involves a process consisting in *m=10* abstract services and *n=20* concrete services per abstract service. The results are depicted in figure 3.8.

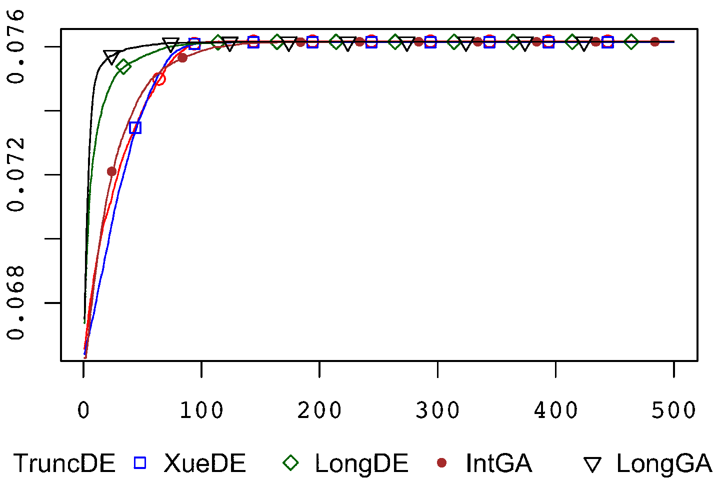


Figure 3.8: The evolution of the best fitness over 500 generations for m=10 abstract services and n=20 concrete services. All algorithms converge within about 100 generations, *LongGA* and *LongDE* are the first to converge. Source: [Pop et al., 2011b]

It may be observed that all algorithms converge within about 100 generations. However, *LongGA* and *LongDE* are the first to converge.

A more complex scenario, involving *m=20* abstract services and *n=40* concrete alternatives is depicted in Fig. 3.9.

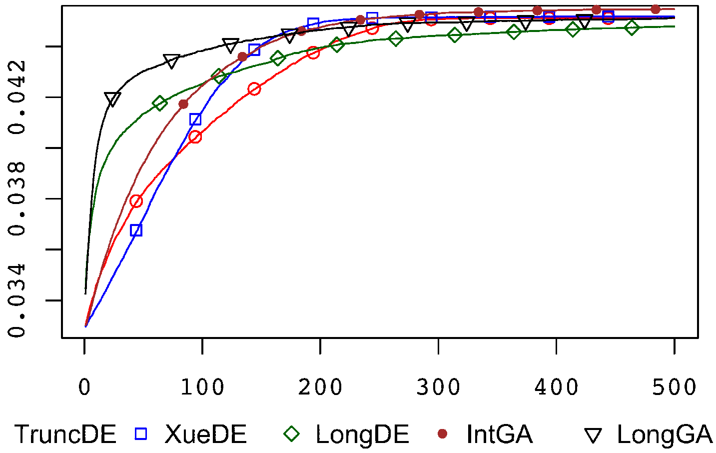


FIGURE 3.9: The evolution of the best fitness over 500 generations for m=20 abstract services and n=40 concrete services. *LongGA* has the best performance. *TruncDE* is the last one that converges. *LongDE* quickly finds good individuals, but then the best fitness increases at a slow rate. Source: [Pop et al., 2011b]

It may be observed that all algorithms require about 200 generations to converge. Compared with the previous experiment, this indicates that, as expected, a higher problem complexity requires more time to find a good solution. *LongGA* has the best performance. *TruncDE* is the last one that converges. *LongDE* quickly finds good individuals, but then the best fitness increases at a slow rate.

The last scenario is based on a process that generates a very complex optimization problem: *m=40* abstract services, each of them having *n=40* alternatives (Fig. 3.10).

The results of this experiment are quite similar to the previous scenarios. However, some differences appear. The convergence time is longer: about 300 generations. This is normal because the search space is much higher. *LongGA* finds the best solutions in short term (until generation no. 220). In long term (more than 300 generations) *LongGA* and *LongDE* are slightly surpassed by *XueDE* and *IntGA*. Comparing with the previous experiments, it may be noticed that, as the complexity increases, *TruncDE*'s rate of convergence decreases.

As a conclusion, the results indicate that the algorithms based on the proposed long genotype (*LongGA*, *LongDE*) outperform the other algorithms for scenarios of low and

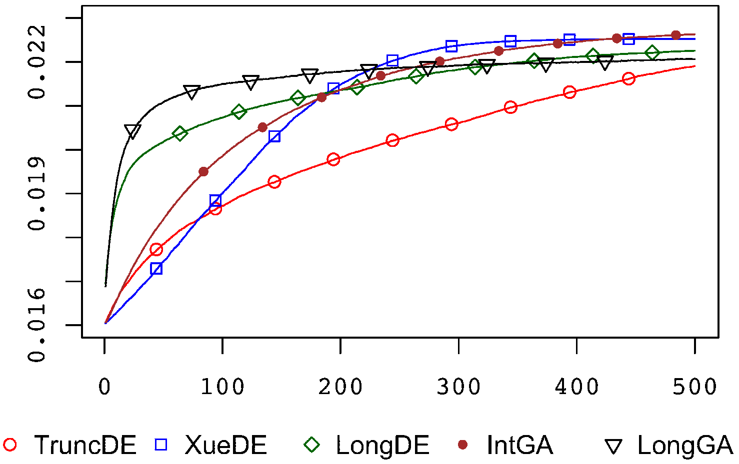


FIGURE 3.10: The evolution of the best fitness over 500 generations for m=40 abstract services and n=40 concrete services. Source: [Pop et al., 2011b]

medium complexity (up to 40 abstract services and 40 concrete implementations per abstract service). Of all implemented DE variants, *XueDE* proved to have the most robust behavior relative to variations of the problem complexity, the range of evolutionary parameters (population size, mutation probability, etc.) and the QoS properties distribution.

3.4.2 Multi-objective Optimization for QoS-aware Service Composition

A multi-objective optimization approach for QoS-aware Service Composition is analyzed in [Cremene et al., 2014a]. Selecting the QoS-optimal services for a composition flow is a multi-objective NP-hard problem. Most of existing approaches reduce this multi-objective problem to a single-objective one. A global evaluation function that combines different criteria into a unique score is considered. However, defining such a function is not a trivial task and the end user solution selected from the Pareto set cannot be predicted. Therefore, the final solution may not reflect the user preferences.

3.4.2.1 Proposed approach

We propose a different approach [Cremene et al., 2014a] based on two steps: a) detect the set of Pareto-optimal solutions and b) use a decision making algorithm for selecting a solution from the Pareto set. Several evolutionary multi-objective algorithms are evaluated. Experiments are based on real QoS data obtained from a public database.

3.4.2.2 Performance evaluation metrics

In order to compare different MO algorithms an objective evaluation measure is necessary. When comparing different algorithms we cannot rely only on one indicator [Okabe et al., 2003]. Therefore, several different metrics are used for comparison [Coello et al., 2007]. The most important are the following:

- The *hypervolume (HV)* defines the coverage area of a known Pareto Front (PF) with respect to the objective space. It represents the sum of all rectangular areas bounded by a reference point and $(f_1(x), ...f_m(x))$. A higher value for *Hypervolume* indicator is desired, meaning that a greater portion of the Pareto front is covered.
- The *spread (S)* measures the spread among the obtained PF and is defined as:

  $$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1) \cdot \bar{d}},$$

  where d_i is the Euclidean distance between consecutive solutions, d_f and d_l are Euclidean distances between extreme solutions and boundary solutions of the obtained PF, and N is the number of solutions from the Pareto front. If d_i has a large variance Δ can be greater than one. In the case of a good distribution (uniform spread of solutions) all distances d_i are equal to $\bar{d}$ and Δ is zero.
- The *set coverage (C)*. If X and Y are two approximations of the optimal PF, the *set coverage* *C(X, Y)* is used to measure the percentage of solutions in Y dominated by at least one solution in X. A value of 1 indicates that all solutions in Y are dominated by some solutions in X. If $C(X, Y) = 0$ then no solution in Y is dominated by a solution in X. When comparing two algorithms it is desirable that the better algorithm has a higher value for $C(X, Y)$ and a lower value for $C(Y, X)$.

The metric is determined as follows:

$$C(X,Y) = \frac{|\{u \in Y | \exists v \in X : v \prec u\}|}{|Y|}$$

.

3.4.2.3 Comparative experimental results

A set of comparative tests meant to compare the state-of-the-art algorithms are conducted. The considered algorithms are: NSGA2 [Deb et al., 2002], SPEA2 [Zitzler et al., 2001], MOEA/D [Zhang and Li, 2007], GDE3 [Kukkonen and Lampinen, 2005b], and POSDE [Chang et al., 1999].

Workflows with different complexities and real QoS data are evaluated. Abstract workflows of various complexity are randomly generated. The complexity is adjusted by increasing the number of abstract services from 10 to 30. A fixed number l of abstract services is considered for each tested scenario. The number of concrete services k is varied from 10 to 90.

Single-objective approaches limitations. Fig. 3.11 and 3.12 compare the solutions obtained using a single-objective optimization algorithm and a multi-objective one. Services composition problems with two and three objectives are considered. The results confirm that single-objective algorithms find unbalanced solutions situated in one extremity of the Pareto set.

For the single-objective approach we use two types of global evaluation functions: a) the function defined in [Canfora et al., 2005] and b) an additive function - a convex combination of each objective. Different weights are used. The global evaluation functions are defined as follows:

$$F_{Canfora} = \frac{w_1 \cdot d + w_2 \cdot a}{w_3 \cdot t},$$
$$F_{aditive} = w_1 \cdot d + w_2 \cdot a + \frac{1}{w_3 \cdot t},$$

where w_i represent predefined weights and the objectives correspond to the QoS properties: throughput - d, response time - t, and availability - a.

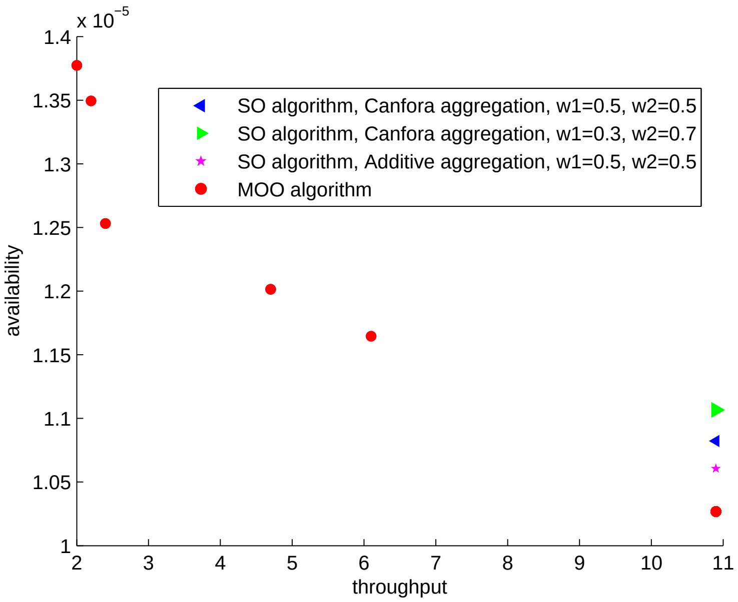


FIGURE 3.11: Solutions found by single-objective and multi-objective optimization algorithms. A two-objective problem is considered. All aggregation functions used for the single-objective algorithms lead to the same type of unbalanced solutions. the Source: [Cremene et al., 2014a]

Hypervolume-based comparisons. Figures 3.13a, 3.13b, and 3.13c present the average values of the normalized Hypervolume indicator for different workflows complexities (higher HV is better). All combinations of $l \in \{10, 20, 30\}$ abstract services and $l \in \{10, 20, ..., 90\}$ concrete services are evaluated. In order to have statistical significance 50 independent runs are executed for each combination.

Fig. 3.13a, 3.13b, and 3.13c indicate that, on average, all tested algorithms behave very similarly (*POSDE* gives slightly higher values).

The next experiment studies the HV dispersion (lower values indicate higher performance). A business workflow with 10 abstract services and 80 concrete services is considered.

Fig. 3.14 and 3.15 present box-plots of the normalized hypervolume values obtained for each algorithm over 50 independent runs for 2 and 3 objectives, respectively. Looking at the standard deviation of the HV average we can see that *POSDE* and *NSGA2* are more stable/robust than the other algorithms. Experiments have shown that this happens for all combinations of abstract and concrete services.

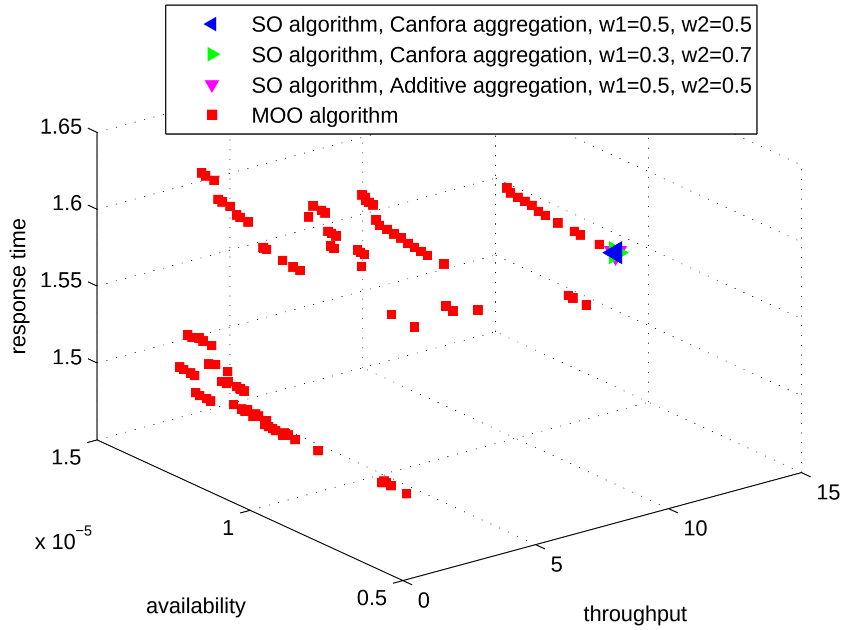


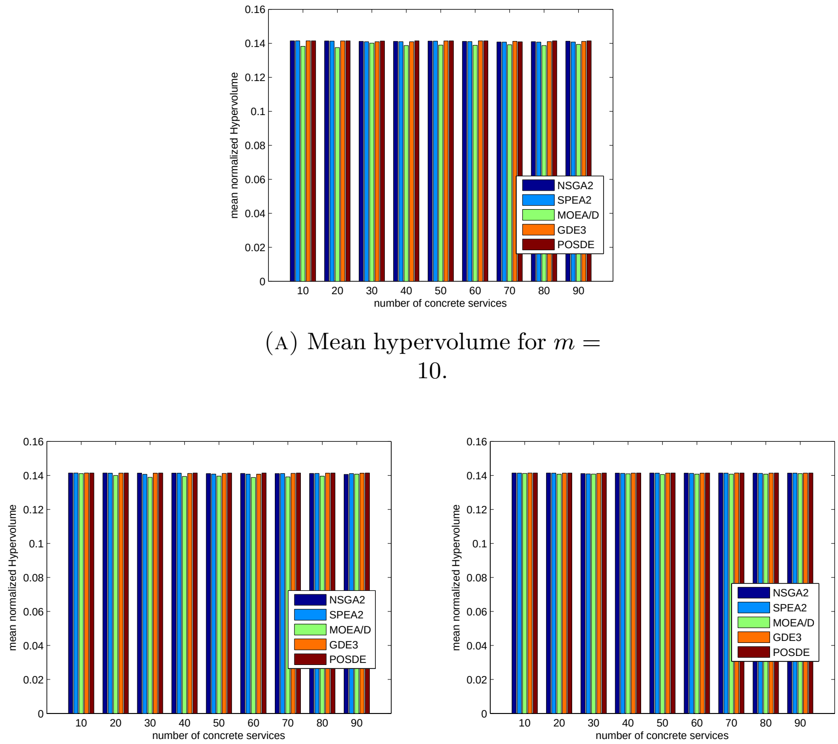
FIGURE 3.12: Solutions found by single-objective and multi-objective optimization algorithm. A three-objective problem is considered. All aggregation functions used for the single-objective algorithms lead to the same type of unbalanced solutions. Source: [Cremene et al., 2014a]

Spread-based comparisons. For the two-objective problem the spread indicator have also been computed. Fig. 3.16 presents the average spread values (lower is better). Results indicate that all algorithms assure a good diversity (POSDE yields slightly better results in terms of spread).

Set Coverage-based comparisons. Set Coverage quality indicator measures the percentage of solutions found by an algorithm A that Pareto dominate the solution set found by an algorithm B. From the average values, Fig. 3.17, we can see that, in general, there is a difference in the solution set. POSDE and NSGA2 are the algorithms that yield the best results. A large percentage of the found solution set Pareto dominates the solution set given by the other optimization algorithms.

Statistical comparison. A statistical comparison between the approximations sets (Pareto fronts) obtained by tested algorithms is also required for an objective conclusion.

Because the data does not follow a normal distribution we performed a *Kruskal-Wallis* analysis over the Hypervolume values obtained by each method over 50 independent



(A) Mean hypervolume for $m =$ 10.

(B) Mean hypervolume for $m =$ 20.

(C) Mean hypervolume for $m =$ 30.

FIGURE 3.13: Average normalized Hypervolume values (50 independent runs) for real data web services composition, workflows with different complexities $l \in \{10, 20, 30\}$ and $k \in \{10, 20, ..., 90\}$: 3.13a - 10, 3.13b - 20, and 3.13c - 50 abstract services, respectively. Higher HV values indicate better performance. All the algorithm behave similarly from this point of view. Source: [Cremene et al., 2014a]

runs. Test results indicate a statistical difference between algorithms. Algorithms have been compared using the *Wilcoxon rank-sum* test.

Experiments reveal that the best algorithms in terms of performance are: POSDE, NSGA2, and GDE3. However, NSGA2 has proven to be also the slowest algorithm. DE approaches represent a good trade-of between time complexity and performance. POSDE proven to be the best choice in our experiments.

3.4.2.4 Decision making

A Decision Making (DM) algorithm based on user preferences is proposed. The purpose of the DM algorithm is to select from the Pareto set a final solution based on user preferences. An hypothesis is that the user expresses his/her preferences for each objective as weights for each QoS attribute.

