

Stochastic Plasticity: Biologically-inspired Approaches In Computer Engineering And Information Technology

by

Lucian PRODAN

Submitted for the degree of

Habilitation

POLITEHNICA UNIVERSITY TIMISOARA
FACULTY OF AUTOMATION AND COMPUTING
POLITEHNICA UNIVERSITY TIMISOARA

June 2025

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

Abstract

This thesis represents a sum of our relevant research since 2006 carried at Politehnica University Timisoara. Essential to any research path, its defining roadway cannot be assimilated to a straight line, with influences and incursions related to a range of fields that are seemingly disjoint. Drawing its vitals from the field of bio-inspired computing, this thesis reunites results from research in the uncertainty-dominated world of stochastic processes, which are crucial for understanding and mastering the reliability of computing processes and systems, with those coming from harnessing the power of information plasticity. Our contributions relate to challenging multi-objective optimization problems, such as that of urban traffic, and machine learning applications providing a better understanding of the complexity of human gestures for detecting violence or monitoring treatment compliance.

The beginning contains post-doctoral research results concerning dependable computing, focusing on the requirements of reliable systems, described by qualities like reliability, availability, and performability, or by fault tolerance strategies implemented by self-testing and self-repairing mechanisms. Our research is motivated by the need for reliable, critical operations, ranging from medical procedures to banking and industrial processes, and machinery needed for prolonged operations in remote, isolated environments, such deep-sea, nuclear, or exploration settings. Going from large to small, dependability remains relevant in the field of nanoelectronics, where environmental aggressiveness and the absence of human intervention at such a scale require engineered reliability rather than reliability as a natural property. Deeply connected to the intricacies of dependability, bio-inspired computing and quantum computing represent two emerging technology vectors that are considered key for the future of computing.

Exploring the limits of engineering ontogenetic processes in digital computing, the Embryonics project has refined a hierarchical organization of bio-inspired computing systems representing a network of parallel entities (the population), made of a multitude of multicellular organisms, each consisting of a finite number of cells (the cellular level), which in turn are made of a finite number of electronic molecules (the molecular level). Therefore, in Embryonics a biological organism is equivalent to a parallel computing system, a biological cell to a processor, while the molecule in biology may be seen as the smallest, programmable element in digital electronics.

If nature employs complex chemical bonds to assemble molecules and then build up

the hierarchy, the world of digital electronics is constrained to existing resources (hardware) that have to be managed through information (software). The physics of quantum computing impose special care, since the microscopic quantum states are prone to frequent errors, which can alter the consistency of the quantum computation processes. The engineering and dependability assessment of such innovative structures is crucial for the hardware-software codesign implied and for the advancement of the fields.

The field of dependable computing extends beyond random events determined by physics or environmental causes, the presence of malicious actors being a modern concern that justifies cryptography and steganography. Cryptographic algorithms must resist a range of cryptanalysis while steganographic techniques must avoid detection by steganalysis attacks. Therefore a next part of the thesis is dedicated to results toward implementing high-speed encryption designs within the AES standard as well as smart data rearrangement for high-capacity steganographic algorithms.

Real life presents quite a large diversity of fields that require multi-objective optimization that could greatly benefit from algorithms, methods and tools that are specific to computer science and engineering. Such situations involve several conflicting objectives, where dependencies on each other result in non-linear variations whenever attempting to maximize any of them. The last part of this thesis is dedicated to presenting our research concerning the application of genetic algorithms and complex networks in the challenging problem of achieving optimal urban traffic flows. Also within the realm of bio-inspired computing, we approached some topics by employing machine learning, in an attempt to provide a better understand of the traffic dynamics, in particular to conditions that lead to congestion build-up.

Understanding human actions in a resource-constrained environment has been explored for the detection of violence in live video streams, targeting applications for processing live video streams from the public surveillance cameras. A dream of our team, designing and implementing real-world devices, has been approached from a health perspective, to provide continuous monitoring of treatment adherence by recognizing hand gestures in real-time. Such an effort is bound to expose the engineering challenges when experimenting with hand gestures, in order to deliver reliable results from a device that produces information that is dense in values but narrow in scope. Delivering the quality level and the attention to detail that would warrant the precise operation of a health-monitoring device remains a challenge worth future investigation.

Acknowledgements

This thesis reunites the main results of the research efforts carried over the various fields of computer science and engineering related to dependable computing, seeking novel, unconventional solutions to well-known, conventional problems. Such the story of this journey is told by one voice, there are many people that influenced and supported it through their extraordinary vision, overwhelming kindness, and enduring patience. Needless to say, without them it would have been impossible to stand where I am today.

My first thankful thoughts go to all my family, who bore patiently with my struggles while keeping me constantly surrounded with all their love and caring. To a true mentor, professor Mircea Vladutiu (also my former PhD advisor), who tenaciously sought after my academic ascension with utmost friendship and caring that was nothing short of paternal.

I am grateful for the foreign experiences made possible by professors Andrew Tyrrell and Gianluca Tempesti from University of York, who gave me the opportunity to join their team and work on a research grant together with professor Gabriel Dragffy and professor Anthony Pipe from University of the West of England. This experience has later brought me in the position of external examiner for several Ph.D. candidates, which chiseled and refined my own research dedication and perseverance.

Back to the lab where much of the action took place, I would like to thank all my friends and colleagues from the Advanced Computing Systems and Architectures lab: Alexandru Iovanovici, Flavius Opritoiu, Alexandru Topirceanu, and also my close collaborators and former Ph.D. candidates: Dacian Avramoni, Cristian Cosariu and Cristian Ruican. At times when plans seemed to fall apart, somebody was always finding reasons for cheering up! A special place for my academic digital twin, now professor Mihai Udrescu, whose deep knowledge and original perspectives helped maintain research course. Many more people helped this achievement, and I am thankful to each one of them. If I have failed to name someone, please accept my heartfelt apologies — your support has not gone unnoticed.

Finally, I want to thank those who financed our activity: our national funding agency UEFISCDI, as well as private companies Nokia and Linde Healthcare.

Table of Contents

I	SCIENTIFIC, PROFESSIONAL AND ACADEMIC ACHIEVEMENTS	IX
1	Introduction	1
1.1	Research Path	3
1.2	Motivation and Outline	11
2	Leveraging Stochastics for Dependability Assessments in Emerging Technologies	13
2.1	Challenging Reliability: Hurdles and Mitigation	19
2.2	Biologically-Inspired Computing	23
2.2.1	The Embryonics Project	25
2.2.2	Reconfiguration Strategies for Hierarchical Fault-Tolerance	28
2.2.3	Partition for Dependability	32
2.3	Quantum Computing	36
2.4	Biologically-Inspired Quantum Computing	39
3	Expanding Dependability: Security Through Steganography and Cryptography	42
3.1	High-Speed AES with Concurrent Error Detection	42
3.2	High-Capacity Steganography	48
4	Leveraging Informational Plasticity for Multi-Objective Optimization	54
4.1	Collaborative Monitoring for Urban Traffic	56
4.2	Softening Intersections for Permeability Enhancement	59
4.3	Leveraging Network Science for Urban Traffic Optimization	62
4.3.1	Employing Complex Networks for Optimal Deployment of Wireless Sensors	63
4.3.2	What Makes Urban Networks Social?	66
4.3.3	Deployment of Traffic Lights and Complex Networks	72
4.3.4	Tracing Congestion by Social Markers	75
4.3.5	Correlating Social Importance with Real Traffic Data	81
4.4	Adding Informational Plasticity with Machine Learning	85
4.4.1	Intelligent Traffic with Epigenetics	85
4.4.2	Applied Epigenetics for Violence Detection in Video Streams	90
4.4.3	Embedded Epigenetics for Gesture Recognition	96
II	RESEARCH AND CAREER PROSPECTS	102
5	Research Directions and Career Developments	103
5.1	Identity	103
5.2	Purpose	106
5.3	Mission	111

5.4 A Final Word	112
References	113

List of Tables

3.1	VHDL synthesis results for main modules	44
3.2	Overhead for SubBytes and InvSubBytes modules	46
3.3	Area overhead for AES verification architectures	47
3.4	Fault detection rate for AES	47
3.5	Performance comparison	52
3.6	Performance comparison update	53
4.1	Aggregated data over the two months testing period	59
4.2	Green time values defined for a specific intersection	61
4.3	Green time values computed for a case-study intersection	62
4.4	The <i>SocialCity</i> algorithm applied over different cities	70
4.5	Topological characteristics for 3 large cities	70
4.6	The <i>SIGS</i> algorithm applied over different cities	72
4.7	Topological characteristics for the city of Timisoara	77
4.8	Quantitative data retrieved from the Here Maps API for each intersection .	82
4.9	Metrics describing the signaling solutions	88
4.10	Accuracy values for different r values	94
4.11	Accuracy values for different classifiers	95
4.12	Data recordings for recognition of pill intake gestures	99
4.13	Accuracy assessment for gestures similar to pill intake	100

Figure table

1.1	Milestones for my academic development	2
2.1	Emerging technology vectors [54].	17
2.2	Estimated vs field-recorded data.	22
2.3	Hierarchical organization in Embryonics [90].	27
2.4	Two-level hierarchical self-repair [69].	28
2.5	Self-repair at the cellular level.	30
2.6	Reliability estimations for logic structures.	31
2.7	Reliability estimations for an organism.	32
2.8	Fault-tolerance strategies in memory structures.	35
2.9	Overview of the QUERIST [90].	39
2.10	Quantum genetic algorithm (left) and chromosome encoding (right) [95].	40
2.11	Flow for the meta-heuristics genetic algorithm (MH-QCS) [96].	41
3.1	The BIST process. [80]	44
3.2	Fault detection applied to AES's multiplicative inversion for 500 injected faults. LFSR, ATPG and binary counter were used as test pattern generators. [80]	45
3.3	Architecture for testing the AES inversion. [81]	46
3.4	A high-speed AES architecture [79].	48
3.5	Our steganographic encoding process. [70]	50
3.6	Matching the secret data and the carrier image [70].	51
3.7	Well-known images used as carriers.	52
4.1	The POE model and placement of focus applications: urban traffic optimization (UTO) and information plasticity with machine learning (IPML).	56
4.2	System architecture (a) and web application interface (b) [49].	58
4.3	Proposed traffic optimization stack [30].	60
4.4	Traffic flow values distributed over a four way intersection [30].	61
4.5	Case study map of Timisoara central area [30].	62
4.6	Vissim simulation results [49].	63
4.7	SIDEWISE optimization of relay node throughput [51].	65
4.8	SIDEWISE optimization of cost and propagation delay [51].	65
4.9	Social fingerprint on transportation [42].	67
4.10	Road network of Budapest, with each node representing an intersection [117].	68
4.11	Betweenness distributions for Budapest [117].	68
4.12	Betweenness for a district in Budapest, before and after applying <i>Social-City</i> [117].	69
4.13	Road network of downtown Los Angeles, with each node representing an intersection [117].	71
4.14	Road network of Sendai, with each node representing an intersection [117].	71
4.15	Hierarchical structure of intelligent traffic lights [50].	73
4.16	Resolution and communities for 5 cities [50].	74
4.18	Recursive application of <i>STiLO</i> onto the community representing the Circumvalatiunii quarter of the city of Timisoara [50].	75

4.17	Urban network communities in Timisoara [50].	75
4.19	Recursive application of <i>STiLO</i> onto the community representing the Mehala quarter of the city of Timisoara [50].	76
4.20	A tech/business district of Timisoara [10].	77
4.21	Queue lengths and number of stops [50].	78
4.22	Simulation results showing oscillating behavior [10].	79
4.23	Overview of the road network for the city of Timisoara [6].	79
4.24	Case-study for the city of Timisoara [6].	80
4.25	Data sample for traffic conditions on each of the crossroads [6].	81
4.26	Deriving the intersection congestion factor Jam Score using Here Maps API [6].	83
4.27	Distribution of the JamScore for the city of Timisoara [6].	83
4.28	JamScore values for the city of Timisoara [6].	84
4.29	A busy intersection in Timisoara: Calea Dorobantilor - Gheorghe Adam [112].	86
4.30	Signaling plan for the intersection, before and after applying the Firefly algorithm [112].	87
4.31	Best fitness values [112].	87
4.32	Daily status for Stan Vidrighin street [7].	90
4.33	One hour Jam Factor predictions for Stan Vidrighin street [7].	90
4.34	Proposed model for violence detection in video streams [121].	92
4.35	Parameter variation [121].	93
4.36	Gesture classification [44].	97
4.37	Evolution of gesture identification performance during training [44].	98
5.1	STEM core domains and their relationship.	105
5.2	Major corridors subject to green wave in the city of Timisoara.	109
5.3	A wearable device for gesture recognition for assessing compliance with medical treatment.	110

Part I

SCIENTIFIC, PROFESSIONAL AND ACADEMIC ACHIEVEMENTS

Chapter 1

Introduction

Contents

1.1 Research Path	3
1.2 Motivation and Outline	11

It would be impossible to even consider the fields of science, technology, engineering and mathematics decoupled from their intrinsic emphasis on innovation, problem-solving, and critical thinking. And since they all come through the employment of computing systems, it is only natural that abilities within these topics transform into a compelling force that nurtures an inquisitive spirit and drives research.

In an age where technology permeates every facet of modern living, the significance of computer science and engineering is proven by the ubiquity of digital devices and infrastructure. Innovations born of computer science and engineering have revolutionized the way we live and interact with the world around us. Choosing to pursue and broaden my education along these fields represented both a pragmatic choice and a bet to a profoundly enriching endeavor.

The main attractors that helped build a solid foundation for my education are represented in figure 1.1, beginning with the vertical (classical) axis that includes concepts that fall under the topics of Computer Architecture and Algorithm Design, which I became interested with as a college freshman. As my time as a student grew by, topics related to digital design and the underpinnings of computer systems became ever more enticing, leading to a later preference for topics related to Computer Reliability and Testing, accompanied by techniques to achieve and implement fault-tolerance. To complete the bachelor degree I went abroad to carry out the work for my graduation thesis at the Swiss Federal Institute of Technology in Zurich (ETHZ), where my knowledge and skills would be put under test by a first real implementation of a working system [Ciressan1998].

The appetite for research was strengthened during a period of studying abroad at the Swiss Federal Institute of Technology in Lausanne (EPFL), when I became dedicated to pursuing research on the novel axis of figure 1.1, more precisely in *bio-inspired* comput-

ing.

What followed was a deep dive into concepts related to pure computing entangling with those coming from biology, physics and bio-chemistry, uncovering scientific nuances and engineering implications that arguably describe the challenges but also the greatness of the pursuit: a successful engineer is bound to combine the ingenuity that leads to making and building with the intellectual and scientific challenge of discovering and understanding. It became clear that achieving the Ph.D. will bring a new direction of personal endeavor, defined by unconventional approaches under the forms of bio-inspired and quantum computing. Even if the milestones depicted in figure 1.1 could be approached individually, however, as I found out, the best results would only come when understanding their synergies.

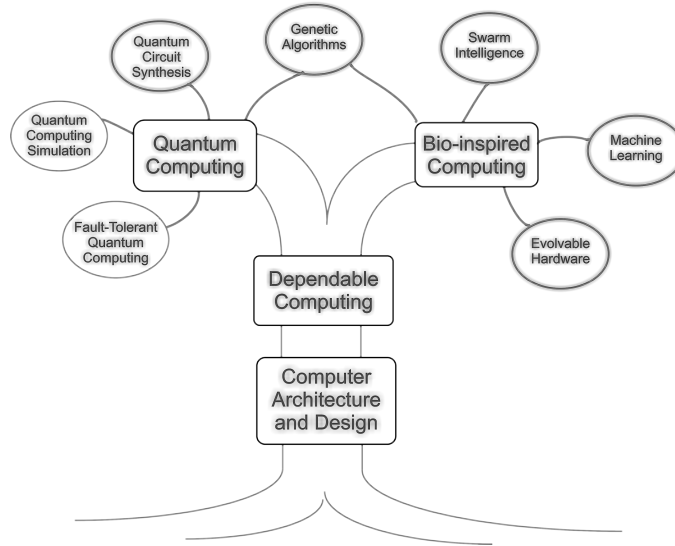


Figure 1.1: Milestones for my academic development.

As a graduate student and research assistant at the EPFL, I became exposed and attracted by the topics of biologically-inspired computing, which left a significant footprint over most of my scientific efforts. Under the supervision of prof. Daniel Mange and his (at the time) Ph.D. student Gianluca Tempesti, I became a member of the Logic Systems Laboratory (<https://infoscience.epfl.ch/entities/orgunit/021941cf-dd76-49bd-851a-b4fe99bdf69b>) and began my explorative journey on the Embryonics project, which will later influence my doctoral thesis. After leaving the EPFL and joining Politehnica University Timisoara, a period of intense research work began under new circumstances, as a doctoral candidate.

Staying within the academia meant a natural continuation of the study efforts, aiming

for the final milestone in any academic pursuit, which is called *habilitation*. This would provide new purpose and energy, and also represent the endowment required for inspiring and leading research through the work of the next generation Ph.D. students.

1.1 Research Path

After defending my doctoral thesis, During this period, my research efforts and results revolve around bio-inspired and quantum computing, with activities being supported by the following research grants:

Grants

- (G1) Daniel Mange (director), Lucian Prodan (member), Hector Fabio Restrepo, Gianluca Tempesti, et al., "Implementation of self-repairing and self-replicating processors in a universal cellular structure", R&D Grant SNF 21-54113.98, years 1998-1999, total value 83932 CHF;
- (G2) Mircea Vladutiu (Director), Lucian Prodan (member), Antonius Stanciu, Mihai Udrescu, et al. "Promotion and Institutionalized Management of Continuous Education Activities (Promovarea si managementul institutionalizat al activitatilor de educatie permanenta)", EU/WB Grant CNFIS 55, years: 1999-2002, total value: 155105.33 USD;
- (G3) Mircea Vladutiu (director), Alexandru Amaricai, Versavia Ancusa, Alin Anton, Oana Boncalo, Lucian Prodan (member), Mihai Udrescu, Nicolae Velciov, "Bioinspired Design of Applications on Reconfigurable Platforms (Proiectarea Bioinspirata a Aplicatiilor pe Platforme Reconfigurabile)", R&D Grant CNCSIS Type A 643/2005, years 2005-2007, total value 67800 ROL;
- (G4) Lucian Prodan (director), Versavia Ancușă, Razvan Bogdan, Marius Cavasdan, Elena Doandes, Cristian Ruican, Lucian Variu, "Estimating Dependability For Emerging Bioinspired Systems With Hierarchical Strategies For Reconfiguration Through Classical And Interdisciplinary Techniques (Estimarea dependabilitatii sistemelor emergente bioinspirate cu strategii ierarhice de reconfigurare prin metode clasice si interdisciplinare)", R&D Grant CNCSIS Type A 371/2007, years 2007-2008, total value 126409 RON;

- (G5) Mircea Vladutiu (director), Alexandru Amaricai, Oana Boncalo, Lucian Prodan, Mihai Udrescu, "Bio-Inspired Computing Architectures for Quantum and Reversible Logical Circuits (Arhitecturi bio-inspirate de calcul pentru circuite logice reversibile si cuantice)", PNII-IDEI-17/2007, years 2007-2009, total value 440011 RON;
- (G6) Tony Pipe, Gabriel Dragffy, Andy Tyrrell, Gianluca Tempesti, Jon Timmis, Prodan Lucian, Mohammad Samie, Anca Popescu, Paul Bremner, Omar Qadir, Yang Liu, "Self-healing Cellular Architectures for Biologically-inspired Highly Reliable Electronic Systems", R&D Grant EPSRC EP/F062192/1, years 2008-2010, total value 353671 GBP;
- (G7) Valentin Muresan (director), Mihai Micea (UPT manager), Cristian Cuna, Vladimir Cretu, Mircea Vladutiu, Horia Ciocarlie, Doru Todinca, Mihai Udrescu, Lucian Prodan (member), et al., "FALX DACIAE: Software Development Tools and Processes for Advanced Multimedia Applications on Mobile Phone Multi-Core Architectures", R&D Project POSCCE/A2-O2.1.1/449/11844, years 2010-2012, total value 506400 EUR;
- (G8) Lucian Prodan (director), Dacian Avramoni, Gabriel Baban, Cristian Cosariu, Alexandru Iovanovici, Flavius Opritoiu, Iosif Szeidert-Subert, Alexandru Topirceanu, Mihai Udrescu, "Experimental Assessment of a Self-Adaptive Intelligent Transportation System (Validare experimentală a unui sistem de transport inteligent bazat pe algoritmi)", R&D Grant UEFISCDI PN-III-P2-2.1-PED-2016-1518, years 2017-2018, total value 468425 RON;

The main research activity results were reported in the following (selected) publications:

Books

- (B1) Lucian Prodan "Self-Repairing Memory Arrays Inspired by Biological Processes", Editura Politehnica, Timisoara, 2009, 211 pages, ISBN 978-973-625-820-6;
- (B2) Mihai Udrescu, Lucian Prodan, "Emerging Computing Systems: Quantum Computing From a Computer Engineering Perspective", Editura Politehnica, Timisoara, 2013, 128 pages, ISBN 978-606-554-684-4;

- (B3) Flavius Opritoiu, Lucian Prodan, Mihai Udrescu, "Arithmetics in Computing Systems: Design and Implementation Issues (Aritmetica în sistemele de calcul: probleme de design și implementare)", Editura Politehnica, Timisoara, 2013, 136 pages, ISBN 978-606-554-566-3;
- (B4) Lucian Prodan, "Microprogrammed Control in Computing Systems (Controlul microprogramat in sistemele de calcul)", Editura Politehnica, Timisoara, 2023, 103 pages, ISBN 978-606-35-0511-9.

Journal papers

- (J1) Lucian Prodan, Mihai Udrescu, Oana Boncalo, Mircea Vladutiu, "Design for Dependability in Emerging Technologies", ACM Journal of Emerging Technologies in Computing, 3(2), Article 6, 2007;
- (J2) Lucian Prodan, "Bioinspiration will always cost more than nature pays", Communications of the ACM, 50(4), 2007;
- (J3) Mihai Udrescu, Lucian Prodan, Mircea Vladutiu, "Simulated Fault Injection Methodology for Gate-Level Quantum Circuit Reliability Assessment", Simulation Modelling Practice and Theory, 23, 1, 60–70, 2012;
- (J4) Alexandru Iovanovici, Dacian Avramoni, Lucian Prodan, "A dataset of urban traffic flow for 13 Romanian cities amid lockdown and after ease of COVID19 related restrictions", Elsevier Data in Brief, ISSN 2352-3409, vol. 32, 2020, DOI: 10.1016/j.dib.2020.106318.

Conference papers

- (C1) Lucian Prodan, Gianluca Tempesti, Daniel Mange, Andre Stauffer. Biology Meets Electronics: The Path to a Bio-Inspired FPGA. Proc. 3rd International Conference on Evolvable Systems (ICES2000), Edinburgh, Scotland, 2000, Springer LNCS 1801, pp. 187-196, ISBN 3-540-67338-5;
- (C2) Lucian Prodan, Gianluca Tempesti, Daniel Mange, Andre Stauffer. Embryonics: Artificial Cells Driven by Artificial DNA. Proc. 4th International Conference on Evolvable Systems (ICES2001), Tokyo, Japan, Oct 3-5 2001, Springer LNCS 2210, pp. 100-111, ISBN 3-540-42671-X;

- (C3) Lucian Prodan, Gianluca Tempesti, Daniel Mange, Andre Stauffer. Embryonics: Electronic Stem Cells. Proc. AAAI 8th International Conference on Artificial Life, (MIT Press, Cambridge MA, 2003), Sydney, Australia, Dec 9-13, 2002, pp. 101-105, ISBN 0-262-69281-3;
- (C4) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Self-Repairing Embryonic Memory Arrays. Proc. IEEE NASA/DoD Conference on Evolvable Hardware (EH'04), Seattle WA, USA, 2004, pp. 130-137, ISBN 0-7695-2145-2;
- (C5) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Self-Repairing Embryonic Memory Arrays. Proc. IEEE NASA/DoD Conference on Evolvable Hardware (EH'04), Seattle WA, USA, 2004, pp. 130-137, ISBN 0-7695-2145-2;
- (C6) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Survivability of Embryonic Memories: Analysis and Design Principles. Proc. IEEE NASA/DoD Conference on Evolvable Hardware (EH'05), Washington DC, USA, 2005, pp. 280-289, ISBN 0-7695-2399-4;
- (C7) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Multiple-Level Concatenated Coding in Embryonics: A Dependability Analysis. Proc. GECCO (ACM-SIGEVO), Washington DC, USA, 2005, pp. 941-948, ISBN 1-59593-010-8;
- (C8) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Reliability Assessment in Embryonics Inspired by Fault-Tolerant Quantum Computation. Proc. 2nd ACM International Conference on Computing Frontiers (CF'05), Ischia, Italy, May 4 - 6 2005, pp.323-333, ACM Press ISBN 1-59593-018-3;
- (C9) Mihai Udrescu, Lucian Prodan, Mircea Vladutiu. Improving Quantum Circuit Dependability with Reconfigurable Quantum Gate Arrays. Proc. 2nd ACM International Conference on Computing Frontiers (CF'05), Ischia, Italy, May 4 - 6 2005, pp. 133-144, ACM Press ISBN 1-59593-018-3;
- (C10) Mihai Udrescu, Lucian Prodan, Mircea Vladutiu. Implementing Quantum Genetic Algorithms: A Solution Based on Grover's Algorithm. Proc. 3rd ACM International Conference on Computing Frontiers (CF'06), Ischia, Italy, May 2 - 5 2006, pp. 71-82, ACM Press, ISBN 1-59593-302-6;

- (C11) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. A Dependability Perspective on Emerging Technologies. Proc. 3rd ACM International Conference on Computing Frontiers (CF'06), Ischia, Italy, May 2 - 5 2006, pp. 187-198, ACM Press, ISBN 1-59593-302-6;
- (C12) Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Fault-Tolerant Memory Design and Partitioning Issues in Embryonics. Proc. 8th International Conference on Evolvable Systems: From Biology to Hardware (ICES 2008), LNCS (Springer), vol. 5216, Prague, Czech Republic, September 21-24 2008, pp. 372-381, ISSN: 0302-9743;
- (C13) Cristian Ruican, Mihai Udrescu, Lucian Prodan, Mircea Vladutiu. Genetic Algorithm Based Quantum Circuit Synthesis with Adaptive Parameters Control. Proc. IEEE Congress on Evolutionary Computation (CEC), Trondheim, Norway, May 18-21 2009, pp. 896-903, ISBN 1-4244-2959-2;
- (C14) Cristian Cosariu, Lucian Prodan, Mircea Vladutiu. Toward Traffic Movement Optimization Using Adaptive Inter-Traffic Signaling. Proc. 14th IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2013), Budapest, Hungary, November 19-21, 2013, pp. 539-544, ISBN 978-1-4799-0194-4;
- (C15) Alexandru Iovanovici, Lucian Prodan, Mircea Vladutiu. Collaborative environment for road traffic monitoring. Proc. 13th International Conference on ITS Telecommunications (ITST 2013), Tampere, Finland, November 5-7, 2013, pp. 232-237, ISBN 978-1-4799-0846-2;
- (C16) Alexandru Iovanovici, Alexandru Topirceanu, Mihai Udrescu, Lucian Prodan, Mircea Vladutiu. Heuristic Optimization of Wireless Sensor Networks Using Social Network Analysis. Proc. 6th International Workshop on Soft Computing Applications (SOFA 2014), Series: Advances in Intelligent Systems and Computing, Chapter: Soft Computing Applications, Vol. 356, 2015, pp 663-671, ISBN 978-3-319-18295-7;
- (C17) Alexandru Topirceanu, Alexandru Iovanovici, Cristian Cosariu, Mihai Udrescu, Lucian Prodan, Mircea Vladutiu. Social Cities: Redistribution of Traffic Flow in Cities Using a Social Network Approach. Proc. 6th International Workshop on Soft Computing Applications (SOFA 2014), Series: Advances in Intelligent

Systems and Computing, Chapter: Soft Computing Applications, Vol. 356, 2015, pp. 39-49, ISBN 978-3-319-18295-7;

- (C18) Alexandru Iovanovici, Cristian Cosariu, Lucian Prodan, Mircea Vladutiu. A hierarchical Approach in Deploying Traffic Lights Using Complex Network Analysis. Proc. 18th International Conference System Theory, Control and Computing (IC-STCC), Sinaia, Romania, October 17-19 2014, pp. 791-796;
- (C19) Cristian Cosariu, Alexandru Iovanovici, Lucian Prodan, Mihai Udrescu, Mircea Vladutiu. Bio-Inspired Redistribution of Urban Traffic Flow Using a Social Network Approach. IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25-28 May 2015, pp. 77-84, ISBN 978-1-4799-7492-4;
- (C20) Gabriel Baban, Alexandru Iovanovici, Cristian Cosariu, Lucian Prodan. Determination of the critical congestion point in urban traffic networks: a case study. 14th International Scientific Conference on Informatics, Poprad, Slovakia, 14-16 November 2017, pp. 18-23, doi: 10.1109/INFORMATICS.2017.8327215;
- (C21) Gabriel Baban, Alexandru Iovanovici, Cristian Cosariu, Lucian Prodan. High Betweenness Nodes and Crowded Intersections: An Experimental Assessment by Means of Simulation. IEEE 12th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, 17-19 May 2018, pp. 339-344 doi: 10.1109/SACI.2018.8440989;
- (C22) Alexandru Iovanovici, Dacian Avramoni, Anca-Maria Moscovici, Lucian Prodan. Traffic micro-simulation as a validation tool for developing genetic algorithms for green wave optimization. IEEE 14th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, 21-23 May 2020, pp. 147-152 doi: 10.1109/SACI49304.2020.9118841.

I also served as Program Committee member for the following conferences, specialized in the topics targeted during the post-PhD research period:

- (PC1) ACM International Conference on Computing Frontiers (CF), years 2006, 2007;
- (PC2) ACM Genetic and Evolutionary Computation Conference (GECCO), year 2007;
- (PC3) ACM Journal on Emerging Technologies in Computing, year 2007;

- (PC4) IEEE International Conference on Evolvable Systems (ICES), year 2008, 2016;
- (PC5) International Conference on Computer and Information Technology (CIT), year 2007, 2008, 2009, 2015, 2016, 2017, 2021;
- (PC6) Intl. Conf. on Dependable, Autonomic and Secure Computing (DASC), years 2015, 2016;
- (PC7) IEEE International Conference on High Performance Computing and Communications (HPCC), years 2018, 2020, 2021
- (PC8) International Conference on System Theory, Control and Computing (ICSTCC), year 2017, 2019, 2020;
- (PC9) IEEE Intl. Symposium on Applied Computational Intelligence and Informatics (SACI), year 2019;
- (PC10) International Conference on Advanced Information Networking and Applications (AINA), years 2021, 2024;
- (PC11) IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), year 2024;

As a recognition for the research activity, I served as reviewer for the following conferences and journals:

- (Rev1) ACM International Conference on Computing Frontiers (CF), years 2006, 2007;
- (Rev2) ACM Genetic and Evolutionary Computation Conference (GECCO), year 2007;
- (Rev3) ACM Journal on Emerging Technologies in Computing, year 2007;
- (Rev4) IEEE International Conference on Computer and Information Technology (ICCIT), year 2010;
- (Rev5) IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), year 2010;
- (Rev6) IEEE International Conference on Evolvable Systems (ICES), year 2010, 2016;
- (Rev7) International Journal of Parallel Programming (IJPP), year 2011;

- (Rev8) IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), year 2012, 2019, 2020;
- (Rev9) IEEE International Symposium on Embedded Multicore SoCs (McSoC), year 2012;
- (Rev10) IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), year 2012;
- (Rev11) IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), year 2013;
- (Rev12) Journal of Information Processing Systems (JIPS), year 2013;
- (Rev13) BioSystems, year 2013;
- (Rev14) IEEE Symposium Series on Computational Intelligence (SSCI), year 2014, 2016, 2020;
- (Rev15) Intl. Conf. on Dependable, Autonomic and Secure Computing (DASC), year 2015;
- (Rev16) Design Automation Conference (DAC), year 2016;
- (Rev17) IEEE Intelligent Transportation Systems Conference (ITSC), year 2016;
- (Rev18) Genetic Programming and Evolvable Machines, year 2015;
- (Rev19) International Conference on Computer and Information Technology (CIT), year 2010, 2011, 2015, 2016, 2017;
- (Rev20) International Conference on System Theory, Control and Computing (ICSTCC), year 2018, 2019, 2020;
- (Rev21) IEEE International Symposium on Parallel and Distributed Processing with Applications, year 2020;
- (Rev22) IEEE Symposium Series on Computational Intelligence (SSCI), year 2016, 2020;
- (Rev23) Special Topic Conference organized by the European Federation for Medical Informatics (STC2020), year 2020;
- (Rev24) IEEE 2023 Congress on Evolutionary Computation (CEC), years 2020, 2021, 2023;
- (Rev25) Genetic Programming and Evolvable Machines, years 2021;

- (Rev26) IEEE International Conference on High Performance Computing and Communications (HPCC), year 2018, 2019, 2020, 2021;
- (Rev27) Transportation Research Record, year 2021;
- (Rev28) International Conference on Advanced Information Networking and Applications (AINA), years 2021, 2022, 2024;
- (Rev29) IEEE Conference on Artificial Intelligence (CAI), year 2024
- (Rev30) IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), year 2024;

Also, I served as an External Examiner for the Ph.D viva for five candidates from Cranfield University (UK), years 2014, 2020, and 2021.

1.2 Motivation and Outline

Following the retirement of prof. Daniel Mange in 2005, the team developing the Embryonics project has become dispersed, with people spreading in other universities across Switzerland and Europe. It became clear that, without the driving force of its leader, further development outside of EPFL would be quite challenging, if possible at all. This event happened to coincide with finalizing the Ph.D. thesis, so a quest for finding new research goals began.

As it turns out, bio-inspired computing extends much further than the Embryonics project, with connections to other fields that are quite generous themselves, such as dependable computing, high-performance computing, and quantum computing. As a consequence, given the intrinsic variety provided by these fields, with so much left untouched or approached partially, the quest for new and challenging research subjects transformed from a revolution into an evolution.

The rest of this thesis follows the natural development of the research timeline, describing some of the most notable findings. Since the majority of our work deals with some degree of uncertainty, understanding stochastic processes and how they reflect in computing processes is paramount. The second chapter is dedicated to understanding reliability and how to harness stochastic processes to ensure valid results. Achieving higher reliability levels through a two-layered hierarchical reconfiguration is discussed

in the general context of the Embryonics project, with some connections to the field of quantum computing, which offers inspiration in error modeling and draws inspiration for achieving quantum genetic algorithms. The third chapter steps away from fault modeling into error control, notably into the fields of cryptography and steganography. Apparently disjunct, these fields are united by aiming to protect information integrity: the former employs a base, binary perspective for protecting information itself by transforming it into a different, illegible form, while the latter aims at protecting information transactions through information embedding onto common photographic file formats. In a leap from uncertainty (information disorganized or affected by errors) to plasticity (information is reorganized to extend meaning or accommodate new knowledge), the fourth chapter presents our perspective in applying bio-inspired computing to some of the most difficult and modern multi-objective optimization problem represented by urban traffic. The application of genetic algorithms and machine learning is presented in the context of finding a global optimum in the control signaling of the urban traffic, with an extension of machine learning perspectives over understanding human gestures. Finally, the fifth chapter is concerned with personal thoughts and aspirations toward future research directions and career developments. The thesis ends with listing some of the relevant references that supported our research.

Chapter 2

Leveraging Stochastics for Dependability Assessments in Emerging Technologies

With modern society heavily dependent on computers, the expectation for their precise and rapid functionality is paramount. Since the inception of computer design, a primary objective has been to maximize performance wherever feasible; currently, designers prioritize dependability when delivering optimal performance. If performance comes under a variety of attributes concurring to produce ever increasing amounts of computing power, delivering this power in a dependable manner represents an ongoing discussion, covering architectures and algorithms in a way that ultimately combines hardware and software with time and information encoding.

Browsing relevant literature for the definitions proposed for dependability, the most recognized proclaim “the ability of a system to avoid service failures that are more frequent or more severe than is acceptable” [4], or “the property of a computer system such that reliance can justifiably be placed on the service it delivers” [66]. Even after a brief analysis, one can sense the inaccuracy of terms like “acceptable”, or “justifiably”, representing a crisp contradiction with the realm of computing, which require further examination upon their meaning.

In practical terms, a reliable system must function with precision for extended durations without encountering any failure or performance decline (described by qualities like reliability, availability, and performability), and it must swiftly recover from errors through fault tolerance strategies (implemented by self-testing and self-repair mechanisms). The term "acceptable" holds crucial, albeit subjective significance within the context of dependability, establishing the upper thresholds of system tolerable damages while retaining functionality or computational accuracy. Dependable systems play a critical role in scenarios where human interventions are restricted or impractical, operating in challenging or hostile environments. Prime examples include systems overseeing critical operations (ranging from medical procedures to banking and industrial processes) and machinery needed for prolonged operations in deep-sea or nuclear settings, as well as space exploration endeavors. However, these specialized applications do not deter the

adoption of a dependability-focused design approach in general-purpose computing. In sectors such as industry, transportation, financial services, and various others, where precise computer functionality is continuously required at all times, any margin for error is intolerable. Furthermore, this approach remains relevant in nanoelectronics, where environmental aggressiveness and the absence of human intervention at such a scale require a robust dependability framework.

As stated in [125], achieving robustness in a digital system begins with its design, which includes: 1) understanding essential aspects affecting dependability so that appropriate design requirements can be formulated; 2) exploring suitable methods for achieving such designs into functional computing systems, and 3) building a proof-of-concept system and validating its performance.

However, engineering dependability is known to be a multifaceted target, uniting different attributes (reliability, availability, maintainability, security, safety, confidentiality), with threats (faults, error, failures) and means (fault prevention, fault tolerance, fault removal, fault forecasting). For these reasons, We will choose our focus on reliability, which represents an essential attribute for dependability that can be objectively estimated.

Throughout their lifespan, electronic devices undergo continuous exposure to various influences, primarily manifesting during transient periods, which result in a range of errors that are collectively termed *transient faults*, *soft errors*, or *single-event upsets* (SEUs). The frequency at which electronic devices encounter these issues, referred to as the *soft error rate* (SER), is quantified in failures per unit time. Due to their reliance on transient phenomena to alter states and logical values, digital electronics exhibit heightened susceptibility to soft errors.

As far as we know, such events affect all digital devices and mainly originate from electromagnetic noise and/or external radiations rather than design or manufacturing flaws [102]. Electronic devices have long been intrinsically reliable due to the protective role of the technology itself. However, as the current technological involvement continues into the nanoelectronics era, the protective role of the technology is wearing out fast, exacerbating the damaging effects of transient events and proving that reliability (and dependability at large) are, in fact, *sine qua non* features. At the same time, there are raising questions about the future of the silicon and post-silicon technologies and the means we should employ to usher these seamlessly. To address these questions and provide answers that would inspire and support the challenges of chip design and test expected

to arise in the next decade, a new focus center research program was created within the academia in December 1998. Called the Gigascale Silicon Research Center (GSRC) [61], its main goal was to approach the long-term research targets required for the continued progress of the semiconductor industry. Some of their vision includes novel architectural vectors, such as defect-tolerant, biologically-inspired, and quantum computing, shown in figure 2.1.

In synchrony with figure 2.1, our focus is represented by two emerging technology vectors, namely biologically-inspired and quantum computing, together with any aspects required for these to be successfully developed, implemented, and deployed. Browsing relevant literature for the definitions proposed for dependability, the most recognized proclaim “the ability of a system to avoid service failures that are more frequent or more severe than is acceptable” [4], or “the property of a computer system such that reliance can justifiably be placed on the service it delivers” [66]. Even after a brief analysis, one can sense the inaccuracy of terms like “acceptable”, or “justifiably”, representing a crisp contradiction with the realm of computing, which require further examination upon their meaning.

In practical terms, a reliable system must function with precision for extended durations without encountering any failure or performance decline (described by qualities like reliability, availability, and performability), and it must swiftly recover from errors through fault tolerance strategies (implemented by self-testing and self-repair mechanisms). The term "acceptable" holds crucial, albeit subjective significance within the context of dependability, establishing the upper thresholds of system tolerable damages while retaining functionality or computational accuracy. Dependable systems play a critical role in scenarios where human interventions are restricted or impractical, operating in challenging or hostile environments. Prime examples include systems overseeing critical operations (ranging from medical procedures to banking and industrial processes) and machinery needed for prolonged operations in deep-sea or nuclear settings, as well as space exploration endeavors. However, these specialized applications do not deter the adoption of a dependability-focused design approach in general-purpose computing. In sectors such as industry, transportation, financial services, and various others, where precise computer functionality is continuously required at all times, any margin for error is intolerable. Furthermore, this approach remains relevant in nanoelectronics, where environmental aggressiveness and the absence of human intervention at such a scale require

a robust dependability framework.

As stated in [125], achieving robustness in a digital system begins with its design, which includes: 1) understanding essential aspects affecting dependability so that appropriate design requirements can be formulated; 2) exploring suitable methods for achieving such designs into functional computing systems, and 3) building a proof-of-concept system and validating its performance.

However, engineering dependability is known to be a multifaceted target, uniting different attributes (reliability, availability, maintainability, security, safety, confidentiality), with threats (faults, error, failures) and means (fault prevention, fault tolerance, fault removal, fault forecasting). For these reasons, We will choose our focus on reliability, which represents an essential attribute for dependability that can be objectively estimated.

Throughout their lifespan, electronic devices undergo continuous exposure to various influences, primarily manifesting during transient periods, which result in a range of errors that are collectively termed *transient faults*, *soft errors*, or *single-event upsets* (SEUs). The frequency at which electronic devices encounter these issues, referred to as the *soft error rate* (SER), is quantified in failures per unit time. Due to their reliance on transient phenomena to alter states and logical values, digital electronics exhibit heightened susceptibility to soft errors [18, 90, 19].

As far as we know, such events affect all digital devices and mainly originate from electromagnetic noise and/or external radiations rather than design or manufacturing flaws [102]. Electronic devices have long been intrinsically reliable due to the protective role of the technology itself. However, as the current technological involvement continues into the nanoelectronics era, the protective role of the technology is wearing out fast, exacerbating the damaging effects of transient events and proving that reliability (and dependability at large) are, in fact, *sine qua non* features. At the same time, there are raising questions about the future of the silicon and post-silicon technologies and the means we should employ to usher these seamlessly. To address these questions and provide answers that would inspire and support the challenges of chip design and test expected to arise in the next decade, a new focus center research program was created within the academia in December 1998. Called the Gigascale Silicon Research Center (GSRC) [61], its main goal was to approach the long-term research targets required for the continued progress of the semiconductor industry. Some of their vision includes novel architectural vectors, such as defect-tolerant, biologically-inspired, and quantum computing, shown in

figure 2.1. In synchrony with the aforementioned technology vectors, our focus is represented by two emerging technology vectors, namely biologically-inspired and quantum computing, together with any aspects required for these to be successfully developed, implemented, and deployed. Browsing relevant literature for the definitions proposed for dependability, the most recognized proclaim “the ability of a system to avoid service failures that are more frequent or more severe than is acceptable” [4], or “the property of a computer system such that reliance can justifiably be placed on the service it delivers” [66]. Even after a brief analysis, one can sense the inaccuracy of terms like “acceptable”, or “justifiably”, representing a crisp contradiction with the realm of computing, which require further examination upon their meaning.

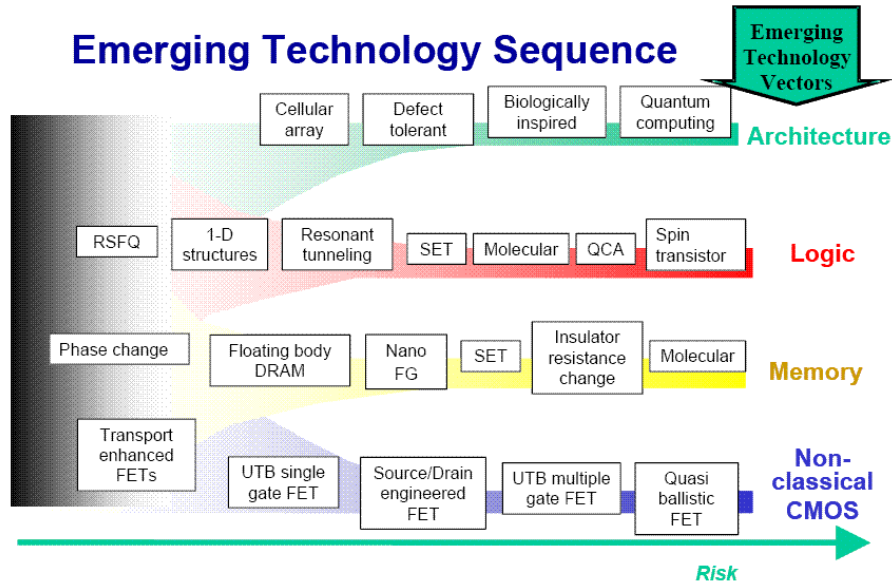


Figure 2.1: Emerging technology vectors [54].

In practical terms, a reliable system must function with precision for extended time intervals without encountering any failure or performance decline (described by qualities like reliability, availability, and performability), and it must swiftly recover from errors through fault tolerance strategies (implemented by self-testing and self-repair mechanisms). The term "acceptable" holds crucial, albeit subjective significance within the context of dependability, establishing the upper thresholds of system tolerable damages while retaining functionality or computational accuracy. Dependable systems play a critical role in scenarios where human interventions are restricted or impractical, operating in challenging or hostile environments. Prime examples include systems overseeing critical operations (ranging from medical procedures to banking and industrial processes) and machinery needed for prolonged operations in deep-sea or nuclear settings, as well

as space exploration endeavors. However, these specialized applications do not deter the adoption of a dependability-focused design approach in general-purpose computing. In sectors such as industry, transportation, financial services, and various others, where precise computer functionality is continuously required at all times, any margin for error is intolerable. Furthermore, this approach remains relevant in nanoelectronics, where environmental aggressiveness and the absence of human intervention at such a scale require a robust dependability framework.

As stated in [125], achieving robustness in a digital system begins with its design, which includes: 1) understanding essential aspects affecting dependability so that appropriate design requirements can be formulated; 2) exploring suitable methods for achieving such designs into functional computing systems, and 3) building a proof-of-concept system and validating its performance.

However, engineering dependability is known to be a multifaceted target, uniting different attributes (reliability, availability, maintainability, security, safety, confidentiality), with threats (faults, error, failures) and means (fault prevention, fault tolerance, fault removal, fault forecasting). For these reasons, We will choose our focus on reliability, which represents an essential attribute for dependability that can be objectively estimated.

Throughout their lifespan, electronic devices undergo continuous exposure to various influences, primarily manifesting during transient periods, which result in a range of errors that are collectively termed *transient faults*, *soft errors*, or *single-event upsets* (SEUs). The frequency at which electronic devices encounter these issues, referred to as the *soft error rate* (SER), is quantified in failures per unit time. Due to their reliance on transient phenomena to alter states and logical values, digital electronics exhibit heightened susceptibility to soft errors.

As far as we know, such events affect all digital devices and mainly originate from electromagnetic noise and/or external radiations rather than design or manufacturing flaws [102]. Electronic devices have long been intrinsically reliable due to the protective role of the technology itself. However, as the current technological involvement continues into the nanoelectronics era, the protective role of the technology is wearing out fast, exacerbating the damaging effects of transient events and proving that reliability (and dependability at large) are, in fact, *sine qua non* features. At the same time, there are raising questions about the future of the silicon and post-silicon technologies and the means we should employ to usher these seamlessly. To address these questions and provide answers

that would inspire and support the challenges of chip design and test expected to arise in the next decade, a new focus center research program was created within the academia in December 1998, led by UC Berkeley and including Carnegie Mellon, Princeton, Stanford, and other universities. Called the Gigascale Silicon Research Center (GSRC) [61], its main goal was to approach the long-term research targets required for the continued progress of the semiconductor industry. Some of their vision includes novel architectural vectors, such as defect-tolerant, biologically-inspired, and quantum computing, shown in figure 2.1.

In synchrony with figure 2.1, our focus is represented by two emerging technology vectors, namely biologically-inspired and quantum computing, together with any aspects required for these to be successfully developed, implemented, and deployed [90]. We will therefore provide a deep dive into design peculiarities of these two technologies of the future computing.

2.1 Challenging Reliability: Hurdles and Mitigation

Any arising faults affecting computer systems can be classified based on broad criteria, but their nature (whether hardware or software) and time behavior (permanent, transient, or intermittent) are crucial. Regarding their time behavior, faults can be categorized as follows:

- **Permanent faults:** These faults continuously affect the normal operation of the device over an indefinite period, essentially persisting permanently. They are also referred to as solid or hard fails and their coverage and modeling are broadly presented in the scientific literature [53, 103, 65].
- **Non-permanent faults:** These faults occur randomly and impact the system's functional behavior for indefinite, yet finite, periods. Due to the variable frequency and duration of their occurrence, they are challenging to model. Non-permanent faults can be further divided into transient and intermittent faults.

Intermittent faults arise from various factors beyond environmental influences, such as component parameter variations, timing issues, and loose connections. These faults typically affect the system for very short periods, complicating the debugging process and requiring accelerated testing under stress conditions in order to render them permanent.

SEUs or transient faults prove much more difficult to tackle, since they originate as effects of environmental conditions that often lie beyond human intervention, such as cosmic rays. Addressing these is essential for improving the reliability of any electronic device, especially when considering the combination of issues outlined in [101]:

- Problems of the small: due to technological shrinkage of the minimum feature size, transistors are becoming more sensitive to outside noise, and therefore more susceptible to SEUs.
- Problems of the large: since smaller transistors can be crammed onto the same die in larger numbers, not only we are witnessing a corresponding increase in SEUs, but also experiencing the challenges of the design complexity.
- Problems of the diverse: as technologies shift and change, bringing new design styles (system-on-a-chip, network-on-a-chip, hybrid analog-digital) and even new devices (quantum computing, memristors), unforeseen reliability aspects will have to be accounted for. In order to minimize their effects, matching the newly developed devices with their corresponding new system architectures promises to be quite beneficial.

Problems of the small represents a class of issues related to technology, which in turn links to production and manufacturing processes. Any failure at this stage is a composite depending on severity (design-dependant), occurrence (production-dependant) and detection (quality dependant), which falls under the methods developed for Failure Mode and Effects Analysis (FMEA) [34]. If design issues seem to be non-dominant, classical CMOS design being well understood, achieving severe quality control and low failure occurrence are characteristic to military grade processes and therefore reserved for special applications. It is perhaps not coincidental that reference values are given in a military document [93].

Design complexity, however, seems to play a different, more important role concerning problems of the large, where the vastness of the transistor numbers requires specific strategies for error mitigation, especially when emerging technologies are concerned. The industry's focus must now shift towards facing the problems of the large, as they require to be efficiently solved in order to sustain further growth.

Since architectural vectors for new technologies are clearly identified in figure 2.1, it is important to also investigate their relationship with reliability. Estimating reliability is

far from trivial, a number of models being available, all based on parameter λ , which is the failure rate. The reliability of a system is a composite based on the reliability of all its subsystems, which are typically heterogeneous. The structure of the system considered notwithstanding, any attempts to provide quantitative figures for the reliability function always equates to assess the failure rate λ , for which values are given in [93]. As one may remark, formal methodologies for precise assessments are not publicly available, being considered to potentially provide a strategic edge and therefore remain protected by military mandates. Until 2018, the only update for this was last performed on 1995, continued interest in the topic leading to an entire family of similar documents [82], all sharing the same issue concerning quantitative assessments for the failure rate λ . Variations of the failure rate are described based on the Arrhenius model (also referred to as the Arrhenius equation), which sees it as a dependence to the operating temperature and activation energy, according to equation (2.1).

$$\lambda = Ke^{-E/K_BT} \quad (2.1)$$

In equation (2.1), K_B is the Boltzmann's constant, T is the absolute temperature, and E is the activation energy for the process, with K being a constant [77, 83]. However, the described dependence does not take into account that the activation energy itself is also dependent on failure mechanisms, while its values have been shown to vary widely [82]. All of the above considered, this model triggered intense criticism due to discrepancies between predicted and measured values [56, 123, 27]. Further missing from the handbook are real and different loading conditions of the devices, such as electrical parameters, heat, and mechanical stress, which leads to a false assumption that all electronic components fail at the same rate, under all of the loading conditions. Since the methods contained by the handbook have been translated into computer software and employed commercially, these software tools are also bound to the same limitations.

A number of experiments for empirically assessing failure rates have been performed. In one such experiment, a board consisting of 149 components (with 18 categories of electronic components) has been subjected to field use in a telecom application to collect actual failure rate values by the International Electronics Reliability Institute (IERI) at Loughborough University, United Kingdom. Discrepancies recorded as opposed to predicted values are shown in figure 2.2a. Another experiment performed by Northrop Grumman collected data from the field between 1993 and 1999 during the Modular Air-

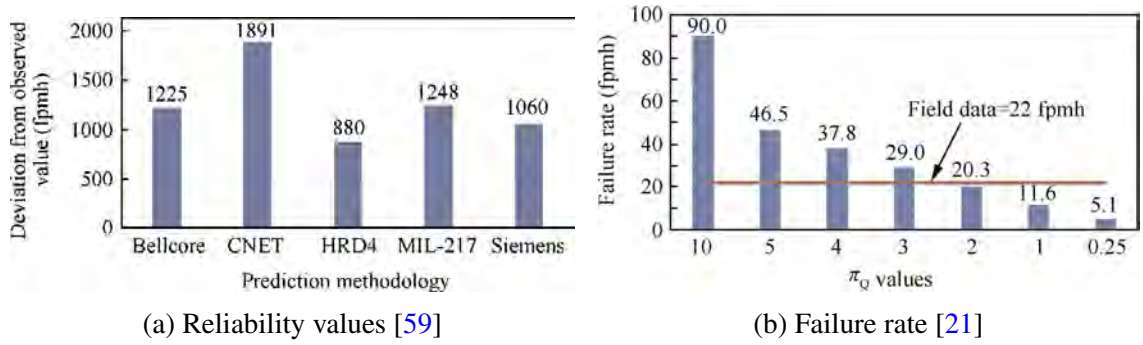


Figure 2.2: Estimated vs field-recorded data.

borne Radar (MODAR) program, military designation AN/APN-241. Failure data (expressed in failures per million hours) accumulated in 300000 operating hours are presented opposed to estimations produced with multiple quality factors in figure 2.2b.

All of the above considered, providing a reliability description for any system requires summarizing the conclusions that may be drawn:

- providing accurate estimations for failure rate values is a difficult task that is strongly dependent on the quality of information used. Unfortunately, MIL-HDBC-217 fails to provide reliable information about electronic devices and the variety of the existing standards and procedures, as well as commercially available software tools, are all part of its progeny [82]. As a consequence, achieving a precise reliability characterisation for any system remains out of reach;
- despite differences between predicted and real values for the failure rate, MIL-HDBC-217 should serve as a starting point for qualitative analyses in order to decide on strategies and subsystem parts that need to be improved regarding reliability. Even if values provided are not accurate, when used consistently they can provide the basis for qualitative assessments for reliability-enhancing measures;
- so far, only internal causes for failure have been discussed. However, external environmental factors can indeed play a significant role by causing soft failures (or soft errors) without permanently damaging the device or system, and therefore should be modeled [85, 90].

New challenges arise with the current downscaling of technology towards nanoelectronics, with more devices being integrated per chip, necessitating a larger amount of redundant/control logic or other measures, with smaller dimensions rendering electronic devices much more sensitive to the influence of stray energetic particles from cosmic rays.

These aspects are essential to the development of the two emerging technologies considered, though for slightly different reasons. From a physical point of view, electronic devices become comparable in size to the very atoms, bio-inspired computing aiming to managing them efficiently while in quantum computing devices become atoms and particles. From a functional perspective, precise computation is not possible in the quantum environment without fault tolerance, while bio-inspired computing leverages fault tolerance to enhance system dependability. Exploring these emerging technologies represents an opportunity to understand the computational processes of the future.

2.2 Biologically-Inspired Computing

If computational processes are omnipresent in the physical universe and studied by natural sciences, some processes are particularly enticing for computing, such as self-replication, how the brain works, different types of evolution (Darwinian, group organization and behavior), self-preservation (the immune system), and developmental processes (morphogenesis). All these processes have been occurring in the living world since the beginning of life itself, being tested and perfected in such a variety of ways that simply are out of reach for human engineering. When the testing period matches the history of life itself, nature appears, without a doubt, the ultimate engineer and architect. Investigating biological organisms reveals the highest level of complexity in its structures, due to the massively parallel cooperation between huge numbers of relatively simple elements, the cells. Their reliability is proven by a life span on the order of hundreds (for the *regnum animale*) or even thousands (for the *regnum vegetabile*) of years.

Would it be possible for such treats to be achieved in digital engineering? Answering this question establishes nature as an obvious source of inspiration, which is probably made famous by John von Neumann, aiming to design a machine that could think (similar to the human brain) and replicate itself [73, 74]. The technology at the time was, however, not ready to take on such ambitious tasks, von Neumann's dream of implementing self-reproducing automata only coming to fruition during the 1990s, when programmable logic became flexible and large enough with the new paradigm of reconfigurable computing. Building bio-inspired digital systems is a twofold challenge: understanding nature is essentially an analytical process [106], while the design and assembly of equivalent structures is a synthetic process [92]. In order for the bioinspired digital devices to become real, at least three major issues require proper addressing [86, 87]:

- would adapting mechanisms from nature in digital engineering also transfer the results?
- were exactly would the two of them, Nature and science/engineering, meet?
- finally, is the process of exporting biological features in computer engineering technically possible?

Understanding the ways nature uses can be structured into a taxonomy of three categories of processes [106]:

1. Phylogenetic processes form the first level of organization of living matter. Their role is to ensure the temporal evolution of the genetic data of all individuals, the result of which represents the evolution of all life into a variety of species. The phylogenetic processes are based on simple mechanisms, such as recombination and mutation, that are nondeterministic. The error rate here ensures nature's diversity.
2. Ontogenetic processes form the second level of organization of living matter. Their role is to implement the temporal evolution of a single individual, determining its development from the stage of a single cell (called the egg, or zygote), through successive cellular divisions and specializations, to the adult stage. Ontogenetic processes are deterministic, any error at this level resulting in malformed or non-viable individuals;
3. Epigenetic processes form the third (and top) level of organization of living matter. Their role is to determine and integrate interactions with the surrounding environment, to provide superior reactions and, ultimately, intelligence. These processes combine a variety of stochastic and deterministic operations and served as inspiration for John von Neumann's universal computer.

This taxonomy makes up the schematic for a model called POE (standing for Phylogeny, Ontogeny, and Epigenesis) that represents the space for new, biologically-inspired systems, also opening new perspectives for achieving superior dependability in computer engineering, both hardware and software [90]:

1. The first in the hierarchy, the phylogenetic processes, correspond to evolutionary computing processes, which include artificial intelligence and soft computing. They can provide highly optimized, approximate solutions by approaching

optimization problems with a combination of population-based trial and error and metaheuristic or stochastic strategies. Using biological terms, a population of solution candidates is subjected to repeated selection, mutation and recombination. As a result, the population is expected to gradually evolve and show an increase in the fitness values, the fitness function corresponding to the desired outcome of the algorithm. Because the population contains *candidates* for an acceptable solution, among which some are better/worse than others, the evolutionary algorithms may be regarded as self-error-injecting, achieving an intrinsic resilience to faults and errors. As a consequence, the phylogenetic processes provide the genetics of living beings, and therefore are essentially informational, corresponding to the software side in computing;

2. The intermediate level in the POE hierarchy, the ontogenetic processes, are concerned with the developmental processes that take part in the living cell, namely cellular division and cellular differentiation. In the living matter, these processes exhibit space uniformity and massive time parallelism, providing the fabric for structural dimension. To achieve such features, field programmable logic (FPGAs) is employed to ensure hardware functionality through software, similar to genetic information governing over living matter. In digital computing, such devices that are able to change their behavior dynamically, based on the particulars of the driving genetic information are known as evolvable or adaptive hardware. They provide the structural blocks for evolvable systems, which will achieve specific computational targets while also maximizing dependability;
3. The top level of the POE model, the epigenetic processes, are the salient representatives of intelligence in the living matter. Due to their capacity of generalizing based on past experiences, these processes provide what we call *learning* (both innate and acquired), therefore ensuring a new level of fault-tolerance. Together with the ontogenetic processes, epigenesis allows for complex strategies of tolerating faults, which ultimately translates into evolution.

2.2.1 The Embryonics Project

The fault-tolerance exhibited by the living matter can be described by two connected stages, situational awareness (where the system perceives and recognizes an external ag-

gression, and predicts its future development) and strategic response (where the system reacts to contain and recover from the aggression and build provisions for future events). Translating these stages in modern computing, a fault-tolerant system has to deal with a succession of steps [1]:

1. Fault detection: the system detects the presence of a fault. This is equivalent with an initial situational awareness that happen at the informational level and can largely be attributed at the ontogenetic level;
2. Fault diagnosis or localization: the system identifies the precise nature and location of a fault. This step raises the situational awareness by extending to the physical level, therefore relating also to the ontogenetic level;
3. Fault limitation or containment: the impact of the fault on the operation of the system is limited to the smallest possible area, so that no further damage results. This is the initial response of the system, which takes place at the physical level, being attributed to the ontogenetic level;
4. Fault repair: the functionality of the system is recovered. The final stage of the system's response re-establishes normal operation while minimizing performance loss and building provisions for similar events in the future. While the reconfiguration required is typically a process that takes place at the ontogenetic level, understanding the measures that need to be taken to perform the reconfiguration is a process that belongs to the epigenetic level. Further strategies for provisioning against future faults are also part of the epigenetic level.

It is clear that all three kinds of processes occur continuously within the living matter, with information processing and storing being carried by molecules and cells through refined mechanisms from chemistry. The living matter certainly makes good use of the tools available at this scale to achieve best adaptability and resilience, processes that belong to the phylogenetic and ontogenetic levels. When it comes to epigenetic processes, however, it seems information complexity reaches to even smaller scales of fundamental particles, involving quantum physics, which will be discussed in the next section.

While developing the theory of self-replicating machines, John von Neumann found a way to map the famous equation in biology "genotype + ribotype = phenotype" so that the machine (described by the phenotype) consists of its complete description (encoded

in the genotype), which is interpreted by a ribosome (encoded by the ribotype). To further explore the potential of biologically-inspired mechanisms by adapting them from nature and incorporating them into novel digital systems, the Embryonics (a contraction for embryonic electronics) project proposes a multiple-level hierarchical model that uses homogeneous, identical elements to achieve extreme reliability, described in figure 2.3. The hardware implementation directly addresses ontogenetic processes, while phylogenetic and epigenetic processes can be added to tackle higher grounds in bio-inspiration [116, 14, 19].

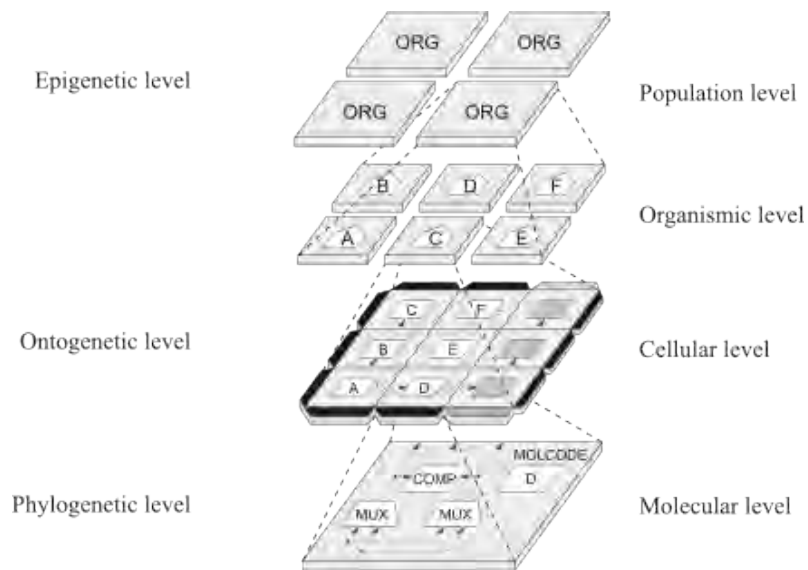
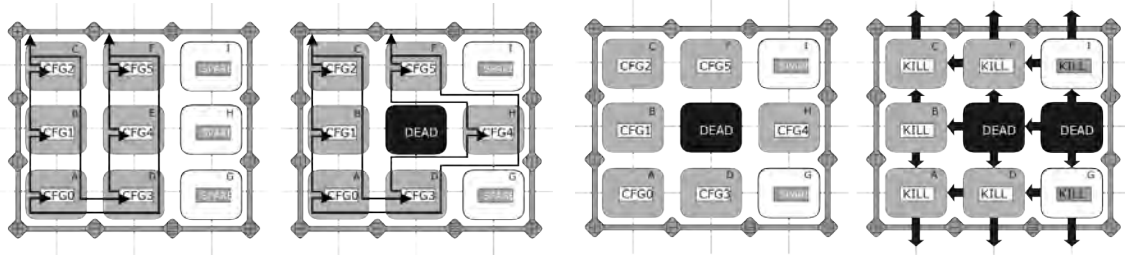


Figure 2.3: Hierarchical organization in Embryonics [90].

The Embryonics perspective proposes a top level representing a network of parallel computing systems (the population), made of a multitude of multicellular organisms, each consisting of a finite number of cells (the cellular level), which in turn are made of a finite number of electronic molecules. If nature employs complex chemical bonds to assemble molecules and then build up the hierarchy, the world of digital electronics is constrained to existing resources (hardware) that have to be managed through information (software). Each cell (starting with the original cell, the zygote) is universal, since it may generate daughter cell(s) through a process called cellular division (growth), with both the parent and the daughter cell(s) sharing the same structure and genetic information (the genome). However, the cells are able to perform different functions due to cellular differentiation, a process through which only a specific part of the genome is executed by each cell. Therefore, the Embryonics perspective correlates a biological organism with a parallel computing system, a biological cell with a processor, while the molecule in biology may be seen as the smallest, programmable element in digital electronics.



(a) Molecular level: faulty molecule E is replaced by spare molecule H, which becomes active

(b) Self-repair failure at molecular level: the entire cell dies, triggering self-repair at the cellular level

Figure 2.4: Two-level hierarchical self-repair [69].

2.2.2 Reconfiguration Strategies for Hierarchical Fault-Tolerance

The hierarchical organization in Embryonics, as shown in figure 2.3, is the result of successive evolution to provide adequate resources for hardware-implemented ontogenetic systems [69]. As a reliability-centric exploration project, Embryonics promotes a hierarchical self-repair approach. To achieve very low failure frequencies, two layers of self-repair are provided: at the molecular level (where programmable logic is prone to soft failures and SEUs) and at the cellular level (where managing soft errors is much more challenging and expensive).

Let us consider a simple 3x3 cell (with one column containing spare molecules), which may experience a variety of failures at the molecular level. Since a column is dedicated to spare resources, any active (or spare) molecule can be repaired through re-configuration in case of a failure, as described in figure 2.4a.

However, such a repair at the molecular level can only be successful if spare resources are available; if not, all molecules from within the cell will "die", thus triggering the second stage of the self-repair mechanism at the superior level. This situation is described in figure 2.4b, where the same cell experiences a successive failure in the same line (molecule B or H) and triggers the self-repair at the upper, cellular level. Similar to repair at the molecular level, which relies on the existence of spare molecules, the repair at the cellular level relies on the existence of spare cells: when a cell dies, the entire column containing the faulty cell (cell C in this example) is deactivated, and its role is taken up by the column to its right, which is itself shifted to the right, and so on until a spare column is reached (figure 2.5a). This strategy was initially considered appropriate due to its low hardware overhead and also the low probability of its employment, despite bringing a steep penalty in resources lost. An evolution of this strategy was implemented

based on the following considerations [90]:

- the repair at cellular level will most likely be required due to the appearance of soft failures, the particle flux responsible being non-uniform and non-isotropic, thus affecting smaller, localized areas, rather than entire columns, therefore suggesting a reconfiguration around the faulty cell only, which would also be more resource-efficient;
- reliable data transfer must also be delivered when repairing memory areas, therefore increasing the probability of subsequent soft failures, as moving data is more susceptible than static data. The additional routing resources required would be justified by the increased flexibility;
- following the observation of the symmetry present in nature, self-repair at the cellular level should be consistent with the self-repair at the molecular level, which would potentially simplify a thorough reliability analysis and multiple-level information coding, such as concatenated coding.

The updated repair strategy is described in figure 2.5b and was also integrated with the memory structures added in Embryonics. Of course, from the implementation perspective, a thorough analysis of various operating modes for the molecules is required to understand the challenges imposed by different operating modes for the molecules. There are two main ways in which a molecule can be configured. When in logic mode, the molecule's configuration register (CREG) stores the binary string that determines its logic function and interconnections, therefore providing grounds for universal computation. When in memory mode, the same CREG is used as a shift register in order to store data; a short memory mode offers 8-bit-wide storage data and configurable interconnections, while a long memory mode offers 16-bit-wide storage data at the expense of fixed interconnections. Variable capacity memory structures can be assembled by chaining together the CREGs from as many molecules as desired. While the operating mode directly influences the molecular reliability function, at the cellular level the reliability function takes advantage of the uniform hierarchical self-repair strategy.

Generally, a cell in Embryonics would include molecules operating in all featured modes, which may rise the question of possibly different reliability profiles, since combinational logic is known to behave differently than sequential logic or memory [55]. In order to derive the reliability function for any cell, one must determine the partitions

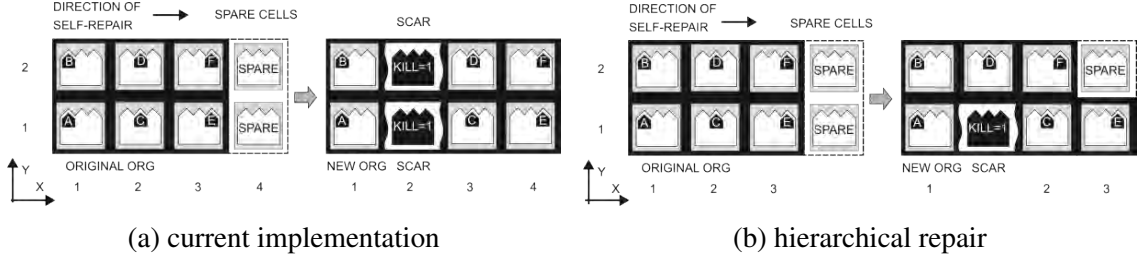


Figure 2.5: Self-repair at the cellular level.

from within the cell, namely the molecular structures that assemble the logic and memory regions. Regardless their operating mode, all structures in Embryonics are rectangular, which allows a systematic approach to deriving the reliability function for a certain cell by taking into account the reliability functions of the cell's specific logic and memory structures.

Let us consider the general case of a logic structure, consisting of M lines and N columns, out of which S are spares. This is actually composed of a number of identical modules of M lines and d columns, out of which n_{Mod} are active and s_{Mod} are spares. These parameters describe a stochastic process and relate to each other as described in equation (2.2).

Therefore the reliability of a module R_{Mod} is given by equation (2.3), where R_{ModRow} is the reliability function of a row that is capable of withstanding at most s_{Mod} errors.