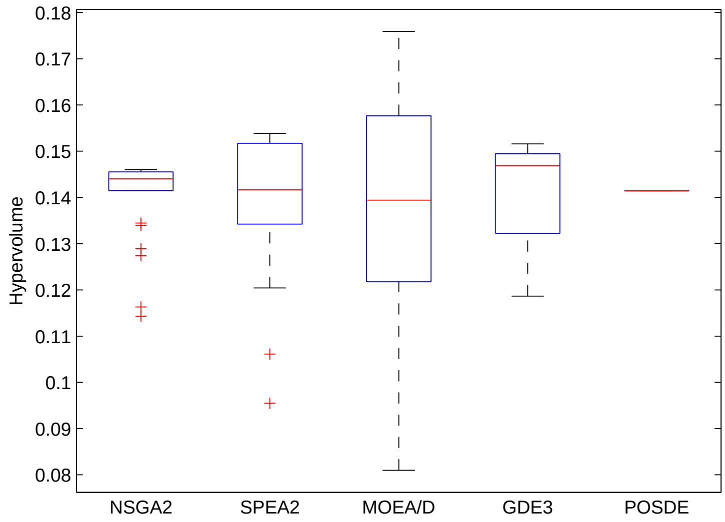


FIGURE 3.14: Normalized Hypervolume box-plots for a 2 objective optimization problem and a workflow with 10 abstract services and 80 concrete services. Higher the box is, higher the dispersion is. The red line in the middle of the box represents the mean value. A lower dispersion indicates a more stable/repeatable behavior. Source: [Cremene et al., 2014a]

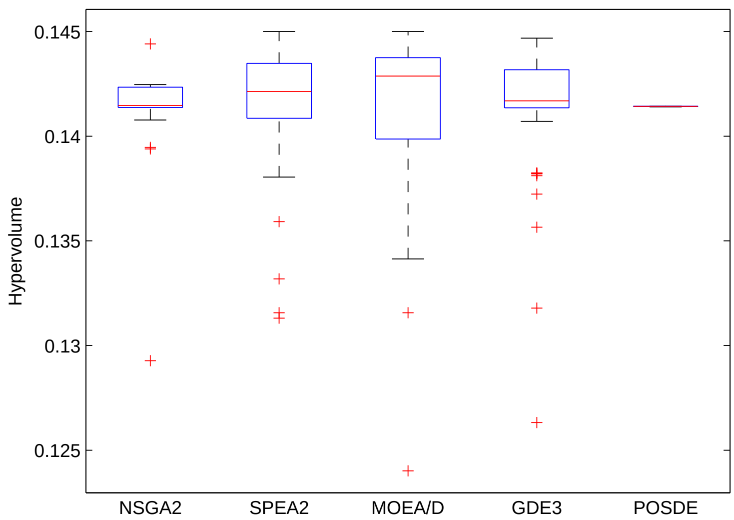


FIGURE 3.15: Normalized Hypervolume box-plots for a 3 objective optimization problem and a workflow with 10 abstract services and 80 concrete services. Higher the box is, higher the dispersion is. The red line in the middle of the box represents the mean value. A lower dispersion indicates a more stable/repeatable behavior. Source: [Cremene et al., 2014a]

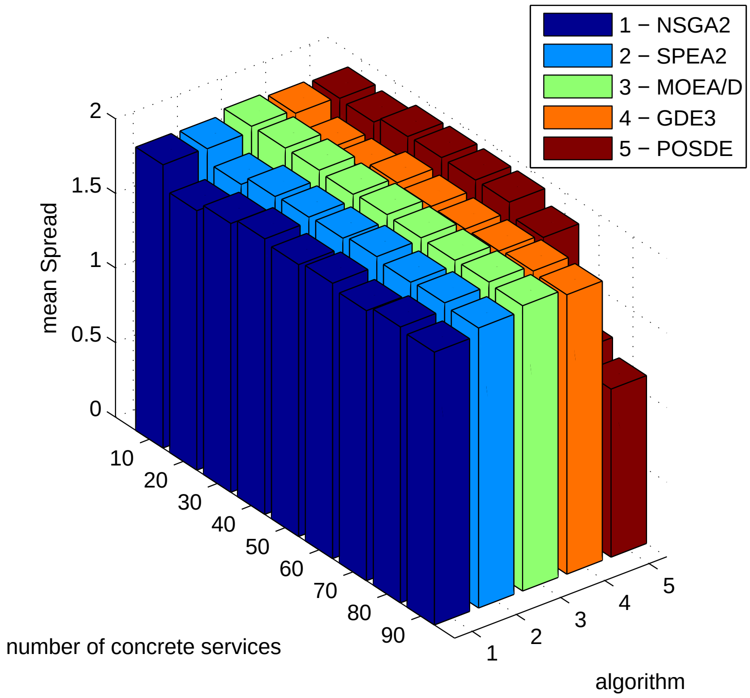


FIGURE 3.16: Mean Spread indicator values for compared algorithms. 50 independent runs are considered. All algorithms assure a relatively good diversity. *POSDE* gives slightly better results.

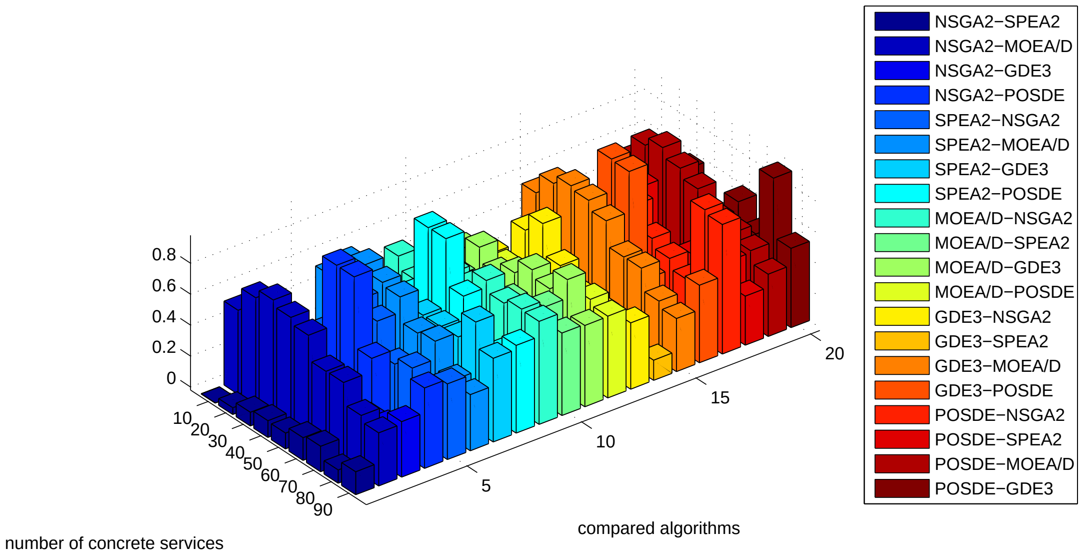


FIGURE 3.17: Mean Set Coverage indicator values for compared algorithms. *POSDE* and *NSGA2* give the best results.

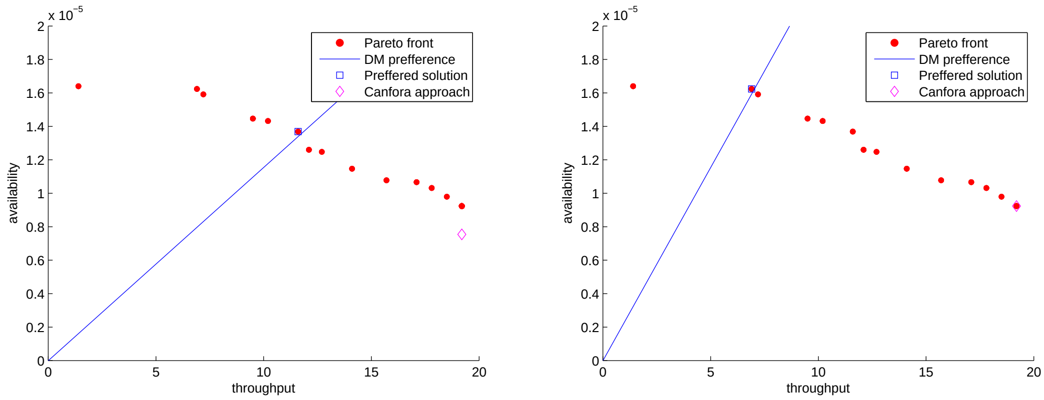
The DM algorithm is depicted bellow:

i. determine the equation of the line passing through the origin point and the point defined by the weights representing the user preferences,

ii. for each solution from the Pareto set compute the distance to the line,

iii. choose the solution having the minimum distance.

Fig. 3.18 illustrates the advantages of the multi-objective approach compared to a single-objective approach. The multi-objective algorithm finds a set of solutions from which it chooses the one that best approximates the user preferences. The weighted single-objective algorithm gives a solution that does not satisfy the user preferences.



(A) $w_1 = 0.5$ and $w_2 = 0.5$. (B) $w_1 = 0.3$ and $w_2 = 0.7$.

FIGURE 3.18: Decision making process for a workflow with $m = 10$ abstract services and $n = 30$ concrete services. The multi-objective approach combined with the DM algorithm is able to find a solution adapted to user preferences while the single-objective approach is unable to find it.

Fig. 3.18a depicts a situation where both objectives have the same importance for the user. In this case, for the first objective (*throughput*) the user assigns the weight $w_1 = 0.5$ and for the second objective (*availability*) the weight $w_2 = 0.5$. The line defined by the origin $(0, 0)$ and the point $(0.5, 0.5)$ represents the user preferences. The solution that is closest to this line is the compromise that best approximates the user's preferences. An SO approach using the same weights does not find a solution that reflects the user preferences.

Fig. 3.18b illustrates another situation where the user favors the QoS property 'availability'. In this case the weights may be $w_1 = 0.3$ and $w_2 = 0.7$. The propose MO approach is able to find a solution that best approximates the user's preferences in this case too.

In conclusion, numerical experiments indicate that the proposed approach outperforms the existing ones in terms of precision in selecting the end user solution. There is however a price to be paid regarding the complexity: usually MO algorithms are slower than SO algorithms. A possible solution may be to parallelize the MO evolutionary algorithms in order to increase the convergence speed.

3.4.3 Equitable Solutions in QoS-aware Service Optimization

The contribution presented in [Suciu et al., 2012] concerns the problem of QoS-aware service composition discussed before but proposes a slightly different approach. Even if the QoS optimization problem is multi-objective most approaches are based on single-objective approaches. Compared to single-objective optimization algorithms, multi-objective optimization algorithms (MOAs) have some advantages: *(i)* the aggregation of objective functions is not necessary and *(ii)* the user has the possibility to select *a posteriori* one of the Pareto optimal solutions.

However, the end user may not be an expert in decision making. Thus, providing a large set of solutions could confuse him/her. We propose to use the concept of Lorenz dominance [Kostreva and Wodzimierz, 1999] in order to reduce the solution set. Lorenz solutions are equitable and well balanced [Kostreva and Wodzimierz, 1999]. An equitable approach based on Lorenz dominance simplify the Decision Maker's choice.

Evolutionary detection of Lorenz solution is preferred because the problem is NP-hard. Several evolutionary heuristics may be considered for detecting Lorenz solutions by replacing Pareto dominance with Lorenz dominance. Proposed Lorenz variants of some common MO algorithms are presented bellow:

L-NSGA2 [Dugardin et al., 2010] is a variant of the very popular NSGA2 algorithm [Deb et al., 2002]. L-NSGA2 is based on a μ, λ selection ($\mu{-}parents{+}\lambda{-}childs$), favoring good solutions through a ranking and sorting system based on Lorenz dominance. For each solution a domination count is determined (how many solutions it dominates). Based on this counter all solutions with rank zero are placed in the non-dominated front and are selected to survive for the next generation.

SLEA. Strength Lorenz Evolutionary Algorithm, SLEA [Suciu et al., 2012], is a variant of SPEA 2 [Zitzler et al., 2001]. SPEA2 is an elitist multi-objective optimization algorithm. This algorithm uses an archive for storing the best solutions found at each generation.

L-GDE3. L-GDE3 [Suciu et al., 2012] represents a multi-objective Lorenz variant of GDE3 [Kukkonen and Lampinen, 2005a] proposed by Kukkonen and Lampinen in 2005. L-GDE3 modifies the selection operator in DE by introducing the concept of Lorenz dominance. The trial vector is selected to replace the decision vector if it dominates the decision vector. Condition $j = rand\, j$ indicates that at least one component of mutated vector v_i is selected. If the vectors are indifferent to each other (neither is better) then both vectors are kept.

L-POSDE. L-POSDE [Suciu et al., 2012] represents a Lorenz variant of POSDE [Chang et al., 1999] - one of the first multi-objective DE approaches. POSDE uses a secondary population to retain the non-dominated solutions found at each generation in the evolutionary process. Diversity is achieved by a distance metric used to alter the fitness of each individual when it is compared with the elements of the archive.

L-DEMO. L-DEMO [Suciu et al., 2012] represents a Lorenz variant of DEMO. DEMO [Robic and Filipic, 2005] is a Differential Evolution multi-objective evolutionary algorithm. For non-constraint optimization problems it is similar to GDE3 algorithm. The candidate solution replaces the parent only if it dominates it. If the parent dominates the candidate, the candidate is discarded. If the candidate and parent solutions are non-dominated by each other the candidate is added to the population. NSGA2 truncation mechanism is used to reduce the population size if by adding candidate solutions the predefined population size is exceeded.

The processes corresponding to composite web services have various complexity and are randomly generated. The complexity is adjusted by increasing the number of abstract services from *10* to *50*.

The results indicate that L-NSGA2 have the best performances demonstrated by the evaluation metrics. But L-NSGA2 proven to be also the slower algorithm. Therefore, DE approaches seem to be a better choice for addressing complex problems. The results show that the Lorenz dominance concept reduces the number of solutions and the decision making process is easier.

3.4.4 Adaptive multi-objective optimization algorithms for QoS-based service composition

An evolutionary multi-objective algorithm based on decomposition is proposed in [Suciu et al., 2013b]. The problem of QoS-aware service composition using an Evolutionary Multi-Objective Optimization Algorithm (EMOA) approach is discussed in [Suciu et al., 2013b] from a slightly different angle. A major challenge that arises is the dynamic nature of the problem of web service composition. Evolutionary algorithms' performance is highly influenced by the parameter settings. A manual tuning of these parameters is not an efficient solution. An algorithm for tuning these parameters may be used instead.

$MOEA/D$ multi-objectve algorithm is considered [Zhang and Li, 2007]. $MOEA/D$ advantages are: scalability with the number of objectives, computational efficiency, and high performance for combinatorial optimization problems. For coping with the dynamic nature of the QoS web service composition problem we endow $MOEA/D$ with an *adaptation mechanism*.

Some simple and powerful adaptation techniques for DE have been proposed in [Neri and Tirronen, 2010]. We propose two adaptive variants of $MOEA/D$ obtained by considering the DE adaptive mechanisms: *SaDE* [Qin et al., 2009] and *CoDE* [Wang et al., 2011]. The new models are called $MOEA/D_C$ and $MOEA/D_S$ [Suciu et al., 2013b].

Algorithm comparison is performed over 50 runs. For the final population, the mean values for the quality indicators Inverted Generational Distance (IGD) [Schuetze et al., 2010] and Hypervolume (HV) [Auger et al., 2009] are computed. The IGD computes the average distance of the reference Pareto set (P^*) to the nearest solution in the solution set found (A). IGD indicates the spread of A. Small values are desirable. The hypervolume represents the surface covered by the solution set and a reference point. High values mean that the front is near to the theoretical front and it assures a good spread.

The results indicate the potential of the adaptive-based approach. Better performance is obtained when the adaptive approach is applied to standard test problems and some business workflows of high complexity [Suciu et al., 2013b].

3.4.5 Scalarization techniques for Evolutionary Multi-Objective Algorithms

When solving a multi-objective problem, Pareto based evolutionary algorithms are usually the preferred choice. Usually they are able to find a good approximation of the theoretical Pareto front and assure a good solution diversity. However, Pareto dominance scales badly with the number of objectives.

Decomposition based algorithms represent a good choice for many-objective problems. Their performance is not affected in such a severe way because they solve multiple one-objective problems. The preferred methods for combining all objectives into one single-objective are *weighted sum* and *weighted Tchebycheff*. In [Suciu et al., 2013a] we study the augmented, modified Tchebycheff and Lp decomposition techniques.

Differential Evolution is used as underlying evolutionary technique. Scalarization techniques behavior is observed. As basis for the comparison we use standard test problem: WFG, DTLZ, and ZDT [Auger et al., 2011]. All problems are real valued unconstrained and require the minimization of objective functions. The hypervolume and inverted generational distance are computed for the obtained solution sets.

Numerical results [Suciu et al., 2013a] indicate that decomposition approaches represent a good choice for solving a many-objective optimization problem. By solving an optimization problem in this way the exponential increase of non-dominated solutions with respect to the number of objectives is avoided [Corne and Knowles, 2007].

3.4.6 Natural language based service composition

The *service-user adequacy* concept may also include the user request versus service function as a type of S-C distance. A simple and natural way for the user to specify a request is to use natural language. If the service is described using semantic concepts it is possible to define a distance based on words/concepts similarity. The papers [Cremene et al., 2009], [Pop et al., 2009] and [Pop et al., 2010a] present two automatic service composition solutions based on this approach.

3.4.6.1 NLSC - natural language service composition

In [Cremene et al., 2009] we propose a natural language service composition method based on composition templates. The use of templates assures that the composition result is always valid. The proposed system, called NLSC (Natural Language Service Composer) is implemented on the top of a service-oriented middleware called *WComp* [Tigli et al., 2009]. An intelligent home environment called *Ubiquarium* [Hourdin et al., 2006] is used for the experiments.

A *conceptual distance* is defined as a measure of similarity between the words used in the user request and the semantic concepts associated with services. A specialized dictionary called WordNet [Fellbaum, 1998] is used in this purpose.

Fig. 3.19 describes the NLSC architecture.

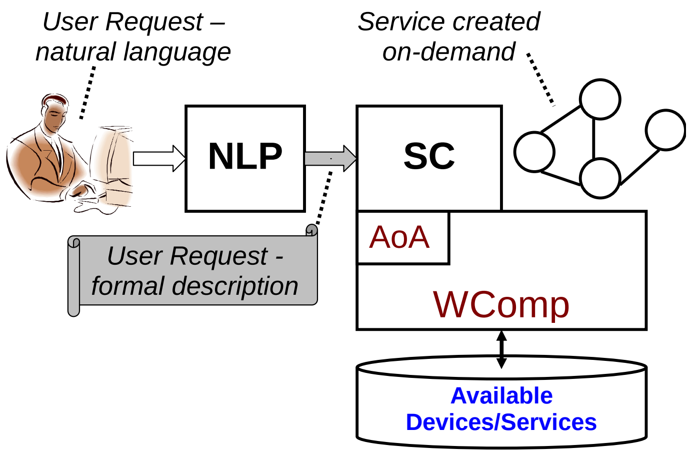


FIGURE 3.19: NLSC architecture. Source: [Cremene et al., 2009]

The architecture includes the following modules: *NLP - Natural Language Processor* - consisting in a set of tools designed for analyzing the user request. A textual or voice-based input is supported. The NLP module transforms the user request (natural language) into a formal request. The formal request is basically a list of concepts (bag of words) extracted from the user request. The formal request is used by the SC - Service Composer - to assemble a web service. SC is built on the top of the WComp platform and it uses *Aspect of Assemblies* (AoA) patterns. AoA patterns may be combined and superposed. This AoA features make possible to compose a large number of valid web services.

The original aspect of our proposal is the mixed approach: semantic and template-based. Thanks to AoA composition templates it is possible to build complex composite services, which are always valid and functional. With other approaches (interface, logic, semantic-based) it is very difficult to create complex service structures/architectures that are valid and work correctly. Additionally, the AoA templates can be composed and this helps us to overcome the limitations of the traditional pattern-based approach.

3.4.6.2 NLSCd - natural language service composition with disambiguation

In order to deal with the ambiguity of the natural language current solutions impose restrictions to user request structure and content. In [Pop et al., 2010a] we propose a service composition system called *NLSCd - Natural Language Service Composer with request disambiguation* (that is an evolution of NLSC [Cremene et al., 2009]). NLSCd detects and compensates the natural language imperfections.

NLSCd system architecture is depicted in Fig. 3.20.