Expanding on equation (2.3), the probability of recovering from i failures is defined as the conditional probability of having $d-i$ molecules operating normally, while at least i spare molecules are ready to become active. The reliability of the logic structure includes all modules, as shown in equation (2.4). Based on the example considered, the reliability for a memory structure can be derived in a similar way, thus leading to the reliability of the cell as presented in equation (2.5). Reliability estimations for a row and for an entire cell are given in figure 2.6 [90].

$$\begin{aligned}
 d &= n_{Mod} + s_{Mod}; \\
 N_{Mod} &= gcd(N, N - S); \\
 N &= N_{Mod} \cdot d;
 \end{aligned}
 \tag{2.2}$$

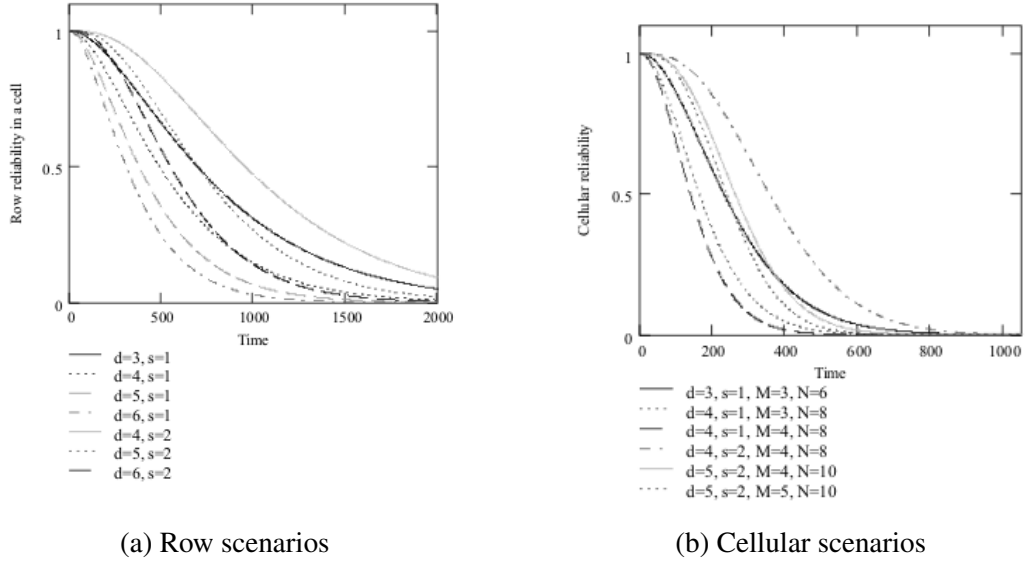


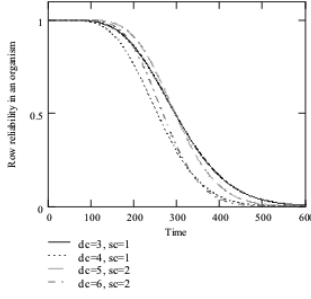
Figure 2.6: Reliability estimations for logic structures.

$$\begin{aligned}
 R_{Mod}(t) &= (R_{ModRow}(t))^M; \\
 R_{ModRow}(t) &= Prob\{no\ fails\}(t) + \sum_{i=0}^{s_{Mod}} Prob\{i\ fails\}(t); \\
 Prob\{no\ fails\}(t) &= R_{Mol}(t)^d; \\
 Prob\{i\ fails\}(t) &= \binom{d}{i} \cdot (1 - R_{Mol}(t))^i \cdot (R_{Mol}(t))^{d-i} \\
 &\quad \cdot \sum_{j=i}^{s_{Mod}} \binom{s_{Mod}}{j} \cdot (R_{Mol}(t))^j \cdot (1 - R_{Mol}(t))^{s_{Mod}-j};
 \end{aligned} \tag{2.3}$$

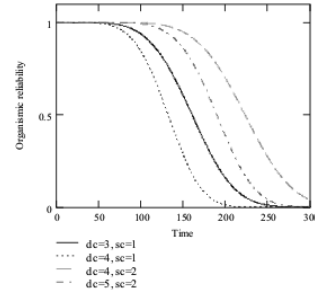
$$R_{LogicStructure}(t) = \prod_{i=1}^{N_{Mod}} R_{Mod_i}(t); \tag{2.4}$$

$$R_{Cell}(t) = \prod_i R_{LogicStructure_i}(t) \cdot \prod_j R_{MemoryStructure_j}(t); \tag{2.5}$$

At the cellular level, the reliability figures depend on the fault-tolerance strategy employed. Due to implementation complexity, in case of a faulty cell the choice was to deactivate the entire column and replace it by a spare. Replacing only the faulty cell would require greater hardware overhead, with potentially reaching superior reliability values at the same time. Let us consider an organism composed of MC rows and DC columns of cells, out of which NC columns are active and SC are spares. The reliability



(a) Column deactivation strategy



(b) Hierarchical strategy

Figure 2.7: Reliability estimations for an organism.

in case of the column deactivation strategy results as described in equation (2.6), whereas for a cell-replacement strategy the reliability function is described in equation (2.7).

$$R_{CellCol}(t) = R_{Cell}^{MC}(t);$$

$$R_{Org}(t) = \sum_{i=0}^{SC} \binom{DC}{i} \cdot R_{CellCol}^{DC-i}(t) \cdot (1 - R_{Mol}(t))^i; \quad (2.6)$$

$$R_{CellRow}(t) = \sum_{i=0}^{SC} \binom{DC}{i} \cdot R_{Cell}^{DC-i}(t) \cdot (1 - R_{Cell}(t))^i;$$

$$R_{Org}(t) = R_{CellRow}^{MC}(t); \quad (2.7)$$

$$R_{Org}(t) = \left(\sum_{i=0}^{SC} \binom{DC}{i} \cdot R_{Cell}^{DC-i}(t) \cdot (1 - R_{Cell}(t))^i \right)^{MC};$$

Comparing the resulting reliability functions, it is to be expected that the more limited strategy based on column deactivation (equation equation (2.6)) generates values that are inferior to the two-stage uniform hierarchical strategy (equation equation (2.7)), shown by plots in figure 2.7a and figure 2.7b, respectively [90]. However, the means to reaching a target reliability value at the organism level are more difficult to observe, since structures in Embryonics are by design heterogeneous and large.

2.2.3 Partition for Dependability

Featuring a four-level hierarchy combined with a two-level hierarchy of self-repairing strategies and two main operating modes for its basic component (the molecule), the Embryonics project actually introduced a new type of FPGA fabric that enables superior

robustness in digital circuits. However, such complexity is also a source of difficulty when targeting specific reliability values, since evaluating reliability strongly depends on the specific organism partitioning [88]. A molecule is able to operate in any of the following modes:

- logic mode: the functionality of the molecule is determined by the entire content of its CREG, offering universal logic (through configurable multiplexers) and universal storage (through a D-type flip-flop);
- memory mode: the functionality of the molecule is determined by the content of its flip-flop, when a smaller part of the CREG is being used as 8-bit user storage with flexible data switching (short memory mode), or when a larger part of CREG is used as 16-bit storage with fixed data switching. When in memory mode, molecules can be chained together to form memory structures or *macro-molecules*.

When in logic mode, any fault is repaired by using the two-level hierarchy of self-repair, which is triggered by self-testing. However, the same self-testing process cannot be employed in case of memory molecules, which require the implementation of error coding and correcting mechanisms. In Embryonics, fault-tolerance can be achieved for memory molecules in two ways:

- at the molecular level: the faulty molecule becomes fault-tolerant by dedicating a significant internal area for redundant coding, an implementation example being presented in figure 2.8a. Each molecule operates in long memory mode, with the CREG being split in two halves. Each half is then employed in a similar way to the complementary DNA strands: one half stores the complementary data of the other, with a continuous comparison between them being made at any given moment. Due to the storage data being continuously shifted, such a comparison introduces a minimal hardware overhead by only involving a few logic gates. Such a strategy would not allow fault repair at the molecular level, the fault detection typically being used to trigger a fault repair at the higher, cellular level. The KILL mechanism, shown in figure 2.4b, is triggered whenever a non-repairable fault occurs. In order to alleviate the steep penalty induced by disabling an entire column of cells, KILL is matched by its opposite, UNKILL mechanism, which reverses the cell to its original status based on the fact that the majority of the memory errors are, in fact, transient [91].

- at the macro-molecular level: macro-molecules are used distinctly to store the redundant code (genome data and control memory); additionally, some dedicated logic structures are also involved, as described in figure 2.8b. A fault-tolerant macro-molecule is made of several structures, involving the user storage and redundant memory arrays, as well as the logic molecules implementing the Hamming error detection and correction. A minimal fault-tolerant macro-molecule includes a 4-column Genome Memory (serving as user storage), a 3-column Control Memory (storing the control data required by the Hamming code), and the Error Correcting Logic (consisting of molecules operating in logic mode that implement the Hamming checking and correcting logic) [88].

Regardless of the fault-tolerance strategy chosen to implement memory structures, the design of two-dimensional arrays in Embryonics presents the challenge of balancing user requirements for logic and memory with the need for redundant resources to enable self-repair. Meeting certain target values for dependability requires careful partitioning of the array into cells and molecules in order to accommodate the application's logical needs with minimal overhead induced by spare molecules and cells. The key issue is determining the optimal cellular configuration, including both active and spare columns. Allocating more spare columns per cell enhances cellular reliability but introduces trade-offs, as it either reduces the logic available to the user (which may be critical) or limits the total number of cells in the array. Since the overall array dimensions are fixed, this limitation on cell count can impact the array's overall reliability. An engineered example discussed in [88] presents an array of 30 columns of molecules, with the following partitioning scenarios:

- 6 columns of cells (in pairs of 2 active followed by 1 spare), each cell composed of 5 columns of molecules (3 active, 2 spare). This case accounts for 12 active columns of molecules, representing 4 active columns of cells, each with 3 active columns of molecules;
- 3 columns of cells (2 active followed by 1 spare), each cell composed of 10 columns of molecules (pair of 4 active, 1 spare). This case accounts for 16 active columns of molecules, representing 2 active columns of cells, each with 8 active columns of molecules;
- 10 columns of cells (pairs of 1 active followed by 1 spare), each cell composed of 3

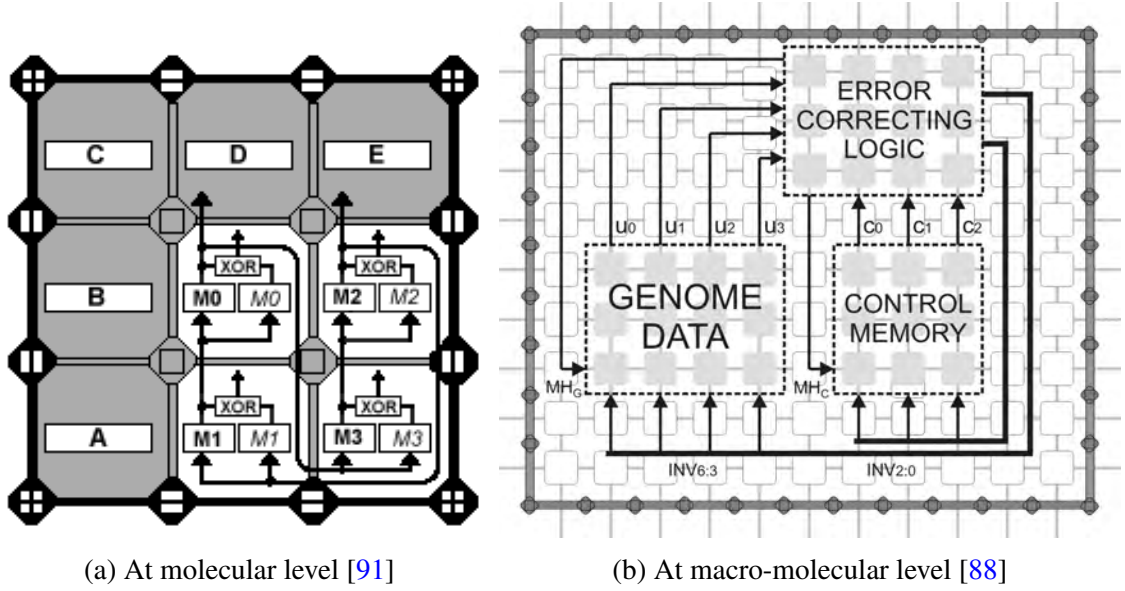


Figure 2.8: Fault-tolerance strategies in memory structures.

columns of molecules (2 active, 1 spare). This case accounts for 10 active columns of molecules, representing 5 active columns of cells, each with 2 active columns of molecules.

Considering the current granularity in Embryonics, a molecule may be viewed as equivalent to a few logic gates or registers, which quickly rises the array dimensions for any real world application. Therefore, as array dimensions increase, the number of potential partitioning options grows significantly, thus rendering the optimization process more complex, and therefore difficult to determine the optimal configuration. Similar challenges are successfully approached through the knapsack problem, which is NP-complete, which aims to find the most valuable set of items that fit in a knapsack of a fixed volume, given a number of items of different values and volumes [88].

Formally, a knapsack of capacity $c > 0$ and N items has to accommodate the best selection δ , each item i being defined by its value $v_i > 0$ and weight $w_i > 0$. Therefore, the goal becomes to maximize the value $\sum_{i=0}^N \delta_i \cdot v_i$, where $\sum_{i=0}^N \delta_i \cdot w_i \leq c$. For our In this analogy, the array to be partitioned corresponds to the knapsack, while the spare molecules and cells parallel the value and weight components of the classic problem. The array's reliability serves as the fitness measure. While various algorithms and programming approaches can solve such problems, genetic algorithms offer a particularly efficient path to rapid convergence.

Unlike the knapsack problem, which can accept a range of partial solutions, any implementation in Embryonics requires exact partitioning, since all cells must have identical

dimensions and completely fill the array. Given its granularity, any real-world application would easily require arrays of the order 10^6 or more molecules, which constitutes a solid argument for the potential employment of genetic algorithms. Our proposed genetic algorithm for partitioning works as follows. It begins with a random population of potential solutions, each solution representing a different partition configuration (the number of active and spare columns within each cell and the number of active and spare columns of cells within the array). The fitness evaluation verifies the configuration meets user requirements, and calculates the array's reliability. Then, crossover and mutation are applied. The algorithm continues iterating until either the target reliability goal is reached, or improvements in reliability become minimal (falling below a specified threshold).

Section 2.2 presented some of the conceptual and design issues that have to be settled when designing a fault-tolerant memory structure within the Embryonics platform. The reliability analyses presented so far are required for determining optimal partitions that provide best system reliability while also accommodating specific user constraints.

2.3 Quantum Computing

Understanding consciousness remains a scientific challenge: while we do understand some of the processes that take place into the brain, building conscious robots remain a fiction. Recent discoveries related to brain functions indicate certain correlations between the short-term memory performance and conscious awareness, which may point to the fact that our brains use, in fact, quantum processes for cognitive functions [60]. Quantum processes occurring in the brain could explain why the human brain is able to outperform (classical) supercomputers under unforeseen circumstances, decision making, or learning something new.

Despite their computational power, conventional computers do have limitations when time is concerned, the class of NP-complete problems being the prime example. On the other hand, quantum computing uses complex particle interactions from quantum physics to manipulate information, which enables them to deliver performance even for NP-complete problems. However, building quantum computers is difficult, since involves mastering both quantum physics (for the mechanics of computing processes) and information theory (for the reliability of computing processes).

Error detection and correction become therefore vital in quantum computing due to environmental interference, which constantly introduces errors. These techniques are es-

essential for maintaining the integrity of quantum computations by limiting error spread. Without such reliability measures, practical quantum computing devices would be unfeasible [84, 104].

Errors in quantum systems stem from two main sources, the first being malfunctioning quantum gates, which cause processing errors. The second source is related to interactions between quantum states and the environment, leading to storing and transmitting errors. The decoherence effect arises due to the environment continuously "measuring" delicate quantum superposition states, which causes either a decay into a projection of the state vector onto a basis (or eigen state) or a modification of quantum amplitudes. To address these issues, researchers employ intrinsic fault tolerance through technology and error-correcting techniques at the quantum circuit level.

However, it was found the accuracy of quantum computations is at risk if the error probability exceeds a critical threshold [2], in a similar way von Neumann theorized the threshold theorem for classical computation [73].

If the bit is the basic unit of information in classical computing, in quantum computing the corresponding unit is the qubit. Expressed in bra-ket notation [76], the qubit is a normalized vector in a Hilbert space $\mathcal{H}^2$, with $|0\rangle, |1\rangle$ the orthonormal basis: $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$, where $a_0, a_1 \in \mathbb{C}$ are the quantum amplitudes, representing the square root of the associated measurement probabilities for the eigen states, with $|a_0|^2 + |a_1|^2 = 1$.

From the representation of the qubit, there are several types of errors that can manifest. Small amplitude errors do not directly destroy the superposition, but can accumulate over time, eventually ruining the computation. Bit flips (interchanging eigen states) and phase errors (reversing eigen states) are correlated due to the way they map onto Hilbert spaces and similar to errors in classical computing, therefore receiving similar attention [87, 90]. However, fault-tolerance techniques have to meet different constraints in the world of quantum computing, which makes their implementation particularly challenging:

- The observation destroys the state (any measurement leads to destroying the state superposition);
- Information copying is not possible, quantum physics renders quantum state copying (or cloning) impossible.

In order to deal with these specific problems, some carefully crafted strategies have to be employed. Measuring the encoded information without destroying it requires the

preparation of some ancilla qubits, which can then be measured. Ancilla qubits are used to duplicate the eigenstate information in such a way that the ancilla become entangled with the useful data, therefore allowing for concatenated (error) coding. However, this is made especially difficult, since setting the ancilla qubits is itself an erroneous process, which leads to limitations to the length of the error recovery process [90]. An alternative approach to concatenated coding is based on using Reconfigurable Quantum Gate Arrays (RQGAs) for a quantum configuration state register, which encodes a simultaneous superposition of distinct error checking and correcting circuits. The measurement of the configuration results in the selection of a single circuit, with distinct advantages concerning the accuracy of the process.

In order to evaluate the characteristics of the fault tolerance algorithms and methodologies in quantum computing, the QUERIST (from QUantum ERror Injection Simulation Tool) was devised, using simulated fault injection techniques. The basics of QUERIST are described in figure 2.9, divided over three cycles that are common to both conventional and quantum computing: initialization, simulation, and data computation [90]. The initialization accepts an HDL model for the circuit and its corresponding error model, which is then subject to a simulation and entanglement analysis, with fault injection being performed via mutants and saboteurs. The final result is the probabilistic accuracy threshold, which is used to validate the fault-tolerance algorithms and methods employed.

While QUERIST may point towards successful implementation of reliable quantum processes, the complexity remains an issue, since entanglement is present in any useful quantum algorithm. For instance, protecting a single logical qubit can be achieved by employing Steane's 7-qubit code and 7 ancillary qubits. As a consequence, protecting a 4-qubit logic circuit requires a 54-qubit ($4 \cdot 14$) register, which relates to a simulation space of 2^{56} classical states for reliability assessment, requiring an enormous computational power. In order to optimize this process, a hybrid simulation-analytic procedure was devised to provide scalability to circuits of arbitrary size, which combines simulated fault injection in a HDL-based framework (QUERIST) with theoretical reliability analysis (based on Markov modeling) [120]. The results indicate that arbitrary-sized quantum circuits cannot be considered unless the noise values are sufficiently low, proving once again that quantum computation is extremely fragile.

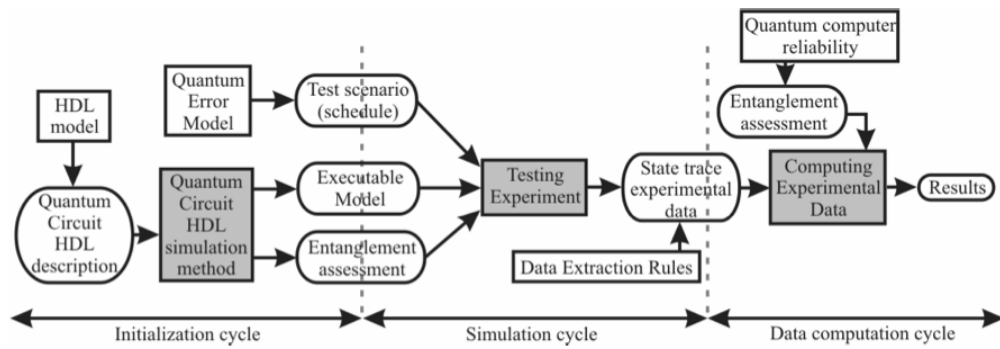


Figure 2.9: Overview of the QUERIST [90].

2.4 Biologically-Inspired Quantum Computing

The emergence of nanoelectronics has the potential to significantly transform the design of conventional computing, not only due to the unprecedented density of devices integrated on a single chip, but also because of the inherent sensitivity of such tiny components. Since computing systems must remain dependable, valuable insights can be drawn from the natural sciences: while biology provides numerous examples of successful fault-tolerance strategies in living systems, physics offers the underlying theoretical foundations required by algorithmic processes (including those found in biological systems). Such common ground suggests quantum computing could benefit even more from biologically-inspired techniques, thus building incentives for further exploration.

Designing quantum circuits requires finding optimal decompositions using a limited set of available basic gates. However, while any useful application may be described with real numbers matrix elements, forming a succession of unitary transformations, determining the details prove computationally challenging. The reasons are manifold: there is no commonly-accepted benchmark for quantum circuit synthesis, the number of quantum gate permutations is enormous, and any change in a quantum topology radically influences the circuit functionality.

Leveraging on inspiration provided by biology in classical computing, the genetic programming and genetic algorithms have penetrated quantum computing to approach the automatic build of complex quantum circuits. In order to employ genetic algorithms, the potential circuit must will be divided into sections and planes (corresponding to vertical and horizontal sections, respectively) to allow chromosome encoding [95]. Each chromosome will be represented as a matrix, the number of rows being limited to the number of qubits, while the columns represent the succession of gates and reflects the

granularity of the synthesis process. The algorithm adaptation to quantum circuits and the chromosome are described in figure 2.10. A single point crossover was implemented and mutations are allowed, the fitness being expressed as a matching percentage to the target output.

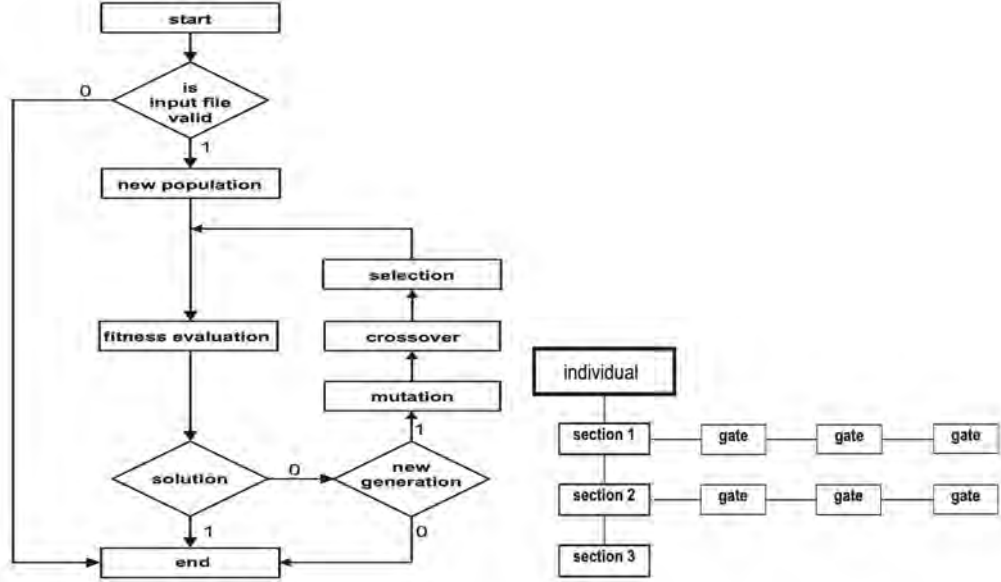


Figure 2.10: Quantum genetic algorithm (left) and chromosome encoding (right) [95].

Experiments were performed in order to generate circuits as large as 5-qubit-wide, using a variety of gates with 1, 2, and 3 inputs, yielding better runtime and faster convergence [95]. The computational resources required are increasing rapidly with the complexity of the output function, due to the mathematical operations involved. The optimal encoding of the chromosome allows dynamic memory allocation, which is then reflected on the quality of the results.

The proposed approach employs mutation and crossover probabilities, population size, and selection type, as means to influence and steer the evolution path. However, statistical information from the current state of the search could be used to modify these parameters, leveraging meta-heuristics to improve handling the search space and ultimately, the convergence of the algorithm [98, 96].

In order to provide any design guidance for improving the synthesis of quantum logic circuits, a thorough understanding of the constraints is essential, the result being a layered software flow that is described in figure 2.11. The adaptive strategy employed uses the fitness results from each population (best, worst, mean) together with operator performance as statistical data during runtime, updating parameter values periodically. Applying this meta-heuristic strategy provided arguments for the parameter tuning vs parameter control

discussion within quantum computing, while enabling our quantum genetic algorithm to provide better runtime and successful synthesis for circuits as large as 7-qubit-wide [96, 94]. Such performance becomes even more valuable since additional computational power is required to assess the statistical significance of the analytical values for the parameters involved and vary their tuning during algorithm runtime. Adapting the crossover and mutation dynamically brings superior performance when compared to conventional genetic algorithms in terms of convergence and number of solutions evolved [94].

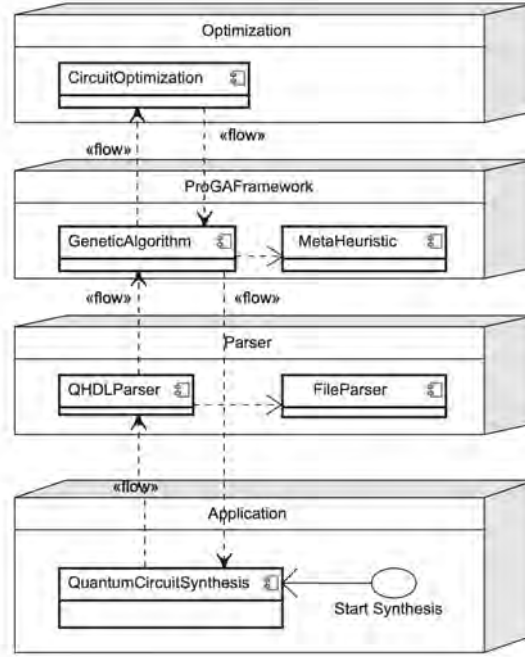


Figure 2.11: Flow for the meta-heuristics genetic algorithm (MH-QCS) [96].

Chapter 3

Expanding Dependability: Security Through Steganography and Cryptography

In 2 we discussed aspects related to assessing dependability in the context of two representatives of emerging technologies, namely bio-inspired and quantum computing. As we know, the microscopic quantum states are prone to frequent errors, which can alter the consistency of the quantum computation processes if the error probability within the basic components (qubits, quantum gates) surpasses a certain value, called the *accuracy threshold*. On the other hand, mission-critical applications require new levels of robustness that can be achieved by employing multi-level self-repairing and self-reconfiguring strategies, which constitute some of the targets of bio-inspired computing. Both computing paradigms share the same error characteristics of being intrinsic to the computation process, without the presence of intentional malevolence.

However, the presence of malicious actors cannot be discarded when dependable systems are concerned, as they often represent a clear and present danger. When considering the fields of cryptography and steganography within the large domain of dependable computing, such a situation is commonly found in both: cryptographic algorithms must resist cryptanalytic attacks while steganographic techniques must avoid detection by steganalysis. Therefore, defense against adversarial threats represents a feature offered by both fields. Furthermore, both rely heavily on mathematical rigor, implied by number theory and algebraic processes, and information theory and signal processing. Our contributions extend toward high-speed encryption within the AES standard as well as smart data rearrangement for high-capacity steganographic algorithms, which will be described next.

3.1 High-Speed AES with Concurrent Error Detection

Rijndael is a symmetric-key encryption algorithm that was selected by NIST from a pool of algorithms to replace the aging Data Encryption Standard (DES), therefore becoming the new standard called AES [57]. At its core, AES is a symmetric, round-based block cipher defined by a block size of 128 bits and key lengths of 128, 192 and 256 bits.

In its most common form, the 128-bit AES organizes information as 16 bytes in a 4x4 matrix, which is altered by each round by using four basic round transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. The encryption requires 11 rounds, the first being dedicated to the encryption key, followed by 10 successive rounds. The decryption consists of the corresponding inverse transformations executed in a reverse order compared to the encryption process. Each AES operation uses operands from the byte processed by an AES operation represents an element of the Galois Field $\mathcal{GF}(2^8)$ generated by the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$.

The SubBytes together with its corresponding inverse, InvSubBytes, are the most complex operations in AES, their complexity representing a source for sophisticated attacks (such as differential and linear cryptanalysis) and also an issue for incorporating testability solutions [80]. The suitability of Built-In Self-Test (BIST) for non-linear operations performed in AES should be therefore analyzed over the SubBytes and InvSubBytes operations (considered as the circuit under test, or CUT, in figure 3.1), since their testability was lacking from the available literature. The ATALANTA automatic test pattern generation tool derives the smallest deterministic test vector set as 152 test patterns for the inversion unit and 164 for SubBytes and InvSubBytes); however, N-detect single stuck-at fault test vector sets are considered the preferable alternative [119]. The test patterns generated by ATALANTA are similar in single stuck-at fault coverage qualities to the sequences provided by an 8-stage linear feedback shift register (LFSR). The Response Compactor (figure 3.1) provides compression of the output vectors to generate a unique signature, in order to keep the hardware overhead required by storage at a minimum. The results yielded by the VHDL synthesis is given in table 3.1.

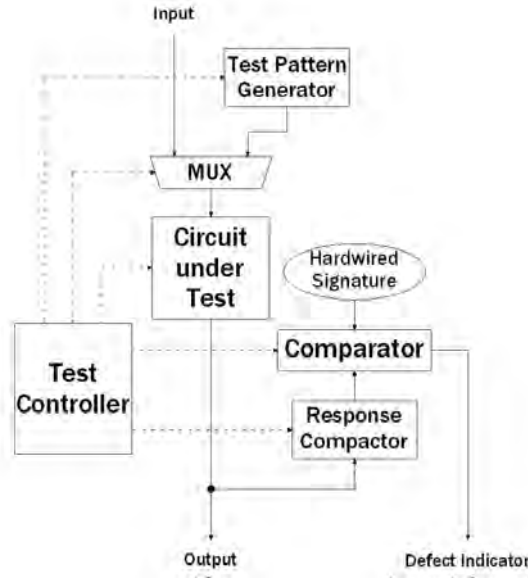


Figure 3.1: The BIST process. [80]

	Gate count	Logic levels	Collapsed faults
SubBytes	651	17	1798
InvSubBytes	680	15	1856
Inversion module	657	16	1765

Table 3.1: VHDL synthesis results for main modules [80].

The testing patterns are generated by a LFSR connected to a multiple input signature register (MISR), with the BIST mechanism being implemented based on a comparison of the signature collected against the pre-computed correct signature. Of course, there are multiple available LFSR-MISR pairs that yield 100% single fault coverage. We evaluated through experimentation all possible LFSR-MISR pairs and concluded that no 8-stage MISR is able to provide 100% single stuck-at fault detection. However, we also determined the 9-stages MISR configurations that offered full coverage, and selected for further investigations the pair consisting of the LFSR based on the $m(x) = x^8 + x^6 + x^5 + x + 1$ polynomial and the MISR based on the $m(x) = x^9 + x^4 + 1$ polynomial.

We compared our BIST proposal with ATALANTA's deterministic test vector set of 152 vectors and determined the fault detection rate for different numbers of stuck-at faults affecting the CUT. Each experiment consisted of 10^6 simulations with the same number of randomly generated faults injected into the circuit for relevant fault coverage. The results shown in figure 3.2 indicate that the deterministic ATPG outperforms the LFSR-MISR strategy in the majority of cases, independent of the number of injected faults, however,

with a significant hardware overhead. Our BIST architecture provides full coverage for single stuck-at defects, while also offering a higher than 99.78% fault detection ratio for multiple stuck-at faults.

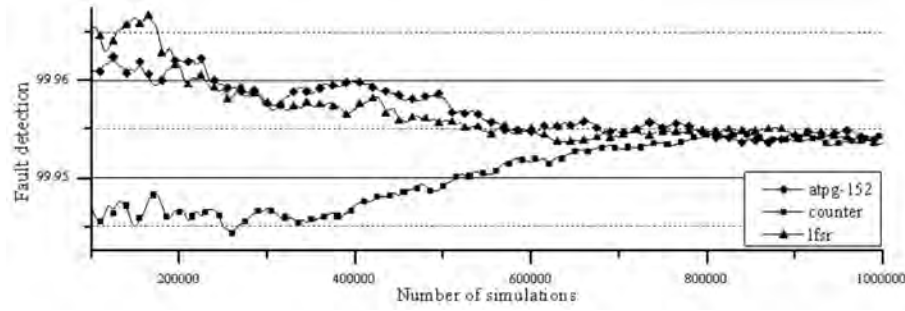


Figure 3.2: Fault detection applied to AES’s multiplicative inversion for 500 injected faults. LFSR, ATPG and binary counter were used as test pattern generators. [80]

The modern need for protecting various information has democratized cryptographic systems; however, it has been determined that cryptographic systems are vulnerable to faults, which also added a side-effect of increased attacks targeting all possible global, physical and electrical failures. Fault injection (also known as fault attacks) can be used to trigger responses that can provide useful insight to further differential fault analysis, with a successful attack on AES only requiring two faulty responses [62, 12]. It becomes therefore crucial that faulty conditions be prevented from delivering the corresponding incorrect responses, therefore requiring protective measures be implemented.

Our proposal represents an architecture for an efficient, on-line testing process that also integrates well with the BIST mechanism for the AES operations [81]. The signature computed from the input of the inversion module is checked against its calculated signature, a process described by figure 3.3. The uniqueness of the checking process is ensured by the fact that given any two elements $\alpha, \beta \in \mathcal{GF}(2^n)$, where $\alpha \neq 0$, $\beta \neq 0$, $\alpha \neq \beta$, and $\alpha \neq \beta^{-1}$ then equation (3.1) holds. This ensures that each signature α is associated with its corresponding verification α^{-1} leading to fault detection. Our experimental results indicate the detection rate is higher than 90% for multiple intermittent faults affecting the inversion module and therefore appropriate for protecting it against fault attacks.

$$\alpha + \alpha^{-1} \neq \beta + \beta^{-1} \quad (3.1)$$

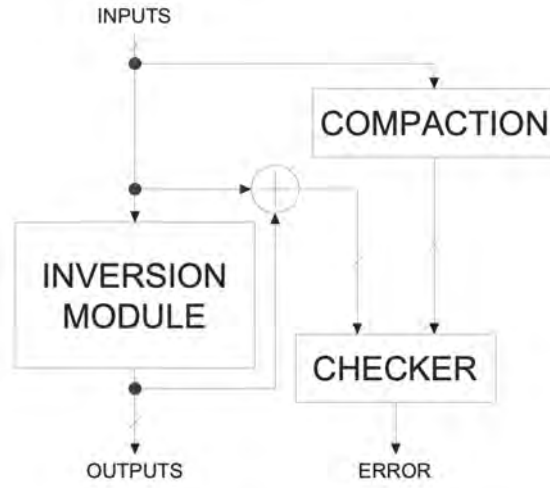


Figure 3.3: Architecture for testing the AES inversion. [81]

For our experiments we modeled the AES operations in VHDL, the synthesis being carried out using the IIT Standard Cell Library for TSMC 0.18 μm . For the error detection we used a base AES design with the inversion module assembled from combinational logic. To provide error detection the SubBytes and InvSubBytes transformations were implemented to include the inversion module, which increases the area by 8% and the latency by 30%, as shown in table 3.2. However, the initial overhead imposed by including the inversion module is compensated by a significant reduction in the area required by the encryption and decryption processes, from 788827 μm^2 to 620971 μm^2 , representing about 21%.

Area [μm^2] / Critical path [ns]	SubBytes	InvSubBytes
Without containing the inversion module	13419 / 2.19	13369 / 2.15
Containing the inversion module	14476 / 2.89	14411 / 2.64

Table 3.2: Overhead for SubBytes and InvSubBytes modules [81].

The addition of our concurrent error detection (CED) strategy introduces a supplementary impact on the area required by the physical implementation. We analyzed several variants corresponding to AES encryption, AES decryption and joint AES encryption-decryption and also implemented a hardware duplication solution in order to assess the area overhead and error detection rate. The results shown in table 3.3 indicate an area overhead of 39.5% as opposed to the 54.35% imposed by the hardware duplication.

	No CED	Hardware duplication	CED
AES encryption	506912	782452	707171
AES decryption	541400	815423	741112
AES encryption/decryption	620971	896083	830283

Table 3.3: Area overhead for AES verification architectures [81].

In order to assess the detection rate for these architectures we used the stuck-at and stuck-open faults and employed a simplifying assumption of faults affecting the inversion module only; this is justified by the fact that the inversion module occupies as much as 67% of all combinational elements of the AES implementation, which drives a correspondingly high probability of it being affected by faults. In this situation, the architecture based on hardware duplication provides a perfect detection rate of 100%. All possible stuck-at faults were derived through the ATALANTA tool while stuck-open and intermittent faults were derived through our own simulation framework. The stuck-at or stuck-open faults were considered to be equally likely. The detection probability for permanent faults shows an exponential increase with the number of rounds performed. The detection rate is shown for different number of faults injected in table 3.4, at any time remaining higher than 94%, while also offering a significant reduction in the area overhead.

Faults injected	1	2	10	100
Detection Rate	99.32%	98.82%	97.14%	94.28%

Table 3.4: Fault detection rate for AES [81].

Having studied the particular aspects of AES encryption and decryption processes, together with adequate BIST and concurrent error detection strategies, a further step is represented by the synthesis of a high-performance cryptocore. Spatial parallelism can be effectively employed in hardware, while also allowing efficient use and sharing of the resources involved. Designing a high-speed AES implementation requires a fine-tuning of the datapath, since each round has to process all 128 bits in parallel, therefore imposing a 128-bit architecture, shown in figure 3.4a. The SubBytes operation is defined as a multiplicative inversion followed by an affine transformation, while InvSubBytes is defined as the inverse affine transformation followed by a multiplicative inversion. The multiplicative inversion represents about 90% from the area required by SubBytes (and

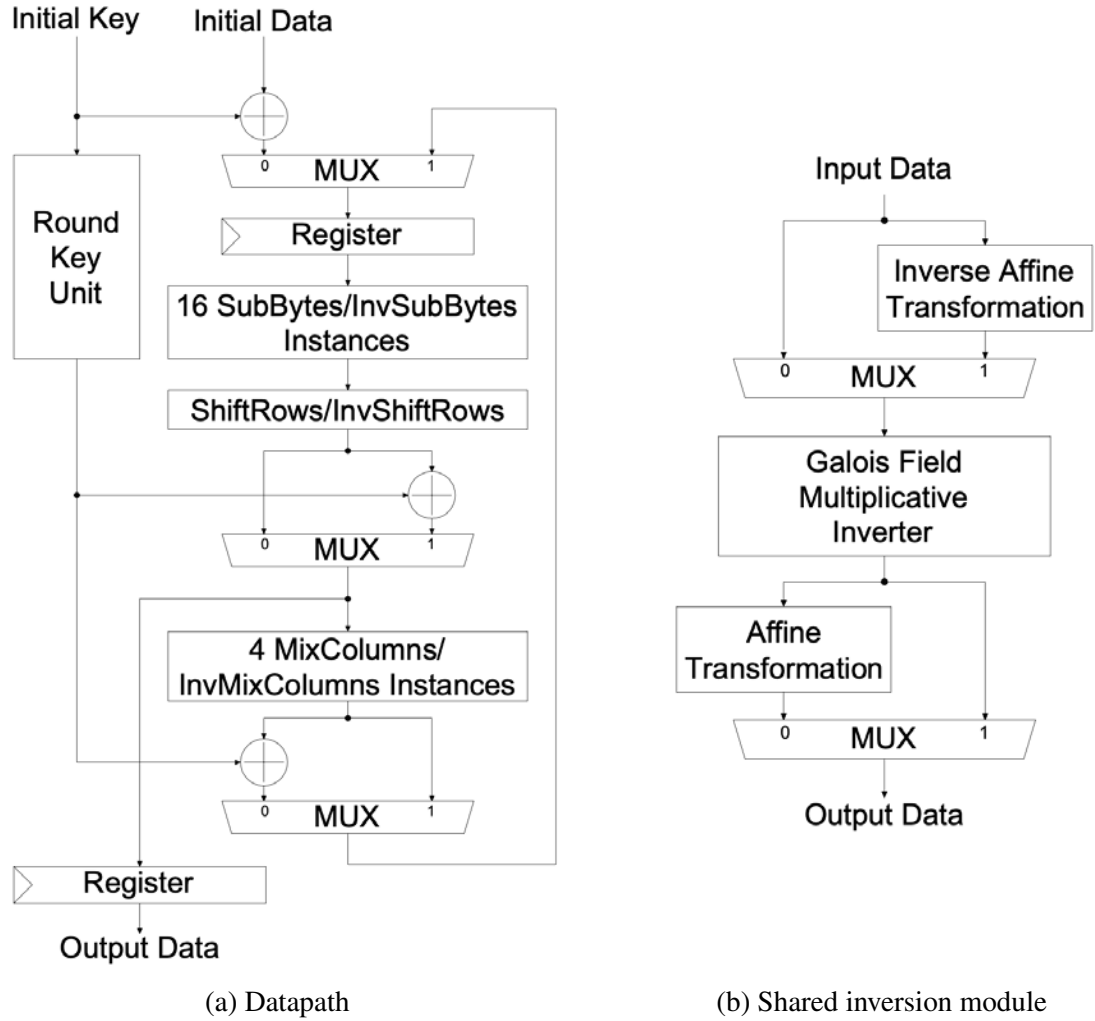


Figure 3.4: A high-speed AES architecture [79].