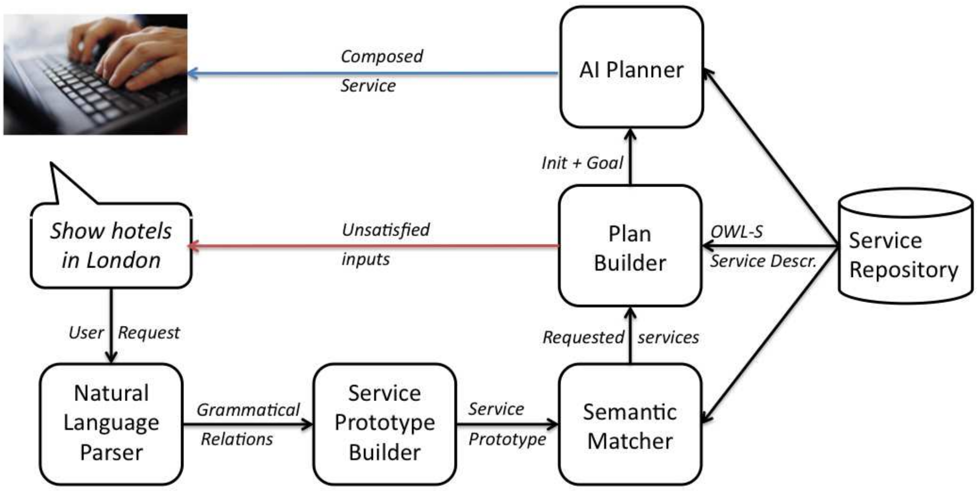


Figure 3.20: NLSCd - Natural Language Service Composer with request disambiguation - architecture. Source: [Pop et al., 2010a]

The *input subsystem* receives a natural language request from the user. A *Natural Language Parser* extracts grammatical relations (GR) that exist between the parts of speech in the request. The *Service Prototype Builder* transforms the user input into a machine readable, formal request - a list of *service prototypes*.

A *semantic tier* is used as a bridge between natural language and the service ontology. In order to identify concrete web services based on their natural language prototypes, we use the *Semantic Matcher*. Services in the *Service Repository* are annotated using OWL-S [Martin et al., 2004].

The list of concrete services identified represents the input for the *Plan Builder*. First, the *Plan Builder* finds other services, on which required services depend on. Second, it matches the input values of the prototypes with the inputs of actual services. Values for the inputs that cannot be satisfied are requested from the user. Finally, an initial state and a goal is defined. An *AI Planner* uses the state and the goal to generate the composed service.

Grammatical relations that link the predicates to the other parts of speech are used to generate '*service prototypes*'. By the means of semantic matching concrete services are discovered, based on their prototypes. The matching operation is based on a semantic *similarity* metric that deals with inaccurate requests. After requested services and their dependencies have been identified by input types and process names, input values are assigned to processes.

The request is supposed to be in a form of imperative. A service prototype only matches a service from the services repository.

In order to evaluate the semantic matching performances, 23 of most common user queries were used. Prototypes based on five of these queries and their similarity measures with existing services are presented in Fig. 3.1. A small semantic distance corresponds to a high similarity between two words/expressions.

Experimental results have shown that inaccurate requests can be matched in more than 95% of user queries. The proposed prototype detects the missing inputs necessary to successfully compose the service and asks the user to supply them. This a a key feature for a natural language based interface since natural language is usually ambiguous and imprecise.

Service URI \ Service Prototype	*book flight alternative (John)*	*book flight (Paris, London)*	*create account (John)*	*find airport (Paris)*	*transport person (John)*
../BookFlight.owl	0.458333333	**0.516666667**	0.683333333	0.783333333	0.495833333
../BookFlight2.owl	0.458333333	0.75	0.833333333	0.789583333	0.633333333
../BookFlightAlternative.owl	**0.404761905**	0.75	0.833333333	0.789583333	0.633333333
../BookMedicalFlight.owl	0.526785714	0.708333333	0.8125	0.7875	0.625
../BookMedicalFlight2.owl	0.609375	0.732142857	0.8125	0.8125	0.651785714
../CreateFlightAccount.owl	0.583333333	0.791666667	**0.666666667**	0.808333333	0.6875
../CreateFlightAccount2.owl	6.61E-01	0.80952381	0.69047619	0.833333333	0.672619048
../CreateMedicalFlightAccount.owl	0.583333333	0.791666667	**0.666666667**	0.808333333	0.697916667
../CreateMedicalFlightAccount2.owl	0.680555556	0.80952381	0.69047619	0.833333333	0.681547619
../CreateMedicalTransportAccount.owl	0.722222222	0.863095238	0.69047619	0.833333333	0.645833333
../CreateVehicleTransportAccount.owl	0.722222222	0.863095238	0.69047619	0.833333333	0.645833333
../CreateVehicleTransportAccount2.owl	0.722222222	0.916666667	0.833333333	0.833333333	0.731770833
../FindNearestAirport.owl	0.895833333	0.7	0.95	**0.75**	0.6
../ProposeFlight.owl	0.5625	0.591666667	0.758333333	0.802083333	0.558333333
../ProposeFlight2.owl	0.601190476	0.8125	0.833333333	0.808333333	0.645833333
../ProposeMedicalFlight.owl	0.601190476	0.8125	0.833333333	0.808333333	0.6875
../ProposeMedicalFlight2.owl	0.708333333	0.827380952	0.833333333	0.833333333	0.672619048
../RegisterPersonWithMedicalTransport.owl	0.736111111	0.880952381	0.833333333	0.833333333	0.606547619
../RegisterPersonWithTransport.owl	0.625	0.666666667	0.708333333	0.786458333	0.420833333
../RequestMedicalTransport.owl	0.654761905	0.875	0.833333333	0.795833333	0.614583333
../RequestTransport.owl	0.583333333	0.666666667	0.708333333	0.786458333	**0.470833333**

TABLE 3.1: Similarity matching results for five queries. A small semantic distance corresponds to a high similarity. Source: [Pop et al., 2010a]

Chapter 4

Decentralized distributed adaptive systems

The most significant author contributions related to distributed adaptive systems field are presented in this chapter. The limitations of the centralized approach have been already discussed in the Introduction chapter. A decentralized distributed approach is more suitable to efficiently address the real world challenges raised by its inherent complexity. In decentralized systems the global coherence is obtained by local, independent actions. How to coordinate multiple autonomous entities (agents) to converge toward a global optimum is a key issue.

4.1 Multi-agent systems

A *Multi-Agent System* (MAS) [Niazi and Hussain, 2011] represents a system composed by multiple interacting entities called *agents* co-located in a specific environment. An agent is an autonomous software (or hardware-software) component incorporating algorithms specific to the particular task meant to be solved by that agent. The *autonomy* is a key agent characteristics - an agent is supposed to be able to solve a certain task without any other external intervention.

A multi-agent system is supposed to be able to solve problems that a single agent/-component/service cannot solve. In some cases, the system is multi-agent by nature if entities having different owners interact or/and share common resources.

The term 'intelligent' used as attribute for an agent indicate the act that Artificial Intelligence specific algorithms are often used (reasoning, search, optimization, evolutionary algorithms, etc.) within agents.

A related term is *Agent-based Model* (ABM), which is meant to study collective behavior of agents. Such agents usually implement very simple rules and they are not necessarily considered 'intelligent'. Usually, ABM approach is used for modeling natural systems (i.e. social systems) while MAS models are used in *Computer* and *Telecommunications* engineering for designing artificial systems [Niazi and Hussain, 2011].

Self-organization is a key feature of MAS. When the agents need to communicate a common protocol/language (i.e. Knowledge Query Manipulation Language - KQML, FIPA's Agent Communication Language - ACL) and common knowledge (represented usually by an ontology) are defined.

Self-recovering and self- management are other common properties of MAS. These features make MAS well suited for designing autonomic systems and complex adaptive systems. One of the most important MAS challenges is to design the intelligent algorithms that an agent should implement.

4.2 A general architecture for a distributed system of services

The idea presented in this section was inspired from the *'smarter planet'* vision launched by IBM [IBM, 2006]. According to this vision, systems such as: power grids, water supply, transport systems, buildings, and others should become 'smarter' in terms of efficiency, energy consumption, etc. Al these systems are interconnected and form *systems of services.*

There is an important challenge in deploying and managing *systems of services* with a large number of services from different domains. Services are parts of a complex system. A store, an enterprise, a university, a city, etc. imply the co-existence and co-interaction of different types of services: human resources, transportation, provisioning, etc. These services are different but they share a common environment and there is an interdependence relation between them. A centralized management is practically impossible in this case. A distributed management, based on autonomous agents, is a more realistic approach.

In order to deploy complex systems of services we need an appropriate infrastructure. This infrastructure should solve various problems such as: interoperability, scalability, extensibility, distributed management, flexibility and composability. A model for such a distributed system should include also the intelligent agent paradigm.

Our study domain concerns in particular web services. Automatic composition and execution of web services is a multidisciplinary problem involving three main areas: SOA, Semantic Web and Computational Intelligence. SOA is an architectural model for creating reliable distributed systems that deliver functionality as services. Semantic Web [Shadbolt et al., 2006] is a vision of a Web that is understandable either by humans and computers.

The Semantic Web has at its core the concept of *ontology*. An ontology describes concepts and relation between these concepts. Semantic web services combine the versatility of web services with the power of semantic technologies. Web services described by semantic data offer new possibilities like automatic discovery, invocation, composition and monitoring. Computational Intelligence offers valuable techniques for knowledge representation and problem solving. A combination of these concepts may lead to rich and powerful applications models.

A general architecture proposed for a system composed by adaptive services is depicted in Fig. 4.1.

The architecture main components are: OA – *Observer Agent* - extracts and provides the Service-Context model (see Chapter 3). MA – *Manager Agent* - adapts the services during their execution. *MA* uses the S-C model obtained from the *OA* and includes reasoning and decision mechanisms. **C** – the *Context* - includes: users, resources, infrastructure elements, environment elements. **S** – the *Services* - may belong to different providers, they communicate with other services.

Each *Manager Agent* may implement an optimization algorithm (as described in Chapter 3). But, if each agent optimizes locally some utility/cost functions specific to a particular service, does it mean that the whole system will arrive to a global optimum? Can we predict what happens at the system level if we know what each agent will do?

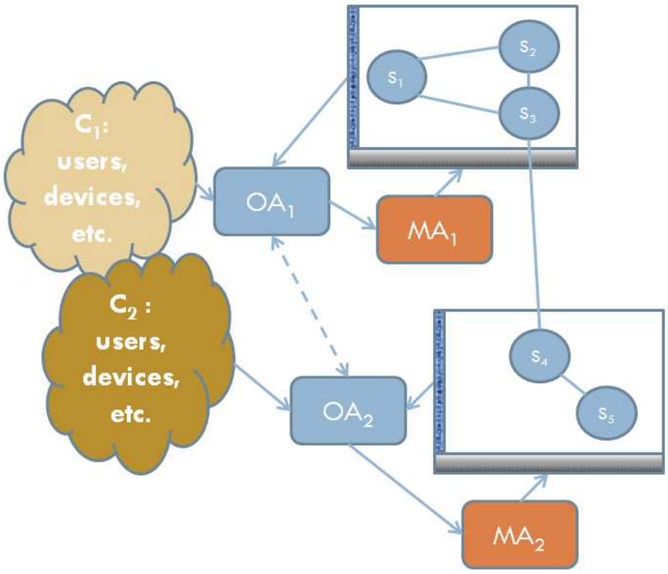


FIGURE 4.1: A general architecture for a system of adaptive services that includes: *OA – Observer Agent* - extracts and provides the Service-Context model (see Chapter 3); *MA – Manager Agent* - adapts the services during their execution; *MA* uses the S-C model obtained from the *OA* and includes reasoning and decision mechanisms; *C* – the *Context* - includes: users, resources, infrastructure elements, environment elements; *S* – the *Services* - may belong to different providers.

The next section provides some answers to these important questions.

4.3 The most relevant contributions to decentralized distributed adaptive systems

A distributed system composed by independent agents may be modeled as a game. Game Theory [Osborne, 2004] may be a useful tool for studying equilibria in games. The engineer role is to design rules that will lead the system to desired equilibria.

4.3.1 Contributions related to Cognitive Radio

Cognitive Radio (CR) [Mitola and Maguire, 1999] is a recent research field having as main study subject the interactions of intelligent radio devices sharing a common resource: the radio spectrum.

Cognitive Radio (CR) is not directly related to web service's field. However, the reason to be interested in CR domain is represented by the large number of existing game theoretical applications, while in *Software Engineering* few GT approaches exist for the moment.

Concretely, a *cognitive radio* is a radio device that can be dynamically programmed. Parameters such as: frequency channel, emission power, modulation type, coding, may be changed while the device is functioning. A cognitive radio is also capable of sensing its environment. For instance a CR can detect the spectrum channels with lower interference. This capabilities enable implementing 'intelligent' algorithms that chose efficiently the best parameters for a radio transmission.

Typical CR applications are dynamic spectrum management and opportunistic communications. Therefore, a CR may be seen as a special kind of agent that acts in a environment specific for radio devices: limited resources, shared channels, noise and interference, multi-path propagation.

In [Cremene et al., 2012a] three oligopoly game models are considered and reformulated in terms of radio access: Cournot [Varian, 2006], Stackelberg [von Stackelberg, 2011], and Bertrand [Bertrand, 1883]. TV white-space access scenarios are considered. Besides Nash and Pareto equilibria, a new equilibrium concept named the Nash-Pareto equilibrium is considered. Equlibria are detected using evolutionary algorithms. The maximum number of channels a CR may access without decreasing its payoff is indicated by the Nash equilibrium.

According to the proposed Cournot-based model, the players are the cognitive radios attempting to access a shared frequency band W. The strategy of each player (CR) i is the number c_i of simultaneous accessed channels. A strategy profile is a vector $c = (c_1, \ldots, c_n)$. The payoff $u_i(c)$ for a player i is given by the difference between a goodput $P(c)c_i$ and a cost of accessing c_i channels Kc_i. The payoff is given by the formula: $u_i(c) = (W - \sum_{k=1}^{n} c_k)c_i - Kc_i$, where K is the cost of accessing one channel. The cost is influenced by the power necessary to transmit and also to interference that may appear between different CR's.

Fig. 4.2 depicts the results obtained with the Cournot game model. Equilibria are detected using an evolutionary-based method.

The simulations for two players indicate one Nash equilibrium when each player uses 3 channels (from 10 available). In this case, the Nash solution is balanced and equitable.

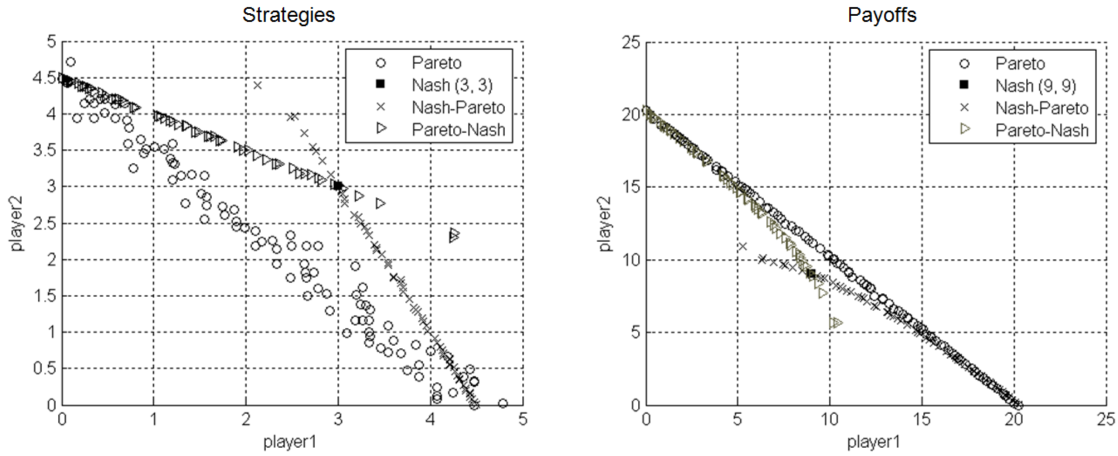


FIGURE 4.2: Cognitive Radios strategies and payoffs for the Cournot game scenario (2 players, W = 10, K = 1). Evolutionary detected equilibria: Nash, Pareto, Nash-Pareto, and Pareto-Nash.

It may be also observed that the Nash equilibrium is very close to the Pareto front. This means that a Nash solution is not an optimal one but it is very close to an optimal one.

Experiments based on Stackelberg model indicate that all users obtain maximum payoffs if the primary users (licensed) are Nash-oriented and the secondary users (opportunistic) are Pareto-oriented.

A crowded spectrum scenario is analyzed using the Bertrand model. These results are described in [Cremene et al., 2012a].

Scenarios for small-cell open access environments are studied from a game theoretical (GT) perspective in [Cremene et al., 2013]. Different small-cells share a common set of frequency channels. Each small-cell is considered to be a cognitive radio entity. Reformulations of the discrete Bertrand game are used to model CR capacity access competition. Stable game situations are described by game equilibria. Situations where Nash equilibrium (NE) is both fair and Pareto-optimal are identified. Sometimes multiple NE solutions may be found. The joint Nash-Pareto solutions capture heterogeneous player behavior. Stable and equitable states are identified for players with different biases (Nash or Pareto).

4.3.2 Adaptive services in the Cloud

Cloud Computing [Armbrust et al., 2010], [Mell and Grance, 2011] have recently gained a very high interest. The 'cloud' is a metaphor for the Internet. Cloud Computing idea is that computational resources may be offered as services. A small enterprise, for instance, which needs a computational infrastructure, may have incentives to rent resources from the Cloud instead of buying its own hardware. The Cloud may host various services that may be accessed remotely.

Three models of Cloud Computing have been proposed: '*software as a service*', '*platform as a service*', and '*infrastructure as a service*'. As network type, the choices are: public, private or hybrid. Examples of cloud vendors are: Google, Amazon, Microsoft Azure, IBM, Oracle Cloud, and others.

4.3.2.1 An adaptive security protocol

Security is an aspect very sensitive for Cloud-based applications because it delegates this responsibility to the Cloud service provider. But can we absolutely trust a Cloud provider?

A different approach is to give more control to the user/consumer. In [Popa et al., 2013] we propose an adaptive framework for Mobile-Cloud applications. Mobile Cloud Computing [Liang et al., 2010] is a model of mobile services that use Cloud resources. For instance, an application may be distributed between a mobile device and the Cloud.

The interest to use mobile services is well known and largely spread. The interest in using Cloud resources is mainly given by the large amount of computational power and storage volume. A mobile device will be always more limited from this point of view.

In [Popa et al., 2013] we analyze the most important security threats. A component-based adaptive security protocol is proposed. Each security property (*authenticity, confidentiality, non-repudiation, encryption*) is implemented as a basic or composite service. Instead of using a classical approach such as HTTPS for instance, a personalized context-adapted security protocol is generated from existing security components. The components are assembled according to a set of rules. Such a solution is more efficient in terms of performances and costs (i.e. energy consumed from the mobile phone's battery).

4.3.2.2 An dynamic market of Cloud-based services

Another important aspect of Cloud Computing is the *economic* one. A dynamic services market is expected to appear in the context of Cloud Computing. A service provider may rent a certain amount of computation resources. Different pricing/cost models may be used [Samimi and Patel, 2011]. All cost models are based on the idea that the cost is proportional to the used resources (CPU time, memory, storage, throughput, requests, etc.).

Let analyze a situation where two services use resources from the Cloud and offer similar functions. Different clients may connect and disconnect dynamically to these services. This scenario offers an example of how Game Theory may be used for studying decentralized adaptive systems.

An architecture describing a dynamic service market in the Cloud is depicted in Fig. 4.3 [Cremene et al., 2012b].