InvSubBytes), therefore allowing a reuse of the field inversion module with a corresponding increase in latency, as shown in figure 3.4b. The encryption and decryption processes are integrated into the same datapath, which is made possible by the fact that SubBytes and InvSubBytes share the same position in the encryption and decryption rounds.

The design choices discussed were successfully synthesized and implemented into an Altera FPGA (Cyclone II EP2C35F672) by using tools such as Altera's Quartus and Mentor Graphics' Modelsim. As a result, our AES core included a number of 6608 Logic Elements and delivered a throughput of 805.24Mbps at a 70Mhz clock. The corresponding resource efficiency was 0.122Mbps/LE [79].

3.2 High-Capacity Steganography

The internet traffic's continuous, exponential growth has intensified the need to protect sensitive information from digital threats, research in cybersecurity remaining a crit-

ical priority. Despite cryptography being synonymous with digital protection, with numerous implementations (as argued in section 3.1), it cannot be sufficient on its own. Offering a complementary approach to information security, steganography conceals information within common file formats (called *carriers*) so that only intended recipients are aware of the hidden data's presence. This field has gained significant attention recently due to the demand for enhanced security channels that not only encrypt sensitive information but also embed it invisibly within standard carriers such as images, audio files, or videos. Since digital images represent the most commonly transmitted media type on the Internet, this makes them the carrier choice for steganography.

The combination of cryptographic encryption and steganographic concealment creates a robust security framework that addresses two fundamental requirements of secure communication: data protection through encryption and invisibility through covert transmission. This dual-layer approach provides superior protection compared to using either technique independently [28].

Due to the fact that the human eye does not perceive small changes in the color tone, substituting the least significant bits (LSBs) of every pixel to embed the secret data represents the preferred method in steganography. To provide the desired amount of stealth, the steganographic algorithm must withstand a range of reverse engineering methods, also known as steganalysis. While a natural image contains random distributions of colors, substituting them with secret data may break this randomness and introduce logic levels that can be detected using advanced steganalysis algorithms. Therefore the main focus of steganography is to provide such a data distribution pattern so the steganographic image appears as natural image.

LSB substitution can be performed two ways. The first, less effective strategy is to embed the secret data irrespective of data loss incurred, which is supposed to be followed by an adjustment of the quality of the resulted image. Raising the number of LSBs employed reduces the quality of the initial image, which has to be compensated by the quality adjustment, leading to a computationally intensive process that does not warrant the best embedding capacity. A second, more recent strategy uses more computational power for the initial embedding process, matching the secret data to the carrier image, which in turn offloads the following post-processing. Either way, producing an image that would be steganalysis-resistant has to maintain some metrics at acceptable values, peak signal-to-noise ratio (PSNR) being the most used [70]. Therefore, steganographic

processes usually combine the optimal LSB substitution (OLSB) [122] with the optimal pixel adjustment process (OPAP) [24]. OLSB uses a bijective function to map the best combination of secret data that produces the minimum amount of distortion on the carrier. The minimum distortion is obtained by OPAP by optimizing the mapping of the secret data onto the carrier.

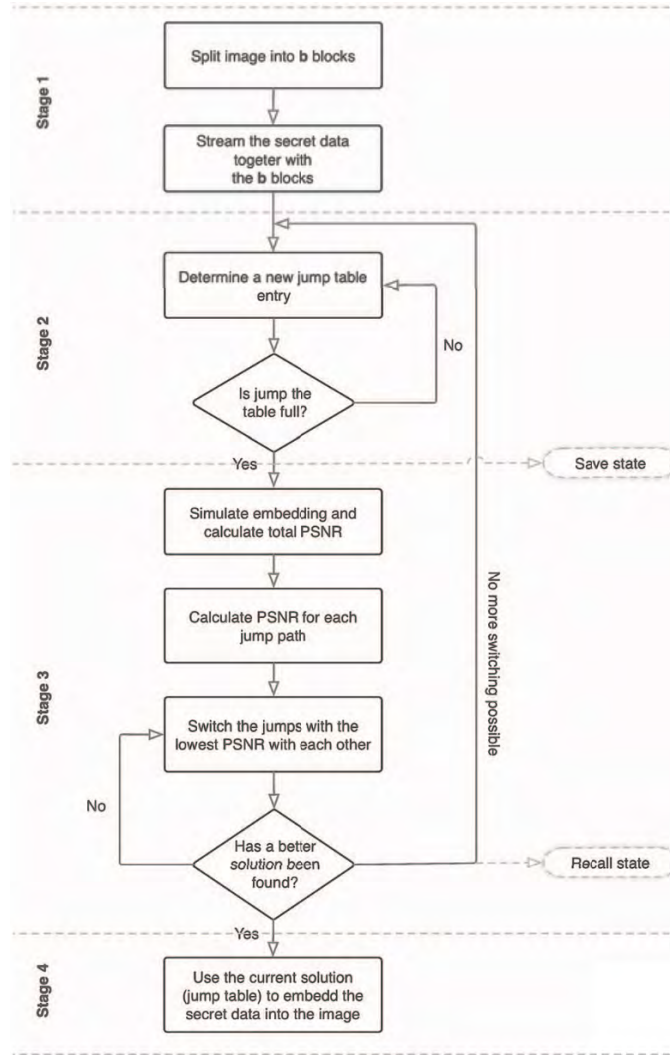


Figure 3.5: Our steganographic encoding process. [70]

Our approach represents a hybridized OPAP that uses a jump table to distribute the secret data within the carrier, the jump table itself being also embedded; we trade a slight reduction in capacity for a better match between the secret data and the carrier, which brings superior PSNR and therefore additional resistance to steganalysis [70]. The algorithm is divided into four successive stages: image segmentation (the carrier is segmented into a set of b equally sized blocks), solution generation and evaluation (the jump table is populated with possible solutions, which will be evaluated based on the resulting PSNR value, keeping the best overall), and embedding both the secret data and the jump table

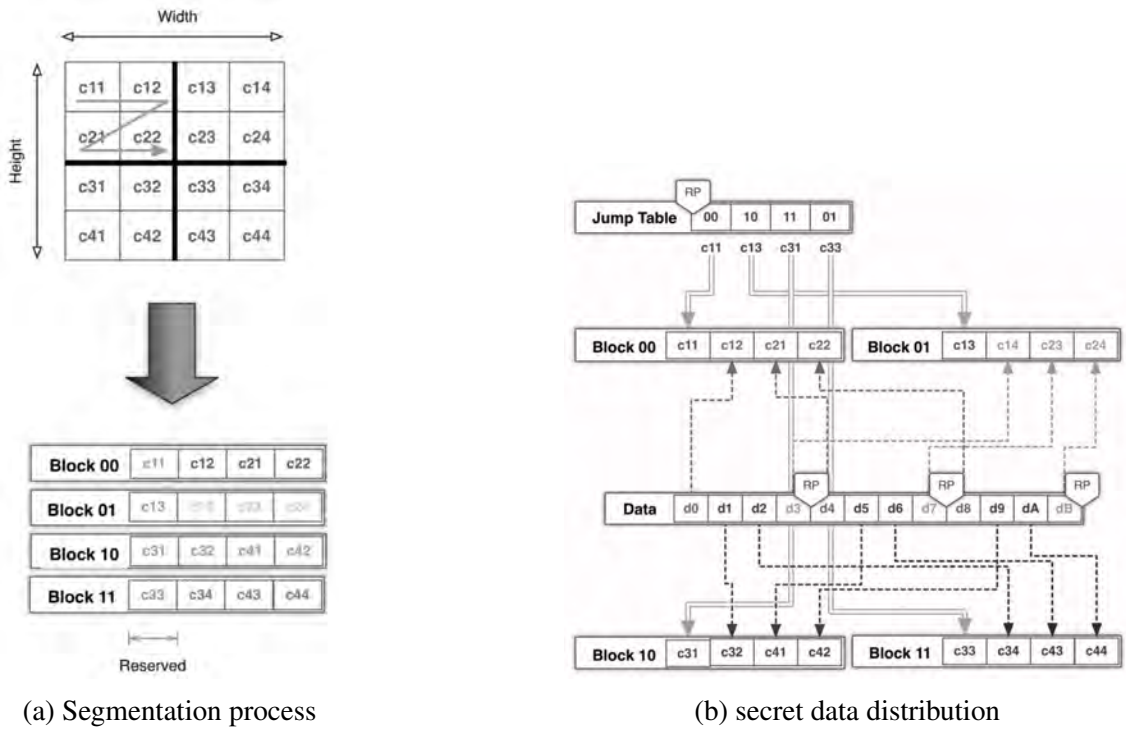


Figure 3.6: Matching the secret data and the carrier image [70].

onto the carrier, as shown in figure 3.5. In equation (3.2), bpp represents the number of bits used for embedding the secret data, the $tileMultiplier$ being used to extend the minimum value b of the segmentation blocks. Parameters $tableSize$ and $tableMultiplier$ from equation (3.3) represent the number of entries in the jump table, and the size of each entry, respectively.

$$b = 2^{bpp+tileMultiplier}, \text{ where } tileMultiplier \geq 0; \quad (3.2)$$

$$tableSize = b \cdot tableMultiplier, \text{ where } tableMultiplier \geq 1; \quad (3.3)$$

The segmentation stage for an example where $bpp = 2$, $tileMultiplier = 0$, and $tableMultiplier = 1$ is described in figure 3.6a, where a 4x4 pixels carrier (with 8 bits per pixel for color depth) is divided into $b = 2^2$ blocks, a stream of pixel information and states being generated for each block. For parameters $bpp = 2$ and $tileMultiplier = 0$ we will have a jump table with 4 entries, corresponding to 4 handling streams carrying both jump and secret data. Each of the handling streams have a capacity of 8 bits, of which 2 are reserved for the jump table, therefore leaving 6 bits available for the secret data. The secret data capacity of the carrier results as 24 bits. Each handle stream begins with one element from the jump table, followed by secret data reordered by the jump table itself, as described in figure 3.6b, all handle streams assembling the steganographic image.



Figure 3.7: Well-known images used as carriers.

We performed multiple experiments on 512x512 pixels carrier images (with a depth of 8 bits/pixel grayscale) using well-known images such as Lena, Baboon, Jet and Peppers, shown in figure 3.7. The value for the number of secret data bits per pixel bpp varied from 1 to 4 in order to determine the best value for the number of blocks b and the size of the jump table. Of course, increasing bpp leads to the increase of b , which in turn increases the size of the jump table. Because our algorithm also embeds the jump table onto the carrier, for comparison purposes we appended the jump table to the secret data used for other state-of-the-art algorithms, such as Simple LSB (SLSB), Optimal LSB (OLSB), and Optimal Pixel Adjustment Process (OPAP). Results are given in table 3.5 indicating our algorithm consistently outperforming the others by at least 2.4dB. This relates to an increase in quality for the resulting steganographic image at the expense of roughly 5% of the total embedding capacity.

bpp value	SLSB	OLSB	OPAP	Our algorithm
1	51.1335	51.4125	53.3104	55.9436
2	44.0167	44.3650	46.1571	48.2192
3	37.9402	38.2194	40.2401	42.5083
4	31.6843	31.9907	34.0843	36.7142

Table 3.5: Performance comparison (values in dB) [70].

We continued to refine our algorithm and extended the LSB matching process to also accommodate color images with a color depth of 24bpp and beyond [71]. It was determined by repeated experiments that for over 78% of the cases the best solution for embedding was found in the first 30 attempts, only a small number of these sustaining further enhancements. Another useful aspect was that a ration of 3:8 is ideal (that is, 3 bits of secret data to be embedded into 8 bits of carrier image). The new algorithm produced even better results, shown in table 3.6.

SLSB	OLSB	OPAP	New algorithm
37.9402	38.2194	40.2401	44.3701

Table 3.6: Performance comparison update(values in dB) [71].

Chapter 4

Leveraging Informational Plasticity for Multi-Objective Optimization

Real life presents quite a large diversity of fields that require multi-objective optimization that could greatly benefit from algorithms, methods and tools that are specific to computer science and engineering. Such situations involve several conflicting objectives, where dependencies on each other result in non-linear variations whenever attempting to maximize any of them. Multi-objective optimization scenarios usually focus on two conflicting objectives, a situation encountered in many key areas:

- computer science and engineering: network design (bandwidth vs. cost), machine learning (accuracy vs. complexity), cloud computing (performance vs. resource usage), database optimization (query speed vs. storage);
- aerospace engineering: aircraft design (weight vs. performance), electronic circuit design (power vs. speed), structural design (cost vs. strength), automotive design (fuel efficiency vs. performance), robot path planning (time vs. energy);
- financial applications: portfolio optimization (risk vs. return), investment strategies (profit vs. risk), resource allocation (cost vs. benefit), trading systems (profit vs. volatility);
- manufacturing: process optimization (quality vs. cost), production scheduling (time vs. resource usage), supply chain optimization (cost vs. delivery time), quality control (defects vs. production speed);
- energy systems: power grid management (cost vs. reliability), renewable energy integration (efficiency vs. stability), energy distribution (cost vs. environmental impact), power plant operation (efficiency vs. emissions);
- environmental management: waste management (cost vs. environmental impact), resource conservation (economic benefit vs. sustainability), land use planning (development vs. preservation), pollution control (economic cost vs. environmental benefit);

- healthcare: Treatment planning (effectiveness vs. side effects), resource allocation (cost vs. coverage), drug design (efficacy vs. toxicity), medical imaging (resolution vs. radiation exposure);
- transportation engineering: traffic management (flow vs. emissions), route planning (time vs. fuel consumption), public transit (coverage vs. cost), logistics (delivery time vs. operational costs).

While far from being exhaustive, this list highlights the importance of adapting information-based processes to provide multiple optimizations simultaneously. Leveraging information plasticity is therefore crucial for determining process parameters in areas that may or may not involve computer science or engineering directly. The oldest successful multi-objective optimization processes are found in nature, exhibited by the living beings, developed and thoroughly tested in a time frame that is unavailable to scientists and engineers. Due to the vast period of time employed for living experiments, the inspiration from nature becomes both necessary and obvious, in a research effort to adapt its mechanisms into engineering and export its proven results into useful processes.

Our current understanding of the living matter, the result of continuously running evolutionary processes, can be represented over three distinct axes of the POE model [107]. The phylogenetic axis (P) is concerned with the temporal evolution of the genetic program of all living beings, the rich biological diversity of all species being the result of its non-deterministic, low-error rate reproduction process. The ontogenetic axis (O) is concerned with the temporal evolution experienced by a multicellular organism, from its early stage of a primordial cell (the zygote) to the final, mature organism. Ontogenetic processes are deterministic, featuring very low error rates and includes cellular division and cellular differentiation. The epigenetic axis (E) involves processes that take place in multi-cellular organisms and exhibit levels of complexity that cannot be integrated by the phylogenetic or ontogenetic processes. The only systems capable of internal topological change and showing the remarkable feature of learning are known to be the nervous, immune, and endocrine systems.

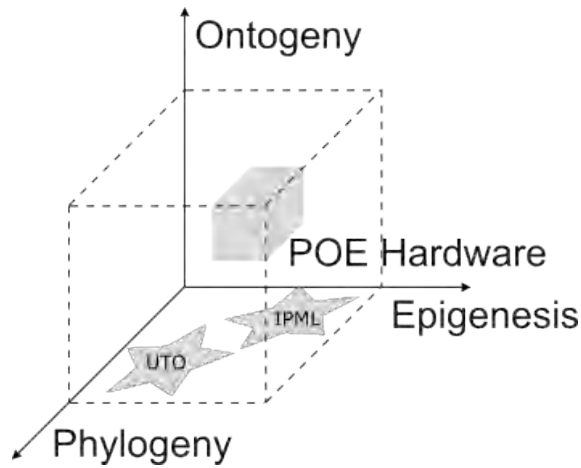


Figure 4.1: The POE model and placement of focus applications: urban traffic optimization (UTO) and information plasticity with machine learning (IPML).

Bio-inspiration has established a wide range of applications, including computer science and engineering, with ontogenetic processes having been analyzed previously [86]. While looking for acceptable solutions into vast search spaces can benefit from leveraging the power of genetic algorithms or complex networks (with applications in urban traffic optimization [49, 50, 51, 117, 31, 10, 11, 6, 112, 7]), the recognition of particular situations and scenarios, with the ability of providing generalizations, can be successfully provided by machine learning to better understand traffic dynamics or interpret human actions and gestures [5, 121], their place within the POE model being shown in figure 4.1. Such applications constitute the focus of this chapter, therefore rendering phylogenetic and epigenetic processes of particular interest for this thesis.

4.1 Collaborative Monitoring for Urban Traffic

Modern society has been continuously experiencing the challenges of increasing urbanization and population growth, and their deep impact brought onto mobility and transportation management. Naturally, advanced technologies are being set to enhance transportation efficiency, safety, and sustainability, all under the common denomination of Intelligent Transportation Systems (ITS).

The efficiency of transportation represents a primary necessity due to growing congestion in urban areas, while matching transportation infrastructure to the requirements of traffic remains an ongoing struggle. ITS provides real-time traffic management capabilities, optimizing traffic flow through adaptive signal control, dynamic routing, and intelligent parking systems. Such optimizations do not describe efficiency alone, but also

warrant sustainability for urban growth by minimizing the environmental impact of pollutant by-products of the transportation processes.

Through advanced sensors, cameras, and communication systems, ITS can monitor dangerous driving behaviors, alert drivers to potential hazards, and could also detect and respond to accidents in a timely manner. Additionally, ITS will integrate autonomous traffic. Since these are challenging tasks, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication systems are being studied in order to enable preventive safety measures, which are crucial for the development of urban communities.

While various approaches exist for in designing an ITS for urban road traffic, they all share the necessity of reliably collecting characteristic data that will assemble the momentary footprint of the traffic itself. Such data, consisting mainly of vehicle trajectory and speed, are usually recorded in a centralized way, by employing V2V and V2I infrastructures. However, building such infrastructures proves to be especially difficult and time consuming since they are altering the very strings of the community. An alternative way that has been building momentum in the last decade is based on the good will of the citizens, using their personal devices and the existing communication infrastructure to gather relevant data in almost real-time. The advent of mobile communication devices (typically smart phones) can provide quality data about road traffic conditions based on the movement of their respective users.

A prime example of such an application is the Waze mobile app (<https://www.waze.com/en-GB/live-map>), which uses a social approach for crowd mapping of traffic conditions. However, if Waze provides a traffic snapshot at that time, it can all be potentially misleading when the quality of data falls, directly dependent on the number of active users in a particular area: having a small number of Waze users moving at low speeds will be misinterpreted as a traffic jam.

In order to provide a long-term qualitative description of the state of urban traffic, a traffic map could be built by employing data from the very actors from the traffic in a collaborative effort [49]. Instead of a momentary snapshot of traffic, which may or may not give an accurate description, such a map would reveal its real attributes, describing tendencies and network flaws, thus becoming the equivalent of open source in computer software. As a key consequence, this traffic map would be able to provide essential insight on how the architects of the urban environment should shape and amenitize it for the future.



Figure 4.2: System architecture (a) and web application interface (b) [49].

The architecture of our system is presented in figure 4.2, with the client side (Harvester) being implemented as a smartphone application performing user-centric, web-based operations and also responsible with sampling and uploading acquired traffic data to the server; the server side performs aggregation of all user data on the map. A typical scenario includes the user with a smartphone running the Harvester application, which acquires and sends samples of traffic-related data (from the on-device GPS) to the Core, which parses the data and computes all the secondary parameters that will be stored in the database. Distance and average speeds are computed between any consecutive sampling points, allowing queries necessary for extracting reports. All testing was performed using HP iPAQ 614c devices running Windows Mobile 5 (for the Harvester side), with the Core side being supported by a SunFire x2200 M2 server. During May-June 2012 we recorded data from daily rides to test the application functionality, results being shown in table 4.1.

As shown in table 4.1, data was recorded over an extended period of 3486.69 hours, which gives an estimation on traffic characteristics over any covered route, at any recorded timestamp in a day. While Waze reacts to traffic changes in real-time, our application employs a statistical approach for data processing and builds a traffic history, which can be particularly useful for describing traffic tendencies over time. Our approach allows true traffic data collection in a collaborative effort, however, from an ITS standpoint this represents one step only. Once traffic data is collected, it must be used to provide efficient traffic coordination.

Week	No. of samples	Recording time (hours)	Recorded distance (km)	Average speed/car (km/h)
1	145712	404.76	2612	51.6
2	152592	423.87	3151	59.5
3	212302	589.73	4127	56
4	132789	368.86	3856	83.6
5	142598	396.11	3217	65
6	82317	228.66	1248	43.7
7	151321	420.34	3278	62.4
8	235569	654.36	3615	44.2
Total	125520	3486.69	25104	58.25

Table 4.1: Aggregated data over the two months testing period [49].

4.2 Softening Intersections for Permeability Enhancement

The previous section described a way of gathering traffic-related data in order to build a traffic map for an urban community. This way, the authorities are able to better understand the hot spots in traffic and react by employing appropriate traffic coordination measures.

Traffic flow analyses consistently show that intersection management is critical for traffic flow, with a particular emphasis on how multiple intersections coordinate to regulate vehicle movement. Traffic lights serve as crucial control points, determining which vehicles can proceed at any given time. By incorporating historical or real-time data, these signals can adapt their timing to match changing traffic or enforce some desired patterns. This way, each traffic light becomes an automated traffic officer, enforcing deterministic behavior over the road network and contributing to precise signaling and efficient vehicle movement. However, predictable traffic flow and static rules were only sufficient for developing urban communities [64]. Modern urban areas face significant traffic disruptions with intersections increasingly becoming overwhelmed. Research effort has therefore been directed to the development of advanced algorithms for automatic traffic signal optimization.

Adapting existing transportation infrastructure to accommodate increasing traffic volume is difficult, given geographical constraints that limit road expansion and the high

costs of new construction. This is the main reason for which optimizing the current road network receives the focus of research strategies, either at network-level or at intersection-level [17]. Intersection-level optimization concentrates on managing local traffic parameters specific to individual intersection environments, while network-level optimization involves subtle modifications to road usage, such as implementing intelligent routing by coordinating signals across multiple intersections. In order to deal with these challenges in order, we devised a three-layer approach [30] that makes use of multiple optimization and structured communication, as described in figure 4.3.

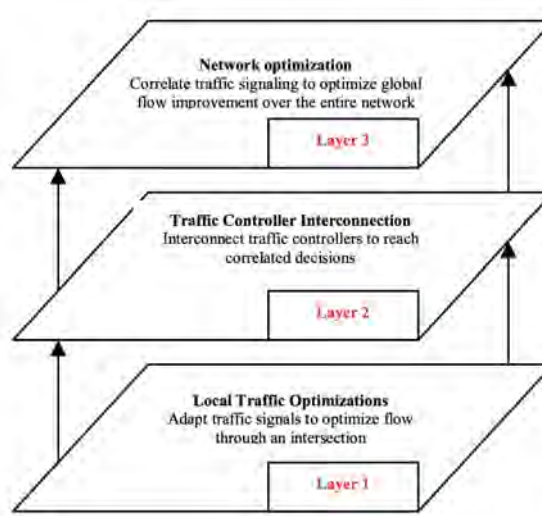


Figure 4.3: Proposed traffic optimization stack [30].

In our view, traffic signaling optimizations occur at Layer 1, finding the optimal phase superposition in order to respond to traffic changes representing a problem that does not seem to have a generally established solution. While several techniques have targeted Level 1, explicit implementations for Layer 2 are difficult to find, especially when taking into consideration that traffic controllers are computationally-constrained. We conceived an energy-efficient implementation of Layer 1, designed to allow and operate in conjunction with Layer 2 by employing a heuristic approach. The main target of Layer 1 represents a continuous traffic flow between key intersections, where any intersection can be seen either as a standalone entity or part of a complex network described by green times, traffic flow and cycle lengths. If intersections operate independently, our algorithm allows for quick derivation of faster flow at intersection level. However, the final goal is to reach synergistic intersection operation to achieve faster flow at a global, road network level, corresponding to Layer 3. Traffic signal timings are derived based on three parameters: green time (traffic allowed to flow), traffic flow (number of vehicles passing on a

specific direction) and cycle length (timeframe between two consecutive green times).

Without any loss of generality, we consider a typical 4 way intersection with directions marked counter-clockwise and traffic flow values indicated for a complete cycle (extending this being quite straightforward). The values considered come as a direct result of a case study conducted on a representative intersection from Timisoara, at 9 o'clock AM, on a typical working day, as shown in figure 4.4. At any particular moment, one direction only is considered to have a green light (for simplicity, we did not cover parallel non-conflicting movements, which can be easily added). The traffic flow matrix Td can then be assembled, where each value represents the measured traffic along the corresponding direction, as shown in table 4.2; going from a certain direction to itself is marked for each $Td[i][i]$, where the value is -1. No trapped vehicles are allowed between consecutive green phases.

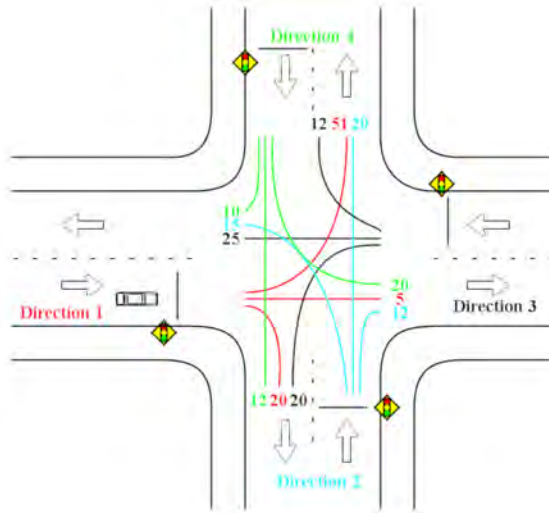


Figure 4.4: Traffic flow values distributed over a four way intersection [30].

Step/direction	1	2	3	4
1	-1	20	5	51
2	15	-1	12	20
3	25	20	-1	12
4	10	12	20	-1
Green time	30	25	30	25

Table 4.2: Green time values defined for a specific intersection [49].

Deriving the green times adjusted to new traffic values holds no consensus in the literature regarding the frequency, the common strategy being to use the *MaximumOutput*

(total traffic value on a direction) and the *Local Minimum* (minimum traffic entering any direction) as variables and allow for slight variations over green times. Our approach is to find the optimal traffic balance for the intersection by searching for the maximum traffic value and using it to determine the increase in green time for that direction, while adjusting the remaining green times accordingly. Running our algorithm produces the results shown in table 4.3. We used PTV VISSIM (<https://www.ptvgroup.com/en>) to evaluate our algorithm [30, 29] over intersections from the central area in the city of Timisoara, VISSIM confirming the tendency to congestion in this area, as shown in figure 4.5.

	Phase 1	Phase 2	Phase 3	Phase 4
Old green times	30	25	30	25
New green times	40	25	30	15

Table 4.3: Green time values computed for a case-study intersection [49].



Figure 4.5: Case study map of Timisoara central area [30].

Using our algorithm for adapting signal timings to traffic values shows that the average queue length figure 4.6a and the maximum queue length figure 4.6b have reduced their values in comparison with the initial setting for the intersection with fixed green cycle times (optimized values shown in red). Further simulations indicate that optimizations performed did not negatively impact on other directions, or the impact was not significant for the average values [30].

4.3 Leveraging Network Science for Urban Traffic Optimization

Complex networks are systems composed of many interconnected elements that exhibit non-trivial topological features, such as scale-free degree distributions, small-world properties, and modular structures. Following the graph theory, from Leonhard Euler's work on the Königsberg bridge problem in 1736, Paul Erdős and Alfréd Rényi developed

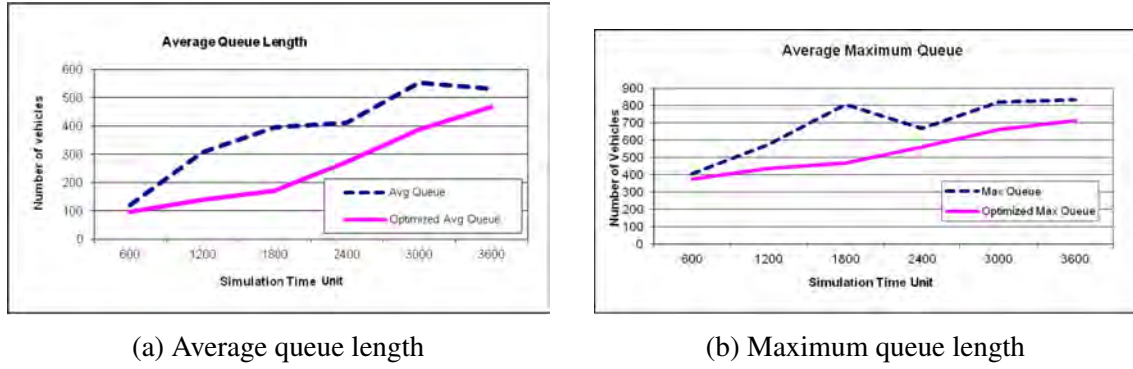


Figure 4.6: Vissim simulation results [49].

random graph theory [38], thus laying the foundation for modern network studies. In 1999, Albert-László Barabási, along with Reka Albert, introduced the scale-free network model in 1999, which describes networks with a power-law degree distribution [13].

Complex networks are essential for modeling interactions in physics, biology, computer science, and social sciences [40]. Computer networks and social networks are considered scale-free, where the degree distribution follows a power law, meaning a few nodes (hubs) have significantly more connections than others. This property enhances robustness against random failures but also increases vulnerability to targeted attacks. Small-world networks display short average path lengths and high clustering coefficients, facilitating efficient information transfer. Some networks from the living matter were found to exhibit small-world properties, examples being the neurons in the brain, connected proteins and transcriptional networks. Other artificial networks also show small-world properties, such as power grid or telephone networks.

As a consequence, it is no surprise that complex networks can at least assist in providing a better understanding of real-life problems, such as optimizing transportation systems, by using network centrality measures and community detection algorithms. While presenting the intricacies of complex networks falls outside the scope of this thesis, we will describe the research carried to understand possible optimizations avenues along the bio-inspired directions of epigenesys and phylogeny.

4.3.1 Employing Complex Networks for Optimal Deployment of Wireless Sensors

Continuous miniaturization and integration of modern hardware, combined with decreasing costs associated unveil a range of applications that employ a variable number of networked, wireless sensors. Example applications include plant automation or disaster management, which require rather large numbers of interconnected nodes (*sensor*,

or *mote* nodes) that operate autonomously and must maintain permanent communication, even in harsh environments (*relay* nodes) with hard power constraints represented usually by batteries. Under these circumstances, the challenging issue becomes the network's topology that has to convey data reliably and with minimal energy impact [51].

We believe the optimal coverage of relay nodes can be determined by measurements of the networks attributes (size and diameter, average path length, clustering coefficient, average degree, density and modularity, distributions of the degrees, betweenness, closeness and centrality) so that throughput and reliability are maximized, while the number of relays and associated costs are minimized.

There exist two main approaches to the placement of sensor network nodes: deterministic and random deployment. While deterministic placement allows for best coverage through precise node positioning, actual conditions from field may prove challenging, leading to random distribution, which can negatively impact network performance metrics. Due to optimal node placement remaining an NP-hard problem, several non-deterministic solutions exist, providing near-optimal results [124, 51].

Our approach is to achieve near-optimal placement of relay nodes through community detection and centrality algorithms derived from network analysis. We called the algorithm performing these tasks SIDEWISE (from SocIally enhanceD WIreless Sensor nEtwork), which processes network data in order to detect communities, each of which will be supplied by a relay node. The sensor network has a hierarchical architecture, in which data is gathered from sensor nodes, and then conveyed by relay nodes up to the sink (master relay) nodes. Of course, the ideal topology would have all relays connected to the sink, which would impact the costs. Instead, our algorithm repeatedly tries to add edges to the minimum spanning tree and uses the centrality of the sink as a fitness function. The process continues until the centrality of the sink becomes 1, the process of network transformation being shown in figure 4.7, where the relay nodes are shown in red, with the sink node in a central, larger position.

The number of relays is dependent on two ad-hoc parameters: modularity resolution (which can vary between 1 and 0) and radius of wireless coverage. The resolution directly impacts the communities detected, for each community a relay being assigned. For the 1000-node topology exemplified in figure 4.7, the maximum resolution yields no less than 9 communities; decreasing the resolution shows an exponential increase for the number of communities identified, with their size decreasing accordingly. When the size of a

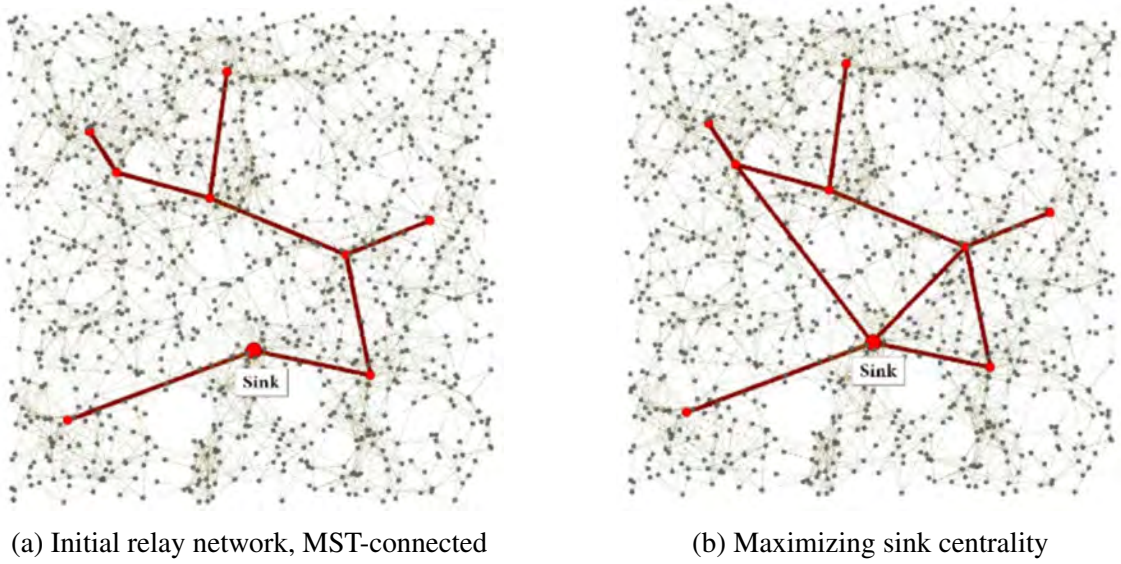


Figure 4.7: SIdewise optimization of relay node throughput [51].