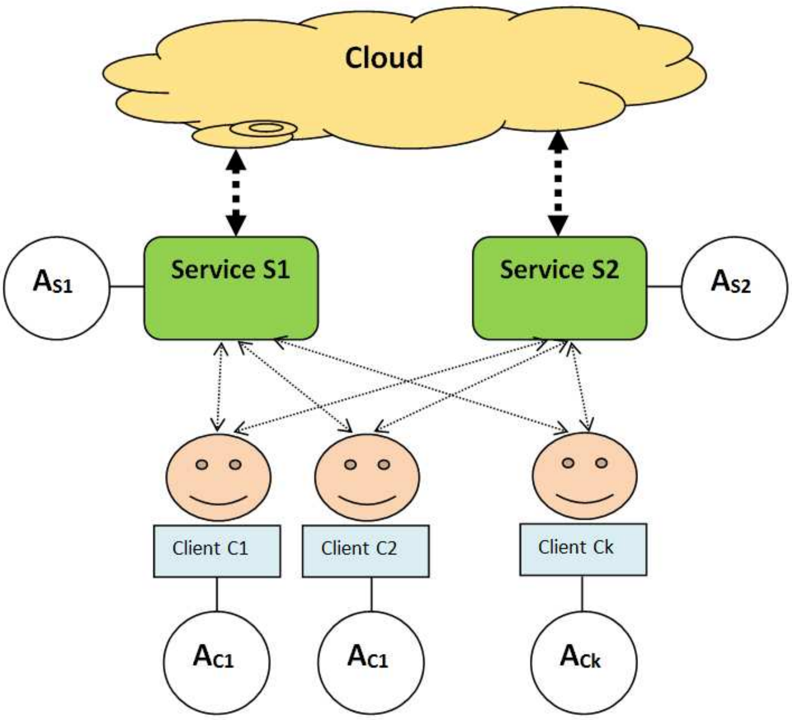


FIGURE 4.3: Cloud services market - a general architecture. Source: [Cremene et al., 2012b]

This architecture includes: the Cloud that provide resources for a specific price (proportional with the amount of resources), services S_1 and S_2 owned by different providers (more than two services may exist), agents A_s1, A_s2 that adapt the services (may change their QoS and decide the amount of resources to buy from the Cloud), a number of k clients C_i that consume the services and client agents A_ci that decide what service to use at a given instant.

Both services provide the same functionality but the QoS is different. If all clients will chose S_1 the service will be overloaded at some moment and the QoS of this service will be degraded. At this moment, the clients will prefer to switch to S_2. Function of how the payoff and the costs are defined different equilibria may appear in this game.

Game Theory is an appropriate tool for studying equlibria in decentralized distributed systems. There are some reasons for studying systems as games: the services compete for the same clients, the profit of one service depends not only on its action but also on the other service actions. An agent responsible with a service may buy more or less resources in advance and may increase or decrease the service's QoS. Increasing the QoS will make the service more interesting for the clients but will also increase the costs.

In particular, Cournot [Varian, 2006] and Bertrand [Bertrand, 1883] game models seem to be adequate for modeling such a situation. This problem is similar to the classical problem with two concurrent factories that produces a same product and should establish a quantity (Cournot) or a price (Bertrand). This research direction is still work in progress.

Chapter 5

Decomposition and unification in adaptive systems

Sometimes, dividing a system in several distinct blocks helps to manage its complexity. However, in other situations is preferred to unify some distinct functions. A system may be decomposed in more than one variant. Alternative ways of dividing a system are discussed in this chapter. A contribution that illustrates the principle of unified computing is presented.

5.1 The limits of 'divide et impera' principle

'*Divide et impera*' it's a simple and practical principle proven to be very effective in many situations where a complex problem needs to be solved. By division, a complex problem is reduced to a sum of simple problems that may be solved independently. Therefore, a kind of structure is introduced for managing the problem. This sense of 'complexity' concept is closer to the sense of 'complicated'.

On the other hand, a new scientific/philosophic domain named *Transdisciplinary* is founded on the idea that the standard approach based on knowledge fragmentation has inherent limits [Nicolescu, 2002]. To be super-specialized in one field means to be ignorant in all the others. Therefore, higher the number of fields, higher the ignorance. Since the number of fields is increasing continuously, as the knowledge accumulates, the consequence is quite obvious and seems unavoidable.

E. de Bono supplies another reason to be critic with respect to the common way of dividing something (the entire science, a domain, a system, a problem, etc.). The '*lateral thinking*' concept [De Bono, 2010] helps us to understand that dividing something in a certain way is just *one* possibility among many other possibilities.

A graphical illustration of this idea is given in Fig. 5.1. A square may be divided in areas with equal surface in many ways, yet only a limited set of patterns (very regularly) is usually found by a person.

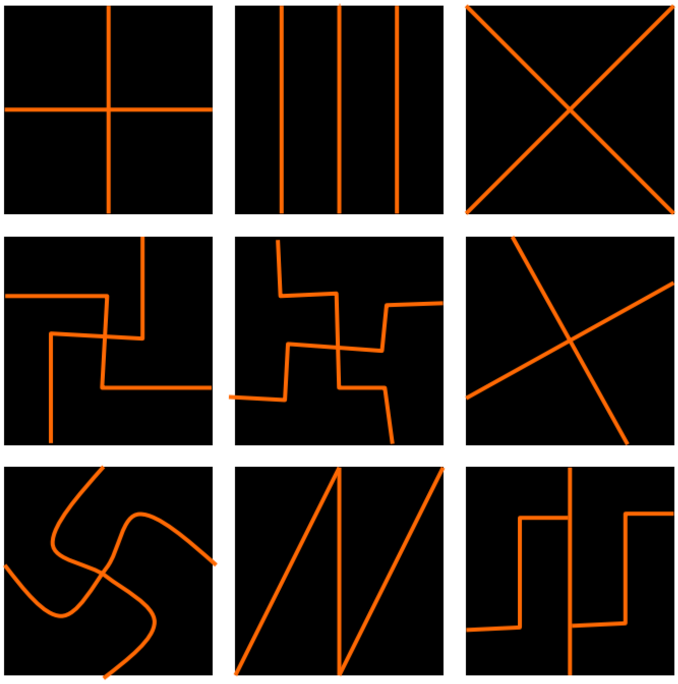


FIGURE 5.1: Graphical illustration of the lateral thinking concept: different ways to split a square in pieces having same form and equal surface.

How can be sure that our division/decomposition method is the best one? It is impossible to answer without evaluating several different alternatives.

5.2 Different perspectives on a same system - examples

System decomposition may be based on different approaches. For instance, when using a *Game Theoretical* approach, multiple way to define what a player is are possible. For instance, a 'player' may be: a user, a service provider, a device, a group/network of devices, etc.

This section discusses two examples where a same system may be seen from different perspectives. The first example concerns *Aspect Oriented Programming* (AOP) and the second one the *Cross Layer Design* principle. The presented examples sustain the idea that different perspectives may bring very useful insights on a system and may help to manage its complexity.

5.2.1 Aspect Oriented Programming

Aspect-oriented Programming (AOP) [Kiczales, 1996] requires to understand first the *Component-oriented Programming* (COP) paradigm [Heineman and Councill, 2001].

COP involves to separate a system in several components. This helps programmers to split a complex application in modules that may be implemented separately and then integrated. This is a typical application of 'divide et impera' principle in software engineering. Component reusability and loose-coupling are highly desired properties.

Service-oriented paradigm is based on the same principle. The idea is to find sets of logical functions - called 'concerns' - that may be separated. An example of concern separation in a software application is to design distinct modules for: *presentation*, *business logic* and *data access*. These are also called *functional* elements. Another way to split an application is to separate the *model*, the *view* and the *controller* (MVC architectural design pattern). When the modules/components are well separated, each module may be changed without affecting the other modules.

However, as the *lateral thinking* principles states, a system may be always separated in different ways.

AOP is a programming paradigm that regards a different kind of separation than the classical component-oriented one. If the separation of *concerns* leads to distinct logical modules, AOP involves a separation of *cross-cutting concerns*. A cross-cutting concern - called *aspect* in AOP - represents a set of program elements that affect other logical concerns. Aspects examples are: caching, logging, monitoring, synchronization, internationalization, security, and persistence. These are also called *non-functional* aspects. An aspect touches several logical modules. If the modules are separated 'vertically', aspects correspond to 'horizontal' planes. For instance, logging is involved in each logical module, security may be related to some vulnerable modules, etc.

AOP comes with a set of languages (i.e. *AspectJ* [Kiczales et al., 2001]) and tools (i.e. aspect *weavers*) meant to help the programmer in programming and changing easily a defined aspect. COP and AOP paradigms offer different complementary perspectives on a same system (software application).

5.2.2 Cross Layer Design

In the IT domain, for instance, a new trend called '*Cross Layer Design*' gained recently an important interest [Fu et al., 2014]. However, the cross-layer approach may be seen as a violation of the common division or protocol stacks in separate layers, the best example being the OSI protocol stack.

Researchers observed that the artificial boundaries established between different layers introduce limitations. Protocols may be globally optimized only if these boundaries are relaxed. In particular, feedback signals should be added between different layers. An example of application domain is the adaptive QoS.

More generally we can state that the optimization of each sub-system (module) separately does not guarantee the optimization of the entire system [Cremene, 2010]. In order to achieve a global optimization, separate modules should exchange information and sometimes cooperate. This means to reconsider the idea of separation. However, the separation should not be a goal by itself but a solution to a problem that cannot be solved in a satisfactory manner otherwise.

5.3 A unified combiner-equalizer module

Sometimes, instead of separating, a better solution may be to *unify* some functions that are usually separated. An adaptive *combiner-equalizer* (CE) module for radio receivers that combines diversity and equalization is proposed in [Cremene et al., 2010], [Cremene and Cremene, 2011]. A national patent was obtained based on this idea [Cremene and Cremene, 2011].

Fig. 5.2 depicts the scheme of the proposed adaptive combiner-equalizer. This is obtained by extending the number of inputs of a standard linear equalizer. The proposed

CE device performs simultaneously two different functions: a Maximum Ratio Combiner (MRC) and an adaptive equalizer.

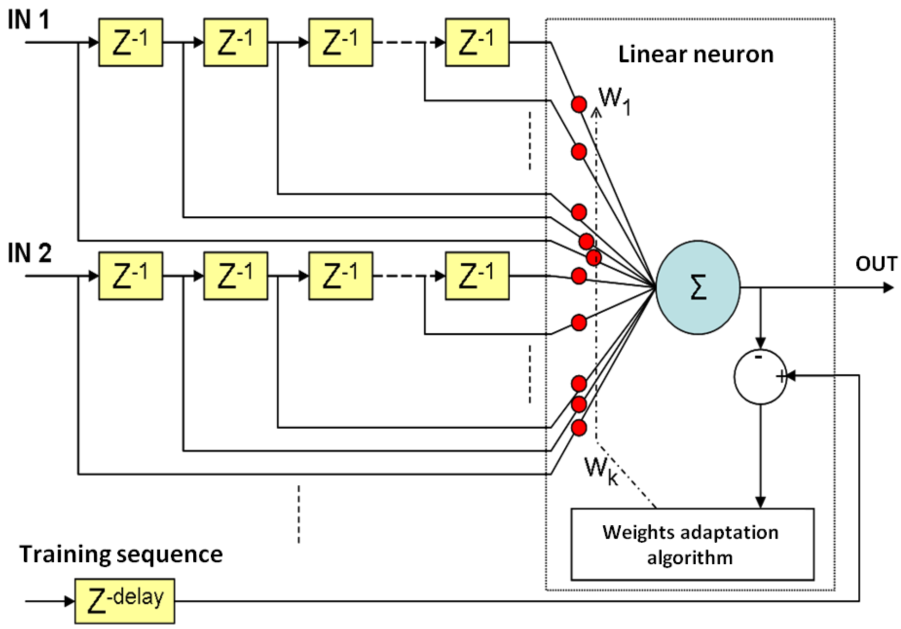


FIGURE 5.2: Electric scheme of the adaptive Combiner-Equalizer. Source: [Cremene and Cremene, 2011].

The proposed *combiner-equalizer* represents a time-domain single-carrier digital signal processing solution. It has multiple inputs representing the received signal (translated in the base-band frequency domain) and one output signal representing the resulted equalized signal. Each input signal is introduced in a chain of delay cells (Z^{-1}). The number of delay cells per chain depends on the maximum delay that a reflected wave may have. Concrete values for maximum delay are specified in standard propagation channel models (WINNER-3GPP) for indoor and outdoor scenarios.

Each version of delayed signal for each input represents the input vector for a linear neuron. Basically, the combiner-equalizer is a multiple-input linear adaptive filter. The neuron weights are dynamically adapted using the *Least Mean Squares* (LMS) algorithm. The adaptation/learning is supervized and it is based on the fact that a known training sequence is periodically transmitted over the channel (the classical approach for training adaptive equalizers).

Experimental results for indoor and outdoor scenarios indicate significant improvements in terms of *Signal to Noise Ratio* (SNR) when multiple inputs are used.

Fig. 5.3 depicts the combiner-equalizer performances for 2 inputs and 3 inputs. A dispersive (Rayleigh) fading is considered.

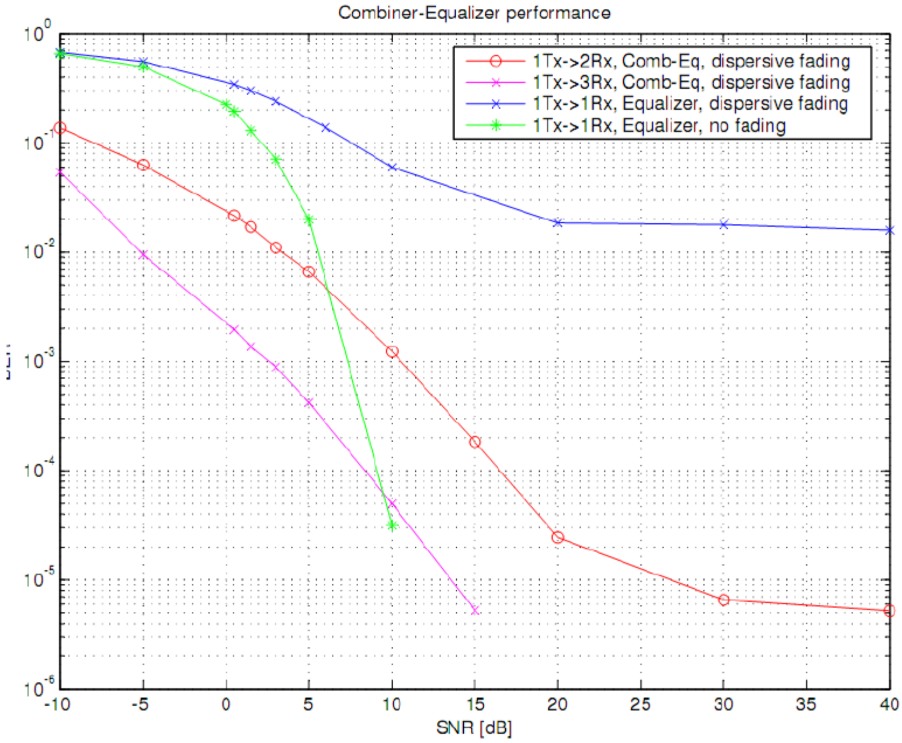


FIGURE 5.3: Adaptive Combiner-Equalizer performances. Comparing to the case of one input (classical solution) - the blue line - an important reduction of the error rate (BER) may be observed when 2 and 3 inputs are used - the red and magenta lines. The green line corresponds to the case of additive Gaussian noise (AWGN) channel. Source: [Cremene, 2010].

The results indicate an important gain in the case of multiple inputs comparing to the single-input situation. The obtained gains have been compared with values reported when the combiner and the equalizer are implemented separately. The proposed combiner-equalizer has better performances [Cremene et al., 2010] and also a simplified implementation since two distinct functions are offered by one block.

Chapter 6

Psycho-social aspects and technical systems

The interactions existent between technical systems and psycho-social aspects are discussed in this chapter. The first section presents several contributions of the author related to *Affective Computing* [Picard, 1997] field. The second section defines the socio-technical metaphor Ropohl [1999]. The third section briefly discusses the trust networks [Ziegler and Lausen, 2005]. The forth section presents a socio-technical inspired model called 'Social Honesty' game.

6.1 Contributions related to Affective Computing

Affective Computing (AC) [Picard, 1997] is a relatively recent research domain which studies technical systems capable of recognizing, interpreting, processing and simulating human affects [Tao and Tan, 2005]. AC may be considered as an extension of *Context-aware Computing* that it is focused on human interactions, with a special interest on emotional aspects. AC is therefore an interdisciplinary field: it concerns *Computer Science*, *Psychology* and *Cognitive Sciences*.

An ontology-based AC model for context-aware applications is proposed in [Benta et al., 2007]. The proposed model enables the expression of complex relations between affective states and other context elements. Several affective spaces, basic and secondary emotional states, location, time, person activity and other context elements are considered. Fuzzy logic expressions are allowed in this model.

OWL web ontology language [McGuinness and van Harmelen, 2004] is used as formal description of the proposed model. A new method for deducing the the secondary affective states [Benta et al., 2010a] from other context elements is proposed. A context-aware museum guide which reacts to users affective states is the example used as proof of concept for this idea [Benta et al., 2007].

An AC based model for a smart home is presented in [Benta et al., 2009]. A control mechanism allowing the smart home to learn new behavior from user's activity is the underlying principle of the proposed model. When the smart home behavior changes, the user do not need to manually change the rules because these rules are automatically updated by learning.

A tool for facial analyses called *FaceReader* [Loijens et al., 2012] is used for extracting emotion-related parameters. An additional module that computes emotional valence levels (positive, negative or neutral) was added to the commercial product named *FaceReader*.

In [Benta et al., 2010b] we propose a personalized ambient intelligent systems that adapts its behavior to user needs. This model is called *BAM – *Behaviour Adaptation Mechanism. *BAM uses an adaptive neural-network as control system. *BAM supervised training is based on user affective response obtained in real-time. The preferred service behavior is deduced from the affective states valence (negative, neutral and positive). User's facial features represent the system input. The number of training examples required in order to train the neural network is analyzed. The evolution of learning parameters when the number of context elements increases is also studied in [Benta et al., 2010b]. As expected, a minimum number of examples is necessary for an efficient learning.

6.2 Socio-technical systems

The socio-technical metaphor proposes to introduce the psycho-social aspect in the framework of complex adaptive systems. *Socio-technical Systems* (STS) is a notion firstly proposed by Ropohl [1999] from Tavistock Institute, London. STS proposes a framework that analyzes the interactions between people and technology and have as goal to improve the relations between humans and machines. Human and technical aspects are both involved in a STS.

A special interest was initially focused on working conditions in industry (i.e. division of labor). Today, this interest is much wider. The number of mobile phones is today equal to the entire population number. Almost every person on this planet interacts directly with the communications and computing technology.

However, we should be aware that, despite the ubiquitous presence of technology in our life, "*engineers tend to ignore the social concerns of their work, and social scientists, on the other hand, do not know very much about technology and are reluctant to consider the artificial reality of technical objects*" Ropohl [1999]. This idea underlines the need to give more attention to interdisciplinary approaches.

For laws governing the systems in general are identified in Ropohl [1999]:

1. The system is more than the sum of its internal elements. This may be explained by the fact that a system includes, not only entities, but also relations between these entities. This represents the *emergence* property and it is specific to complex systems (see Chapter 2).

2. The system's function is determined by its internal structure. In fact we already used this property in the S-C model where the S-C adequacy attributes of the composite service obtained by aggregation is determined by the internal architecture of the composite service.

3. Different structures may provide similar functions (principle of equifunctionality). Therefore, several alternative solutions may exist for a given function. This idea underlines the importance of using different perspectives (lateral thinking).

4. One level of hierarchy is insufficient to describe the system - is the principle of excluded reductionism. The existence of a hierarchy means that the system may be

seen as an element of a superstructure, and every element of the structure may be seen as a subsystem. From the social point of view, this means that understanding an individual is not possible without regarding also his/her social context. This property may be related to the aggregation property of CAS discussed by Holland [Holland, 1995] (see Chapter 2).

It may be observed that some of these properties are common with the Complex Adaptive System properties described in Chapter 2.

6.3 Trust networks

Virtual communities and networks have become attractive to many people. However, virtual networks involve a higher degree of anonymity, which also creates uncertainty and risk. This fact motivates the interest in examining reputation and trust that characterizes interpersonal communication [Abdul-Rahman and Hailes, 2000], [Cook, 2005]. Trust is also related to norms and social structure. Finding trustworthy partners is important because it leads to more stable and reliable relations [Kollock, 1994].

A special case is the *buyer-seller* relation. Studies indicate that asymmetric information between buyers and sellers may create problems with the quality of services and products. This is known as the '*lemons*' problem [Akerlof, 1970].

The anonymity existent in virtual communities makes things even worse. A consequence is that bad services/products may eliminate the good ones (Gresham's law).

The trust problem may be also formulated for a community of artificial agents. Let consider a decentralized distributed system composed by agents having different owners. An example may be a set of Cognitive Radio (CR) devices managed by different providers. Since each agent is autonomous it means that it has the possibility to chose actions that may be unfavorable to other agents. For instance, an agent may cheat or report a misleading information to other agents in order to increase his benefit (in the CR case this will be the bandwidth).

The *trust* problem was also identified in the context of *semantic web* where human and artificial agents exchange information [Ziegler and Lausen, 2005]. Credibility is an essential aspect about this information. Trust metrics represent computational means used to evaluate relationships between agents [Ziegler and Lausen, 2005]. A trust network may be represented as a directed graph where the nodes represents agents and the edges represent a measure of trust between agents.

Reputation systems [Fouss et al., 2010] may help establishing more reliable relations. Such an approach supposes to have access to identity and reputation related information. However, a reputation system will not work when such an information is not available or most interactions are 'one-shot'.

Alternative approaches are based on Game Theory (i.e. Prisoner's Dilemma game), where the evolution of cooperation is studied [Axelrod, 2006]. Community norms (i.e. an agent A punishes an agent B who cheated on another agent C) represent an efficient cooperation enforcement mechanism [Greif, 1989].

6.4 Strategic social interactions in technical systems

We recently proposed a model based on Evolutionary Game Theory which analyzes the dynamics of social honest behavior [Cremene et al., 2014b]. A typical scenario is based in a society where each agent interacts with other agents by providing/using services. Human and artificial agents may be involved. Agents may establish an informal contract (Service Level Agreement - SLA) for providing/using services. An honest behavior consists in offering a service at the expected quality, whereas a dishonest behavior means cheating by deliberately providing lower quality [Cremene et al., 2014b].

A punishment for dishonest behavior seems to be a necessary condition for promoting and supporting the honest behavior [Axelrod, 2006], [Sigmund, 2007], [Szolnoki and Perc, 2013]. The punishment effect on the honest/dishonest behavior dynamics is analyzed in the strategic interaction framework of Cellular Automata and Evolutionary Game Theory (GT) [Smith, 1982]. In a game, one player's payoff depends on the actions of all the other players. We propose a social dilemma game, called the Social Honesty (SH) game [Cremene et al., 2014b]. A player may chose between two strategies: 'honest' (H) or 'dishonest' (D).

6.4.1 Social Honesty game definition

For convenience, we call an 'H-player' a player using the honest strategy and a 'D-player' a player using the dishonest strategy. We assume that an incentive towards dishonest behavior exists, yet there is also an associated risk: a probabilistic punishment for D-players. When both players chose the dishonest strategy only one of them will win, yet the punishment may be applied to both of them. Dishonest behavior in one player causes a lower payoff for the honest player with whom he/she interacts.

The payoff matrix of the Social Honesty game is depicted in table 6.1:

Table 6.1: **SH game normal-form.**

Player1/Player2	Honest (H)	Dishonest (D)
Honest (H)	c, c	$0, A$
Dishonest (D)	$A, 0$	B, B'

The payoff matrix of the Social Honesty game. Two-player normal-form game.

When two H-players interact each player gets a payoff $c(c > 0)$, which is a constant. When an H-player interacts with a D-player, the H-player gets a payoff equal to 0. The D-player is punished with probability p_1. S denotes the punishment severity ($S > 0$). The unpunished D-players get a payoff a. a is the D-player's advantage in an (H, D) interaction. D-player's payoff is expressed as a discrete random variable A that takes the value $(-S)$ with probability p_1 and the value a with probability $1 - p_1$:

$$A = \begin{pmatrix} -S & a \\ p_1 & 1 - p_1 \end{pmatrix}$$

In a (D, D) interaction, the payoffs are assigned to players according to the following rule: •

- (i) each D-player is punished independently with probability p_2;
- (ii) if no player is punished, one player gets zero and the other one gets $b, (b > 0)$, with a probability equal to 0.5. D-players cannot win b and zero simultaneously.

In this case, the payoffs for the D-players may be expressed as discrete random variables B and B':

$$B = \begin{pmatrix} -S & 0 & b \\ p_2 & \frac{1-p_2}{2} & \frac{1-p_2}{2} \end{pmatrix}$$

and

$$B' = \begin{pmatrix} -S & b & 0 \\ p_2 & \frac{1-p_2}{2} & \frac{1-p_2}{2} \end{pmatrix}$$

To the best of our knowledge probabilistic utility is a new approach.

A standard $N \times N$ lattice model is considered. Each lattice cell represents a player. In the majority of our experiments players are arranged on a regular lattice with joint boundaries.

6.4.2 Experiments with Social Honesty game

The following parameters of the game are used in the experiments: $a = 3, c = 2, b = 2$, and a punishment probability $p = p1 = p2$. The agent's world size is 100×100. Results are averaged over 100 runs in order to have statistical relevance.

The influence of local social interactions on the spreading of a particular behavior is studied. We use numerical simulations to explore the contagion dynamics of honest and dishonest strategies in the agent population.

6.4.2.1 System evolution in time

Fig. 6.1 depicts the evolution of the agent's world (system) in time (rounds). Cluster formation may be observed after a dramatic fall of H-player rate in the first rounds. In time, the player rate remains almost constant for a given punishment probability and severity.

6.4.2.2 Punishment probability and severity importance

Experiments indicate that punishment probability is a more important parameter than punishment severity. These results are in accordance with the empirical evidence based

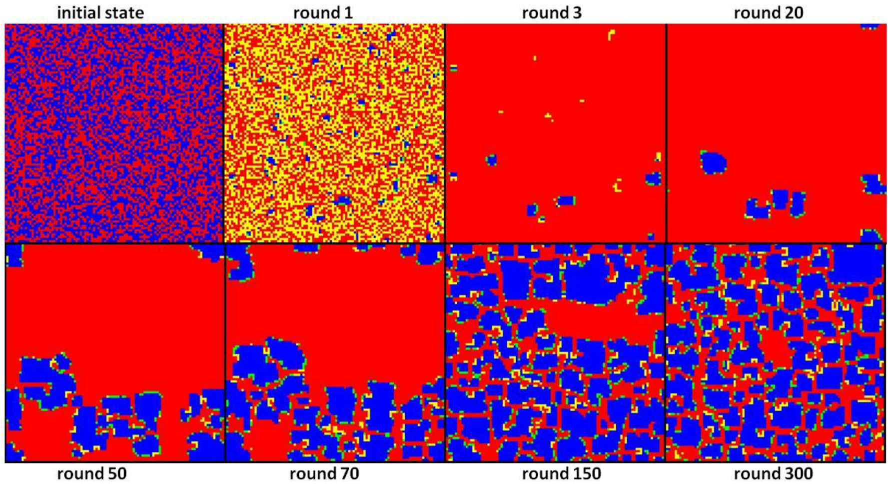


FIGURE 6.1: **H-cluster formation in a 100×100 population after $1, 3, 20, 50, 70, 150$, and 300 rounds ($S = 2, p = 0.15$).** The initial population contains 50%, randomly distributed, D-players. Few small clusters of H-players appear in the very first rounds, containing only 2% H-players after round 3. After 50 rounds the H-clusters become larger (25% H-players). After 150 rounds H-clusters may be found all over the population (57% H-players). The color code is: blue - is honest/was honest; red - is dishonest/was dishonest; green - is honest/was dishonest; yellow - is dishonest/was honest. Source [Cremene et al., 2014b].

on real world observations [von Hirsch et al., 1999], [Nagin and Pogarsky, 2001]. Transition intervals for punishment probability and severity have been observed in all experiments.

Fig. 6.2 depicts the transition interval obtained for punishment probability p when the punishment severity S is constant.

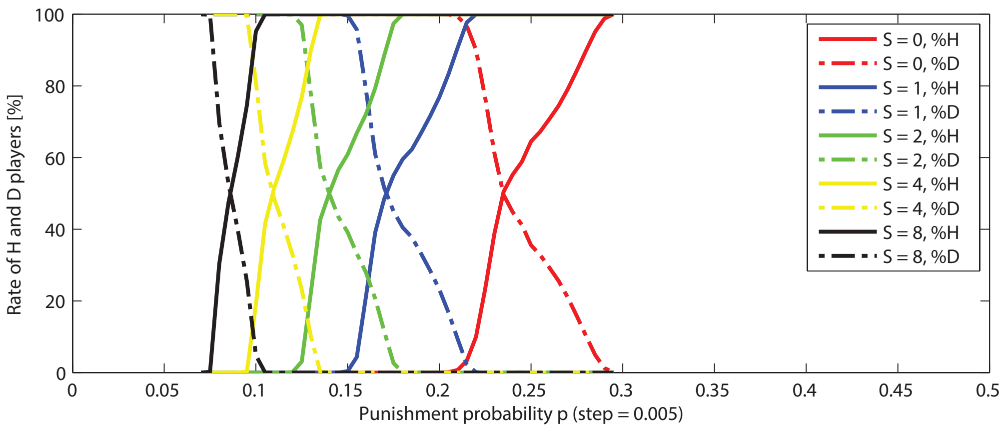


FIGURE 6.2: **Average variation of H and D-player rate with punishment probability p.** Averaged values for 100 runs are observed after 500 game rounds. The game starts with 50%, randomly distributed, D-players. Punishment severity varies: $S = 0, 1, 2, 4, 8$. For lower S, p-transition intervals become wider and translated to higher values. The H rate becomes 100% if the punishment probability p is higher than a specific value: 0.295 for $S = 0$; 0.22 for $S = 1$; 0.18 for $S = 2$; 0.135 for $S = 4$, and 0.105 for $S = 8$. Source: [Cremene et al., 2014b].

6.4.2.3 Neighborhood topology impact

The impact of different types of neighborhoods: von Neumann, Moore, well-mixed, and scale-free is analyzed in this experiment. Von Neumann and Moore neighborhoods may have different radia. In a 'well-mixed' case everybody is neighbor with everybody. In a 'scale-free' a spatial power-law based graph is mapped on the lattice.

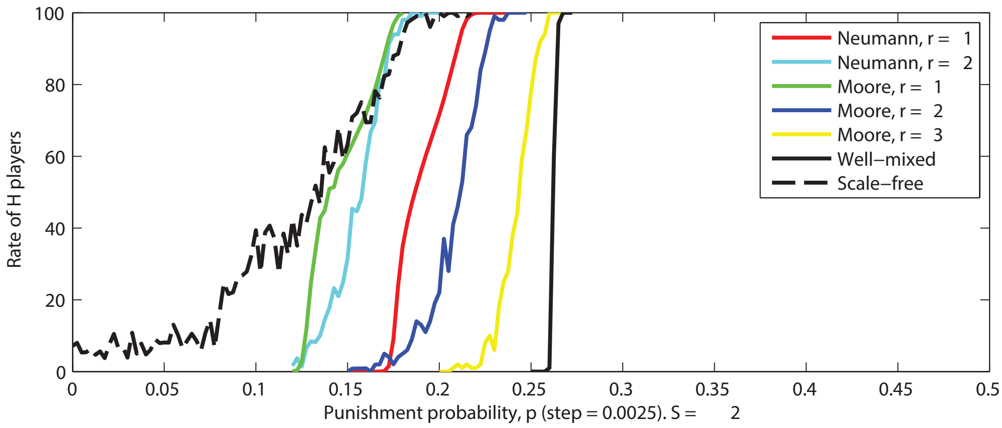


FIGURE 6.3: ***p*-transition intervals for different types of neighborhoods**: von Neumann (r = 1,2), Moore (r = 1,2,3) well-mixed (everybody is neighbor with everybody) and scale-free (power law connections mapped on the lattice). Averaged values for 100 runs are observed after 500 game rounds.

6.4.2.4 Asynchronous updating impact

Asynchronous updating impact on the *p*-transition interval is analyzed in this experiment. In the previous experiments all players interact synchronously: each players plays each round with *all* his/her neighbors. However, real world complex systems provide strong evidence that players do not activate synchronously. Instead, they use different types of activation mechanisms.

Different asynchronous updating methods have been applied on cellular automata and spatial games Cornforth et al. [2002] Newth and Cornforth [2009]. The most important are described bellow:

The *Random Order* scheme defines randomly the players who will activate for every round. If a player is not active then no interactions with other players are possible.

The *Random Independent* scheme defines one player to be activated for every round. Every player is chosen randomly. Every selection is independent of any other.

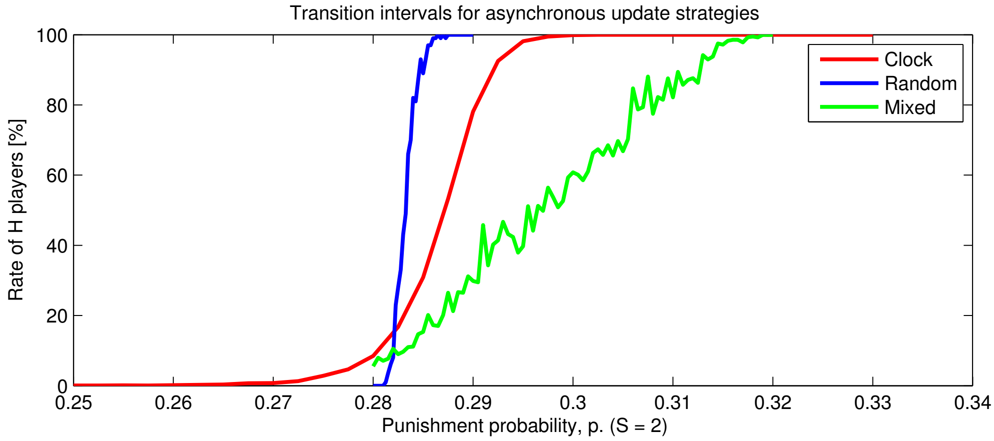


FIGURE 6.4: **Effects of asynchronous updating schemes on the p-transition interval.** Moore (r = 1) neighborhood is considered. Punishment severity is $S = 2$. The initial population contains 50%, randomly distributed, H-players. Averaged values for 100 runs are observed after 10,000 game rounds. The three asynchronous updating methods are: *Clock*, *Random*, and a combination of these two methods called 'mixed' (half players use '*Clock*' and the rest '*Random*').

The *Clock* scheme defines a period of activation for each player by assigning a timer. The period of activation is set at random for each player. Every round is considered to be an increase of the timer for every player. If the timers have exceeded the period defined, the states are updated and the timer is set to zero. The clock scheme is more commonly encountered in biological systems.

In the *Cyclic* scheme, a player is chosen at each round according to a fixed update order. This order is decided at random during initialization of the game. This method is more common met in engineering systems rather than in nature.

Fig. 6.4 illustrates the transition intervals for three asynchronous methods. Comparing to the synchronous updating strategy, an H rate stability is achieved after a longer time, since there are less interactions between players.

The most significant effect of the asynchronous updating is an important modification of the p-transition interval. While for synchronous updating scheme the p-transition interval is $(0.13, 0.18)$ (when $S = 2$), for all asynchronous update strategies the intervals are translated to higher values. The *Random* scheme presents a narrower interval and the mixed scheme a wide interval with noise (player rate instability between different simulation runs).

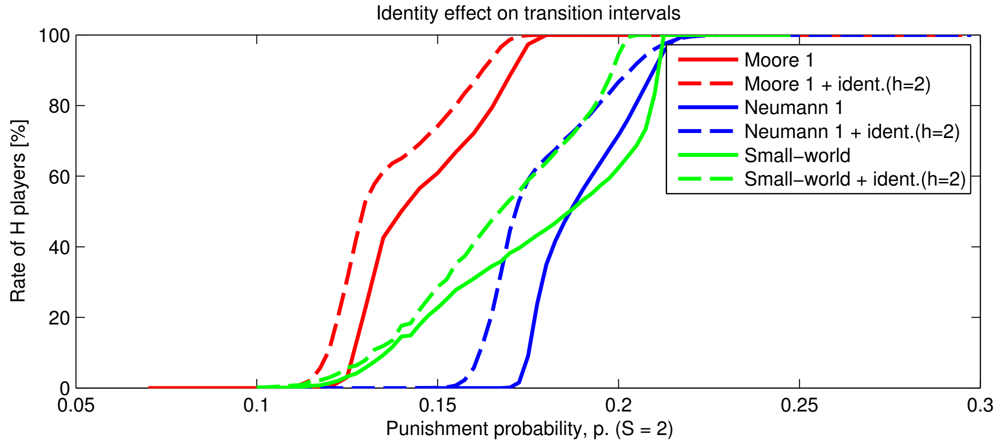


FIGURE 6.5: **Effects of player's identity on the p-transition interval.** Moore (r = 1), Neumann (r = 1) and small-world neighborhoods are considered. Punishment severity is $S = 2$. A synchronous updating scheme is used. The initial population contains 50%, randomly distributed, H-players. Averaged values for 100 runs are observed after 500 game rounds. The memory size is $h = 2$.

6.4.2.5 Effect of player's identity on the p-transition interval

The *identity* plays an important role in human society and it influences significantly people's decisions Akerlof and Kranton [2010]. In this experiment we introduce the player memory and we define a player *identity*. The identity is considered as a resistance of a player in changing his/her strategy. An honest player will tend to be remain honest and a dishonest one to remain dishonest except if a repeated experience shows that the other strategy is more successful. According to this principle, a player will imitate the best neighbor strategy only if in 2, 3 or more rounds the same type of strategy wins, otherwise it keeps its strategy unchanged.

Fig. 6.5 depicts the p-transition intervals for several types of Neighborhoods. The graphics with solid line correspond to the cases without identity and the dotted lines to cases with identity (the memory/history size is $h = 2$).

From these results it may be observed that the identity slightly changes the p-transition interval to lower values, which favors the H players. What happens if the memory size increases?

Fig. 6.6 depicts the p-transition intervals for a memory size (h) of 2 and 3. This means a player changes its strategy if three times consecutively the other strategy wins, otherwise it keeps the same strategy. For Moore (r = 1) we also tested a case when in 2 of 3 cases

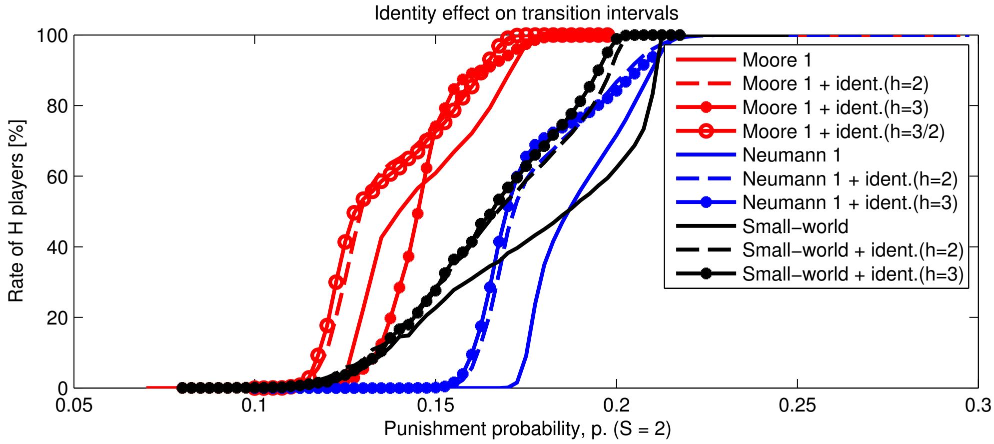


FIGURE 6.6: **Effects of player's identity on the p-transition interval.** Moore (r = 1), Neumann (r = 1) and small-world neighborhoods are considered. Punishment severity is $S = 2$. A synchronous updating scheme is used. The initial population contains 50%, randomly distributed, H-players. Averaged values for 100 runs are observed after 500 game rounds. Memory sizes $h = 2$ and $h = 3$ are considered.

the same strategy wins. The results in this case were very close to the situation when the memory size is $h = 2$.