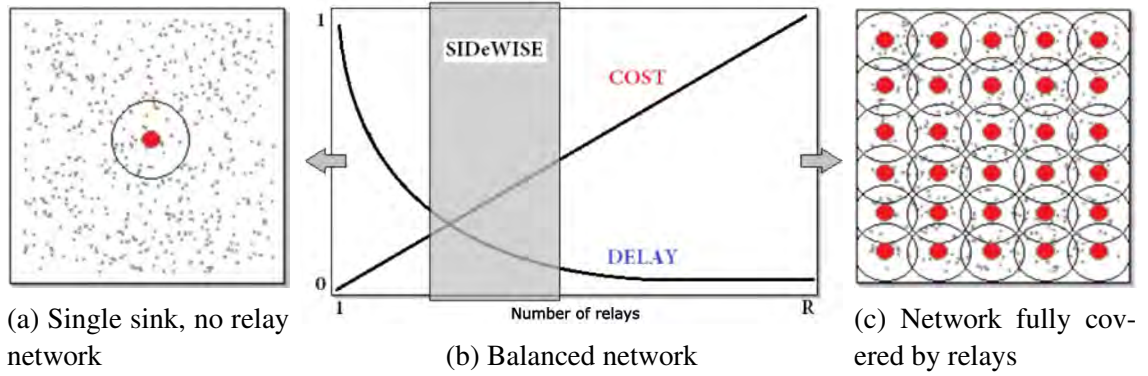


Figure 4.8: SIdewise optimization of cost and propagation delay [51].

community falls below a certain percentage λ of the network (the division factor), the respective community becomes irrelevant, and therefore being covered by a relay does not justify anymore. Of course, the number of relays not only determines costs, it also influences the propagation delay. Therefore, the ideal network has to find the optimal placement between costs and delays.

For our example network we have computed the extremes shown in figure 4.8. To the left (figure 4.8a) we have the lowest cost (one relay node, which is also the sink), but also the worst propagation delay (6.98τ); to the right (figure 4.8c), we have full coverage by relay nodes, which ensures best propagation delay (1τ), covered by 100 relays and sinks, which drives the cost up. The middle ground is represented by a linear increase in costs with the number of relays, while the propagation delay shows an exponential decreasing tendency. The optimal middle (figure 4.8b) determined by the SIdewise algorithm has a cost of 7 relay nodes and a propagation delay of 3.62τ , which represents

a 92% improvement in propagation delay over (figure 4.8a) with only 7% of the relays required by (figure 4.8c).

SIDeWISE represents a novel approach of making adjustments to the topology of a sensor network by employing a social perspective. The results indicate a logarithmic behavior for SIDeWISE, which is of particular importance for networks with large node numbers.

4.3.2 What Makes Urban Networks Social?

Modern transportation is rapidly evolving, with the infrastructure transitioning from passive systems (based on conventional traffic lights) to intelligent, interactive networks, in a preparation for the upcoming deployment of fully autonomous vehicles and experimental virtual drivers. Modern transportation systems allow bidirectional communication between vehicles and infrastructure, enabling real-time information exchange that influences decision-making. The next evolutionary step is developing infrastructure that autonomously responds to changing traffic conditions, making real-time adjustments to maintain efficient vehicle movement without human intervention.

But are there any other force that impact traffic, other than momentary, fortuitous conditions that are quite limited in time? It seems people purposes change during a day, which impacts on transportation usage [42, 43]. While work (and, possibly education) are primarily driven by professional obligations, the remaining types of travel are heavily influenced by social needs, motivated by the desire for human interaction and social engagement, as shown in figure 4.9a. Furthermore, since the transportation means of choice for the majority of travels is the car, as shown in figure 4.9b, this constitutes proof of the social fingerprint on urban transportation. Therefore, using social network analysis to better understand and improve the urban road network represents a natural step [117], where the SIDeWISE algorithm could also play a role in pointing the intersections that require optimization, as described previously in section 4.2.

In modern urban environments, expanding the road network seems difficult, if not impossible: geography limits the construction of new segments, while economy limits the opportunities for new passages and infrastructure overlays. However, a change of perspective from conventional optimizations [58] could prove beneficial by allowing a reconsideration of the network topology itself and the flexibility it allows.

One possible approach to optimizing urban traffic relies on improving road network

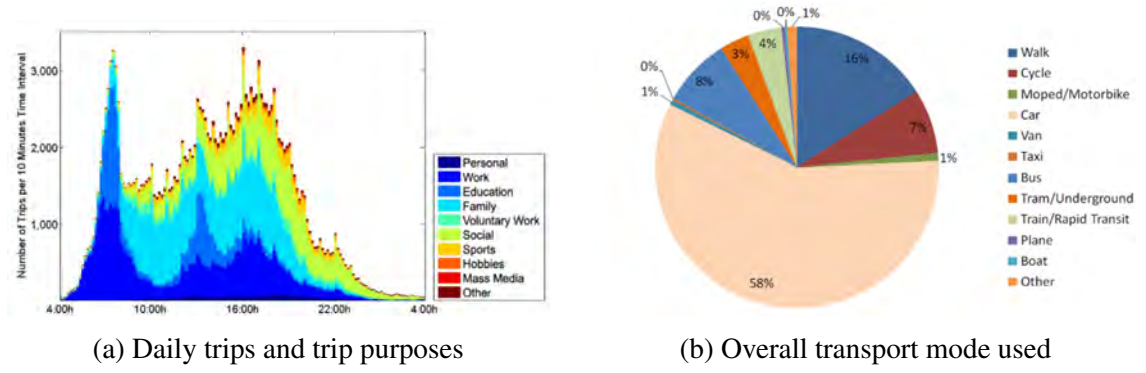


Figure 4.9: Social fingerprint on transportation [42].

topology by applying complex network analysis techniques. The urban road network is composed of intersections as nodes and streets as directed edges, which are then subject to a strategic street redirection strategy in order to create a more balanced distribution of network connectivity. This way, without building any additional road infrastructure, the traffic throughput can be maximized. Before any decision can be made, some network attributes have to be measured: network size (nodes and edges), average path length and degree, clustering coefficient, network diameter, density and modularity [75], distributions of the degrees, betweenness, closeness, and centrality. This analysis builds a network social snapshot in which the purpose is to perform an evening of the betweenness distribution in the graph, by applying community detection and centrality algorithms [117].

Key to this approach is to influence the betweenness centrality, that is, a node's influence in a graph; the higher the betweenness, the higher the influence of that intersection in the road network, thus highlighting intersections that may become, or are right now, affected by congestion. For our case study we chose Budapest, a large capital city halved by the river Danube and featuring variable terrain and landforms. Budapest has 1.7 million inhabitants and a surface area of 525 square kilometers.

We used OpenStreetMap [78] and Gephi [15] to process all road network data, then divided the road network into neighborhoods, with results being shown in figure 4.10. The road network graph comprises 12,038 nodes (corresponding to intersections) and 17,309 edges (corresponding to directions along streets). After applying community detection and sizing communities for different modularity values ($\lambda=1$ for community/district level and $\lambda=2$ for neighborhood level), the result can be seen in figure 4.10a. The communities can be regarded as smaller networks, for which our algorithm can be applied in parallel. Inside communities one can observe nodes have different betweenness values, influenc-

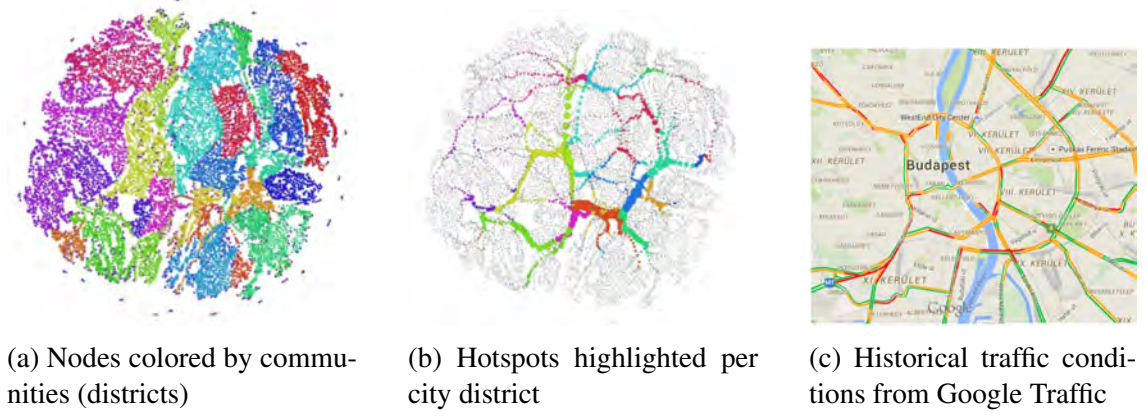


Figure 4.10: Road network of Budapest, with each node representing an intersection [117].

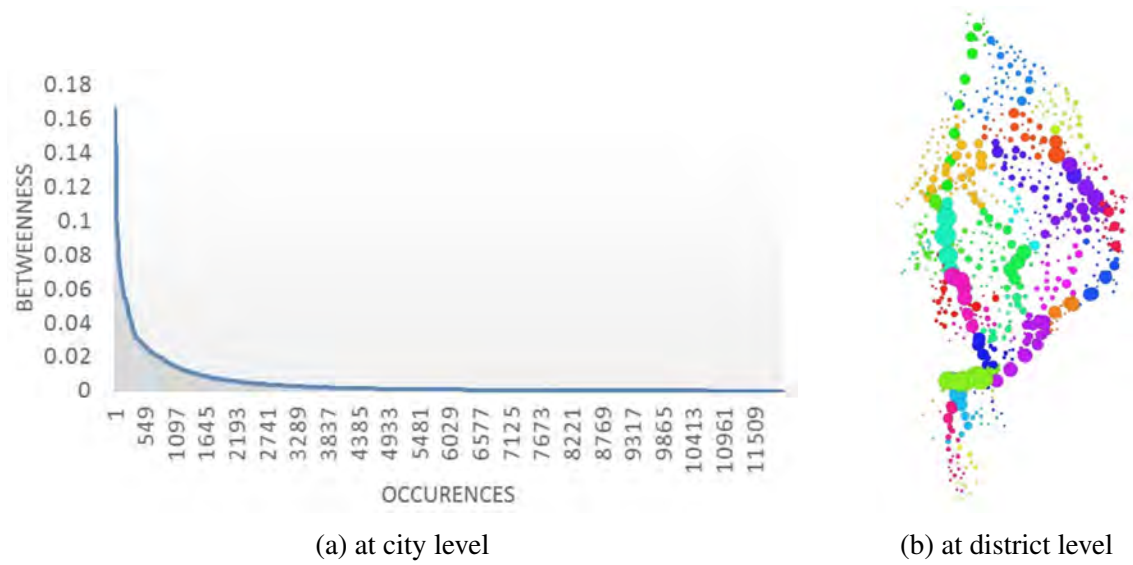


Figure 4.11: Betweenness distributions for Budapest [117].

ing the daily driving habits and choice of a route (a snapshot being given in figure 4.10c), which in turn produces tendencies for congestion, as shown in figure 4.10b.

The initial analysis of the road network graph for Budapest reveals a betweenness distribution (figure 4.11a) that closely resembles a power law, which is typical for social networks (and complex networks, at large). However, such betweenness distribution also indicates the intersections with a high influence over urban traffic and, therefore, prone to congestion. The algorithm we devised, called *SocialCity* [117] attempts to even the graph in figure 4.11a by distributing the importance of intersections and therefore resulting in a more even traffic. *SocialCity* implements a genetic algorithm whose purpose is to optimize betweenness through redistribution at multiple levels (neighborhood, district, and city levels), by adjusting street directions. The new urban road network, with recon-

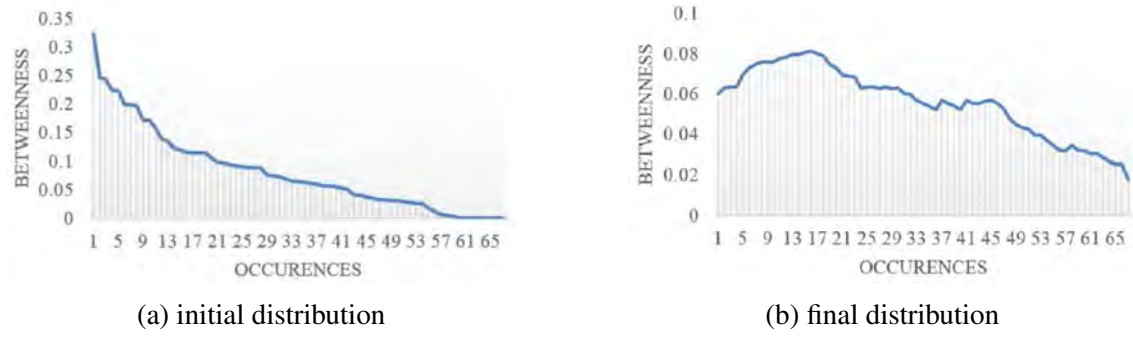


Figure 4.12: Betweenness for a district in Budapest, before and after applying *SocialCity* [117].

figured directions, without any new roads added, should lead toward reducing congestion or eliminating it altogether. In order to expedite our experiments, we ran *SocialCity* over a central district in Budapest, for which the initial analysis is described in figure 4.11b, with nodes colored by community and sized by their betweenness values.

To the best of our knowledge, redesigning the existing road network by adapting the street directions represents the first approach targeting the evening of the betweenness values. The results look good, starting from the initial distribution that looks similar to a power law (shown in figure 4.12a), *SocialCity* managed to produce a final distribution that certainly does not look like a power law, but much flattened, as shown in figure 4.12b. In order to measure, we approximate the graphs from figure 4.12 as linear functions, the resulting slopes being $s=-0.0035$ (figure 4.12a), and $s=-0.0008$ (figure 4.12b), respectively, which accounts for an improvement of 4.375 times. Looking for further validation, we ran *SocialCity* over other cities for which we found relevant data, with results (shown in table 4.4) showing improvements ranging from 2.62 to 4.67. A simplified interpretation would be to say that regions with heavy-loaded traffic were improved several times, which appears significant. Of course, *SocialCity* produces different results due to particular characteristics of each city, but the fact that improvements were possible to every city represents an achievement worth considering when optimizing a city's road network at small costs.

City	Nodes	Edges	Initial s	Optimized s	Improvement
Augsburg	6097	7929	-0.0028	-0.0006	4.670
Budapest	12038	17309	-0.0035	-0.0008	4.375
Constanta	2794	3994	-0.0040	-0.0011	3.630
Sibiu	2260	3056	-0.0021	-0.0008	2.620

Table 4.4: The *SocialCity* algorithm applied over different cities [117].

If a resettlement of the road network by changing street directions yields significant improvement (performed by the *SocialCity* algorithm), it is perhaps worth trying to squeeze any potential for additional improvements by refining the approach and taking into consideration not only street directions, but also available lanes. This extends the road network graph considered in figure 4.10, allowing our *SIGS* (for Social Intersection Genetic Shuffler) algorithm to attempt redistribution of traffic over road segments that are available for additional traffic [31]. For this purpose, we will use the same basic network metrics as before: network size (nodes and edges), average path length, clustering coefficient, average degree, network diameter, density and modularity, degree distribution, betweenness, closeness and centrality. We extended the analysis performed onto the city of Budapest to also accommodate two large cities from USA and Japan, namely Los Angeles and Sendai, presented by comparison in table 4.5.

City	Budapest	Los Angeles	Sendai
Nodes	12038	29759	42807
Edges	17309	44226	59224
Communities	147	122	336
Modularity	0.979	0.972	0.983
Average path length	30.175	73.044	47.094
Average degree	1.428	1.486	1.384
Population density	3.31	3.2	1.31

Table 4.5: Topological characteristics for 3 large cities [31].

A second city under analysis was Los Angeles, an urban metropolis with an extensive road infrastructure (figure 4.13a), comprising regular streets and freeways/highways spanning the entire area and providing superior connectivity, which will later be proven indirectly by our *SIGS* algorithm. Los Angeles is also significantly larger than Budapest,

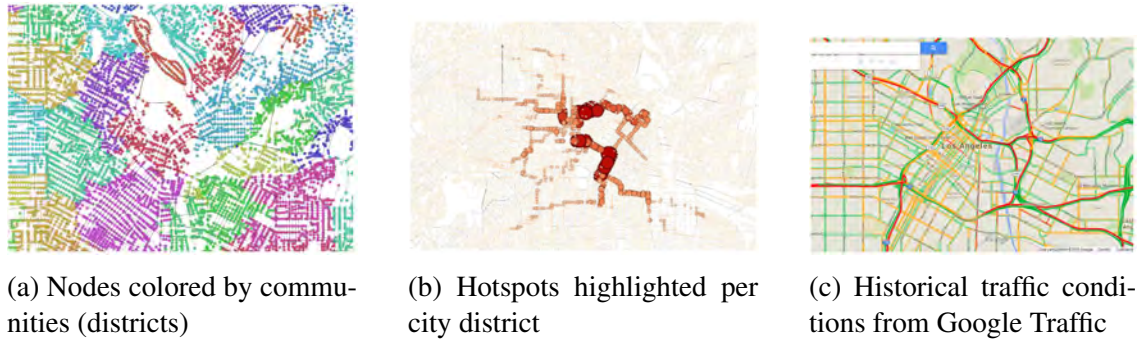


Figure 4.13: Road network of downtown Los Angeles, with each node representing an intersection [117].

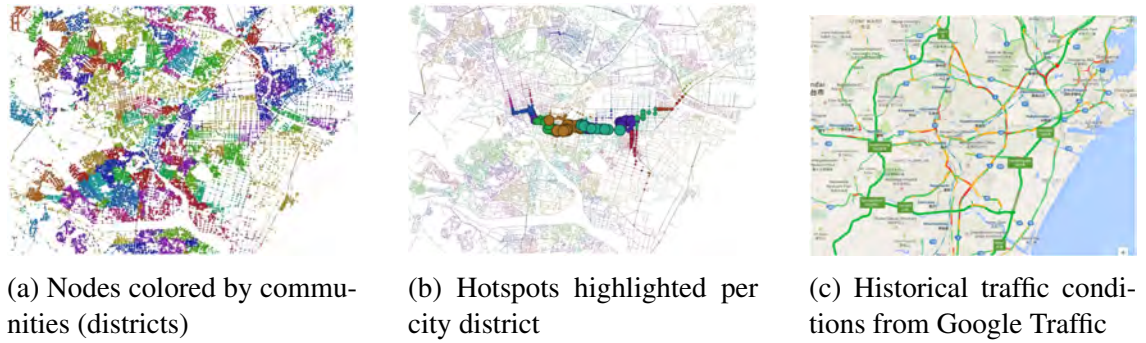


Figure 4.14: Road network of Sendai, with each node representing an intersection [117].

with 3.8 million inhabitants and a surface area of 1302 square kilometers, therefore bound to traffic jams (shown in figure 4.13b and figure 4.13c).

The last city in our analysis is Sendai, which is the smallest of the three by population count, with 1.1 million inhabitants, but the middle by surface, with about 786.3 square kilometers. Sendai makes for a particular case, with its high-efficiency mass transportation by over- and underground trains, thus allowing the road network to cope quite well with traffic challenges.

Since large cities have specific topologies and tend to be cleaved by natural or man-made features, we applied the *SIGS* algorithm over two different sets of data for each city, representing different parts, the results being shown in table 4.6. For Budapest, dataset 1 was covering the historical, central area, while dataset 2 referred to the financial district, both datasets yielding significant improvements. For Los Angeles, dataset 1 had 18% of its area covered by highways, which absorb a great part of the traffic, therefore leaving no overcrowding tendencies for the less important streets. For the second Los Angeles dataset, representing the downtown area, the street topology resembles a rectangular mesh, intrinsically resistant to congestion. Sendai represents a special situation, both

datasets showing a number of small communities (exhibiting high modularity and low clustering coefficient), which cope with internal traffic quite well due to an efficient public transportation, therefore preventing the *SIGS* algorithm from achieving cross-community optimization.

As a conclusion, the *SocialCity* and *SIGS* algorithms we developed represent useful tools for resetting and recreating the urban road network to better respond to traffic challenges. Improvements shown in both table 4.4 and table 4.6 could be viewed as equivalent to lowering the traffic load, which allows for some consistent provisions for the years to come, until new features or means of transportation will take over.

City	Initial s	Optimized s	Improvement
Budapest 1	-0.220	-0.0534	4.119
Budapest 2	-0.377	-0.0372	10.134
Los Angeles 1	-0.388	-0.132	2.939
Los Angeles 2	-0.172	-0.0714	2.408
Sendai 1	-0.118	-0.0917	1.286
Sendai 2	-0.184	-0.145	1.268

Table 4.6: The *SIGS* algorithm applied over different cities [117].

4.3.3 Deployment of Traffic Lights and Complex Networks

Shaping the urban road network was determined by continuous historical processes, which were contradicting at times, and therefore hurting its efficiency. While the primary role of traffic lights is to determine traffic to stop and go alternately along all available directions, thus constraining traffic flow to impose order, they also impact on driver's responses and habits [36]. This raises the crucial need of understanding not only how streets should connect (revisiting street directions was discussed in section 4.3.2) but also how they should interact, through a systematic analysis of the relationship between street patterns and traffic flow. Such an analysis could greatly help traffic network planning and identify key intersections that require traffic lights in order to maintain steady traffic flows: too many traffic lights can disrupt traffic and lead to slow-downs or congestion, while too few of them can be dangerous and lead to the same effects. We believe that lessons learned from complex network analysis could be applied for optimal deployment of traffic lights [50].

All intersections from the road network may implement traffic lights (semaphores), with each traffic being subject to local optima values in terms of traffic flow (level 1 in figure 4.3). However, communication and coordination between intersections are part of more complex processes (level 2 in figure 4.3) in order to target Pareto optimality at a global (network) level.

Our approach to maximizing traffic flow is based on establishing continuous communication and negotiation between intersections that play different roles (master and slave). Master nodes are determined dynamically at community level and are able to take decisions that impact slave nodes in terms of green phase durations.

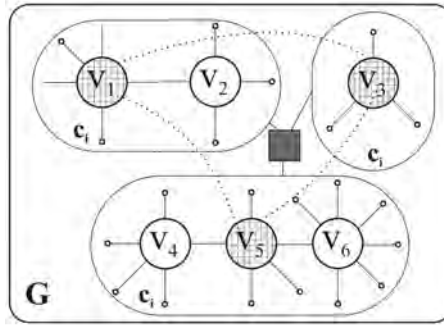


Figure 4.15: Hierarchical structure of intelligent traffic lights [50].

Let us consider the example from figure 4.15, where a simplified road network consists of 3 communities, with master nodes (V_1 , V_3 , V_5 , grayed-out) coordinating slave nodes in their respective communities. Such coordination serves the following purpose: master nodes assemble the population (level 3 in figure 4.3) that will be subjected to optimization by means of a genetic algorithm [23] in order to determine the best configuration for global traffic flow; this configuration would then propagate down to level 1 and drive traffic as discussed in section 4.2. However, for this plan to work, the master and slave nodes have to be identified: if a node does not implement a traffic light, such a traffic light should be added; if a node does implement a traffic light that is not necessary, then the respective traffic light can be removed (or made redundant) with no consequences for the traffic flow. In our view, master nodes serve communities, while slave nodes contribute by enforcing the policy decided by the master. When a master node decides to adjust its timing plan, it will notify all master nodes directly connected and wait for acknowledge messages confirming the new timing is achievable.

The *STiLO* (from Social Traffic Light Optimization) algorithm uses two parameters, the resolution *RES* for community identification and threshold *TRESH* value for ending

the recursion process. The number of communities identified is dependent on parameter *RES*, a graphical representation revealing an inflection point from where its impact decreases significantly. We have studied the cities of Augsburg, Bratislava, Cluj-Napoca, Iasi, and Timisoara, to better understand the effects of resolution on the number of communities, the results being shown in figure 4.16. The region containing the point of inflection is clearly visible, pointing to the optimal resolution beyond which only a few (if any) new communities are identified.

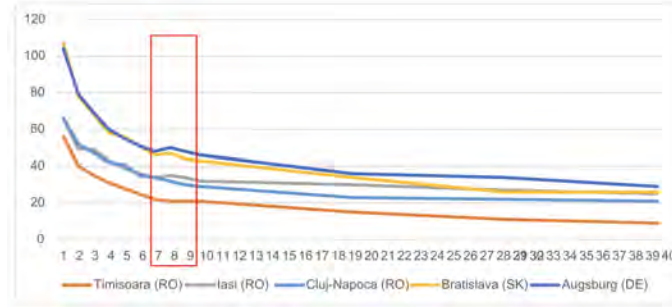


Figure 4.16: Resolution and communities for 5 cities [50].

After detecting communities, nodes are ranked based on their betweenness values, with the highest determining the master. The case study was performed over the city of Timisoara, its urban complexity being shaped by centuries of evolution and multiculturalism. The communities identified at the selected resolution are presented in figure 4.17, appearing to map quite well over the city's traditional neighborhoods (or quarters), which confirms features of complex networks and therefore opportunity for corresponding analysis.

Running *STiLO* over two distinct communities in Timisoara produces a relevant superposition with the quarters of Circumvalatiunii (figure 4.18), and Mehala (figure 4.19). The nodes colored in red (figure 4.18a and figure 4.19a) represent the master nodes, while each community may be recursively broken into smaller communities ((figure 4.18b and figure 4.19b). The Circumvalatiunii quarter holds the highest average degree (3.309), which indicates a large number of intersections and, therefore, a rich structure of sub-communities. Nodes shown in figure 4.18 and figure 4.19 have high betweenness values, indicating they have to support equally high traffic values that could turn in potential congestion spots. The *STiLO* algorithm provides a systematic match between intersections perceived as hotspots in daily traffic and master nodes that require optimization. This way, a hierarchical placement and assignment of traffic lights can be implemented as a network of master and slave nodes.

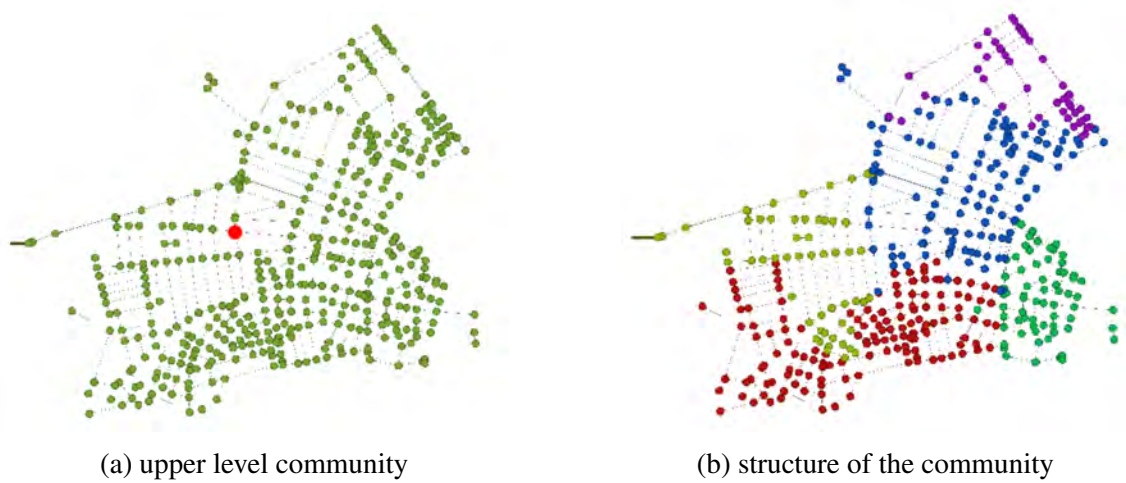


Figure 4.18: Recursive application of *STiLO* onto the community representing the Circumvalatiunii quarter of the city of Timisoara [50].

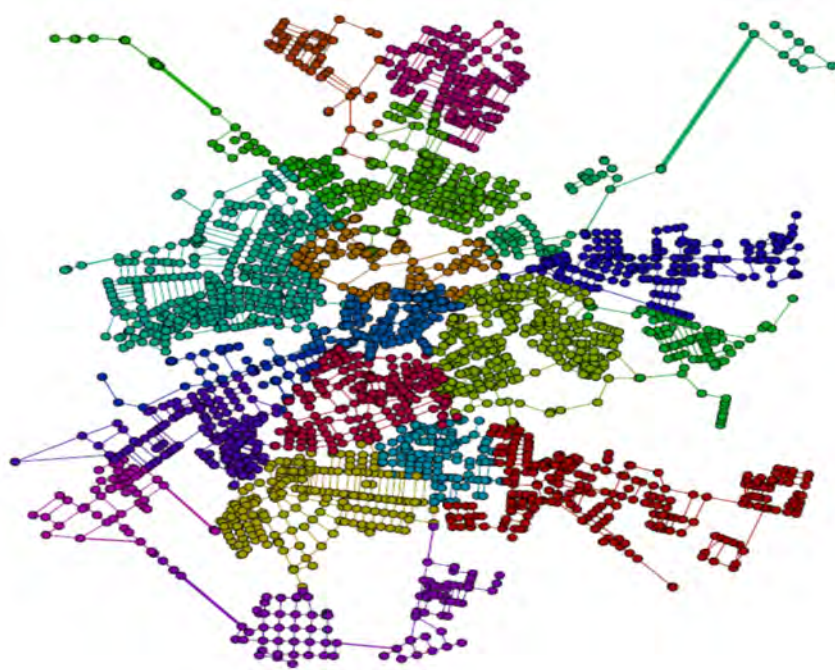


Figure 4.17: Urban network communities in Timisoara [50].

4.3.4 Tracing Congestion by Social Markers

Year by year, urban road networks grow increasingly busy, confronting with traffic jams becoming a daily routine for most commuters. Assessing queues in almost real-time is implemented in a variety of systems [17] and used to forewarn traffic participants in order to avoid certain hot spots where queuing intensifies, trips take longer, and vehicles travel at lower speeds (if any). Such conditions are part of defining congestion and it is generally acknowledged they can be addressed through adaptive traffic signal control

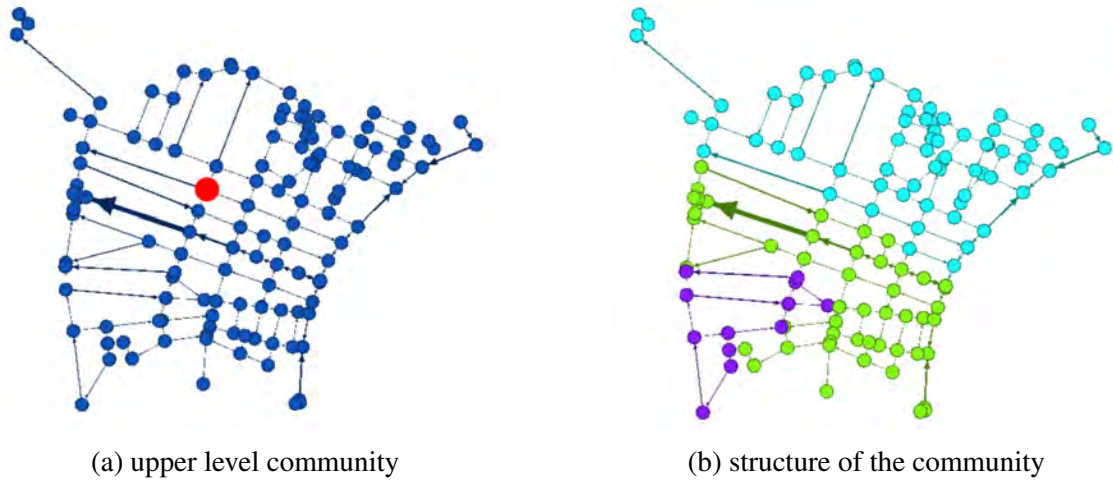


Figure 4.19: Recursive application of *STiLO* onto the community representing the Mehala quarter of the city of Timisoara [50].

strategies in conjunction with a variety of vehicular communications [41]. However, as pointed out by section 4.3, better understanding the social phenomena driving traffic could yield great benefits for understanding congestion and possible approaches for its optimization.

In order to provide a better argue and benefit from direct experience with the traffic, we decided to analyze the city of Timisoara, the heart of one of the largest metropolitan areas in Romania [10]. Our goal was to apply some of the concepts and tools from complex network analysis to identify intersections that are currently prone to congestion, as well as deriving measures that would prevent congestion emergence. Some correlations between network degree and centrality attributes and congestion probability have been made previously [46, 126], indicating that the more connected a node is, or the higher its betweenness centrality, the greater the probability it will be affected by congestion. The conclusion is that random and scale-free networks show more resilience to congestion than tree-like or regular networks, with valuable advice for the managers of the road network: nodes with higher betweenness should be accompanied by shorter green cycles, as they have the highest potential to upset traffic onto connected nodes with lower betweenness values.

The area under scrutiny represents one of the tech/business districts of the city (shown in figure 4.20a), which, as always, has experienced a rapid new developement over the existing road network for which there were no changes. As a consequence, the area can become quite busy and consistently affected by congestion during rush hours. Applying the community detection algorithm provided the results shown in figure 4.20b, with nodes

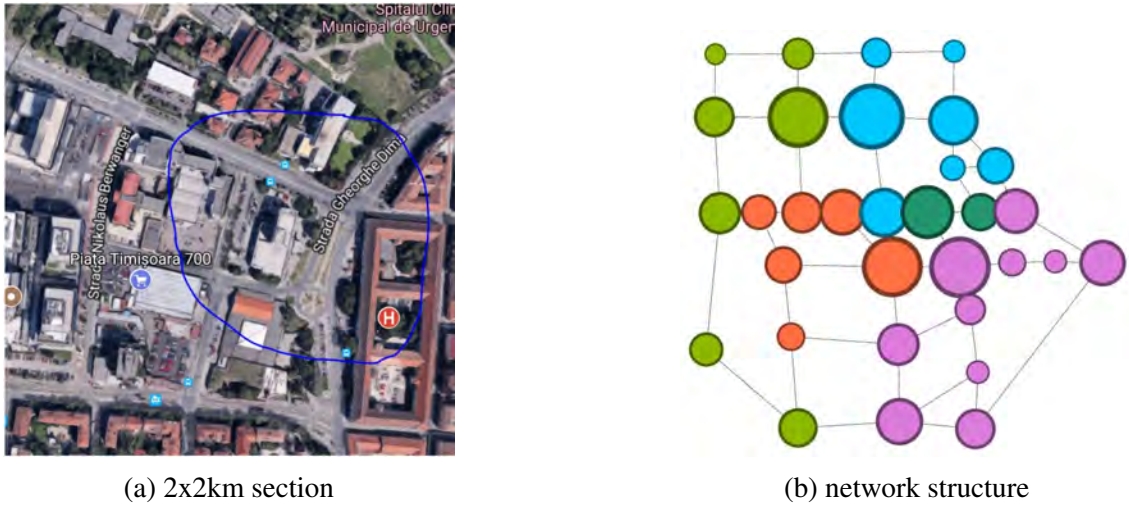


Figure 4.20: A tech/business district of Timisoara [10].

(corresponding to intersections) being scaled to their relative betweenness centrality and colored to highlight distinct communities. Out of all centrality metrics available, we employ the betweenness centrality because it reflects the number of shortest paths from the network that traverse a specific node, thus giving out the subjective importance of an intersection; such intersections tend to be chosen by many drivers, therefore increasing the likelihood of congestion. The topological characteristics highlighted in table 4.7 show an average degree that is specific to flat areas, and a high modularity showing a well-defined community structure.

Metric	Value
No. of nodes	4170
No. of edges	6212
Diameter	63
Modularity	0.938
Average path length	43.127
Average degree	1.44
Population density	2.91

Table 4.7: Topological characteristics for the city of Timisoara [10].

If the subjective experience in this area confirms congestion happening at rush hours, we would like to also derive it by using complex network analysis and VISSIM simulation. After calibrating the data for VISSIM, we plotted queue lengths and total number of stops for all the cars in the simulation, which serve as an indication for the current

state of operation of the network, with the ideal situation allowing for short queues and stopping times. In figure 4.21 the dotted line shows the queue length while the dashed line shows the number of stops for the simulation performed for a congestion-affected intersection (streets Gh. Lazar and Gh. Dima), with some regions of interest in different colors [10]. The green area (left) describe traffic before reaching the maximum capacity for the intersection, when traffic remains light still. The situation worsens in the yellow area, with traffic intensifying while the intersection barely keeps being permeable. Ideally, this would be the moment for a proactive system to respond to traffic dynamics and compensate in order to prevent congestion. However, the district under study does not implement any advanced ITS system, so the intersection naturally becomes congested, a situation marked with red; the congestion rapidly leads to a situation in which no more cars can enter the intersection, with traffic being effectively halted. The other green area in figure 4.21(right) shows the congestion clearing out, since cars already inside the intersection are finding their way out while no additional cars are able to enter still.

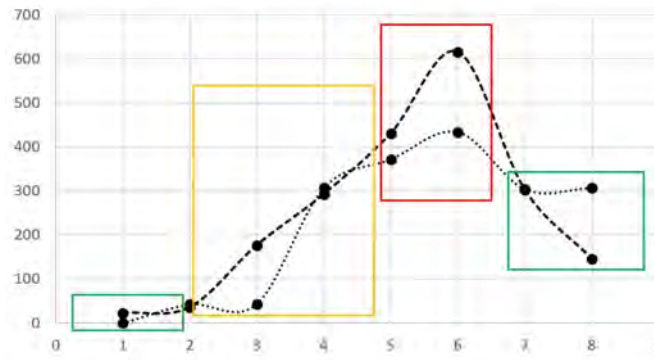


Figure 4.21: Queue lengths and number of stops [50].