Fig. 6.7 depicts the world state for different neighborhood typologies and memory sizes ($h = 2, 3$, and 4). It was observed that when identity is present the H clusters are larger and more stable. It also may be observed that, for each neighborhood type, the form of clusters reflects the neighborhood form - some fractal like behavior appears.

6.4.2.6 Noise effect

By 'noise' we understand here a mutation applied on the player strategy. Each round, with a certain probability, a player will switch (mutate) its strategy. This fact will introduce some kind of noise. The noise effect is analyzed in this experiment for different types of neighborhood topologies. Figure 6.8 depicts the noise effect on p-transition intervals.

For Moore and small-world neighborhoods a translation and also an extension of the p-transition interval in the presence of noise may be observed. The 'SF small-world' topology, which is a combination between scale-free and Moore topologies, presents a wider transition interval in the presence of noise. The most interesting result is that, for the scale-free topology the noise does not make a significant difference. This kind of topology seems to be more robust to noise comparing to the others. This fact may

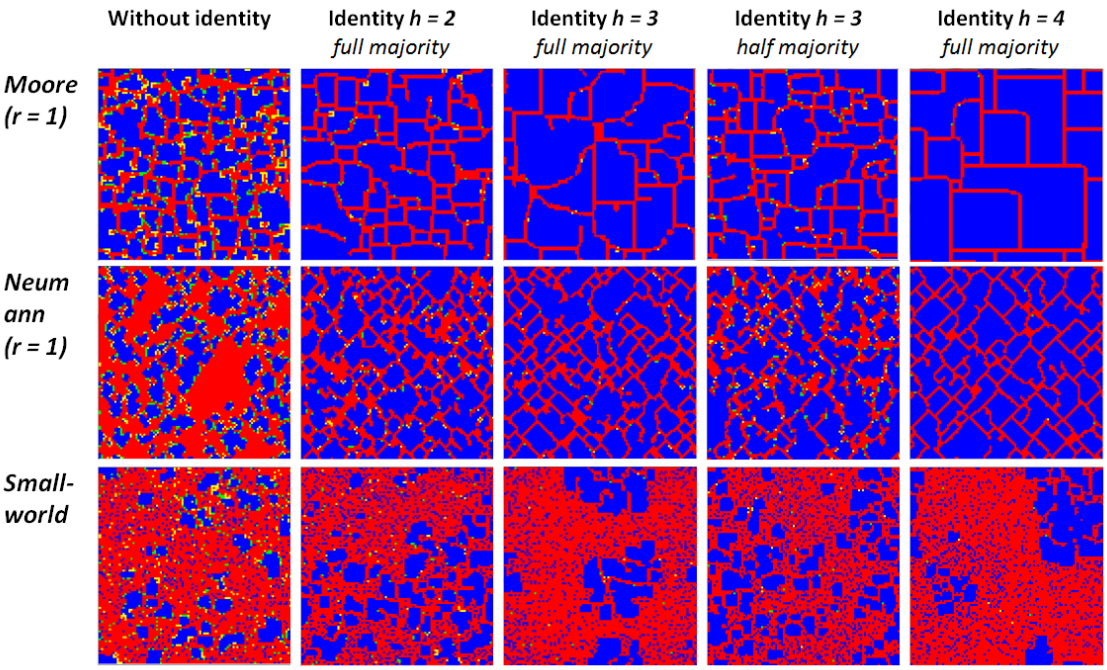


FIGURE 6.7: **World states for non-identity and identity-based players with memory sizes of 2, 3, and 4.** The initial population contains 50%, randomly distributed, H-players. The punishment probability is kept constant ($p = 0.15$ for Moore (r = 1) and Small-world neighborhoods, and $p = 0.18$ for Neumann (r = 1)). The world snapshots are taken when the rate evolution stabilizes (in this case, after 1500 rounds for $h = 2$ and 3 and after 5000 rounds for $h = 4$).

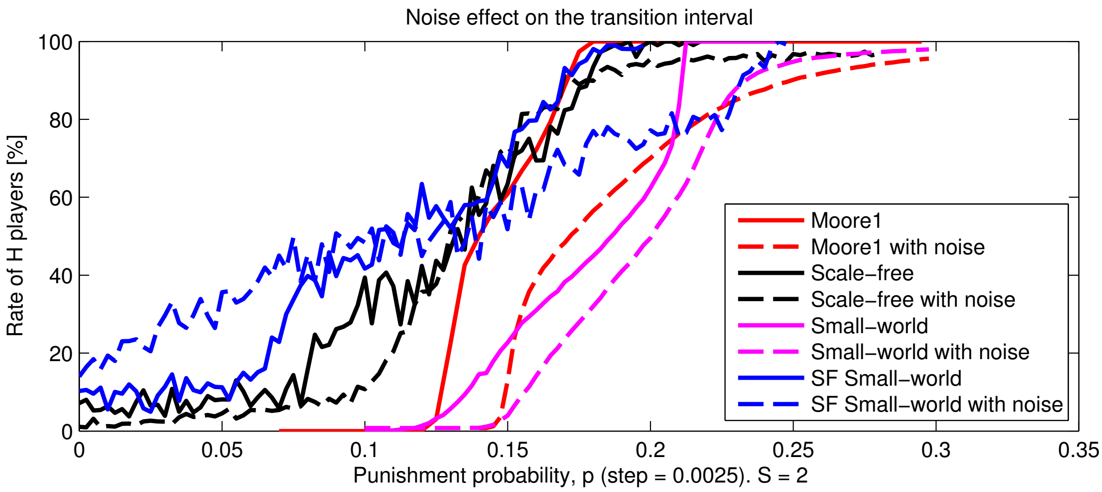


FIGURE 6.8: **Noise effect on the p-transition intervals for different neighborhood topologies.** The initial population contains 50%, randomly distributed, H-players. Moore (r = 1), Scale-free, Small-world, and 'SF small-world' neighborhoods are considered. Punishment severity is $S = 2$. A synchronous updating scheme is used. Averaged values for 100 runs are observed after 500 game rounds. The mutation probability is 0.0075. The noisy variants of p-transition intervals are depicted with dotted line.

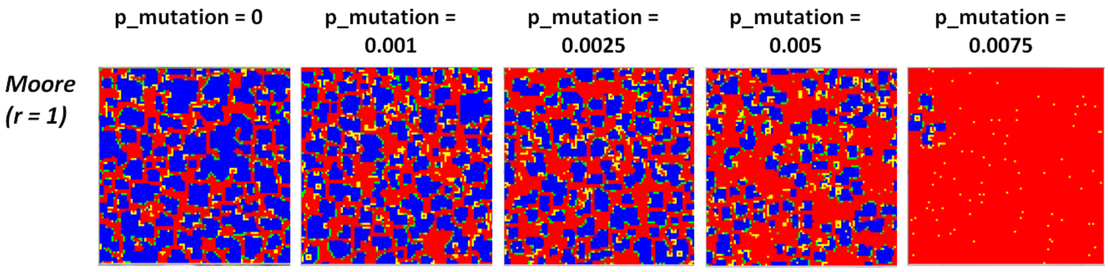


FIGURE 6.9: **World states for different noise levels after 500 rounds.** The initial population contains 50%, randomly distributed, H-players. Moore (r = 1) neighborhood is considered. The punishment severity is $S = 2$ and the punishment probability is $p = 0.15$. A synchronous updating scheme is used. Different mutation probabilities are considered in the range $0 - 0.0075$.

be explained by the presence of a minority of *hubs* in the scale-free networks, which will tend to impose their strategies (they are more interconnected than other player and therefore they play more games accumulating a higher payoff).

Fig. 6.9 illustrates the effect of the noise on the cluster structure.

The effect of the noise on the cluster structure is visible in Fig. 6.9. When the mutation probability increases the clusters become smaller in size and the H players rate decreases. In other words, the noise is not in the favor of honest players. This fact is also confirmed by the effect on the p-transition interval depicted in Fig. 6.8.

6.4.2.7 Final remarks about experiments with Social Honesty game

Experimental results with the SH game [Cremene et al., 2014b] illustrate how the behavior of a population of interacting individuals may be influenced by setting an adequate punishment severity and applying it with a certain probability. Punishment probability proven to be more important than punishment severity.

Transition intervals for punishment probability and severity have been identified in all experiments. Punishment severity proves to be ineffective when punishment probability is very low. Higher punishment probability makes it possible to reduce significantly the punishment severity, with the same effect on the honest/dishonest population rate. These results may be related to the observations about the U.S. Prohibition period when, despite a high punishment severity, the punishment probability was very low and thus ineffective Binmore [2006]. A higher punishment probability is necessary when the dishonest's advantage is higher.

Since the asynchronous updating is closer to real situations than the synchronous one, a conclusion may be that asynchronous interactions favor D players.

Player's identity changes the p-transition intervals but not significantly. There are somehow some changes induced by the identity but from other points of view. For instance, as expected, when the memory increases the evolution speed decreases because player will not change their strategy unless several times the same opposite strategy wins.

The noise modifies the p-transition interval. The modification is different for each different neighborhood topology. In general, the transition interval becomes wider in the presence of noise.

Honest strategy survival depends on the players' ability to form clusters. The initial proportion of D and H-players is less important than the initial cluster formation (groups are stronger against aggression than isolated individuals). Small size populations seem to be less predictable and less sensitive to punishment.

New dynamics, with a new relation between punishment probability and punishment severity, have been revealed through the proposed model. An epidemic of honesty is possible if model parameters are finely tuned and the cluster formation is triggered. Policy makers, various groups and organizations, law enforcement institutions may use such a model for fine tuning punishment severity and certainty towards favoring honest behavior contagion.

Chapter 7

Conclusions and future work

This chapter gives an overview of the main contributions and of the envisaged lines of inquiry following the present research.

7.1 Conclusions

An interdisciplinary unifying vision on adaptive systems, which includes also psycho-social aspects, is proposed. Services, artificial and human agents, resources and other context-related aspects are included in the proposed models. Single-objective and multi-objective evolutionary optimization algorithms, semantic tools and game-based models are used for solving various issues of adaptive systems. The concept of *complexity* is systematically analyzed for each proposed model in its many fold aspects. Computational complexity and complex systems complexity are analyzed for centralized and distributed control models. NP-hard and complex real world phenomena are analyzed.

Concerning web service composition and adaptation, we propose a general approach, that of *modeling the adaptation problem as an optimization one* [Cremene et al., 2007], [Cremene et al., 2012c]. This approach is based on concepts such as *service-context model* and *service-context distance* [Cremene et al., 2007], [Cremene et al., 2012c]. Thus, a vague concept as 'service adaptation' is turned into a clearly defined optimization problem having a very precise mathematical definition.

The *service-context adequacy* concept is very general. QoS attributes are included in this concept as a measure of user satisfaction [Pop et al., 2011b], [Cremene et al., 2014a].

The physical distance is also included in a proposed solution related to medical services optimization [Pop et al., 2011a].

Natural-language-based service composition is another field where a user-service adequacy is defined. A dictionary based word similarity metric is used [Cremene et al., 2009], [Pop et al., 2009], [Pop et al., 2010a] in order to measure the user-service distance. Two solutions for natural language based service composition are presented. The first one uses composition templates to ensure a valid result. The second one is focused on the language ambiguity and detects the missing parts form the user request.

The *service-context distances* represent functions to be minimized (or maximized) by optimization algorithms. Evolutionary approaches represent efficient approaches for problems with large search spaces. Comparative experiments with single-objective [Pop et al., 2011b] and multi-objective optimization algorithms [Cremene et al., 2014a] are conducted for several service composition/adaptation scenarios. Model checking, AI planning, and optimization techniques are combined in order to produce complete service composition and adaptation solutions [Pop et al., 2010a].

A new encoding scheme for Genetic Algorithms, based on a long genome, is proposed in [Pop et al., 2011b]. The proposed encoding scheme increases the GA convergence speed. This fact is important especially when a sub-optimal solution is satisfactory.

Several types of evolutionary algorithms have been tested. Comparative experiments are presented in [Pop et al., 2011b] and [Cremene et al., 2014a]). Results indicate that single-objective and multi-objective Differential Evolution-based algorithms provide a good trade-off between performance and time/space complexity.

It was proven that multi-objective algorithms combined with a decision making algorithm provide an approach that is superior to the classical single-objective approach in term of precision of selecting a solution in accordance with user preferences [Cremene et al., 2014a]. Performance metrics and statistical analysis have been used in order to compare the performances of different optimization algorithms.

Approaches based on Lorenz dominance have been also experimented in [Suciu et al., 2012]. This type of approach reduces the number of final solutions found by a multi-objective algorithm to a limited set of equitable, wel balanced solutions. Thus, the decision making process is facilitated.

A general problem of evolutionary algorithms is parameter tuning. An automatic tuning method is proposed in [Suciu et al., 2013b]. The results indicate performance improvements when the proposed adaptive approach is applied to standard test problems and some composite web services of high complexity.

The case of many-objective problems is analyzed in [Suciu et al., 2013a]. The result indicate that decomposition approaches represent a good choice for solving many-objective optimization problems.

Another important domain analyzed in this thesis regards the distributed decentralized systems. Contributions concerning the field of *Cognitive Radio* are presented in [Cremene et al., 2012a] and [Cremene et al., 2013]. Standard game models such as Cournot, Stackelberg and Bertrand are used to model situations where radio devices compete for a limited shared resource - the radio spectrum. Several types of game equilibria are investigated and interpreted.

A framework based on a security adaptive protocol with applications in *Mobile Cloud Computing* is presented in [Popa et al., 2013]. A game-based model for a dynamical market of web services residing in the Cloud is proposed [Cremene et al., 2012b]. The role of *Game Theory* in predicting the evolution of a distributed system is analyzed. Nash equilibrium may be achieved using a distributed algorithm [Pradelski and Peyton, 2012].

The principle of using multiple perspectives in decomposing a system is discussed and exemplified. A proposed *combiner-equalizer* module [Cremene et al., 2010], [Cremene and Cremene, 2011] for radio receivers illustrates the idea of combining two operations in one unique block. The performances of the proposed combiner-equalizer indicate important gains when multiple inputs are used.

A holistic interdisciplinary approach may not overlook the psycho-social aspect [Ropohl, 1999]. The service-context model caters to this by including the user and its interactions with services and context elements. Contributions to *Affective Computing* [Benta et al., 2009], [Benta et al., 2010b] based on ontologies and neural networks are presented. The social aspect is also thoroughly analyzed in a game theoretical analysis of behavior contagion. Social dynamics are studied using a new model called 'Social Honesty' game

[Cremene et al., 2014b]. Experiments indicate interesting dynamics based on transition intervals and cluster emergence.

7.2 Proposed vision and future lines of inquiry

The proposed vision is based on the idea that future development of technical systems should take into account an *ecological* and *holistic* approach. Many technical areas are strictly bounded to their own field, which is artificially separated from other areas. However, as we can learn from multi-agent systems and game-based models, local optimization does not necessarily lead to an optimal global state. Therefore, a more in depth analysis of the technical systems implications on other systems and on human society.

A main line of inquiry to be further pursued regards the use of *Computational Intelligence* and *Game Theory* techniques in approaching *Software Engineering* and *Telecommunications* problems.

Future work:

- Investigating a hybrid approach for an adaptation control mechanism: function-based (utility, cost) and rule-based. In some cases it is easier to define rules and in other cases it is preferable to define distance functions.
- Analyzing and experimenting complex adaptation scenarios, where a large number of service-context distances are involved. Scalability of search/optimization algorithms also needs to be studied.
- Applying the service-context model to decentralized distributed scenarios. Analyzing and comparing various distributed algorithms for complex adaptive systems.
- Further development of the proposed model for the dynamic market based on concurrent services in the Cloud and evaluate the impact of different price models. Analyze the different game equilibria that may appear in such a system.
- Finding a general method for designing decentralized algorithms based on given game equilibria. The evolutionary approach used in the proposed solutions is based

on a centralized control mechanism. However, an agent-based implementation is usually needed in practice.

- New experiments using Cognitive Radio devices ('SmartRadio' boards).
- Finishing a mobile application for psychological therapy.
- Performing additional experiments with the Social Honesty game. New results have already been obtained. Additional real world validation experiments are necessary.
- A more in depth analysis of socio-technical interactions and proposal of new game-based models.

Appendix A

Academic contributions

A.1 PhD theses co-adviser

The author is/was co-adviser to several PhD theses:

1. *Emil Bucur*, in progress, dir. Prof.dr.mat. Dumitru Dumitrescu, started in 2014.
2. *Daniela Popa*, 'Security of Mobile Cloud Applications', dir. Prof.dr.ing. Monica Borda, defended in 2013.
3. *Suciu Mihai Alexandru*, 'Evolutionary Optimization and Strategic Interactions', dir. Prof.dr.mat. Dumitru Dumitrescu, defended in 2013.
4. *Todica Valeriu*, 'Compunerea si Executia Automata a Serviciilor Web', dir. Prof.dr.ing. Mircea-Florin Vaida, defended in 2012.
5. *Florin-Claudiu Pop*, 'Compunerea si optimizarea serviciilor orientate pe nevoile utilizatorilor', dir. Prof.dr.ing. Mircea-Florin Vaida, defended in 2011.
6. *Kuderna-Iulian Benta*, 'Sisteme Senzitive la Context Personalizate Afectiv', dir. Prof.dr.ing. Costin Miron, defended in 2010.

Most of PhD students enumerated before have been also involved in research projects directed by the thesis author. This fact helped them to have a better financial support and to publish more papers.

A.2 Adaptive Systems Laboratory

The author is head of the '*Adaptive Systems Laboratory*' (website: http://asl.utcluj.ro/asl) affiliated to the Technical University of Cluj-Napoca.

The laboratory research topics, having as central metaphor '*adaptive systems*', concern application domains such as: Service-oriented Computing, Mobile Computing, Wireless Communications, Cognitive Communications, and Affective Computing.

Appendix B

Future development strategy

Several future research lines have been already proposed in the *Conclusions* chapter. The intention is to continue the inquiry lines regarding the applications of optimization and *Game Theory* in software and hardware-software multi-agent adaptive systems, keeping the complexity aspect in mind.

One must be aware that PhD coordination is a also form of *education*. Therefore, the *pedagogical* aspect should also be taken into account in addition to pure research aspects.

My future development strategy for education and research is based on several key concepts that are: '*critical thinking*', '*creative thinking*', '*personalized education*' and '*motivation-oriented education*'. These concepts are defined in the next section.

B.1 Critical and creative thinking for a personalized motivation-based learning

The key concepts used to define the personal strategy, which also represent educational values, are the following:

'*Critical thinking*' is first of all an interrogative attitude. Socrates offers one of the best examples of critical thinking within his famous dialogues. Later, K. R. Popper introduced the concept of '*critical rationality*' [Popper, 1971], which refers to the validation of scientific affirmations through the test of falsification [Popper, 2002]: only those affirmations that are potentially falsifiable can be considered as belonging to an empirical science.

In the education field, critical thinking [Browne and Keeley, 1986] may be considered as the science of asking the right questions. A practical application of critical thinking is offered by D. Rowntree in his book '*Learn how to study*' [Rowntree, 1998]. The very simple yet powerful study techniques proposed in this book represent an ideal thinking toolbox for students and researchers.

'*Creative thinking*' is an approach for generating new valuable ideas. A frequent obstacle when searching for new ideas are old ideas. Existing ideas may block new ideas that are very different from what was established as the 'common sense'. Creativity is considered a process that depends more on intuition than on reason. Therefore, creativity may appear as a mysterious process, that is difficult to control. However, E. de Bono proposes a new concept called '*lateral thinking*' [De Bono, 2010]. This concept comes with a set of thinking methods that increase the chances to produce new ideas [De Bono, 2010]. The most important principle of lateral thinking is to generate alternative perspectives on a same problem/object. This is equivalent to using a kind of '*breadth first search*' (lateral thinking) instead of '*depth first search*' (vertical thinking). The creativity is very important for research and, thanks to E. de Bono, we have today several practical tools for deliberately stimulating it.

'*Fine grained time control.*' One should be aware that exploring multiple alternatives may be costly. However, a strict and *fine grained time control* may be a solution to this problem (i.e. trying to solve a creative task in only 5 to 10 minutes). This kind of time constrained exercises values spontaneity.

'*Personalized education*' is an widely accepted ideal for any education system. It is well known that different people have different learning styles (visual, kinetic, etc.). In order to determine the personal learning style some psychological tests may be used. With the help of *e-learning platforms* such evaluation tests may be easily implemented. The first step is to be aware of the student's learning style. In a second step the learning process may be adapted to specific student needs. In general, personalized education implies additional effort and time. However, when working with a small group of master or PhD students a personalized approach is feasible.

'*Motivation-oriented education*' is a new paradigm proposed by the Thesis author, based on the idea that motivation is the first condition to any human action. A human being is never purely rational but also emotional. Motivation is strongly related to emotions. The student may be helped to identify his/her motivational profile (or motivational DNA [Lowe, 2009]). Similarly to the case of learning style, some psychological tests may be used to identify a person's motivational profile. A student aware of its inner motivation profile is more efficient. The same is true for a professor.

'*Positive thinking.*' An important motivational factor is *positive thinking* [Shahbazzadegan et al., 2013]. Beside criticism, it is highly important to *encourage* a student in order to lead him/her to success or high performance. To give and to receive feedback is a very known value for managers and leaders [Bloxham, 2003] and should be also used in education, in both directions. Negative and positive feedback needs to be wisely balanced in order to increase one's motivation. The feeling of having an important social role may offer a higher *sense* of life for a young researcher and thus may generate a strong and stable motivation.

B.2 Education: new culture, techniques, courses, and directions

B.2.1 Cultural aspects

At this moment, traditional engineering education in Romania is in general purely technical-oriented. Therefore, it might be a surprise for a Romanian engineer to hear that in Japan, for instance, the *arts* are also included in standard engineering formation. And Japan has one of the best rated educational system.

Research is strongly related to intuition and creativity. Since creativity is mainly based on producing new unexpected connections [De Bono, 2010], it might be useful for PhD students to study other domains that are not strictly related to their narrow research subject.

An aspect existing in other academic cultures but almost absent in ours (Romanian) is the culture of informal scientific discussions. For instance, in USA, France and other high developed countries researchers have dedicated places (around cafe machines or discussion rooms) for informal interactions. A culture based on free idea sharing exists in such places. New ideas appear from these informal spontaneous discussions. Explaining our ideas to a person coming from a completely different domain is a useful experience.

B.2.2 Educational techniques

My intention is to continue to use the instruments of critical [Rowntree, 1998] and creative thinking [De Bono, 2010] in my courses. The large availability of learning materials (articles, books, video tutorials, free courses available on Internet) moves the educational focus more on *how the students learn* rather than what course or book they are reading.

A new mobile communication course, presented from a critical thinking perspective, is in progress.

I also intend to develop (or reuse) an e-leaning web platform where motivational and learning profile assessment modules will be integrated.

B.2.3 New courses

I propose the following new courses for engineering formation in our faculty/department:

- *Learn how to study.* Students should become more aware that it is highly important to assume their own education. A course presenting efficient tools for learning should be presented from the very first year.
- *Pattern recognition.* Numerous engineering applications are based on pattern recognition. Such a course may be introduced in the last year, as master or doctoral course.
- *Optimization.* Numerous engineering applications requires optimization. Such a course may be introduced in the last year, as master or doctoral course.

B.2.4 New directions

In my opinion, *Interdisciplinary* and *Transdiscliplinary* [Nicolescu, 2002] approaches deserve more attention. Numerous new ideas are based on a model application from one domain to another domain. Courses for PhD students should be proposed on these topics.

B.3 Research development strategy

As a professor, even more important than being socially recognized as 'successful' is to help other people to become successful (in a deeper sense, no just in a strict material one). Therefore, a research/education strategy for a professor is to be as efficient as possible in helping students to become *honest successful* people. However, aspects as 'honesty' and 'success' may be sometimes contradictory. The next sections discuss these two issues in the academic context.

B.3.1 Research ethic

The *honesty* aspect is very important in research, especially because an important number of *fake results* are discovered with regularity in literature. Surprisingly, the highest rated journals are not an exception. Sometimes it is the case of unintentional errors. In other cases, the self-manipulation plays also an important role in committing 'useful errors'. The pressure '*to publish or perish*' leads to an important number of low quality scientific results. The number of low quality conferences and journals has increased in recent years. All these facts indicate that, when the volume is increased the quality may suffer. What finally counts is the benefit for the society.

In order to promote honesty in research some measures need to be taken. Examples of proposed actions are the following:

- Exposing examples of what is and what is not *ethic*. Explain to students concepts such as 'plagiarism' and 'intellectual property'. Discuss cases of fake results reported by scientists. Remember the fact that, in science, fake results are always discovered - it's only a matter of time.

- Develop a cultural identity based on honesty. Make honesty a value more important than other superficial values such as: fame, social status, and material benefits. It is highly important being a consistent personal example.

- Applying procedures that enable other researchers to repeat the experiments in order to validate the reported results. For instance, publishing the source code used for experiments. When possible, validating the results published by other researchers by repeating their experiments.

B.3.2 Research quality

A general *goal* of my personal development strategy is to contribute to a meaningful, high quality education and research. A higher sense of research, related to social and psychological aspects, should be pursued.

A first condition to achieve quality in research is to be aware of the *motivation* driving the person involved in the educational process (professors and the students). The second condition is to use thinking tools for an efficient learning process. Critical and creative thinking paradigms offer very powerful study instruments, which make the learning process more predictable and fruitful. A good awareness about the personal learning style may increase learning efficiency.

The following actions are examples of support measures for sustaining a high quality research:

- Investigating high impact research problems. Curiosity is an important motivational factor but it is more efficient when applied on relevant problems/challenges.
- Target high quality journals and conferences even if the volume of the scientific production may decrease by doing that.
- Maintaining connections with other researchers that may provide complementary skills and expertise. Our laboratory has active collaborations with: '*Centre for the Study of Complexity*' of Babes-Bolyai University, '*Rainbow*' and '*LEAT*' research teams from University of Nice, '*SysCom*' team from University of Savoie, and '*SensorLab*' team Josef Stefan Institute, Ljubljana.
- Promoting a stimulating environment within the laboratory. Each member should have a personal benefit and the opportunity to develop oneself. Exchange and flow of ideas is stirred in periodic discussions. Permanent laboratory members are

encouraged to propose and lead their own research projects. The most important role of the leader is to create a favorable environment for the other team members.

- Enhancing creativity by promoting an interdisciplinary vision and a personal inner equilibrium. This may be improved by sports, arts, and meditation. A researcher is not a machine publishing articles but a sensitive and complex human being. Psychological and social aspects may be sometimes more important than the strictly technical ones.

- Systematically applying to public calls for national and international research grants and projects. Proposals for Horizon 2020 are in progress. Collaborations with the *IT Cluster* of Cluj-Napoca have been initiated.

In conclusion, my development strategy follows these three essential aspects: a) continuing the research lines presented in this Thesis, b) promoting the educational values stated previously by taking the stated actions, and c) initiating new projects, identifying the research opportunities and applying for funding.

Bibliography

Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6 - Volume 6*, HICSS '00, pages 6007–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0493-0. URL http://dl.acm.org/citation.cfm?id=820262.820322.

Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag. ISBN 3-540-66550-1. URL http://dl.acm.org/citation.cfm?id=647985.743843.

George A. Akerlof. The market for 'lemons': Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84(3):488–500, 1970.

George A. Akerlof and Rachel E. Kranton. *Identity Economics: How Our Identities Shape Our Work, Wages, and Well-Being*. Princeton University Press, 2010.

A. Altland and B. Simons. *Condensed Matter Field Theory*. Cambridge University Press, 2006. ISBN 9780521845083. URL http://books.google.ro/books?id=0KMkfAMe3JkC.

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, April 2010. ISSN 0001-0782. doi: 10.1145/1721654.1721672. URL http://doi.acm.org/10.1145/1721654.1721672.

Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA '09, pages 87–102, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-414-0. doi: 10.1145/1527125.1527138. URL http://doi.acm.org/10.1145/1527125.1527138.

Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75–103, March 2011. doi: 10.1016/j.tcs.2011.03.012. URL http://hal.inria.fr/inria-00638989.

R.M. Axelrod. *The Evolution of Cooperation*. Basic Books, 2006. ISBN 9780465005642.

Robert M. Axelrod. *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press, 1997.

A. Bartha and D. Dumitrescu. Perturbation in population of pulse-coupled oscillators leads to emergence of structure. *International Journal of Computers Communications and Control*, 6(6):222–226, 2011.

A. Bartha, D. Dumitrescu, and M. Cremene. Patent pending a 00987/30.09.2011: Identificarea partilor sumei pentru probleme np complete de tip 'subset-sum' folosind un dispozitiv electronic digital, 2011.

Kuderna-Iulian Benta, Anca Rarău, and Marcel Cremene. Ontology based affective context representation. In *Proceedings of the 2007 Euro American conference on Telematics and information systems*, page 46. ACM, 2007.

Kuderna-Iulian Benta, Marcel Cremene, and Valeriu Todica. Towards an affective aware home. In *Proceedings of ICOST 2009*, 2009.

Kuderna-Iulian Benta, Marcel Cremene, Nicoleta Ramona Giba, Ulises Xolocotzin Eligio, and Anca Rarau. Secondary emotions deduction from context. In Tarek Sobh, editor, *Innovations and Advances in Computer Sciences and Engineering*, pages 165–170. Springer Netherlands, 2010a. ISBN 978-90-481-3657-5. doi: 10.1007/978-90-481-3658-2_29. URL http://dx.doi.org/10.1007/978-90-481-3658-2_29.

Kuderna-Iulian Benta, Marcel Cremene, and Amalia Hoszu. Training the behaviour preferences on context changes. *ECEASST*, 28, 2010b.

J. Bertrand. Book review of theorie mathematique de la richesse sociale and of recherches sur les principles mathematiques de la theorie des richesses. *Journal de Savants*, 67: 499–508, 1883.

K. Binder and W. Kob. *Glassy Materials and Disordered Solids: An Introduction to Their Statistical Mechanics*. World Scientific, 2011. ISBN 9789814350174. URL http://books.google.ro/books?id=lEOWIZeFSP4C.

K. Binmore. The origins of fair play. Papers on Economics and Evolution 2006-14, Max Planck Institute of Economics, Evolutionary Economics Group, Nov 2006.

Ken Binmore. *Game theory - a very short introduction*. Clarendon Press, 2007. ISBN 978-0-19-921846-2.

Giulio Biroli, Simona Cocco, and Rémi Monasson. Phase transitions and complexity in computer science: an overview of the statistical physics approach to the random satisfiability problem. *Physica A: Statistical Mechanics and its Applications*, 306:381–394, 2002. URL http://www.sciencedirect.com/science/article/B6TVG-4562998-4/1/a5e119ab4782ef5a662e849f3b4a2aae.

E. Bloxham. *Economic value management: applications and techniques*. Wiley Finance. J. Wiley, 2003. ISBN 9780471354260. URL http://books.google.ro/books?id=3PYpAQAAMAAJ.

Stephen Blundell and Katherine M. Blundell. *Concepts in thermal physics*. Oxford University Press, 2006. ISBN 9780198567691. URL http://www.worldcat.org/isbn/9780198567691.

F. Brauer and C. Castillo-Chavez. *Mathematical models in population biology and epidemiology*, volume 40. Springer Verlag, New York, 2001. ISBN 978-1-4614-1686-9. URL http://www.google.com/search?client=safari&rls=en-us&q=Mathematical+models+in+population+biology+and+epidemiology&ie=UTF-8&oe=UTF-8.

N. Browne and S.M. Keeley. *Asking the right questions: a guide to critical thinking.* Prentice-Hall, 1986. ISBN 9780130494382. URL http://books.google.ro/books?id=m0hYAAAAMAAJ.

Stephen G. Brush. History of the lenz-ising model. *Rev. Mod. Phys.*, 39(4):883–893, oct 1967. doi: 10.1103/RevModPhys.39.883. URL http://link.aps.org/doi/10.1103/RevModPhys.39.883.

Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani. An approach for qos-aware service composition based on genetic algorithms. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO'05, pages 1069–1075, New York, NY, USA, 2005. ACM. ISBN 1-59593-010-8. doi: http://doi.acm.org/10.1145/1068009.1068189. URL http://doi.acm.org/10.1145/1068009.1068189.

C.S. Chang, D.Y. Xu, and H.B. Quek. Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system. *IEE Proceedings - Electric Power Applications*, 146(5):577–583, 1999. doi: 10.1049/ip-epa:19990481.

Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, April 1936. ISSN 00029327. doi: 10.2307/2371045. URL http://dx.doi.org/10.2307/2371045.

Barry A. Cipra. An introduction to the ising model. *Am. Math. Monthly*, 94(10):937–959, December 1987. ISSN 0002-9890. doi: 10.2307/2322600. URL http://dx.doi.org/10.2307/2322600.

Carlos A. Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer, 2007.

Karen S. Cook. Networks, norms, and trust: The social psychology of social capital. *Social Psychology Quarterly*, 68(1):4–14, March 2005. ISSN 0190-2725. URL http://www.ingentaconnect.com/content/asoca/spq/2005/00000068/00000001/art00002.

David W. Corne and Joshua D. Knowles. Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In *Proceedings of* *the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, pages 773–780, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-697-4. doi: 10.1145/1276958.1277115. URL http://doi.acm.org/10.1145/1276958.1277115.

D. Cornforth, D.G. Green, D. Newth, and M. Kirley. Do artificial ants march in step? ordered asynchronous processes and modularity in biological systems. In Russell K. Standish, Mark A. Bedau, and Hussein A. Abbass, editors, *Artificial Life*, volume 8, page 28. MIT Press, 2002.

L. Cremene and M. Cremene. Patent ro126731-a2: Adaptive method and device for the unified processing of signals for radio receivers. Derwent Primary Accession No.: 2011-Q12597, 2011.

L.C. Cremene, D. Dumitrescu, R. Nagy, and M. Cremene. Game theoretical modelling for dynamic spectrum access in tv whitespace. *CoRR*, abs/1209.5387, 2012a.

L.C. Cremene, N. Gaskó, M. Cremene, and D. Dumitrescu. A game theoretical perspective on small-cell open capacity sharing in cognitive radio environments. In *Proceedings of NEW2AN*, pages 247–259, 2013.

Ligia Cremene. *Tehnici adaptive in sisteme de comunicatii wireless*. Editura Casa Cartii de Stiinta, 2010.

Ligia Chira Cremene, Nicolae Crisan, and Marcel Cremene. An adaptive combiner-equalizer for multiple-input receivers. In *Novel Algorithms and Techniques in Telecommunications and Networking*, pages 385–390. Springer Netherlands, 2010.

M. Cremene, A. Bartha, and D. Dumitrescu. Patent pending a 00884/12.09.2011: Dispozitiv electronic pentru rezolvarea problemei np complete 'subset-sum', 2011a.

M. Cremene, M. Suciu, N. Gasko, and L. Cremene. Solutii echilibrate în optimizarea multi-criterială a serviciilor. mecanisme de adaptare distribuite. Technical report, Technical University of Cluj-Napoca, 2012b. URL http://helios.utcluj.ro/te252/rapoarte/2012/predare/ro/RT2_TE252_2012.pdf.

M. Cremene, M. Suciu, D. Pallez, and D. Dumitrescu. Multi-objective optimization for qos-aware service composition. *IEEE Transactions on Services Computing (under review)*, 2014a.

Marcel Cremene, Michel Riveill, and Christian Martel. Autonomic adaptation solution based on service-context adequacy determination. *Electronic Notes in Theoretical Computer Science*, 189:35–50, 2007.

Marcel Cremene, J.-Y. Tigli, Stéphane Lavirotte, Florin-Claudiu Pop, Michel Riveill, and Gaëtan Rey. Service composition based on natural language requests. In *Services Computing, 2009. SCC'09. IEEE International Conference on*, pages 486–489. IEEE, 2009.

Marcel Cremene, Kuderna-Iulian Benta, and Michel Riveill. Une approche de type optimisation pour l'auto-adaptation de services mobiles. In *7es journées francophones Mobilité et Ubiquité - Ubimob 2011*, 2011b.

Marcel Cremene, Michel Riveill, Anca Rarau, Costin Miron, Benta Iulian, and Valeriu Todica. Adaptation mechanism based on service-context distance for ubiquitous computing. *International Journal of Computers, Communications & Control*, 7(1), 2012c.

Marcel Cremene, D. Dumitrescu, and Ligia Cremene. A strategic interaction model of punishment favoring contagion of honest behavior. *PLOS ONE*, 9(1):e87471, 2014b.

Martin Davis. The church-turing thesis: Consensus and opposition. In *Lecture Notes in Computer Science*, volume 3988, pages 125–132. Springer, 2006. ISBN 3-540-35466-2. URL http://dblp.uni-trier.de/db/conf/cie/cie2006.html#Davis06.

Robyn M. Dawes. *Social dilemmas, economic self-interest and evolutionary theory*, pages 53–79. Springer, New York, 1991.

E. De Bono. *Lateral Thinking: Creativity Step by Step*. Harper colophon books. Harper Collins, 2010. ISBN 9780062043276. URL http://books.google.ro/books?id=v7dv3DASmSAC.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

Frederic Dugardin, Farouk Yalaoui, and Lionel Amodeo. New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *European Journal of Operational Research*, 203(1):22–31, May 2010.

D. Dumitrescu, R.I. Lung, and T.D. Mihoc. Evolutionary equilibria detection in non-cooperative games. In *Applications of Evolutionary Computing*, volume 5484 of *Lecture Notes in Computer Science*, pages 253–262. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01128-3. doi: 10.1007/978-3-642-01129-0_29. URL http://dx.doi.org/10.1007/978-3-642-01129-0_29.

D. Dumitrescu, A. Bartha, and M. Cremene. Patent ro128241-a2: Process for solving subset sum problem in the field of the theory of computational complexity. Derwent Primary Accession Number: 2013-F20991, 2013.

Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. The MIT Press, http://wordnet.princeton.edu/, 1998.

Sujuan Feng and Eiko Seidel. Self-organizing networks (son) in 3gpp long term evolution. Technical report, Nomor Research GmbH, Munich, Germany, 2008.

François Fouss, Youssef Achbany, and Marco Saerens. A probabilistic reputation model based on transaction ratings. *Inf. Sci.*, 180(11):2095–2123, June 2010. ISSN 0020-0255. doi: 10.1016/j.ins.2010.01.020. URL http://dx.doi.org/10.1016/j.ins.2010.01.020.