After simulating the area of interest (figure 4.20a) in VISSIM, the *STiLO* algorithm was applied to identify relevant communities (figure 4.20b) and also pinpoint congestion-prone hotspots. The results presented in figure 4.22 indicate an unstable behavior of the intersection, matching the subjective opinion based on daily driving within the area. In figure 4.22a the total travel time oscillates (dotted line) while the average vehicle speed shows an almost constant value (dashed line) with a negative slope, suggesting a tendency for traffic worsening. Travel time for individual vehicles is also oscillating, indicating a congested intersection where waiting times for halted traffic are longest. An equilibrium can be reached when the vehicle platooning evolves to a slow movement crossing the intersection, with the effect of a pseudo-green wave. Establishing such equilibrium is sought by the *SIGS* and *SocialCity* algorithms to dynamically adapt lanes and streets to

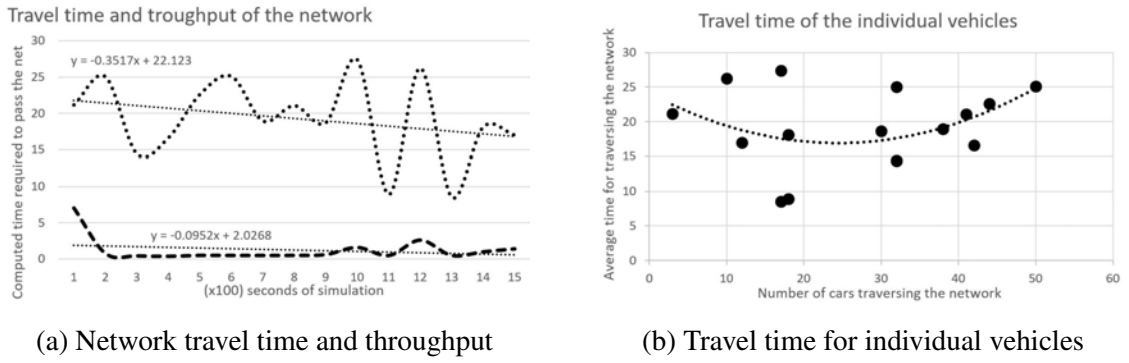


Figure 4.22: Simulation results showing oscillating behavior [10].

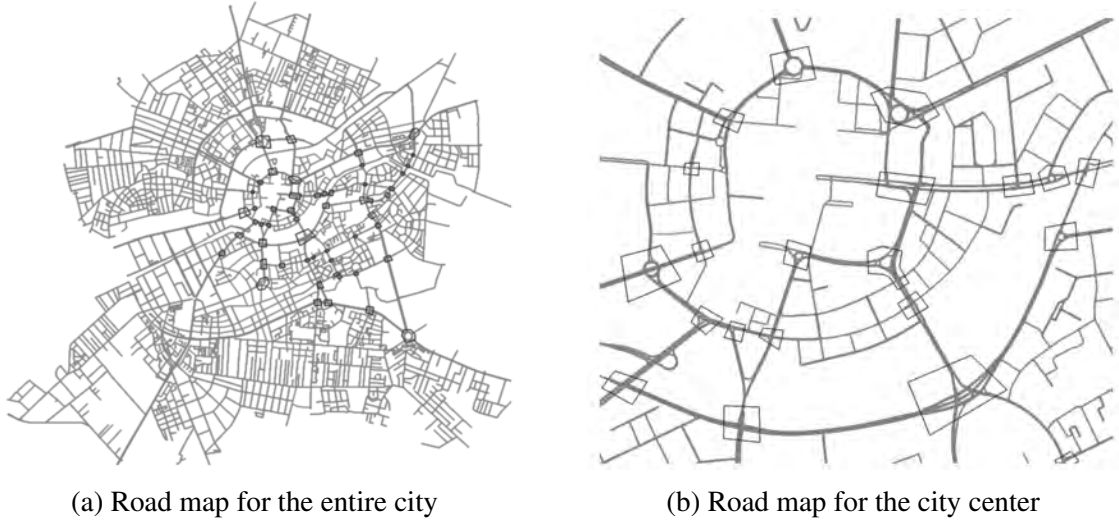


Figure 4.23: Overview of the road network for the city of Timisoara [6].

temporary increase capacity over some preferred directions.

While figure 4.20a depicts a 2x2km section, it would be worth extending the section and attempt to assess the entire city by means of simulation in VISSIM, the maximum allowed section being 10x10km. The complete road map for Timisoara can be retrieved online from OpenStreetMap, for which we then employed a custom Python script to create the complete network topology with all the intersections (nodes) and corresponding streets (edges). The main metrics of interest are available by using Gephi. The distribution of the betweenness centrality for the city of Timisoara, shown in figure 4.24a, indicates a power law distribution, which is specific to urban and other complex networks. We were interested in validating the subjective perception of the most crowded intersections by doing a city-wide simulation in VISSIM and investigating the top 100 intersections that are perceived as the most problematic (figure 4.24).

A clear perspective over the road map is given by figure 4.23, where the full city is shown in figure 4.23a and its center is shown in figure 4.23b, the intersections considered

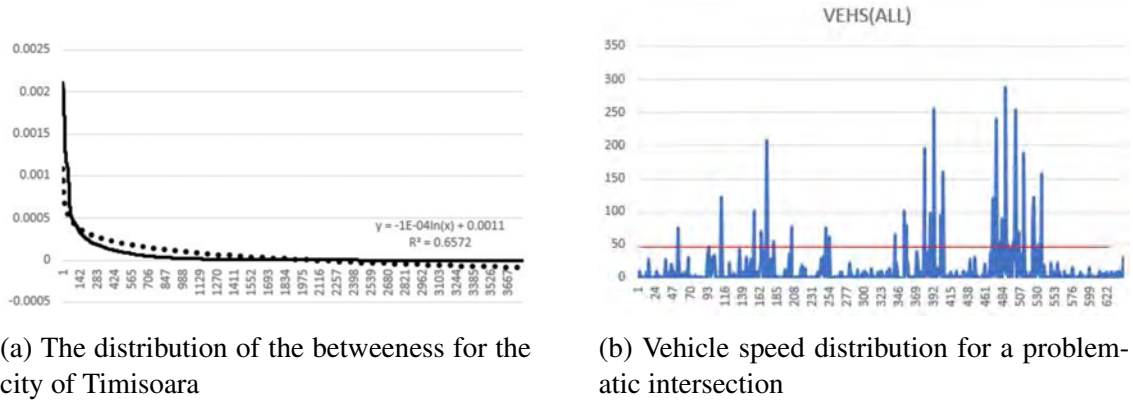


Figure 4.24: Case-study for the city of Timisoara [6].

for simulation purposes being highlighted as polygons. The simulations were performed with real data collected from traffic at different moments, which brings phenomena observed as close to reality as possible. Of course, real data was only available for a very limited number of intersections, chosen for their subjective importance and confirmed by the associated betweenness values. While simulating the entire city remains a challenge for the future, plotting vehicle speed for a certain intersection at different moments can point to what we socially perceive as being "bad traffic": figure 4.24b shows significant variation in speed for vehicles passing through the same intersection at different moments. At some particular moments, the intersection allows traffic to move freely, with higher speeds; however, most of the time the speeds recorded are slow, with strangled traffic during extended periods for the day.

The case-studies performed allowed the derivation of a workflow that uses a range of external tools and customized scripts to collect and parse online data describing the urban environment providing incentives for custom road traffic analysis and simulation (OpenStreetMap, VISSIM, Gephi) [10, 11]. Our results confirm a matching between the complex network analysis, traffic simulation and subjective experiences from real traffic concerning the intersections that are prone to congestion. These case-studies aimed to provide a different perspective and better understanding on how the road network topology impacts the likelihood of congestion in urban traffic. The motivation is manifold, the main target being to employ quantifiable metrics (from complex network analysis) and provide a formal description of congestion phenomena that is usually only subjectively assessed, so that a new perspective over means available to proactively manage ITS systems for maintaining traffic continuity is established. A second target offered by a better understanding of traffic would be designing a resilient road network [3] so that proactive

measures by taken to mitigate the effect of shortages caused by fortuitous events (accidents, repair sites) or even criminal activity (sabotaging traffic for intentional diversion).

4.3.5 Correlating Social Importance with Real Traffic Data

Analyzing the urban road network for the implementation, planning, and deployment of any intelligent transportation system relies on accurate identification of major intersections, which is often subjective and based not only on the geometrical complexity of the intersection but also on its social importance, which equates to driver preferences. Tracing the social markers has been discussed in previous sections (4.3.3 and 4.3.4), however the correlation between the subjective social importance of any intersection with data from real traffic remains to be analyzed.

Reliable acquisition of high quality data from real time traffic is not trivial, since existing systems are centralized, maintained by local authorities and therefore not open to public. Alternatively, some collaborative crowd-sourcing platforms have started to record such data, such as OpenStreetMap, Google Maps/Waze, or Here Maps. We chose Here Maps (<https://maps.here.com>) based on the features offered by this platform-as-a-service for location solutions and a successful prior connection to the company. Through the Here API, we were able to develop a custom Python script to record data every 5 minutes for the entire city of Timisoara [6]. A typical data sample is presented in figure 4.25, with attributes explained in more detail in table 4.8. Street classification is also acquired from the Here platform, while the road topology is processed from raw data coming from OpenStreetMap.

```
Time_stamp,Main_street,Secondary_street,QD,SU,CN,SP,JF,Temperature,Description,Pressure
2018-05-20T14:03:51Z,DN58,Anina,+,41.0,0.7,41.0,1.01587,23.66,Clear,1017
2018-05-20T14:03:51Z,DN58,Carasova,+,48.0,0.7,48.0,1.203,23.66,Clear,1017
2018-05-20T14:03:51Z,DN58,Doman,+,53.01,0.7,53.01,1.10599,23.66,Clear,1017
2018-05-20T14:03:51Z,DN58,Strada 1950,+,39.0,0.7,39.0,1.79001,23.66,Clear,1017
2018-05-20T14:03:51Z,DN58,Strada Spitalului,+.21.2,0.71,21.2,3.14159,23.66,Clear,1017
```

Figure 4.25: Data sample for traffic conditions on each of the crossroads [6].

Value	Interpretation
QD	Queuing Direction - value that can be "+" or "-", indicating the direction of traffic queuing corresponds or is opposed to the driving direction. Traffic queuing is considered a marker of congestion.
SU	Speed Uncut - value indicating the average speed for the road segment. The speed limit for the road segment is ignored, all speeds are included in the average.
CN	Confidence Number - value between 0.0 and 1.0, indicating the percentage of real time data included in the speed calculation. A value greater than 0.7 and less than or equal to 1.0 indicates real time speeds.
SP	Speed - value for the average speed for the road segment. Speeds over the limit are included, but trimmed down to limit value.
JF	Jam Factor - value between 0.0 and 10.0 describing the expected quality of travel. As the value increases, the quality of travel degrades, 10 representing road closure (or full traffic stop). As an exception, value -1.0 indicates that a Jam Factor could not be calculated.

Table 4.8: Quantitative data retrieved from the Here Maps API for each intersection [6].

With data being collected and assembled, our aim was to derive some quantitative representation to be correlated next with relevant attributes from complex network analysis that would express the importance of each intersection. More precisely, we targeted a comparison between our algorithm-derived representation (see figure 4.26), called *Jam Score*, and a specific attribute for congestion provided by Here Maps under the name of *Jam Factor* (as shown in table 4.8) [6]. Here Maps assigns a Jam Factor value for each intersection, therefore providing an instant value and a history record; these are essential for characterizing the quality of traffic, which is in direct conjunction with the average speed that can be maintained over a certain street segment. Within the city of Timisoara, the default limit is 50km/h, with very few segments where the limit is slightly higher (60km/h), which makes a steady speed of about 35km/h perfectly tolerable and equivalent to a congestion-free road segment. In terms of values for the *Jam Factor*, this equates to values under 3, whereas values above 3 indicate the presence of some form of congestion.

From our experience, *Jam Factor* values above 3 in a 24-hour period match the

hotspots around the city, or intersections that are often crowded or congested and constitute ideal picks for any optimization efforts. By applying our algorithm, shown in figure 4.26, the *Jam Score* is computed by integrating the *Jam Factor* weighed by the corresponding confidence value, also reported by the Here platform. As a result, we manage to build a reliable description of the urban traffic tendency for longer periods of time starting from its momentary snapshots.

```

A: Intersection data retrieval
1 :FOREACH Samplei WHERE Samplei.IID = IID DO:
3 : IF Samplei.timeStamp ∈ [timeStart, timeStop] THEN
4 :   JFI ← JFI + Samplei.JF
5 :   SUI ← SUI + Samplei.SU
6 :   CFI ← (Samplei.CF >= 5)? + 1 : -1
7 :   CNT ← CNT + 1
7 : ENDIF
9 :ENDFOR

B: Intersection JamScore computation
10 :JamScore(IID, JFI, SUI, CFI, CNT){
11 : IF (CFI > 0.5 * CNT) THEN
12 :   IJS ← JFI/CNT
13 : ELSE
14 :   IF (SUI/CNT >= 35) THEN
15 :     corr ← SUI/CNT
16 :     IJS ← IJS + 10/corr
17 :   ENDIF
18 : ENDIF

```

Figure 4.26: Deriving the intersection congestion factor Jam Score using Here Maps API [6].

We have run the algorithm described in figure ?? for a period of 48 days during the months of April and May 2018, the results being recorded in figure 4.28 and looking quite similar to those from figure 4.24a. Differences arise from the fact that betweenness values result from complex network analysis as mathematically computed function values with no dependence on time, whereas the *Jam Score* values represent an averaging of actual traffic conditions over a significantly long period of time.

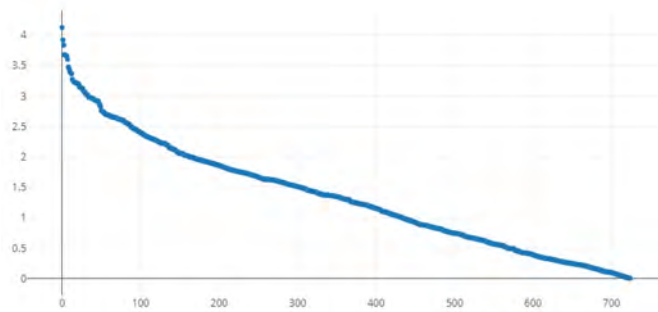


Figure 4.27: Distribution of the JamScore for the city of Timisoara [6].

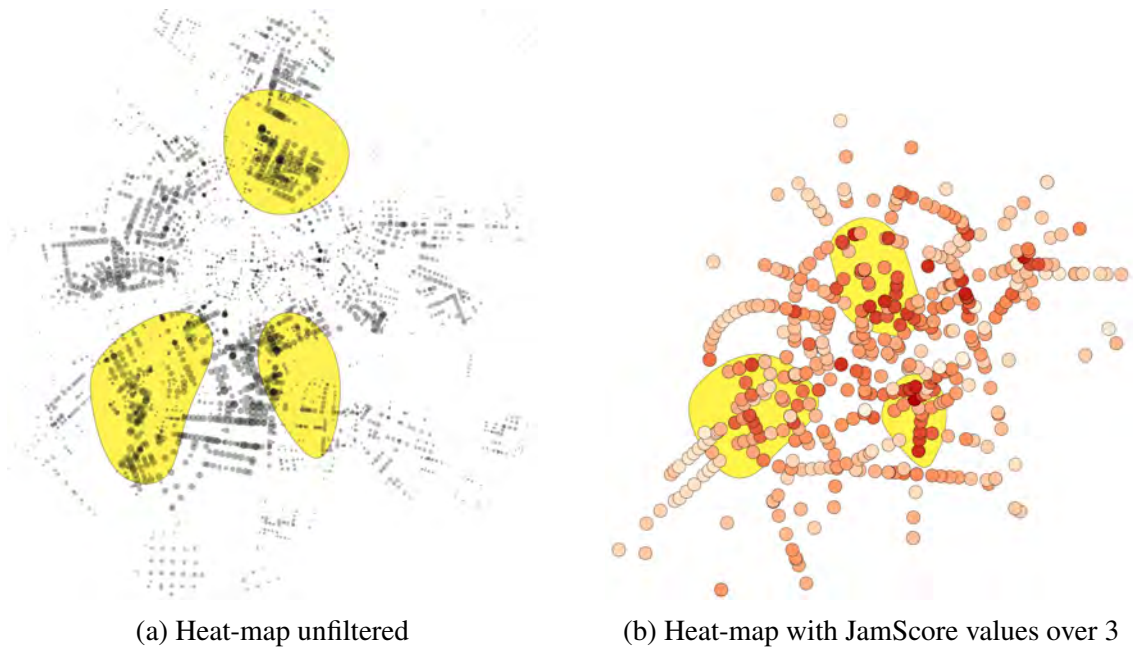


Figure 4.28: JamScore values for the city of Timisoara [6].

Building a heat map for *Jam Score* values with a city map overlay, shown in figure 4.28 presents the results in a more visually informative way. In figure 4.28a we have each intersection shown as gray shade indicating its value, with 3 major areas being constantly affected by congestion. The area marked to the north corresponds to the neighboring dorm-village of Dumbravita (up) and one of the city's shopping mall, which extends to a large residential and commercial area spanning across 4 major road segments, where any traffic disturbance translates quickly to queues and delays. The area marked to the south-west corresponds to other major road segments, introducing some speed and traffic restrictions as they pass under a rail bridge, also routing commercial heavy traffic towards E70 and the other city's shopping mall. The area marked to the south-east represents a concentrated business area, headquarter of a large employer, Continental Automotive. Frequent traffic jams and delays are recorded periodically during the morning and afternoon rush hours. In figure 4.28b we have filtered *Jam Score* values below 3 in order to show the most significant hotspots within the city.

By introducing the *Jam Score* we succeeded in introducing a numerical equivalent of the degree of congestion of an intersection and performed a correlation with the values indicated by the complex network analysis and the real life driving experience. Once again, we were able to confirm that intersections prone to congestion show high values for betweenness and the *Jam Factor*.

4.4 Adding Informational Plasticity with Machine Learning

Finding biological inspiration in nature has been serving computer science well, along all its three defining axes, as shown in figure 4.1. In particular, the epigenetic axis represents a valuable host for leveraging the potential of digital systems for endowing them with the gift of knowledge, either innate or acquired. The epigenetic dominant feat, represented by the ability of learning, can be used to enhance the abilities of a system to perform difficult tasks autonomously. Directions approached include the automatic detection of violence in real-time video streaming and gaining the ability to optimize and forecast traffic conditions based on current developments. Our approaches on both directions will be addressed shortly.

Additionally, an enticing avenue is fueled by the fact that epigenetic phenomena could be used not only to perfect digital systems, but also to use them as a helping tool when loss of epigenetic functions occur in biological organisms. Significant results on this research direction will also be presented next.

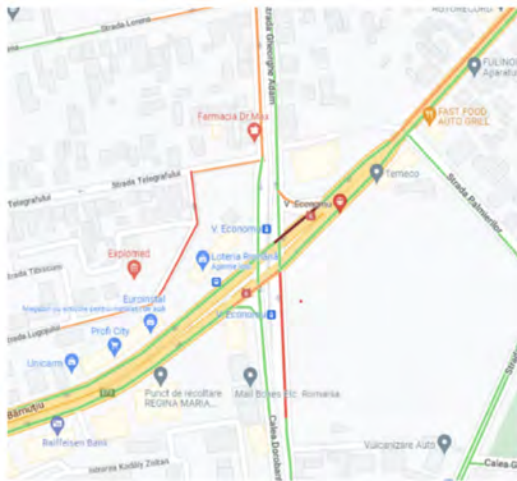
4.4.1 Intelligent Traffic with Epigenetics

Bio-Inspired Intersection Optimization

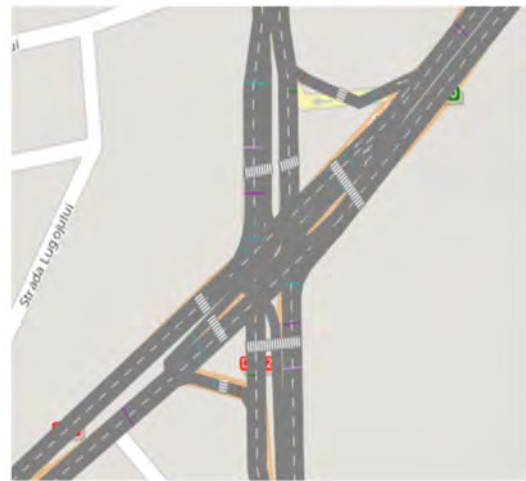
In section 4.2 we discussed some of the conventional approaches and provided a method for optimizing intersections as a first step in a global traffic optimization strategy. However, bio-inspiration offers a wide range of algorithms and methods and may serve as a powerful optimization tool. For this purpose, we employed evolutionary computation, which starts from an initial candidate solution and continues the search for an acceptable solution by refining it through stochastic recombination and mutation. A particular case is called *swarm intelligence*, which has been used previously in urban traffic optimization by adjusting signal timing within an intersection to minimize stops and delay time [Fu2021, NGUYEN2021, 17].

Specifically from the swarm intelligence family, we chose to experiment with the *Firefly* algorithm, known to perform well against other metaheuristic algorithms [Yang2012]. The Firefly algorithm is based on the observed mating behavior of the tropical fireflies, and implements a few simple rules [Abohashima2021, SABAR201745]:

- two fireflies can be attracted to each other;



(a) High level view provided by Google Maps



(b) Corresponding model in Vissim

Figure 4.29: A busy intersection in Timisoara: Calea Dorobantilor - Gheorghe Adam [112].

- the attraction between fireflies is directly proportional to their individual light emission and inversely proportional to their distance;
- any firefly must move towards the brightest firefly, otherwise, the movement is random.

We implemented the Firefly algorithm in Matlab (<https://www.mathworks.com/products/matlab.html>) and used data provided by the Vissim for a crowded intersection in the city of Timisoara, shown in figure 4.29. The experiments were performed on the available ASUS ROG STRIX G712LU laptop, with Intel I7 CPU at 2.6 GHz and 8GB RAM running Windows 10 Pro. The intersection signaling was considered as input data for the Vissim simulation, which provided corresponding values for fuel consumption and exhaust gases. The fitness function was calculated as the inverse of the sum of fuel consumption and exhaust gases values. Seeking a minimal footprint of fuel burnt and exhausts released through the intersection, the initial and final signaling are presented in figure 4.30. Encodings were used as T=time, Lt=Left, Fw=Foreward, Rt=Right, with traffic light colors designated as R=Red, G=Green, Am=Amber, i-G=intermittent Green.

T	Current plan												Optimized plan											
	Road 1			Road 2			Road 3			Road 4			Road 1			Road 2			Road 3			Road 4		
	Lt	Fw	Rt	Lt	Fw	Rt	Lt	Fw	Rt	Lt	Fw	Rt	Lt	Fw	Rt	Lt	Fw	Rt	Lt	Fw	Rt	Lt	Fw	Rt
0	R	R	i-G	G	G	R	R	R	R	R	R	R	R	R	i-G	G	G	R	R	R	R	R	R	R
22	R	R	i-G	Am	Am	R	R	R	i-G	R	R	R	R	R	R	Am	Am	R	R	R	R	R	R	R
26	R	R	i-G	R	R	R	R	R	i-G	R	R	R	R	R	R	R	R	R	R	R	R	G	G	G
30	R	R	R	R	R	R	R	R	R	G	G	R	R	R	i-G	R	R	R	R	R	i-G	G	G	G
48	R	R	i-G	R	R	R	R	R	i-G	G	G	R	R	R	i-G	R	R	R	R	R	i-G	Am	Am	Am
58	R	R	i-G	R	R	R	R	R	i-G	Am	Am	G	R	R	i-G	R	R	R	G	R	i-G	R	R	R
62	R	R	i-G	R	R	R	R	R	i-G	R	R	Am	R	R	i-G	R	R	Am	R	R	i-G	R	R	R
64	R	G	G	R	R	R	G	G	R	R	R	R	R	R	i-G	R	R	R	R	R	i-G	R	R	R
86	R	Am	Am	R	R	R	Am	Am	R	R	G	R	R	R	i-G	R	R	G	R	R	i-G	R	R	R
90	R	R	i-G	R	R	R	R	R	i-G	R	R	Am	R	R	i-G	R	R	Am	R	R	i-G	R	R	R
92	G	R	i-G	R	R	G	R	R	R	R	R	G	G	R	R	R	R	G	G	R	R	R	R	R
101	Am	Am	R	R	R	Am	R	R	R	R	R	G	Am	R	R	R	R	G	Am	R	R	R	R	R
105	R	R	i-G	R	R	R	R	R	i-G	R	R	G	G	R	R	R	G	G	R	R	R	R	R	R
120	R	R	i-G	G	G	R	R	R	i-G	R	R	Am	G	R	R	R	Am	G	R	R	R	R	R	R
142	R	R	i-G	Am	Am	R	R	R	R	R	R	G	i-G	R	R	R	R	G	i-G	R	R	R	R	R
146	R	R	i-G	R	R	R	R	R	i-G	R	R	Am	i-G	R	R	R	Am	i-G	R	R	R	R	R	R
150	R	R	R	R	R	R	R	R	R	G	G	R	R	R	i-G	R	R	R	R	R	i-G	G	G	G
168	R	R	i-G	R	R	R	R	R	i-G	G	G	R	R	R	i-G	R	R	R	R	R	i-G	Am	Am	Am
178	R	R	i-G	R	R	R	R	R	i-G	Am	Am	R	G	i-G	R	R	R	R	G	i-G	R	R	R	R
182	R	R	i-G	R	R	R	R	R	i-G	R	R	G	G	R	R	R	G	G	R	R	R	R	R	R
184	R	G	G	R	R	R	G	G	R	R	Am	Am	R	R	R	Am	Am	R	R	R	R	R	R	R
206	R	Am	Am	R	R	R	Am	Am	R	R	R	R	R	R	i-G	G	G	R	R	R	i-G	R	R	R
209	R	R	i-G	R	R	R	R	R	i-G	R	R	R	R	R	i-G	G	G	R	R	R	i-G	R	R	R
212	G	R	i-G	R	R	G	R	R	R	R	R	R	R	R	Am	Am	R	R	R	R	R	R	R	R
231	Am	R	i-G	R	R	Am	R	R	i-G	R	R	R	R	R	R	R	R	R	R	R	R	G	G	G
234	R	R	i-G	R	R	R	R	R	i-G	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
240	R	R	i-G	G	G	R	R	R	i-G	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Figure 4.30: Signaling plan for the intersection, before and after applying the Firefly algorithm [112].

Our experiments show the best solution seemingly having been generated quite early, during the second generation, as shown in figure 4.31, which we believe is a consequence of initialization the Firefly with the existing signaling for the intersection, which is supposedly already been optimized by the city planners. Instead of initializing from a random choice of parameters and let the Firefly algorithm work in time, our experiment attempted a jump-start by using the existing signaling in order to evolve a better solution.

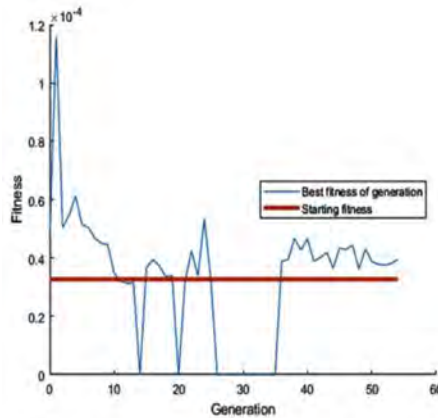


Figure 4.31: Best fitness values [112].

The values describing the impact of signaling are shown in table 4.9, so a direct comparison between the existing and Firefly-derived solution can be made. Our result shows significant improvements of overall values for CO (carbon monoxide), NO_x (nitrous oxides), VOC (volatile organic compounds) and fuel, which we to be worth bearing the slight increase in the overall queue. It is difficult to assess the impact of such an opti-

mization applied city-wide, since our experiment completely decoupled the intersection from the network, where Pareto optimality is bound to constrain local optimization possibilities. However, providing local optimizations should be possible, sometimes providing unexpected and significant impact over fuel usage and exhaust emissions, which certainly impacts the urban environment.

Solution	CO	NOx	VOC	Queue	Fuels
Existing	21430	4170	4967	229	307
Best	6051	11177	1402	245	87
Improvement	71.76%	71.77%	71.77%	-6.99%	71.66%

Table 4.9: Metrics describing the signaling solutions [112].

Predicting Traffic Congestion

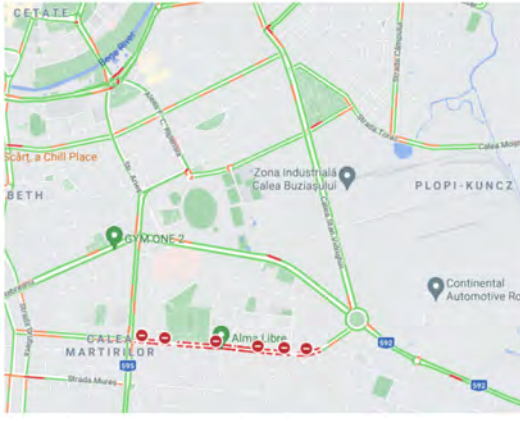
Urban traffic faces continuous pressure from its very dynamics, but in recent years a new component arises to augment its effects: not only the vehicle number is increasing, but new technologies are being adopted, leading to new urban mobility (a variety of scooters and their derivatives) with its pinnacle represented by the arriving of autonomous vehicles. New urban mobility is equivalent to small, usually personal, vehicles, imposing reduced speeds to other participants. Similarly, autonomous vehicles do not warrant a speed increase; on the contrary, the absence of the human factor means slower response in order to mitigate any possible hazardous reaction. Under these circumstances, keeping in mind that expanding the existing infrastructure offers limited avenues, the only response to the increasing pressure is represented by the advent of the *smart cities*, a paradigm that combines an efficient vehicular infrastructure with precise coordination brought by information systems. Providing a closely coupled match between the two is a slow, trial by error process, whose maturing is bound to experience evermore frequent congestion. Perhaps technology can be used to only to coordinate traffic, but also to predict its tendencies in order to provide faster responses, more adequate to local specifics.

It is possible that by using data gathered from real-life traffic, the concepts of machine learning could be put to work for identifying patterns and predicting traffic congestion caused by a range of possible anomalies, ranging from erratic driver behavior to accidents. If data is available, comparing the prediction against reality should be possible, which leads us to a case study over the city of Timisoara [7].

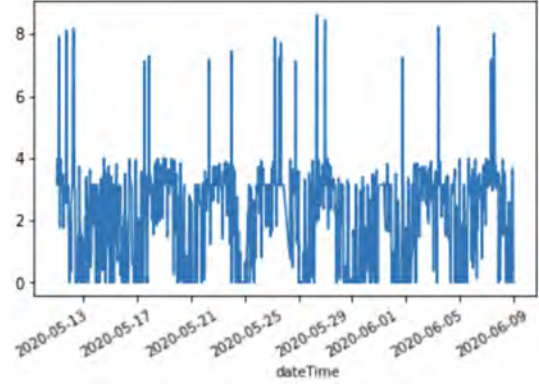
Predictions can rely on real-time or historical data, therefore the literature mentions a variety of approaches based on either or both. Modeling traffic to allow predictions has been done using a variety of algorithms, methods and tools, mixing statistical analysis, Markov chains and neural networks [68, 99, 64, 105]. Some strategies involve gathering real-time traffic data to derive strategies for regulating flow to avoid the rise of congestion, which require significant resources and therefore difficult to implement. A simpler, more straightforward way to provide traffic predictions would be to understand the ripple effect of anomalies, such as accidents or special situations, such as those imposed by the COVID-19 period.

The hardware used for training the model was a lab workstation with Gen 8 Intel i3 quad core at 3.6 Ghz boost frequency, equipped with 24GB RAM and a Nvidia GTX 1080Ti GPU. We used Tensorflow and Keras (<https://www.tensorflow.org>), together with other libraries such as NumPy (<https://numpy.org>), Pandas (<https://pandas.pydata.org>) and Scikit-Optimize (<https://scikit-optimize.github.io/stable/>). For our case-study we used the insight provided by Here Maps within the periods between 15th of May and 8th of June 2021, marked by the COVID-19 restriction measures taken nation-wide, which resulted in reduced traffic, and the upcoming wave of eased restrictions and increased traffic flow starting 1st of June. Data was sampled every 15 minutes and included all the important streets from the city of Timisoara [7]. Of course, some streets are more susceptible to develop congestion than others, the street of choice for our study being Stan Vidrighin, one of the important urban arteries connecting the south with the center and east parts of the city. Despite being a multiple-lane street that provides for smooth traffic, many vehicles using this street daily leads to sensitivity to anomalous events, such as accidents, vehicle failure or erratic driving, shown in figure 4.32a.

Our main goal was to allow prediction for traffic flow congestion for up to one hour (given by Here Maps under the form of Jam Factor table 4.8), which was then to be compared with actual data recorded. The actual measurements shown in figure 4.32b show some spiking variations indicating localized traffic jams. Our predictions (in orange) are shown together with real data recordings (in blue) in figure 4.33, indicating a good fit for baseline values and ability to indicate tendencies of increased traffic and possible jams; however, our predictions were not able to match the extent of the spikes seen in figure 4.32b. The correlation between the predicted values and the actual data resulted at



(a) Localized traffic jams



(b) Jam Factor data

Figure 4.32: Daily status for Stan Vidrighin street [7].

0.673, a value slightly below 0.8, considered to be a threshold to indicate a strong connection between the two data sets. We believe this value can be considered close enough (at 84%) to consider these results relevant. Further development to improve the prediction accuracy should include more training data, as well as an increased sampling rate.

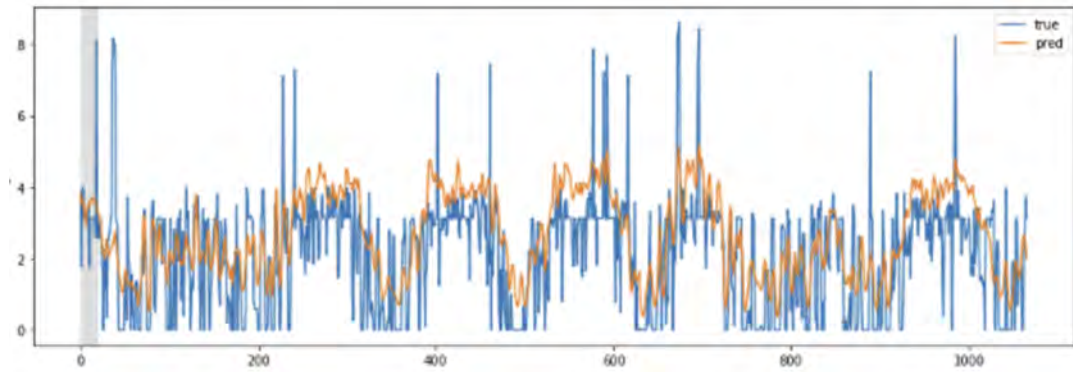


Figure 4.33: One hour Jam Factor predictions for Stan Vidrighin street [7].

4.4.2 Applied Epigenetics for Violence Detection in Video Streams

A palpable effect of the digital age is represented by the omnipresent spread of sensing devices, from temperature and pollution sensors to surveillance cameras. As a natural consequence, the vast amount of such active devices (about 770 million CCTV cameras operational in 2021 worldwide [16]) translates into challenges related to their purpose and management, leading to the resulting data streams potentially being used to infer higher meanings, ranging from crime prevention to traffic monitoring and observing environments unsuitable for humans. The evidence of violence and violent behavior date back from the dawn of human civilization and still constitute dark components of our society.

As a contribution to violence mitigation efforts, our approach was to provide automatic identification of violent acts from live video streams by employment of a deep learning model onto a cost-effective, resource constrained platform [121].

Generally, automatic identification in video streaming implies detecting events that are rare/unexpected, which is commonly known as *anomaly detection*. Automatic anomaly detection requires adequate datasets and extensive training, violence detection receiving significant research efforts for the assembly of relevant datasets, such as RWF-2000 [25] and UCF-Crime [111]. Following the build of an appropriate dataset, a Deep Multi Instance Learning (MIL) Ranking Model was developed that considers anomaly detection as a regression problem and employs a multiple instance ranking objective function that differentiates between normal and anomalous events by building a chart by assigning them different scores [111]. While this used a CNN (convolutional neural network), a next step was to replace it with a ResNet (residual neural network) and use the videos split into separate groups, containing normal and abnormal instances [37].

Sensing gestures and understanding the course of action automatically is difficult and computationally intensive, so using optical flows [25] or any other criteria to narrow the time frame where violence could happen is much desired. Performance was improved through the implementation of a SepConvLSTM (separable convolutional long short-term memory), using a separate convolution of each channel with one filter followed by the recombination of information across channels, which was considered efficient from both computational and memory perspectives [52]. Our analysis indicated this architecture to provide the best results, so it was only natural to attempt a comparison with our solution.

For our implementation we decided to use a type of metric learning, called Siamese Neural Network [20], where two identical sub-networks are joined at their outputs, providing two signatures that are then measured by their distance; with a modified loss function, this type of network was previously proven to work well for anomaly detection [113, 115].

Processing time-series data (such as videos) can produce better results if focus is given to the relevant periods; for our purpose, violence is more likely to appear in certain frames, such as those containing 2 people close to each other, possibly with fast movements. Our proposed architecture is shown in figure 4.34, with backbone models used for experiments being Mobilenet v2 [100] and DenseNet [47] and trained on the ImageNet dataset [35]. The model consists of two streams, *Feature stream* (describing the scenario,

in red) and *Feature difference stream* (capturing temporal changes, in green), which are then combined by the Merge layer, the whole sequence being learned with the help of a single long short-term memory (LSTM) or gated recurrent unit (GRU) cell.