Bo Fu, Yang Xiao, Hongmei Deng, and Hui Zeng. A survey of cross-layer designs in wireless networks. *Communications Surveys Tutorials, IEEE*, 16(1):110–126, 2014. ISSN 1553-877X. doi: 10.1109/SURV.2013.081313.00231.

D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991. ISBN 9780262061414. URL http://books.google.ro/books?id=pFPHKwXro3QC.

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.

Malcolm Gladwell. *The Tipping Point: How Little Things can Make a Big Difference*. Little Brown, Boston, 2000.

Avner Greif. Reputation and coalitions in medieval trade: Evidence on the maghribi traders. *The Journal of Economic History*, 49(04):857–882, 1989. doi: 10.1017/s0022050700009475. URL http://dx.doi.org/10.1017/s0022050700009475.

G.R. Grimmett. *Percolation*. Grundlehren der mathematischen Wissenschaften. Springer, 2010. ISBN 9783642084423. URL http://books.google.ro/books?id=LjWDcgAACAAJ.

Mark Harman, S. Afshin Mansouri, and Yuanyuan Zhang. Search-based software engineering: Trends, techniques and applications. *ACM Comput. Surv.*, 45(1):11:1–11:61, December 2012. ISSN 0360-0300. doi: 10.1145/2379776.2379787. URL http://doi.acm.org/10.1145/2379776.2379787.

George T. Heineman and William T. Councill, editors. *Component-based Software Engineering: Putting the Pieces Together*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001. ISBN 0-201-70485-4.

John H. Holland. *Hidden Order: How Adaptation Builds Complexity*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1995. ISBN 0-201-40793-0.

Vincent Hourdin, Daniel Cheung-Foo-Wo, Stéphane Lavirotte, and Jean-Yves Tigli. Ubiquarium informatique: Une plate-forme pour l'étude des équipements informatiques mobiles en environnement simulé. In *Troisième Journées Francophones Mobilité et Ubiquité(UbiMob), poster*, page 2, Paris, France, 2006.

IBM. Smarterplanet vision website, 2006. URL http://www.ibm.com/smarterplanet/us/en/overview/ideas/.

Klamroth Kathrin and Jorgen Tind. Constrained optimization using multiple objective programming. *Journal of Global Optimization*, 37:325–355, 2007. doi: 10.1007/s10898-006-9052-x. URL http://dx.doi.org/10.1007/s10898-006-9052-x.

Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, January 2003. ISSN 0018-9162. doi: 10.1109/MC.2003.1160055. URL http://dx.doi.org/10.1109/MC.2003.1160055.

G. Kiczales. Aspect-oriented programming. *ACM Comput. Surv.*, 28(4es), December 1996. ISSN 0360-0300. doi: 10.1145/242224.242420. URL http://doi.acm.org/10.1145/242224.242420.

Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold. An overview of aspectj. In *Proceedings of the 15th European Conference on Object-Oriented Programming*, ECOOP '01, pages 327–353, London, UK, UK, 2001.

Springer-Verlag. ISBN 3-540-42206-4. URL http://dl.acm.org/citation.cfm?id=646158.680006.

P. Kollock. The emergence of exchange structures: An experimental study of uncertainty, commitment and trust. *Am J Soc*, 100(2):313–345, 1994.

M. M. Kostreva and O. Wodzimierz. Linear optimization with multiple equitable criteria. *RAIRO - Operations Research*, 33:275–297, 1999. ISSN 1290-3868. doi: 10.1051/ro:1999112. URL http://dx.doi.org/10.1051/ro:1999112.

Saku Kukkonen and Jouni Lampinen. Gde3: the third evolution step of generalized differential evolution. In *IEEE Congress on Evolutionary Computation*, pages 443–450, 2005a. doi: 10.1109/CEC.2005.1554717.

Saku Kukkonen and Jouni Lampinen. GDE3: the third evolution step of generalized differential evolution. In *IEEE Congress on Evolutionary Computation*, pages 443–450, 2005b.

Jouni Lampinen and Ivan Zelinka. *Mechanical engineering design optimization by differential evolution*, pages 127–146. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999. ISBN 0-07-709506-5. URL http://portal.acm.org/citation.cfm?id=329055.329094.

D.P. Landau and K. Binder. *A Guide To Monte Carlo Simulations In Statistical Physics*. Cambridge University Press, 2005. ISBN 9780521842389. URL http://books.google.ro/books?id=HbxQxS7tHiYC.

Alberto Leporati and Miguel A. Gutiérrez-Naranjo. Solving subset sum by spiking neural p systems with pre-computed resources. *Fundam. Inform.*, 87(1):61–77, 2008. URL http://dblp.uni-trier.de/db/journals/fuin/fuin87.html#LeporatiG08.

Hongbin Liang, Dijiang Huang, and Daiyuan Peng. On economic mobile cloud computing model. In Martin L. Gris and Guang Yang 0001, editors, *MobiCASE*, volume 76 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 329–341. Springer, 2010. ISBN 978-3-642-29335-1. URL http://dblp.uni-trier.de/db/conf/mobicase/mobicase2010.html#LiangHP10.

L.W.S. Loijens, J.M.M. Theuws, A.J. Spink, P. Ivan, and M. Den Uyl. Analyzing facial expressions with facereader: Evaluation of improvements to the software for exploring consumer behavior. In *Proceedings of the 5th European Conference on Sensory and Consumer Research*, page 8, 2012.

Tamara Lowe. *Get Motivated!: Overcome Any Obstacle, Achieve Any Goal, and Accelerate Your Success with Motivational DNA*. Crown Business, 2009. ISBN 978-0385524698. URL http://books.google.ro/books?id=Ts6aRHzK_PMC.

S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. Wiley-Interscience series in discrete mathematics and optimization. J. Wiley & Sons, 1990. ISBN 9780471924203. URL http://books.google.ro/books?id=0dhQAAAAMAAJ.

D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, TR Payne, et al. Owl-s: Semantic markup for web services, 2004.

Jose Antonio Martin H., Javier de Lope, and Dario Maravall. Adaptation, anticipation and rationality in natural and artificial systems: computational paradigms mimicking nature. *Natural Computing*, 8(4):757–775, December 2009. doi: 10.1007/s11047-008-9096-6. URL http://dx.doi.org/10.1007/s11047-008-9096-6.

Deborah L. McGuinness and Frank van Harmelen. Owl web ontology language overview. Technical Report REC-owl-features-20040210, W3C, 2004.

Peter Mell and Timothy Grance. The nist definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST), September 2011. URL http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

D. R. Merkin. *Introduction to the Theory of Stability*. Texts in Applied Mathematics. U.S. Government Printing Office, 1997. ISBN 9780387947617. URL http://books.google.ro/books?id=kNVoqQ9gE9MC.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Dordrecht, 1999.

J. Mitola and Jr. Maguire, G.Q. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, Aug 1999. ISSN 1070-9916.

Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124 – 143, 1996. ISSN 0899-8256. doi: http://dx.doi.org/10.1006/game.1996.0044. URL http://www.sciencedirect.com/science/article/pii/S0899825696900445.

Daniel S. Nagin and Greg Pogarsky. Integrating celerity, impulsivity, and extralegal sanction threats into a model of general deterrence: Theory and evidence. *Criminology*, 39(4):865–892, 2001. ISSN 1745-9125.

J.F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

Z. Néda, E. Ravasz, Y. Brechet, T. Vicsek, and A.L. Barabási. Self-organizing processes: The sound of many hands clapping. *Nature*, 403(6772):849–850, 2000. doi: 10.1038/35002660.

Zoltán Néda, Razvan Florian, Mária Ravasz, András Libál, and Géza Györgyi. Phase transition in an optimal clusterization model. *Physica A: Statistical Mechanics and its Applications*, 362(2), Apr 2006. URL http://www.sciencedirect.com/science/article/B6TVG-4H27B02-5/1/eb533cd07e362a168564a95e2252f7fd.

Ferrante Neri and Ville Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artif. Intell. Rev.*, 33(1-2):61–106, 2010. ISSN 0269-2821. doi: 10.1007/s10462-009-9137-2. URL http://dx.doi.org/10.1007/s10462-009-9137-2.

David Newth and David Cornforth. Asynchronous spatial evolutionary games. *Biosystems*, 95(2):120–129, 2009. ISSN 0303-2647. doi: http://dx.doi.org/10.1016/j.biosystems.2008.09.003. URL http://www.sciencedirect.com/science/article/pii/S0303264708001949.

Muaz Niazi and Amir Hussain. Agent-based computing from multi-agent systems to agent-based models: A visual survey. *Scientometrics*, 89(2):479–499, November 2011. ISSN 0138-9130. doi: 10.1007/s11192-011-0468-9. URL http://dx.doi.org/10.1007/s11192-011-0468-9.

B. Nicolescu. *Nous, la particule et le monde.* Editions du Rocher, 2002. ISBN 9782268038247. URL http://books.google.ro/books?id=8-rUPAAACAAJ.

Tatsuya Okabe, Yaochu Jin, and Bernhard Sendhoff. A critical survey of performance indices for multi-objective optimisation. *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, 2:878–885, 2003. doi: 10.1109/CEC.2003.1299759.

Mihai Oltean and Oana Muntean. Solving the subset-sum problem with a light-based device. *Natural Computing*, 8(2):321–331, 2009. URL http://dblp.uni-trier.de/db/journals/nc/nc8.html#OlteanM09.

Martin J. Osborne. *An introduction to game theory.* Oxford Univ. Press, 2004. ISBN 978-0-19-512895-6.

E. Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action.* Political Economy of Institutions and Decisions. Cambridge University Press, 1990.

Rosalind W. Picard. *Affective Computing.* MIT Press, Cambridge, MA, USA, 1997. ISBN 0-262-16170-2.

F.-C. Pop, M. Cremene, M.-F. Vaida, and A. Serbanescu. Medical services optimization using differential evolution. In *International Conference on Advancements of Medicine and Health Care through Technology*, pages 72–77. Springer Berlin Heidelberg, 2011a.

Florin Pop, Marcel Cremene, M-F Vaida, and Michel Riveill. On-demand service composition based on natural language requests. In *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, pages 45–48. IEEE, 2009.

Florin-Claudiu Pop, Marcel Cremene, Mircea Vaida, and Michel Riveill. Natural language service composition with request disambiguation. In *Service-Oriented Computing*, pages 670–677. Springer Berlin Heidelberg, 2010a.

Florin-Claudiu Pop, Marcel Cremene, and Mircea-Florin Vaida. Workflow based service composition using genetic algorithms. *Acta Technica Napocensis*, 51(2), 2010b.

Florin-Claudiu Pop, Denis Pallez, Marcel Cremene, Andrea Tettamanzi, Mihai Suciu, and Mircea Vaida. Qos-based service optimization using differential evolution. In

Proceedings of the 13th annual conference on Genetic and evolutionary computation, pages 1891–1898. ACM, 2011b.

Daniela Popa, Marcel Cremene, Monica Borda, and Karima Boudaoud. A security framework for mobile cloud applications. In *Roedunet International Conference (RoEduNet), 2013 11th*, pages 1–4. IEEE, 2013.

K. R. Popper. *The Open Society and Its Enemies: The spell of Plato*. Princeton paperbacks. Princeton University Press, 1971. ISBN 9780691019680. URL http://books.google.ro/books?id=v-yrLmQNRvsC.

K. R. Popper. *The Logic of Scientific Discovery*. Routledge Classics. Taylor & Francis, 2002. ISBN 9780203994627. URL http://books.google.ro/books?id=T76Zd20IYlgC.

Bary S.R. Pradelski and Young H. Peyton. Learning efficient nash equilibria in distributed systems. *Games and Economic Behavior*, 75(2):882 – 897, 2012. ISSN 0899-8256. doi: http://dx.doi.org/10.1016/j.geb.2012.02.017. URL http://www.sciencedirect.com/science/article/pii/S0899825612000383.

A.K. Qin, V.L. Huang, and P.N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *Evolutionary Computation, IEEE Transactions on*, 13(2):398–417, 2009.

Jinghai Rao and Xiaomeng Su. A survey of automated web service composition methods. In *Proceedings of the First International Conference on Semantic Web Services and Web Process Composition*, SWSWPC'04, pages 43–54, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-24328-3, 978-3-540-24328-1. doi: 10.1007/978-3-540-30581-1_5. URL http://dx.doi.org/10.1007/978-3-540-30581-1_5.

Anca Rarau, Kuderna-Iulian Benta, and Marcel Cremene. Multifaceted based language for pervasive services with deterministic and fully defined behavior. In *Pervasive Services, IEEE International Conference on*, pages 76–79. IEEE, 2007.

Stanley Reiter and Leonid Hurwicz. *Designing Economic Mechanisms*. Cambridge Press, 2006.

Tea Robic and Bogdan Filipic. Demo: Differential evolution for multiobjective optimization. In *EMO*, pages 520–533, 2005.

Günter Ropohl. Philosophy of socio-technical systems. *Society for Philosophy and Technology*, 4:59–71, 1999.

Michael G. Rosenblum, Arkady S. Pikovsky, and Jürgen Kurths. From phase to lag synchronization in coupled chaotic oscillators. *Phys. Rev. Lett.*, 78:4193–4196, Jun 1997. doi: 10.1103/PhysRevLett.78.4193. URL http://link.aps.org/doi/10.1103/PhysRevLett.78.4193.

Don Ross. Game theory, 2010. URL http://plato.stanford.edu/archives/fall2010/entries/game-theory/.

D. Rowntree. *Learn how to Study: A Realistic Approach*. Time Warner paperbacks, 1998. ISBN 9780751520880. URL http://books.google.ro/books?id=7RmOQwAACAAJ.

P. Samimi and A. Patel. Review of pricing models for grid amp; cloud computing. In *Computers Informatics (ISCI), 2011 IEEE Symposium on*, pages 634–639, March 2011.

Oliver Schuetze, Xavier Equivel, Adriana Lara, and Carlos A. Coello Coello. Some comments on gd and igd and relations to the hausdorff distance. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '10, pages 1971–1974, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0073-5. doi: 10.1145/1830761.1830837. URL http://doi.acm.org/10.1145/1830761.1830837.

Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, May 2006. ISSN 1541-1672. doi: 10.1109/MIS.2006.62. URL http://dx.doi.org/10.1109/MIS.2006.62.

Bita Shahbazzadegan, Mehdi Samadzadeh, and Moslem Abbasi. The relationship between education of emotional intelligence components and positive thinking with mental health and self-efficacy in female running athletes. *Procedia - Social and Behavioral Sciences*, 83(0):667 – 671, 2013. ISSN 1877-0428. doi: http://dx.doi.org/10.1016/j.sbspro.2013.06.126. URL http://www.sciencedirect.com/science/article/pii/S1877042813011932.

Karl Sigmund. Punish or perish? retaliation and collaboration among humans. *Trends in Ecology and Evolution*, 22(11):593 – 600, 2007. ISSN 0169-5347.

John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

L. Smith. *Chaos: A Very Short Introduction*. Very Short Introductions. OUP Oxford, 2007. ISBN 9780192853783. URL http://books.google.ro/books?id=0fsrTBuLb3cC.

R.V. Sole. *Phase Transitions*. Primers in Complex Systems. Princeton University Press, 2011. ISBN 9780691150758. URL http://books.google.ro/books?id=8RcLuv-Ll2kC.

Jim Spohrer. Service science progress and directions (keynote talk). ICSOC 2010, 2010. URL http://disi.unitn.it/~icsoc/icsoc10/keynotes.html.

Mihai Suciu, Marcel Cremene, Florin Pop, and Dumitru Dumitrescu. Equitable solutions in qos-aware service optimization. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*, pages 1537–1538. ACM, 2012.

Mihai Suciu, Marcel Cremene, and Dumitru Dumitrescu. Exploring some scalarization techniques for emoas. In *Proceeding of the fifteenth annual conference companion on Genetic and evolutionary computation conference companion*, pages 99–100. ACM, 2013a.

Mihai Suciu, Denis Pallez, Marcel Cremene, and Dumitru Dumitrescu. Adaptive moea/d for qos-based web service composition. In *Evolutionary Computation in Combinatorial Optimization*, pages 73–84. Springer Berlin Heidelberg, 2013b.

Attila Szolnoki and Matjaž Perc. Correlation of positive and negative reciprocity fails to confer an evolutionary advantage: Phase transitions to elementary strategies. *Phys. Rev. X*, 3:041021–1–11, Nov 2013. doi: 10.1103/PhysRevX.3.041021. URL http://link.aps.org/doi/10.1103/PhysRevX.3.041021.

Jianhua Tao and Tieniu Tan. Affective computing: A review. In Jianhua Tao, Tieniu Tan, and RosalindW. Picard, editors, *Affective Computing and Intelligent Interaction*, volume 3784 of *Lecture Notes in Computer Science*, pages 981–995. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-29621-8. doi: 10.1007/11573548_125. URL http://dx.doi.org/10.1007/11573548_125.

Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey, Vincent Hourdin, Daniel Cheung-Foo-Wo, Eric Callegari, and Michel Riveill. Wcomp middleware for ubiquitous computing: Aspects and composite event-based web services. *Annals of Telecommunications (AOT)*, 64(3):197–214, 2009.

Valeriu Todica, M.-F. Vaida, and Marcel Cremene. Formal verification in web services composition. In *Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on*, pages 195–200. IEEE, 2012.

H.R. Varian. *Intermediate microeconomics: a modern approach*. WW Norton & Co., 2006.

Andrew von Hirsch, Anthony E. Bottoms, Elizabeth Burney, and P-O Wikstrom. *Criminal Deterrence and Sentence Severity: An Analysis of Recent Research*. Oxford: Hart Publishing, 1999.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, Second Edition, 1947.

Heinrich von Stackelberg. *Market Structure and Equilibrium: 1st Edition Translation into English*. Springer, 2011.

Beibei Wang, Yongle Wu, and K.J. Ray Liu. Game theory for cognitive radio networks: An overview. *Comput. Netw.*, 54(14):2537–2561, October 2010. ISSN 1389-1286. doi: 10.1016/j.comnet.2010.04.004. URL http://dx.doi.org/10.1016/j.comnet.2010.04.004.

Yong Wang, Zixing Cai, and Qingfu Zhang. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evolutionary Computation*, 15(1):55–66, 2011. URL http://dx.doi.org/10.1109/TEVC.2010.2087271.

Feng Xue, A.C. Sanderson, and R.J. Graves. Multi-objective differential evolution and its application to enterprise planning. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 3535–3541, 2003. doi: 10.1109/ROBOT.2003.1242137.

Osman N. C. Yilmaz, Jyri Hämäläinen, and Seppo Hämäläinen. Self-optimization of random access channel in 3rd generation partnership project long term evolution.

Wirel. Commun. Mob. Comput., 11(12):1507–1517, 2011. ISSN 1530-8669. doi: 10.1002/wcm.1217. URL http://dx.doi.org/10.1002/wcm.1217.

Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11:712–731, 2007.

Yan Zhang and Mohsen Guizani, editors. *Game Theory for Wireless Communications and Networking*. CRC Press, 2011.

Zhongshan Zhang, Keping Long, Jianping Wang, and Falko Dressler. On Swarm Intelligence Inspired Self-Organized Networking: Its Bionic Mechanisms, Designing Principles and Optimization Approaches. *IEEE Communications Surveys and Tutorials*, 16(1):513–537, February 2014. doi: 10.1109/SURV.2013.062613.00014.

Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011. URL http://dblp.uni-trier.de/db/journals/swevo/swevo1.html#ZhouQLZSZ11.

Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, December 2005. ISSN 1387-3326. doi: 10.1007/s10796-005-4807-3. URL http://dx.doi.org/10.1007/s10796-005-4807-3.

Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.2440.