Our hybrid architecture combines a deep neural network (DNN) and k-nearest neighbors (kNN) classifier. The DNN consists of two fully connected layers, the first layer with 128 units and ReLU activation, and the second as a single unit with sigmoid activation. These are integrated through a Twin Neural Network (TNN) design [20], allowing the model to perform classification via the DNN while simultaneously learning to separate positive and negative samples in the feature space to optimize the kNN algorithm's performance.

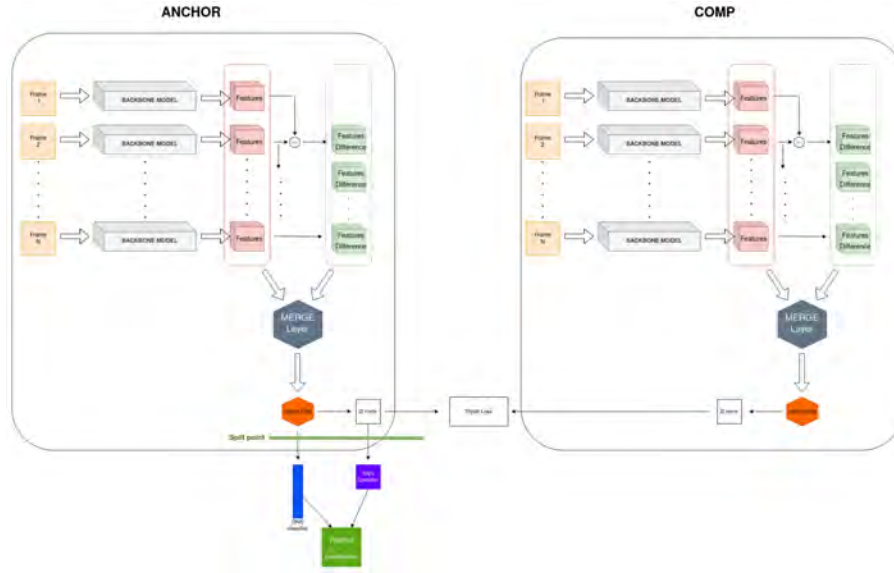


Figure 4.34: Proposed model for violence detection in video streams [121].

We used input frames reshaped to a 224x224 resolution, with pixel values adjusted for the Mobilenet V2 and DenseNet models in search for the optimal hyper-parameters. Additional experiments were performed at a higher resolution of 320x320, which proved to capture more information despite backbone models not being trained for this specifically. In order to capture the temporal dynamics between sequential frames, we used the feature difference that also preserves time dimensionality across both streams. The twin network architecture encodes video content (violent or non-violent) into a 256-dimensional embedding space, which is then l2-normalized before being processed by both classifiers. For the DNN classifier we utilized a binary cross-entropy loss function. Parameter r was used to vary the merging layer from a single Feedforward Neural Network with a DNN classifier ($r=0$), to a Twin Neural Network (TNN) with a KNN classifier ($r=1$).

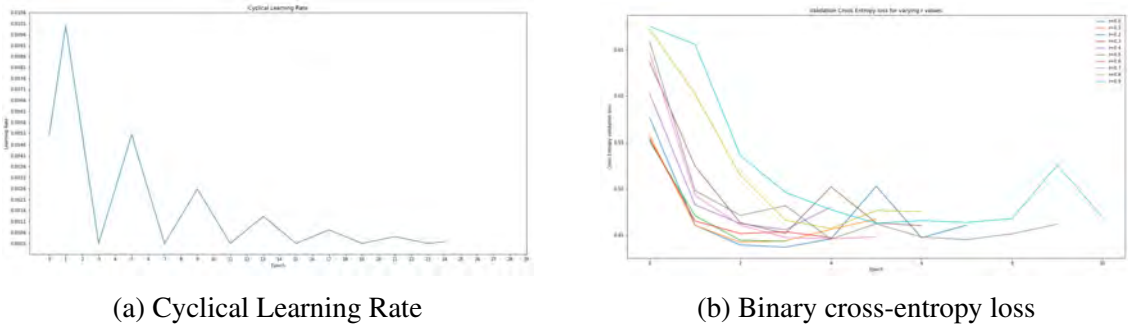


Figure 4.35: Parameter variation [121].

The model was trained for a maximum of 50 epochs, with a batch size of 8, using early stopping with a threshold of 5 epochs and monitoring the validation weighted loss. The alternative approach was to keep the training parameters and split the model in two parts to train them separately, starting with TNN and then using the results to train the DNN classifier. All experiments were performed on a system with an Intel(R) Xeon(R) E5-1620 v3 @ 3.50 GHz CPU, 64GB RAM and a NVIDIA GeForce RTX 3060 GPU. The model was implemented using Tensorflow and Keras Tuner (<https://www.tensorflow.org>) and run over the RWF-2000 dataset.

The learning rate over the epochs varied from a minimum of 10^{-4} to a maximum of 10^{-2} , shown in figure 4.35a. Then we kept the learning rate at a fixed value of 10^{-4} and varied r in increments of 0.1 and observed the values for the binary cross-entropy loss, shown in figure 4.35b. Next, we aimed at obtaining best values for the accuracy, trying different combinations of the two classifiers, DNN and KNN. The results show that a weighted sum sometimes outperforms any single classifier, the results being shown in table 4.10.

r values	DNN	KNN	weighted
0.0	82.25	-	82.25
0.1	82.75	80.5	83
0.2	81.5	78.75	80.25
0.3	81.5	82.25	82.75
0.4	81.25	81	81.25
0.5	82	80.75	81
0.6	82	81.75	81.5
0.7	81.75	81.5	81.75
0.8	81	81.5	81.5
0.9	81	79.75	79.75
1.0	83.75	-	83.75

Table 4.10: Accuracy values for different r values [121].

For comparison purposes we assessed accuracies for different models with a threshold of 0.5. This allows a direct comparison between the state-of-the-art values (upper half) and our results (lower half), the complete overview being presented in table 4.11. The results highlighted represent the best accuracies, delivered by SepConvLSTM-M, and densenet169_320_SLSearch. As expected, we could not reach the state-of-the-art result; however, we feel our best result justifies further analysis.

Model	Accuracy
I3D (Flow only) [22]	75.50
ConvLSTM [110]	77.00
I3D (TwoStream) [22]	81.50
C3D [118]	82.75
I3D (RGB only) [22]	85.75
Flow Gated [25]	87.25
SPIL Convolution [109]	89.30
SepConvLSTM-M [52]	89.75
mobilenet_224_DNN	78.50
densenet169_224_DNN	80.75
densenet121_224_DNN	80.75
densenet201_320_DNN	81.25
densenet201_224_DNN	81.50
densenet169_320_DNN	82.00
densenet121_320_DNN	82.25
densenet169_320_weighted	83.00
densenet169_320_SL	83.70
densenet169_320_SLSearch	85.25

Table 4.11: Accuracy values for different classifiers [121].

We replicated the learning processes on our system in order to give a fair comparison between the two best results. Our system was able to perform the training for the SepConvLSTM-M in about 45 minutes per iteration, with the best accuracy of 89.75% being reported on iteration 114 from a total of 140, giving a total time frame of 85 hours and 30 minutes. Training for our model required 24 minutes per iteration, with the best accuracy being reported on iteration 9, giving a total time frame of 3 hours and 36 minutes. Our result may have been increased by fine tuning the backbones, which we did not consider due to memory constraints. To conclude, our approach ran much faster, requiring 4.21% of the time required by the best result. In addition, while our training system was a PC built with components available commercially off-the-shelf, the best result was achieved by using Google Colab with an allocated Tesla P100-16GB GPU, which is significantly more powerful than our Nvidia RTX 3060. This way, we opened the path for

performing violence detection under a resource-constrained environment, which could further benefit from other techniques, such as background suppression, object-level feature extraction.

4.4.3 Embedded Epigenetics for Gesture Recognition

With modern society continuously aiming to raise the living standards, the effects of aging have become ever more challenging, the loss of cognitive functions, such as thinking, remembering, and reasoning often leading to severe interference with a person's daily life. Such conditions are commonly known as *dementia*, ranging in severity up to the point when a person becomes completely dependent on others for even the most basic activities. Dementia is known to be chronically degenerative, with no known medical cure; treatment does exist to manage and contain symptoms, slowing down degenerative effects, but its efficacy is strongly dependent upon strict adherence from the patient. According to medical studies, 50% of the patients fail to medicate correctly, with 30% of hospital admissions over the age of 65 directly related to non-adherence to medical regimen [108]. Key to these facts is represented by dementia actually preventing appropriate adherence to treatment, which would in turn slow its evolution, thus favoring appropriate adherence to treatment.

Breaking this recursive dependence can only be performed by monitoring adherence to medical regimen, ideally by means of an inexpensive wearable device. As a consequence, we propose an Internet-of-Things (IoT)-connected wearable device, placed on the hand of the patient like any modern smart watch, that would perform real-time gesture recognition for monitoring pill ingestion. Such a device would allow autonomous and systematic monitoring of adherence to medical treatment, requiring no outside intervention [5].

In order to recognize gestures, it is crucial to understand their complexity and range, in addition to their relation to the face (raising eyebrows, winks, nostril/cheek movements, head shakes, expressions) or body posture (actions performed with the hands or feet). We are specifically interested in recognizing hand gestures, which have been classified as shown in figure 4.36 [44]. While gesture recognition may be performed best by a vision-based system, achieving accuracy up to 97% [45], our aim is represented by a portable system able to operate autonomously and be located with the patient at all time. All of the above indicate a sensor-based device would be better suited, favoring a wrist wearable

endowed with inertial sensing and some communication capabilities. Previous research confirms both the feasibility and performance of the method, with up to 92.5% accuracy [72, 63, 26].

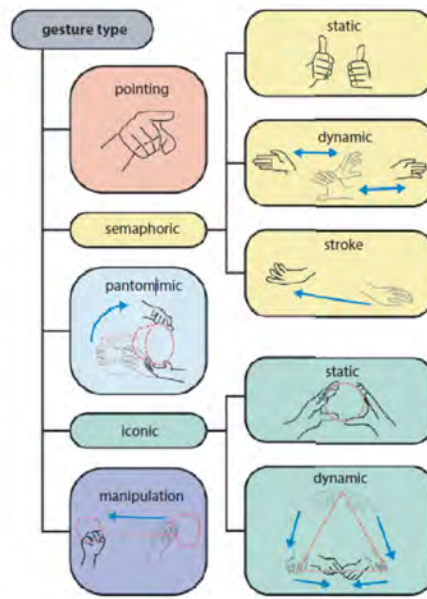


Figure 4.36: Gesture classification [44].

For our purpose we decided on LILYGO TTGO T-Watch, which is an open source device running ESP32 [67, 39] and providing an integrated IMU (accelerometer and gyroscope) and WiFi/Bluetooth connectivity. Gesture-related data acquisition is performed by the wearable device in real-time, followed by additional processing for detecting hand gestures. We targeted three types of hand gestures, associated with pill intake, casual hand movement, and still hand. The detection is performed by a fully-connected neural network comprising 3 layers: 50 neurons for the input layer, a single hidden layer with 20 neurons, and 3 neurons for the output layer.

The neural network was subjected to three categories of training data containing acceleration measurements with 120 samples each, supplied by the LILYGO device at a sampling rate of 100Hz. The first category provides measurements for the gesture of interest, which implies raising the hand for the purpose of ingesting a pill. The second category provides measurements of other (casual) gestures performed with the same hand. The third category relates to the body posture, providing measurements of the same hand during various relaxed and relatively still positions. Data was collected from 14 people with no health issues affecting the hand gestures specifically (7 persons 18-24 years old, 2 persons 24-44 years old, 5 persons 44-74 years old). For training, testing and validation purposes, data was shuffled and partitioned using 60%, 20%, and 20% splits.

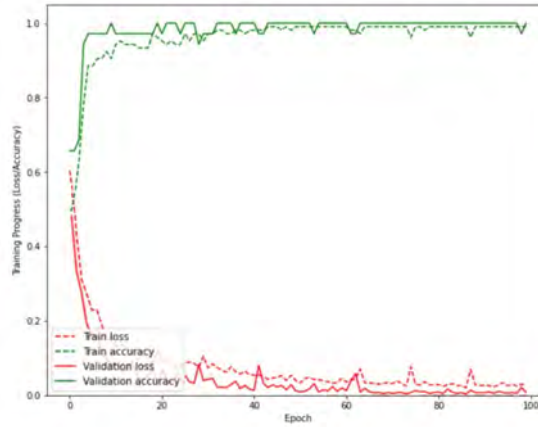


Figure 4.37: Evolution of gesture identification performance during training [44].

After the initial training (shown in figure 4.37) we performed three tests. The first test involved the training and validation splits, with accuracy values of over 99% for each of the datasets, and error (loss) values under 5%. For the second test we used the testing splits, yielding accuracy values of 100% and losses of about 0.2%. The third test was different, involving 5 persons of which 2 were previously involved in collecting data used for the training. The other 3 persons were new for the system, in an attempt to assess its operating accuracy. Each person was subject to 5 separate test, each involving 10 gestures associated with pill intake. The final result is considered to be the mean value of the 10 gesture runs, denoted as M in table 4.12, indicating accuracy values over 97%.

A valid issue is represented by the features that make a credible pill intake gesture, which triggered additional experimenting in order to assess the relevant threshold. For this purpose, gestures that should partly mimic the pill intake, such as raising the hand up to the opposite elbow or shoulder, and reaching for the nose, mouth, or an eye. We employed 2 persons to carry out gestures similar (but different or incomplete) to pill intake, the results being shown in table 4.13. When gestures imply raising the hand to the opposite elbow level, our algorithm does not identify the pill intake gesture, the values recorded being too small. The values show significant increase when raising the hand up to the opposite elbow, but remaining under 45% accuracy and therefore meaning that the gestures are still not identified as pill intake. However, reaching to the mouth/nose/eye is very similar to the pill intake, which triggers an incorrect identification of a pill intake gesture, generating false positives. Such situations are very confusing for gesture recognition, since it is very difficult to fully characterize specific gestures only from acceleration values.

After reviewing the results, we conclude that adherence to medical regimen can be

Table 4.12: Data recordings for recognition of pill intake gestures (mean values shown as M) [5].

	Attempt 1	Attempt 2	Attempt 3	Attempt 4	Attempt 5
Subject A	0.989802	1.000000	0.999823	0.996998	0.999983
	1.000000	0.999974	0.999999	0.998863	0.857413
	0.996036	1.000000	1.000000	0.999967	1.000000
	0.990195	0.866578	0.999650	0.999969	0.960832
	1.000000	0.992302	1.000000	1.000000	0.990579
	0.999648	0.999966	0.998800	0.999511	1.000000
	0.999991	0.999999	0.999395	1.000000	0.999999
	1.000000	0.934785	0.987594	0.999926	0.978256
	0.999964	1.000000	1.000000	0.999873	0.999996
	0.999671	0.972247	0.999999	0.991738	1.000000
	$M=0.997522$	$M=0.9765851$	$M=0.998526$	$M=0.998684$	$M=0.978795$
Subject B	1.000000	1.000000	0.989976	1.000000	0.999886
	0.999998	0.999648	0.998894	0.999976	0.998652
	0.970230	1.000000	1.000000	0.991948	0.999549
	1.000000	0.999930	1.000000	1.000000	1.000000
	1.000000	0.999999	0.998131	0.999511	1.000000
	0.998972	0.999996	0.978722	0.972337	1.000000
	0.979291	0.999988	0.999998	0.996116	0.988832
	0.999893	1.000000	0.999966	1.000000	0.990021
	1.000000	1.000000	0.992322	0.989855	0.999853
	1.000000	0.987996	0.999999	0.990579	0.999648
	$M=0.994838$	$M=0.998755$	$M=0.995800$	$M=0.994032$	$M=0.997644$
Subject C	0.999996	0.969864	1.000000	0.999988	0.988948
	0.987992	0.988995	0.979868	0.992209	0.999986
	0.989980	0.990224	0.979980	0.989943	0.981342
	1.000000	0.998893	0.988656	0.992745	0.965998
	0.999998	1.000000	0.992921	0.999999	0.989766
	0.999968	0.998987	1.000000	0.998972	0.990906
	0.999297	0.979304	0.990924	0.999291	0.984011
	0.999895	0.993209	0.988993	0.999453	0.999294
	1.000000	0.999893	0.986889	0.999996	0.998469
	0.999893	0.998887	0.989973	0.999988	0.998893
	$M=0.997701$	$M=0.991821$	$M=0.989814$	$M=0.997258$	$M=0.989826$
Subject D	0.999886	0.997546	0.988775	0.999648	0.992302
	0.999998	0.896578	0.888998	0.999991	0.999966
	0.986789	0.989882	0.989999	0.986789	0.998131
	0.969782	0.989905	0.999965	0.969782	0.978722
	0.990029	0.999996	0.990025	0.999754	0.891429
	0.988741	0.989976	0.990189	0.896578	0.988741
	0.999384	0.999882	0.999648	1.000000	0.999384
	0.999887	0.998801	0.999996	0.960832	0.999930
	0.999999	0.999999	0.999999	0.999650	0.999999
	0.989964	0.999999	1.000000	1.000000	0.989912
	$M=0.994925$	$M=0.986256$	$M=0.984759$	$M=0.981302$	$M=0.983961$
Subject E	0.989802	0.999648	0.990029	1.000000	0.988948
	1.000000	0.999991	0.999998	0.999991	0.999986
	0.866578	0.998972	0.970230	0.999754	0.999921
	0.992302	0.979291	0.989882	0.990924	0.977233
	0.999395	0.891998	1.000000	0.979304	0.992302
	0.987594	0.989999	0.998972	0.999966	0.998972
	0.979291	0.986789	0.990025	0.979341	0.988741
	0.999893	0.969782	0.984011	0.999903	0.999993
	0.978256	0.993209	0.999968	0.988993	0.998131
	0.999996	0.999893	0.969782	0.984023	0.999029
	$M=0.979310$	$M=0.980957$	$M=0.989289$	$M=0.992213$	$M=0.994325$

Table 4.13: Accuracy assessment for gestures similar to pill intake [5].

		Right hand- left elbow	Right hand- left shoulder	Right hand- mouth, nose, eyes
Subject A (sitting)	Attempt #1:	0,000001	0.013052	1.000000
	Attempt #2:	0.000006	0.009023	0.999874
	Attempt #3:	0.047611	0.284041	1.000000
	Attempt #4:	0.023761	0.008311	0.999853
	Attempt #5:	0.001781	0.191394	0.999999
	Attempt #6:	0.000006	0.002629	0.999961
	Attempt #7:	0.000108	0.224519	0.999995
	Attempt #8:	0.000010	0.104929	0.984945
	Attempt #9:	0.000000	0.203976	0.898478
	Attempt #10:	0.000002	0.442386	1.000000
Subject A (standing)	Attempt #1:	0.000000	0,99189	1.000000
	Attempt #2:	0.000000	0.998435	1.000000
	Attempt #3:	0.000001	0.999196	1.000000
	Attempt #4:	0.000004	0.993023	0.999999
	Attempt #5:	0.000000	0.999541	0.999999
	Attempt #6:	0.000000	0.970640	1.000000
	Attempt #7:	0.000000	0.992034	1.000000
	Attempt #8:	0.000000	0.970640	1.000000
	Attempt #9:	0.000001	0.804532	1.000000
	Attempt #10:	0.000000	0.798429	1.000000
Subject B (sitting)	Attempt #1:	0.000000	0,99582	1.000000
	Attempt #2:	0.000000	0.991447	1.000000
	Attempt #3:	0.000000	0.901634	1.000000
	Attempt #4:	0.000000	0.922864	1.000000
	Attempt #5:	0.000000	1.000000	1.000000
	Attempt #6:	0.000001	0.939847	0.999999
	Attempt #7:	0.000000	0.983276	0.999995
	Attempt #8:	0.000000	0.999687	1.000000
	Attempt #9:	0.000000	0.968693	1.000000
	Attempt #10:	0.000000	0.963857	1.000000
Subject B (standing)	Attempt #1:	0.000044	0,177954	1.000000
	Attempt #2:	0,000001	0,178253	1.000000
	Attempt #3:	0,000101	0,209255	0.999992
	Attempt #4:	0,000091	0,348146	0.999968
	Attempt #5:	0,000197	0,073302	0.999988
	Attempt #6:	0,000001	0,137356	1.000000
	Attempt #7:	0,000558	0,097247	1.000000
	Attempt #8:	0.000000	0,035561	1.000000
	Attempt #9:	0.000021	0,000853	0.999995
	Attempt #10:	0.000003	0,00014	0.984956

performed by machine learning running on an inexpensive wearable device that produces no interference with other medical devices. The performance is based on processing acceleration values only to deliver high accuracy in identifying the pill ingestion gesture. Further differentiation between gestures could be enhanced by adding full gyroscope data, an addition that should not significantly raise expenses for the device.

Part II

**RESEARCH AND CAREER
PROSPECTS**

Chapter 5

Research Directions and Career Developments

The world at large may be regarded as comprising of two main categories. On one hand, there is the category of *knowledge creators* (or scientists, for short), that unites all those who seek to understand the mysteries of Nature. According to Moshe Sipper [Sipper2002], the scientist is a detective who “analyzes natural processes, wishing to explain their workings, ultimately seeking to predict their future behavior”. On the other hand, there is the category of *applied technical professionals* (or engineers, for short), that includes all those dedicated to making things work for the betterment of society and mankind. The engineer is therefore a builder, trusting his own revelations, reliant on science and constrained by economics and society dynamics. It would seem the two categories are intertwined, the scientist requiring tools built by the engineer, while the engineer ultimately makes those tools and creates technology advancements by using scientific knowledge. However, the intertwining does not equate exact contemporaneity: casting some doubt, Lewis Wolpert [Wolpert1998] considers that technology is actually much older than science, with scientists always having relied onto the existing technology, while engineers only making use of modern science, that is, from the nineteenth century onward.

No inquisitive mind, be it of a scientist or an engineer, escapes the great existential questions [ExistentialQ], which may be reduced to who we are (identity), what are we supposed to do (mission), and why are we supposed to do it (purpose). My attempts to give some meaningful answers will make up the rest of this chapter, since I believe these will definitely imprint on our future research directions, together with the development of the academic career.

5.1 Identity

The modern-day society has coined the term *STEM* to emphasize the importance of Science, Technology, Engineering and Mathematics, for its continued development, something that can only be achieved through education. The reasons STEM education

is considered vital are manifold: it fosters innovation, drives economic growth, and enhances our understanding of the world; it allows individuals develop highly-valued skills, cultivates critical thinking and problem-solving abilities, and stimulates large-scale collaboration. However, while interconnected they represent distinct disciplines, different by their focus, methods and goals. In other words, while unifying, they also fuel a debate on the concept of identity, therefore eliciting further analysis.

Science targets the understanding of the natural world by approaching some key questions, such as "What is this?" and "Why does it behave that way?". Through orchestrated observation, experimentation, hypothesis testing, and theory formulation, the scientist is constantly attempting to explain how and why everything works the way it is observed or predicted. Therefore, the scientist is involved with the great discovery of nature laws and fundamental truths. Mathematics represents the essentials of logic, focusing on abstract patterns, structures, and relationships to provide formal proof, modeling and algorithms. The mathematician attempts to understand the logical sequence of steps required to model and solve problems. Science and mathematics are bonded together by being sometimes oblivious to the practical, immediate application of their results, which sets the scene for the following two contenders.

Technology is mainly concerned with applying the scientific knowledge, aiming to improve on daily tasks by creating better tools and products. When engineering is concerned the main focus is building reliable, efficient, and cost-effective solutions and devices by leveraging science and mathematics. There is a perceptible symbiosis between technology and engineering, since one cannot exist without the other; however, the higher position in the hierarchy belongs to the engineer, who is responsible for the overall perspective and *mise en place* so that useful results are achieved reliably, safely, and efficiently. It is precisely at this point when the identity of the engineer becomes blurry: the engineer has to assemble elaborate ways in which cutting edge knowledge from science and mathematics are put to practical use by using technology, to lay ground for continued progress in all directions. The engineer has therefore to show strong abilities in all science, mathematics, and technology! As a consequence, one may view the relationship between the STEM domains as shown in figure 5.1.

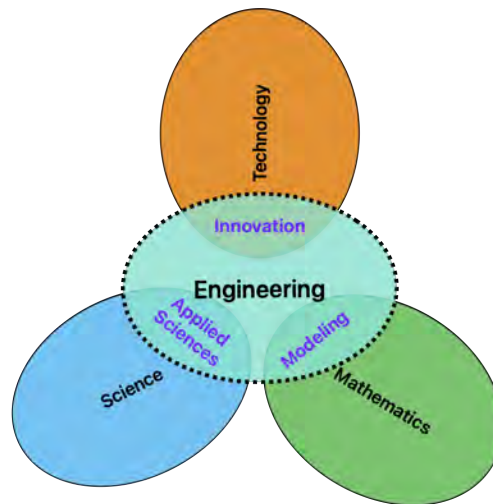


Figure 5.1: STEM core domains and their relationship.

The superposition with other core domains is easy to explain: engineering combined with science gives applied sciences, with mathematics gives modeling, while in combination with technology gives innovation. Would this argued multivalence cast a lesser importance for the engineer compared to the scientist or mathematician? It seems William Shakespeare’s centuries-old famous quote: “A jack of all trades is a master of none, but oftentimes better than a master of one” can still stir some debates, however alleviated by the dualistic nature of the modern engineer, coined as a *scigineer* [Sipper2002].

But however close the engineer and the scientist got, some clear distinctions between the academic research and industrial R&D remain. The academic research, while including both science and engineering, aims at advancing human knowledge and understanding by pursuing fundamental questions regardless of immediate practical application. Long-term, high-risk perspectives are often undertaken by the academia as investigations are performed with no immediate pressure and usually become embedded into educational processes. With an accent toward intellectual inquisitiveness and scientific rigor, academia seeks to publish findings, being open for peer review and scrutiny, measuring success by citations, theoretical contributions, and recognition. On the other hand, industrial R&D approaches specific problems to solve them and create competitive advantage and commercial value, with a tendency to focus on practical applications with a clear potential for return on investment. Here the success is measured by revenue and, quite opposite to academia, by creating protected intellectual property through patents and trade secrets. Industrial R&D must balance innovation with reliability and manufacturability, with a much lower risk tolerance and significantly shorter timelines. The schism between the

two worlds is only apparent, with quite a few fellows from the academia becoming successful in private businesses and viceversa, with Intel and Google founders being prime examples.

5.2 Purpose

As previously stated and as an engineer I wholeheartedly share the belief that above all, the engineer has to be a builder. A solid education clearly brought an important contribution to the desire to build, regardless of its form: if conceiving algorithms and providing high-quality data are essential, implementing services and devices do not yield a lesser importance, while building an academic career represents both an obligation and an opportunity.

Much of the research from the last period revolved around approaches to bio-inspired optimization of urban traffic; as such, it is only natural that we will continue our research efforts along this topic. The main motivation is to bring our contribution to a fundamental optimization challenge in transportation engineering, which is equivalent to an equilibrium network design problem [ABDULAAL197919, Marcotte1986]. In essence, the equilibrium network design problem is about how to provide an optimal design or set of modifications to a transportation network while accounting for the users' route-choice behavior [Friesz1981]. The complexity lies in the fact that two interrelated decisions are involved: network design decisions (made by a central planner) and route choice decisions (made by individual users). Users will prefer to choose routes based on their preference and experience, typically shortest travel time, which leads to reaching an equilibrium state that affects the overall network performance. Therefore, any change of the network design variables reflects on the traffic flow satisfying the user equilibrium conditions, making it a bi-level optimization problem that is NP-hard and non-convex, with exact solutions difficult to derive for large networks such as the urban road networks present in large cities. The equilibrium is reached when Wardrop's first principle is satisfied, that is, the delay times on any of the used routes become less or equal than on any unused routes [Wardrop1952]. A possible solution is described by the Performance Index (PI), which is to be solved as mathematical optimization problem given by equation (5.1):

$$\min_{\psi \in \Omega_0} \text{PI}(\psi, \mathbf{q}^*(\psi)) = \sum_{a \in \mathbb{L}} (\mathbb{W}_a D_a(\psi, \mathbf{q}^*(\psi)) + \mathbb{K}_a S_a(\psi, \mathbf{q}^*(\psi))) \quad (5.1)$$

Therefore, the objective is to find the optimum value for PI (and therefore the set of signal timings ψ) that minimizes the total weighted sum of delays and stops across all links, accounting for how those timings influence drivers' route choices and hence the flow patterns, subject to the following parameters:

- ψ : the signal timing vector, including green times, cycle times, and offsets for all signalized intersections;
- Ω_0 : the set of all feasible signal timing vectors ψ , including constraints such as minimum green time and maximum cycle time;
- $\mathbf{q}^*(\psi)$: the equilibrium traffic flows corresponding to signal settings, typically derived using a Stochastic User Equilibrium (SUE) model, meaning that drivers choose routes probabilistically based on perceived travel times, which depend on signal timings;
- $\mathbb{W}_a D_a(\psi, \mathbf{q}^*(\psi))$: the weighted average delay on link a , important for reducing delays on that link, depending on both the signal timing and the equilibrium flow;
- $\mathbb{K}_a S_a(\psi, \mathbf{q}^*(\psi))$: the weighted average number of stops on link a , importance for reducing stops on that link;

Research efforts continue to be invested in optimizing urban traffic by finding new ways of deriving optimal signaling [Nagurney2001, Lin2015, HUANG2022102719]. However, with all the progress reported in the literature, finding the Pareto-optimal signaling for a whole urban road network remains a challenge to this day, mainly due to the fact that urban traffic may not be strictly represented by a well-defined mathematical problem, but by a mix of socially-determined conditions whose dynamics are very difficult to collect and understand. To contribute, our efforts will span two main directions:

1. partitioning the network into smaller areas will bring a significant reduction in computational efforts together with essential opportunities for providing optimization at this (smaller, local) level. We started an incursion toward this direction in 4.3, where the social relevance of communities is also relevant for managing significant slices of the total traffic;

2. following the partitioning of the network into smaller areas (communities) will potentially ease the opportunities for creating green waves;
3. collecting relevant data to be used for traffic simulation and urban network design;

As stated in item 1, the communities are smaller partitions of the urban network that exercise a clear influence onto the overall traffic but is also a container for self-generated traffic, that is, traffic generated within the community remains confined within the boundaries of the same community. While not necessarily allowing for Pareto-optimality, specifically optimizing at the community level may prove beneficial at different moments of the day, or during impactful events.

The *green wave* mentioned by item 2 represents an optimization allowing traffic to flow unobstructed through several successive intersections. However, for this to be even possible, the network design is constrained, which reduces the leeway for finding the best coordination for traffic lights and builds pressure toward relaxing some of the constraints (such as the speed for which the green wave is designed). We aim for a better understanding on how to leverage bio-inspired computing techniques to implement community green waves that operate in parallel, a research direction discussed in 4.2 and 4.3.3. To the best of our knowledge, there isn't a publicly available, dedicated map illustrating the green wave traffic signal coordination specifically for the city of Timisoara; however, the major roads from the city make up for a solid choice for corridors subject to becoming green waves, as shown in figure 5.2 (in gray/black). Some of them are already implemented, but the dynamics of the city imposes a constant pressure for adjusting the existent green waves and implementing as many new ones as possible. Our aim is to continue the research and engineering over green waves, with a case-study platform represented by the city of Timisoara.

Collecting high-quality data from traffic, listed as 3, will help reaching new levels in understanding the social footprint of communities and the requirements and constraints of interconnected communities. At this moment, the equilibrium traffic flows are typically derived using a Stochastic User Equilibrium (SUE) model, meaning drivers choose routes probabilistically based on perceived travel times, which depend on signal timings [Prashker1999, PRASHKER2000277]. Therefore this leaves the social constraints of the drivers and their footprint over urban traffic completely unaccounted for, a fact that could significantly change the traffic simulation results, and by consequence, the way traffic signaling is determined and implemented. Acquiring high-quality data would

therefore solve social implications, determining how drivers choose their routes with increased precision, therefore allowing the creation of new benchmarks for assessing the traffic performance in urban networks, ultimately allowing the creation of digital twins for the unique urban fingerprint of each of our cities. We truly believe the better way for understanding urban traffic is represented by a shift from stochastic-based to social-inclusive models. While we do not expect a shrinking of the search space that would trivialize finding acceptable solutions, we feel that bio-inspired techniques such as genetic algorithms will continue to increase their quality.

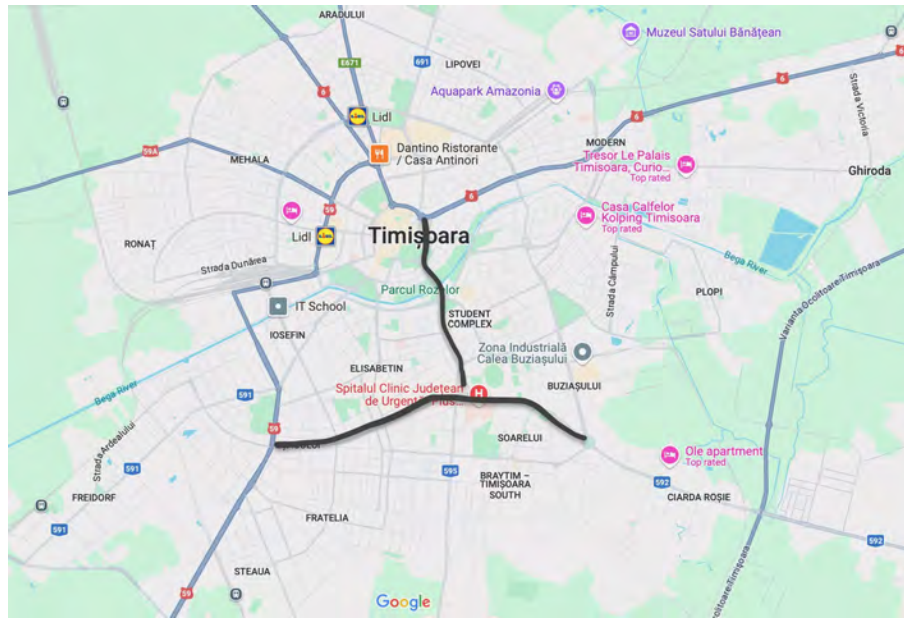


Figure 5.2: Major corridors subject to green wave in the city of Timisoara.

Our research efforts also point towards building and developing public services: a more fluid traffic could be served not only by multiple, parallel green waves, but also by a dynamic management of crisis situations, providing reserves when facing the full spectra of congestion associated with any major event (sports, accidents, public rallying, disturbances due to infrastructure maintenance, etc). The city management authorities need to provide adequate reaction for the traffic signaling systems fast, which requires both reactive measures (a swift derivation of new signaling) and pro-active measures (a choice of pre-compiled scenarios for foreseeable situations). We believe our contribution could assist in both directions, therefore providing fuel for a close relationship between the academia and the city authorities.



Figure 5.3: A wearable device for gesture recognition for assessing compliance with medical treatment.

Everything discussed so far belongs to the realm of algorithms and data, which make up the land of software. However, the world of computing is binary, a fact that would not be complete without its full expression of the duality software-hardware: whether considering abstract vs physical or functionalism vs materialism, some properties that exist in one realm do not exist into the other, leaving any computer engineering implementation incomplete if both perspectives are not provided. Therefore we do not plan for restraining to the software side, hardware implementations are being also targeted. These are the reasons for which we designed the traffic stack from figure 4.3 or the alternative intersection optimization from figure 4.30, as well as results reported for violence detection under constrained environments (in table 4.11) and gesture recognition (in table 4.13). Implementing these as embedded systems exposes the engineer to unforeseen situations, even when using pre-developed devices that provide some standard functionality: the wearable device we used for experimenting with gestures had some strange moments, reporting unusual energy levels, as shown in figure 5.3. Such situations certainly do not deliver the quality level and attention to detail that would warrant the precise operation of a health-monitoring device!

5.3 Mission

When active within the fields of computer science and engineering, one has to convey toward students ways and models for computational thinking that are both simple to acquire and resilient on time. While a fundamental skill, computational thinking involves multiple levels of abstraction aimed at problem solving and designing systems (what is the best way of solving this?) and understanding human behavior (how difficult is this to solve?).

In the coming age of automation, we should go beyond programming, toward conceptualizing and heuristic reasoning (planning, learning and scheduling in the presence of uncertainty), stimulating the mixing of natural and technical sciences in a coherent structure, with a recognition and focus on valid models and quality ideas. In other words, we should teach innovation to students. However, innovation is delicate in nature, being neither a science, nor an art, but a practice. As a consequence, we should talk to our students about how to:

1. recognize and cherish an opportunity;
2. recognize forms of innovation;
3. identify settings that make it possible;
4. write clear specifications and then build hardware and software using modern tools and components;
5. help adopt and sustain the new systems.

In the quest to remain at the forefront of science and technology, every so often, the teaching staff is subject to the challenging task of updating or even proposing new courses for students. We believe there are two fundamental qualities for a successful course, which we would love to take if we were students, and we would love to teach if given the chance. We were fortunate to have had this privilege, with courses introducing state-of-the-art technologies, while also spending great efforts to maintain the proper atmosphere that would invite to achieving excellence.

5.4 A Final Word

Despite choosing deliberately to minimize any talk about difficulties, their presence is certain and they will most likely continue to exist, as current developments concerning traffic optimization have yet to settle this challenging problem. From a scientific perspective, we will therefore continue to work along this topic, also exercising the intellectual freedom of joining any other topic that will ignite interest and fuel collective research efforts. However, building and maintaining a research team represents a challenge of its own.

Our final thoughts go toward one intriguing quote from a well-known modern author, who is also an accomplished engineer, Nassim Taleb from Tandon School of Engineering, New York University, stating “Mathematics offer certainties but not effectiveness. Engineering offers effectiveness but not certainties. Social Science offers neither certainties nor effectiveness.” It would seem that in order to score any scientific win, one has to exercise the full potential of own social skills, which are apparently endorsed by a science that can deliver neither certainties nor effectiveness. Uncertainties when motivating a team’s efforts, the disorder implied by not knowing exactly where your efforts will lead to, which is actually an essential flavor of research, represent stressors that render research quite fragile [114]. Staying within academia and continuing research research efforts further prove that anti-fragility is born from fragility exposed to stressors, two ingredients that we have an abundance of.

References

- [1] Mostafa Abd-El-barr. *Design And Analysis of Reliable And Fault-tolerant Computer Systems*. GBR: Imperial College Press, 2006. ISBN: 1860946682.
- [2] D. Aharonov and M. Ben-Or. “Fault-tolerant quantum computation with constant error”. In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*. STOC ’97. El Paso, Texas, USA: Association for Computing Machinery, 1997, pp. 176–188. ISBN: 0897918886. DOI: [10.1145/258533.258579](https://doi.org/10.1145/258533.258579). URL: <https://doi.org/10.1145/258533.258579>.
- [3] Réka Albert, Hawoong Jeong, and Albert-László Barabási. “Error and attack tolerance of complex networks”. In: *Nature* 406.6794 (2000), pp. 378–382. DOI: [10.1038/35019019](https://doi.org/10.1038/35019019). URL: <https://doi.org/10.1038/35019019>.
- [4] A. Avizienis et al. “Basic concepts and taxonomy of dependable and secure computing”. In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 11–33. DOI: [10.1109/TDSC.2004.2](https://doi.org/10.1109/TDSC.2004.2).
- [5] Dacian Avramoni et al. “Detection of Pill Intake Associated Gestures using Smart Wearables and Machine Learning”. In: *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)*. 2022, pp. 000251–000256. DOI: [10.1109/CINTI-MACRo57952.2022.10029615](https://doi.org/10.1109/CINTI-MACRo57952.2022.10029615).
- [6] Dacian Avramoni et al. “Real Time Urban Traffic Data Pinpointing Most Important Crossroads”. In: *Soft Computing Applications*. Ed. by Valentina Emilia Balas et al. Cham: Springer International Publishing, 2021, pp. 294–306. ISBN: 978-3-030-51992-6. DOI: [10.1007/978-3-030-51992-6_23](https://doi.org/10.1007/978-3-030-51992-6_23).
- [7] Dacian Avramoni et al. “Short term traffic congestion prediction using publicly available traffic data: a case study on Timisoara”. In: *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2022, pp. 000121–000126. DOI: [10.1109/SAMI54271.2022.9780813](https://doi.org/10.1109/SAMI54271.2022.9780813).
- [8] F. Awesome. *Frank*. 2005.

- [9] F. Awesome. *frank, but lowercase*. 2006.
- [10] Gabriel Baban et al. “Determination of the critical congestion point in urban traffic networks: A case study”. In: *2017 IEEE 14th International Scientific Conference on Informatics*. 2017, pp. 18–23. DOI: [10.1109/INFORMATICS.2017.8327215](https://doi.org/10.1109/INFORMATICS.2017.8327215).
- [11] Gabriel Baban et al. “High Betweenness Nodes and Crowded Intersections: An Experimental Assessment by Means of Simulation”. In: *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. 2018, pp. 000339–000344. DOI: [10.1109/SACI.2018.8440989](https://doi.org/10.1109/SACI.2018.8440989).
- [12] H. Bar-El et al. “The Sorcerer’s Apprentice Guide to Fault Attacks”. In: *Proceedings of the IEEE* 94.2 (2006), pp. 370–382. DOI: [10.1109/JPROC.2005.862424](https://doi.org/10.1109/JPROC.2005.862424).
- [13] A.L. Barabási and M.Á. Pá3sfai. *Network Science*. Cambridge University Press, 2016. ISBN: 9781107076266. URL: <https://books.google.ro/books?id=iLtGDQAAQBAJ>.
- [14] W. Barker et al. “Fault tolerance using dynamic reconfiguration on the POEtic tissue”. English. In: *IEEE Transactions on Evolutionary Computation* 11.5 (Oct. 2007), pp. 666–684. ISSN: 1089-778X. DOI: [10.1109/TEVC.2007.896690](https://doi.org/10.1109/TEVC.2007.896690).
- [15] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. “Gephi: An Open Source Software for Exploring and Manipulating Networks”. In: *Proceedings of the International AAAI Conference on Web and Social Media* 3.1 (Mar. 2009), pp. 361–362. DOI: [10.1609/icwsm.v3i1.13937](https://doi.org/10.1609/icwsm.v3i1.13937). URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/13937>.
- [16] Paul Bischoff. *Surveillance Camera Statistics: Which City has the Most CCTV?* — *comparitech.com*. <https://www.comparitech.com/vpn-privacy/the-worlds-most-surveilled-cities/>. [Accessed 09-04-2025]. 2023.
- [17] Transportation Research Board, Engineering National Academies of Sciences, and Medicine. *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*. Washington, DC: The National Academies Press, 2010. ISBN: 978-0-309-14304-2. DOI: [10.17226/14364](https://doi.org/10.17226/14364). URL: <https://nap.nationalacademies.org/catalog/14364/adaptive-traffic-control-systems-domestic-and-foreign-state-of-practice>.

- [18] Pascal Bouvry et al. “Resilience within Ultrascale Computing System: Challenges and Opportunities from NESUS Project”. English. In: *Supercomputing frontiers and innovations* 2.2 (2015), pp. 1–18. ISSN: 2409-6008. DOI: <http://superfri.org/superfri/article/view/46>.
- [19] P Bremner et al. “SABRE: a bio-inspired fault-tolerant electronic architecture”. In: *Bioinspiration Biomimetics* 8.1 (Jan. 2013), p. 016003. DOI: [10.1088/1748-3182/8/1/016003](https://doi.org/10.1088/1748-3182/8/1/016003). URL: <https://dx.doi.org/10.1088/1748-3182/8/1/016003>.
- [20] Jane Bromley et al. “Signature verification using a "Siamese" time delay neural network”. In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. NIPS’93. Denver, Colorado: Morgan Kaufmann Publishers Inc., 1993, pp. 737–744. DOI: [10.5555/2987189.2987282](https://doi.org/10.5555/2987189.2987282).
- [21] L.M. Brown. “Comparing reliability predictions to field data for plastic parts in a military, airborne environment”. In: *Annual Reliability and Maintainability Symposium, 2003*. 2003, pp. 207–213. DOI: [10.1109/RAMS.2003.1181927](https://doi.org/10.1109/RAMS.2003.1181927).
- [22] João Carreira and Andrew Zisserman. “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4724–4733. DOI: [10.1109/CVPR.2017.502](https://doi.org/10.1109/CVPR.2017.502).
- [23] Halim Ceylan and Michael G.H Bell. “Traffic signal timing optimisation based on genetic algorithm approach, including drivers’ routing”. In: *Transportation Research Part B: Methodological* 38.4 (2004), pp. 329–342. ISSN: 0191-2615. DOI: [https://doi.org/10.1016/S0191-2615\(03\)00015-8](https://doi.org/10.1016/S0191-2615(03)00015-8). URL: <https://www.sciencedirect.com/science/article/pii/S0191261503000158>.
- [24] Chi-Kwong Chan and L.M. Cheng. “Improved hiding data in images by optimal moderately-significant-bit replacement”. In: *Electronics Letters* 37 (16 2001), pp. 1017–1018. DOI: [10.1049/el:20010714](https://doi.org/10.1049/el:20010714). eprint: <https://digital-library.theiet.org/doi/pdf/10.1049/el%3A20010714>. URL: <https://digital-library.theiet.org/doi/abs/10.1049/el%3A20010714>.
- [25] Ming Cheng, Kunjing Cai, and Ming Li. “RWF-2000: An Open Large Scale Video Database for Violence Detection”. In: *2020 25th International Confer-*

- ence on Pattern Recognition (ICPR)*. 2021, pp. 4183–4190. DOI: [10.1109/ICPR48806.2021.9412502](https://doi.org/10.1109/ICPR48806.2021.9412502).
- [26] Yen-Cheng Chu et al. “Recognition of Hand Gesture Sequences by Accelerometers and Gyroscopes”. In: *Applied Sciences* 10.18 (2020). ISSN: 2076-3417. DOI: [10.3390/app10186507](https://doi.org/10.3390/app10186507). URL: <https://www.mdpi.com/2076-3417/10/18/6507>.
- [27] Ernest O Codier. “Reliability prediction—Help or hoax?” In: *Annual Symposium on Reliability*. 1969, pp. 383–390.
- [28] Eric Cole. *Hiding in Plain Sight: Steganography and the Art of Covert Communication*. Wiley, 2003. ISBN: 978-0471444497.
- [29] Cristian Cosariu. *TACTICS: adaptive framework for reactive control of road traffic systems*. eng. Timișoara, 2015.
- [30] Cristian Cosariu, Lucian Prodan, and Mircea Vladutiu. “Toward traffic movement optimization using adaptive inter-traffic signaling”. In: *2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*. 2013, pp. 539–544. DOI: [10.1109/CINTI.2013.6705256](https://doi.org/10.1109/CINTI.2013.6705256).
- [31] Cristian Cosariu et al. “Bio-inspired redistribution of urban traffic flow using a social network approach”. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. 2015, pp. 77–84. DOI: [10.1109/CEC.2015.7256877](https://doi.org/10.1109/CEC.2015.7256877).
- [32] National Research Council. *Reliability Growth: Enhancing Defense System Reliability*. Washington, DC: The National Academies Press, 2015. ISBN: 978-0-309-31474-9. DOI: [10.17226/18987](https://doi.org/10.17226/18987). URL: <https://nap.nationalacademies.org/catalog/18987/reliability-growth-enhancing-defense-system-reliability>.
- [33] Tim Croy et al. “Stop chasing the AI illusion”. In: *Communications of the ACM* 50.4 (Apr. 2007), pp. 7–9. DOI: [10.1145/1232743.1232757](https://doi.org/10.1145/1232743.1232757).
- [34] Alec Feinberg Dana Crowe. *Design for Reliability*. CRC Press, 2001. ISBN: 084931111X.
- [35] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).

- [36] Leandro L. Di Stasi et al. “The effect of traffic light spacing and signal congruency on drivers’ responses at urban intersections”. In: *Transportation Engineering* 8 (2022), p. 100113. ISSN: 2666-691X. DOI: <https://doi.org/10.1016/j.treng.2022.100113>. URL: <https://www.sciencedirect.com/science/article/pii/S2666691X22000112>.
- [37] Shikha Dubey, Abhijeet Boragule, and Moongu Jeon. “3D ResNet with Ranking Loss Function for Abnormal Activity Detection in Videos”. In: *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*. 2019, pp. 1–6. DOI: [10.1109/ICCAIS46528.2019.9074586](https://doi.org/10.1109/ICCAIS46528.2019.9074586).
- [38] Paul Erdos and Alfred Renyi. “On the evolution of random graphs”. In: *Publ. Math. Inst. Hungary. Acad. Sci.* 5 (1960), pp. 17–61.
- [39] *ESP the open-source SoC platform*. <https://www.esp.cs.columbia.edu/>. [Accessed 02-04-2025].
- [40] Ernesto Estrada. *The Structure of Complex Networks: Theory and Applications*. USA: Oxford University Press, Inc., 2011. ISBN: 019959175X.
- [41] Ryan Florin and Stephan Olariu. “A survey of vehicular communications for traffic signal optimization”. In: *Vehicular Communications* 2.2 (2015), pp. 70–79. ISSN: 2214-2096. DOI: <https://doi.org/10.1016/j.vehcom.2015.03.002>. URL: <https://www.sciencedirect.com/science/article/pii/S2214209615000121>.
- [42] Regine Gerike and Tina Gehlert. “Spatial structures and travel time analysis of the German time use survey”. In: *1st International Time Use Observatory Workshop*. 2009.
- [43] Regine Gerike, Tina Gehlert, and Friedrich Leisch. “Time use in travel surveys and time use surveys – Two sides of the same coin?” In: *Transportation Research Part A: Policy and Practice* 76 (2015). Emerging data and methodological considerations in time-use analysis, pp. 4–24. ISSN: 0965-8564. DOI: <https://doi.org/10.1016/j.tra.2015.03.030>. URL: <https://www.sciencedirect.com/science/article/pii/S0965856415000828>.
- [44] Neela Harish et al. “Vision-Based Gesture Recognition: A Critical Review”. In: *Artificial Intelligence and Data Mining Approaches in Security Frameworks*. John Wiley Sons, Ltd, 2021. Chap. 14, pp. 247–260. ISBN: 9781119760429. DOI:

- <https://doi.org/10.1002/9781119760429.ch14>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119760429.ch14>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119760429.ch14>.
- [45] Neela Harish et al. “Vision-Based Gesture Recognition: A Critical Review”. In: *Artificial Intelligence and Data Mining Approaches in Security Frameworks*. John Wiley Sons, Ltd, 2021. Chap. 14, pp. 247–260. ISBN: 9781119760429. DOI: <https://doi.org/10.1002/9781119760429.ch14>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119760429.ch14>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119760429.ch14>.
- [46] PETTER HOLME. “CONGESTION AND CENTRALITY IN TRAFFIC FLOW ON COMPLEX NETWORKS”. In: *Advances in Complex Systems* 06.02 (2003), pp. 163–176. DOI: [10.1142/S0219525903000803](https://doi.org/10.1142/S0219525903000803). eprint: <https://doi.org/10.1142/S0219525903000803>. URL: <https://doi.org/10.1142/S0219525903000803>.
- [47] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- [48] Alexandru Iovanovici, Dacian Avramoni, and Lucian Prodan. “A dataset of urban traffic flow for 13 Romanian cities amid lockdown and after ease of COVID19 related restrictions”. In: *Data in Brief* 32 (2020), p. 106318. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2020.106318>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340920312129>.
- [49] Alexandru Iovanovici, Lucian Prodan, and Mircea Vladutiu. “Collaborative environment for road traffic monitoring”. In: *2013 13th International Conference on ITS Telecommunications (ITST)*. 2013, pp. 232–237. DOI: [10.1109/ITST.2013.6685551](https://doi.org/10.1109/ITST.2013.6685551).
- [50] Alexandru Iovanovici et al. “A hierarchical approach in deploying traffic lights using complex network analysis”. In: *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*. 2014, pp. 791–796. DOI: [10.1109/ICSTCC.2014.6982515](https://doi.org/10.1109/ICSTCC.2014.6982515).

- [51] Alexandru Iovanovici et al. “Heuristic Optimization of Wireless Sensor Networks Using Social Network Analysis”. In: *Soft Computing Applications*. Ed. by Valentina Emilia Balas, Lakhmi C. Jain, and Branko Kovačević. Cham: Springer International Publishing, 2016, pp. 663–671. ISBN: 978-3-319-18296-4.
- [52] Zahidul Islam et al. “Efficient Two-Stream Network for Violence Detection Using Separable Convolutional LSTM”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–8. DOI: [10.1109/IJCNN52387.2021.9534280](https://doi.org/10.1109/IJCNN52387.2021.9534280).
- [53] C. Mani Krishna Israel Koren. *Fault-Tolerant Systems*. Morgan Kaufmann Publishers, 2007. ISBN: 0-12-088525-5.
- [54] *International technology roadmap for semiconductors, emerging research devices*. 2004. URL: https://web.archive.org/web/20070711213328/http://www.itrs.net/Links/2004Update/2004_05_ERD.pdf.
- [55] H. Jahanirad. “Efficient reliability evaluation of combinational and sequential logic circuits”. In: *Journal of Computational Electronics* 18.1 (Mar. 2019), pp. 343–355. ISSN: 1572-8137. DOI: [10.1007/s10825-018-1288-4](https://doi.org/10.1007/s10825-018-1288-4). URL: <https://doi.org/10.1007/s10825-018-1288-4>.
- [56] Christopher Jais, Benjamin Werner, and Diganta Das. “Reliability predictions-continued reliance on a misleading approach”. In: *2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS)*. IEEE. 2013, pp. 1–6.
- [57] Vincent Rijmen Joan Daemen. *The Design of Rijndael. AES - The Advanced Encryption Standard*. Springer Berlin, Heidelberg, 2002. ISBN: 978-3-540-42580-9. DOI: [10.1007/978-3-662-04722-4](https://doi.org/10.1007/978-3-662-04722-4).
- [58] David Easley Jon Kleinberg. *Networks, Crowds, and Markets ; Reasoning about a Highly Connected World*. Cambridge University Press, 2010. ISBN: s9780521195331 0521195330.
- [59] J. Jones and J. Hayes. “A comparison of electronic-reliability prediction models”. In: *IEEE Transactions on Reliability* 48.2 (1999), pp. 127–134. DOI: [10.1109/24.784270](https://doi.org/10.1109/24.784270).

- [60] Christian Matthias Kerskens and David López Pérez. “Experimental indications of non-classical brain functions”. In: *Journal of Physics Communications* 6.10 (Oct. 2022), p. 105001. DOI: [10.1088/2399-6528/ac94be](https://doi.org/10.1088/2399-6528/ac94be). URL: <https://dx.doi.org/10.1088/2399-6528/ac94be>.
- [61] K. Keutzer and A.R. Newton. “The MARCO/DARPA Gigascale Silicon Research Center”. In: *Proceedings 1999 IEEE International Conference on Computer Design: VLSI in Computers and Processors (Cat. No.99CB37040)*. 1999, pp. 14–19. DOI: [10.1109/ICCD.1999.808257](https://doi.org/10.1109/ICCD.1999.808257).
- [62] Chong Hee Kim and Jean-Jacques Quisquater. “Faults, Injection Methods, and Fault Attacks”. In: *IEEE Design Test of Computers* 24.6 (2007), pp. 544–545. DOI: [10.1109/MDT.2007.186](https://doi.org/10.1109/MDT.2007.186).
- [63] Minwoo Kim et al. “IMU Sensor-Based Hand Gesture Recognition for Human-Machine Interfaces”. In: *Sensors* 19.18 (2019). ISSN: 1424-8220. DOI: [10.3390/s19183827](https://doi.org/10.3390/s19183827). URL: <https://www.mdpi.com/1424-8220/19/18/3827>.
- [64] Victor L. Knoop. *Traffic Flow Theory: An introduction with exercises*. https://www.victorknoop.eu/research/book/Knoop_Intro_traffic_flow_theory_edition3.pdf. 2021.
- [65] Parag K. Lala. *Digital Circuit Testing and Testability*. Academic Press, 1997. ISBN: 0-12-434330-9.
- [66] J. C. Laprie. “Dependability: Basic Concepts and Terminology”. In: *Dependability: Basic Concepts and Terminology: In English, French, German, Italian and Japanese*. Ed. by J. C. Laprie. Vienna: Springer Vienna, 1992, pp. 3–245. ISBN: 978-3-7091-9170-5. DOI: [10.1007/978-3-7091-9170-5_1](https://doi.org/10.1007/978-3-7091-9170-5_1). URL: https://doi.org/10.1007/978-3-7091-9170-5_1.
- [67] *LILYGO® TTGO T-Watch-2020*. <http://www.lilygo.cn/>. [Accessed 02-04-2025].
- [68] Nikolaos Limnios and Bei Wu. “Estimation of stationary probability of semi-Markov Chains”. In: *Statistical Inference for Stochastic Processes* 25.2 (July 2022), pp. 355–364. ISSN: 1572-9311. DOI: [10.1007/s11203-021-09255-3](https://doi.org/10.1007/s11203-021-09255-3). URL: <https://doi.org/10.1007/s11203-021-09255-3>.
- [69] D. Mange et al. “Toward robust integrated circuits: The embryonics approach”. In: *Proceedings of the IEEE* 88.4 (2000), pp. 516–543. DOI: [10.1109/5.842998](https://doi.org/10.1109/5.842998).

- [70] Septimiu Fabian Mare, Mircea Vladutiu, and Lucian Prodan. “Decreasing Change Impact Using Smart LSB Pixel Mapping and Data Rearrangement”. In: *2011 IEEE 11th International Conference on Computer and Information Technology*. 2011, pp. 269–276. DOI: [10.1109/CIT.2011.73](https://doi.org/10.1109/CIT.2011.73).
- [71] Septimiu Fabian Mare, Mircea Vladutiu, and Lucian Prodan. “High capacity steganographic algorithm based on payload adaptation and optimization”. In: *2012 7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*. 2012, pp. 87–92. DOI: [10.1109/SACI.2012.6249981](https://doi.org/10.1109/SACI.2012.6249981).
- [72] Sara Moccia et al. “Automated classification of hand gestures using a wristband and machine learning for possible application in pill intake monitoring”. In: *Computer Methods and Programs in Biomedicine* 219 (Mar. 2022), p. 106753. DOI: [10.1016/j.cmpb.2022.106753](https://doi.org/10.1016/j.cmpb.2022.106753).
- [73] J. von Neumann. “Probabilistic Logics and the Synthesis of Reliable Organisms From Unreliable Components”. In: *Automata Studies. (AM-34), Volume 34*. Ed. by C. E. Shannon and J. McCarthy. Princeton: Princeton University Press, 1956, pp. 43–98. ISBN: 9781400882618. DOI: [doi : 10.1515/9781400882618-003](https://doi.org/10.1515/9781400882618-003). URL: <https://doi.org/10.1515/9781400882618-003>.
- [74] John Von Neumann, Paul M. Churchland, and Klara Von Neumann. *The Computer and the Brain*. 2nd. USA: Yale University Press, 2000. ISBN: 0300084730.
- [75] M. E. J. Newman. “Modularity and community structure in networks”. In: *Proceedings of the National Academy of Sciences* 103.23 (2006), pp. 8577–8582. DOI: [10.1073/pnas.0601602103](https://doi.org/10.1073/pnas.0601602103). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.0601602103>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.0601602103>.
- [76] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [77] Patrick O’Connor. *Practical Reliability Engineering*. John Wiley & Sons, 2002. ISBN: 0-470-84462-0.
- [78] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. <https://www.openstreetmap.org>. 2017.

- [79] Flavius Opritoiu et al. “A high-speed AES architecture implementation”. In: *Proceedings of the 7th ACM International Conference on Computing Frontiers*. CF '10. Bertinoro, Italy: Association for Computing Machinery, 2010, pp. 95–96. ISBN: 9781450300445. DOI: [10.1145/1787275.1787300](https://doi.org/10.1145/1787275.1787300). URL: <https://doi.org/10.1145/1787275.1787300>.
- [80] Flavius Opritoiu et al. “Built-in self test applicability for the non-linear operations of Advanced Encryption Standard”. In: *2009 5th International Symposium on Applied Computational Intelligence and Informatics*. 2009, pp. 307–312. DOI: [10.1109/SACI.2009.5136262](https://doi.org/10.1109/SACI.2009.5136262).
- [81] Flavius Opritoiu et al. “Concurrent Error Detection for Multiplicative Inversion of Advanced Encryption Standard”. In: *2010 10th IEEE International Conference on Computer and Information Technology*. 2010, pp. 582–588. DOI: [10.1109/CIT.2010.121](https://doi.org/10.1109/CIT.2010.121).
- [82] Guru Prasad PANDIAN et al. “A critique of reliability prediction techniques for avionics applications”. In: *Chinese Journal of Aeronautics* 31.1 (2018), pp. 10–20. ISSN: 1000-9361. DOI: <https://doi.org/10.1016/j.cja.2017.11.004>. URL: <https://www.sciencedirect.com/science/article/pii/S100093611730239X>.
- [83] Michael Pecht, Pradeep Lall, and Edward Hakim. “Temperature as a Reliability Factor: We have a headache with Arrhenius”. In: *Thermal Management of Electronic Systems II: Proceedings of EURO THERM Seminar 45, 20–22 September 1995, Leuven, Belgium*. Springer. 1995, pp. 27–41.
- [84] JOHN PRESKILL. “FAULT-TOLERANT QUANTUM COMPUTATION”. In: *Introduction to Quantum Computation and Information*. World Scientific, 1998, pp. 213–269. DOI: [10.1142/9789812385253_0008](https://doi.org/10.1142/9789812385253_0008). eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789812385253_0008. URL: https://www.worldscientific.com/doi/abs/10.1142/9789812385253_0008.
- [85] L. Prodan, M. Udrescu, and M. Vladutiu. “Self-repairing embryonic memory arrays”. In: *Proceedings. 2004 NASA/DoD Conference on Evolvable Hardware, 2004*. 2004, pp. 130–137. DOI: [10.1109/EH.2004.1310821](https://doi.org/10.1109/EH.2004.1310821).

- [86] Lucian Prodan. “Self-Repairing memory Arrays Inspired by Biological Processes”. Doctoral Dissertation. Universitatea Politehnica Timisoara, 2004.
- [87] Lucian Prodan, Mihai Udrescu, and Mircea Vladutiu. “A dependability perspective on emerging technologies”. In: *Proceedings of the 3rd Conference on Computing Frontiers*. CF '06. Ischia, Italy: Association for Computing Machinery, 2006, pp. 187–198. ISBN: 1595933026. DOI: [10.1145/1128022.1128049](https://doi.org/10.1145/1128022.1128049). URL: <https://doi.org/10.1145/1128022.1128049>.
- [88] Lucian Prodan, Mihai Udrescu, and Mircea Vladutiu. “Fault-Tolerant Memory Design and Partitioning Issues in Embryonics”. In: *Proceedings of the 8th International Conference on Evolvable Systems: From Biology to Hardware*. ICES '08. Prague, Czech Republic: Springer-Verlag, 2008, pp. 372–381. ISBN: 9783540858560. DOI: [10.1007/978-3-540-85857-7_33](https://doi.org/10.1007/978-3-540-85857-7_33). URL: https://doi.org/10.1007/978-3-540-85857-7_33.
- [89] Lucian Prodan et al. “Biology Meets Electronics: The Path to a Bio-inspired FPGA”. In: *Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware*. ICES '00. Berlin, Heidelberg: Springer-Verlag, 2000, pp. 187–196. ISBN: 3540673385.
- [90] Lucian Prodan et al. “Design for dependability in emerging technologies”. In: *J. Emerg. Technol. Comput. Syst.* 3.2 (July 2007), 6–es. ISSN: 1550-4832. DOI: [10.1145/1265949.1265952](https://doi.org/10.1145/1265949.1265952). URL: <https://doi.org/10.1145/1265949.1265952>.
- [91] Lucian Prodan et al. “Embryonics: Artificial Cells Driven by Artificial DNA”. In: *Proceedings of the 4th International Conference on Evolvable Systems: From Biology to Hardware*. ICES '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 100–111. ISBN: 354042671X. DOI: [10.5555/645510.656932](https://doi.org/10.5555/645510.656932).
- [92] Lucian Prodan et al. “Embryonics: electronic stem cells”. In: *Proceedings of the Eighth International Conference on Artificial Life*. ICAL 2003. Cambridge, MA, USA: MIT Press, 2002, pp. 101–105. ISBN: 0262692813.
- [93] *Military Handbook Reliability Prediction of Electronic Equipment*. 1991. URL: <https://www.quanterion.com/wp-content/uploads/2014/09/MIL-HDBK-217F.pdf>.

- [94] Cristian Ruican et al. “Adaptive vs. Self-adaptive Parameters for Evolving Quantum Circuits”. In: *Evolvable Systems: From Biology to Hardware*. Ed. by Gianluca Tempesti, Andy M. Tyrrell, and Julian F. Miller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 348–359. ISBN: 978-3-642-15323-5. DOI: [10.1007/978-3-642-15323-5_30](https://doi.org/10.1007/978-3-642-15323-5_30).
- [95] Cristian Ruican et al. “Automatic Synthesis for Quantum Circuits Using Genetic Algorithms”. In: *Adaptive and Natural Computing Algorithms*. Ed. by Bartłomiej Beliczynski et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 174–183. ISBN: 978-3-540-71618-1. DOI: [10.1007/978-3-540-71618-1_20](https://doi.org/10.1007/978-3-540-71618-1_20).
- [96] Cristian Ruican et al. “Genetic algorithm based quantum circuit synthesis with adaptive parameters control”. In: *2009 IEEE Congress on Evolutionary Computation*. 2009, pp. 896–903. DOI: [10.1109/CEC.2009.4983040](https://doi.org/10.1109/CEC.2009.4983040).
- [97] Cristian Ruican et al. “Performance Analysis for Genetic Quantum Circuit Synthesis”. In: *Artificial Intelligence and Soft Computing*. Ed. by Leszek Rutkowski et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 205–212. ISBN: 978-3-642-13232-2. DOI: [10.1007/978-3-642-13232-2_25](https://doi.org/10.1007/978-3-642-13232-2_25).
- [98] Cristian Ruican et al. “Quantum Circuit Synthesis with Adaptive Parameters Control”. In: *Proceedings of the 12th European Conference on Genetic Programming*. EuroGP '09. Tübingen, Germany: Springer-Verlag, 2009, pp. 339–350. ISBN: 9783642011801. DOI: [10.1007/978-3-642-01181-8_29](https://doi.org/10.1007/978-3-642-01181-8_29). URL: https://doi.org/10.1007/978-3-642-01181-8_29.
- [99] Julien Salotti et al. “Comparison of Traffic Forecasting Methods in Urban and Suburban Context”. In: *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. 2018, pp. 846–853. DOI: [10.1109/ICTAI.2018.00132](https://doi.org/10.1109/ICTAI.2018.00132).
- [100] Mark Sandler et al. “ MobileNetV2: Inverted Residuals and Linear Bottlenecks ”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2018, pp. 4510–4520. DOI: [10.1109/CVPR.2018.00474](https://doi.org/10.1109/CVPR.2018.00474). URL: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474>.
- [101] *Gigascale Silicon Research Center Annual Report 1999*. 2000. URL: <https://apps.dtic.mil/sti/tr/pdf/ADA377260.pdf>.

- [102] P. Shivakumar et al. “Modeling the effect of technology trends on the soft error rate of combinational logic”. In: *Proceedings International Conference on Dependable Systems and Networks*. 2002, pp. 389–398. DOI: [10.1109/DSN.2002.1028924](https://doi.org/10.1109/DSN.2002.1028924).
- [103] Martin L. Shooman. *Reliability of Computer Systems and Networks*. John Wiley & Sons, 2002. ISBN: 0-471-29342-3.
- [104] P.W. Shor. “Fault-tolerant quantum computation”. In: *Proceedings of 37th Conference on Foundations of Computer Science*. 1996, pp. 56–65. DOI: [10.1109/SFCS.1996.548464](https://doi.org/10.1109/SFCS.1996.548464).
- [105] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. “A Comparison of ARIMA and LSTM in Forecasting Time Series”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 1394–1401. DOI: [10.1109/ICMLA.2018.00227](https://doi.org/10.1109/ICMLA.2018.00227).
- [106] M. Sipper et al. “A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 83–97. DOI: [10.1109/4235.585894](https://doi.org/10.1109/4235.585894).
- [107] M. Sipper et al. “A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 83–97. DOI: [10.1109/4235.585894](https://doi.org/10.1109/4235.585894).
- [108] Felicity Smith et al. “Assisting people with dementia with their medicines: experiences of family carers”. In: *International Journal of Pharmacy Practice* 23.1 (Oct. 2014), pp. 44–51. ISSN: 0961-7671. DOI: [10.1111/ijpp.12158](https://doi.org/10.1111/ijpp.12158). eprint: <https://academic.oup.com/ijpp/article-pdf/23/1/44/36164521/ijpp12158.pdf>. URL: <https://doi.org/10.1111/ijpp.12158>.
- [109] Yukun Su et al. “Human Interaction Learning on 3D Skeleton Point Clouds for Video Violence Recognition”. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 74–90. ISBN: 978-3-030-58548-8. DOI: [10.1007/978-3-030-58548-8_5](https://doi.org/10.1007/978-3-030-58548-8_5).
- [110] Swathikiran Sudhakaran and Oswald Lanz. “Learning to detect violent videos using convolutional long short-term memory”. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2017, pp. 1–6. DOI: [10.1109/AVSS.2017.8078468](https://doi.org/10.1109/AVSS.2017.8078468).

- [111] Waqas Sultani, Chen Chen, and Mubarak Shah. “Real-World Anomaly Detection in Surveillance Videos”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6479–6488. DOI: [10.1109/CVPR.2018.00678](https://doi.org/10.1109/CVPR.2018.00678).
- [112] Larisa Szatmari et al. “Road Intersection Optimization with Resource-Constrained Metaheuristic: A Case Study”. In: *2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*. 2022, pp. 000261–000266. DOI: [10.1109/SISY56759.2022.10036312](https://doi.org/10.1109/SISY56759.2022.10036312).
- [113] Hironori Takimoto et al. “Anomaly Detection Using Siamese Network with Attention Mechanism for Few-Shot Learning”. In: *Applied Artificial Intelligence* 36.1 (2022), p. 2094885. DOI: [10.1080/08839514.2022.2094885](https://doi.org/10.1080/08839514.2022.2094885). eprint: <https://doi.org/10.1080/08839514.2022.2094885>. URL: <https://doi.org/10.1080/08839514.2022.2094885>.
- [114] Nassim Nicholas Taleb. *Antifragile: Things That Gain from Disorder (Incerto)*. Random House Publishing Group, 2014. ISBN: 978-0812979688.
- [115] Xian Tao et al. “Unsupervised Anomaly Detection for Surface Defects With Dual-Siamese Network”. In: *IEEE Transactions on Industrial Informatics* 18.11 (2022), pp. 7707–7717. DOI: [10.1109/TII.2022.3142326](https://doi.org/10.1109/TII.2022.3142326).
- [116] Gianluca Tempesti et al. “Self-Replicating Hardware for Reliability: The Embryonics Project”. English. In: *Acm journal on emerging technologies in computing systems* 3.2 (July 2007). ISSN: 1550-4832. DOI: [10.1145/1265949.1265955](https://doi.org/10.1145/1265949.1265955).
- [117] Alexandru Topirceanu et al. “Social Cities: Redistribution of Traffic Flow in Cities Using a Social Network Approach”. In: *Soft Computing Applications*. Ed. by Valentina Emilia Balas, Lakhmi C. Jain, and Branko Kovačević. Cham: Springer International Publishing, 2016, pp. 39–49. ISBN: 978-3-319-18296-4.
- [118] Du Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2015, pp. 4489–4497. DOI: [10.1109/ICCV.2015.510](https://doi.org/10.1109/ICCV.2015.510). URL: <https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.510>.
- [119] Chao-Wen Tseng et al. “An evaluation of pseudo random testing for detecting real defects”. In: *Proceedings 19th IEEE VLSI Test Symposium. VTS 2001*. 2001, pp. 404–409. DOI: [10.1109/VTS.2001.923469](https://doi.org/10.1109/VTS.2001.923469).

- [120] Mihai Udrescu, Lucian Prodan, and Mircea Vlăduțiu. “Simulated fault injection methodology for gate-level quantum circuit reliability assessment”. In: *Simulation Modelling Practice and Theory* 23 (2012), pp. 60–70. ISSN: 1569-190X. DOI: <https://doi.org/10.1016/j.simpat.2012.01.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1569190X12000032>.
- [121] Catalin Vladu, Lucian Prodan, and Alexandru Iovanovici. “Resource Constrained, Fast Convergence Training for Violence Detection in Video Streams”. In: *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)*. 2022, pp. 000239–000244. DOI: [10.1109/CINTI-MACRo57952.2022.10029428](https://doi.org/10.1109/CINTI-MACRo57952.2022.10029428).
- [122] Ran-Zan Wang, Chi-Fang Lin, and Ja-Chen Lin. “Image hiding by optimal LSB substitution and genetic algorithm”. In: *Pattern Recognition* 34.3 (2001), pp. 671–683. ISSN: 0031-3203. DOI: [https://doi.org/10.1016/S0031-3203\(00\)00015-7](https://doi.org/10.1016/S0031-3203(00)00015-7). URL: <https://www.sciencedirect.com/science/article/pii/S0031320300000157>.
- [123] Alan P Wood and Jon G Elerath. “A comparison of predicted MTBFs to field and test data”. In: *Proceedings of Annual Reliability and Maintainability Symposium (RAMS)*. IEEE. 1994, pp. 153–156.
- [124] Mohamed Younis and Kemal Akkaya. “Strategies and techniques for node placement in wireless sensor networks: A survey”. In: *Ad Hoc Networks* 6.4 (2008), pp. 621–655. ISSN: 1570-8705. DOI: <https://doi.org/10.1016/j.adhoc.2007.05.003>. URL: <https://www.sciencedirect.com/science/article/pii/S1570870507000984>.
- [125] Y. Yu, B. Bastien, and B.W. Johnson. “A state of research review on fault injection techniques and a case study”. In: *Annual Reliability and Maintainability Symposium, 2005. Proceedings*. 2005, pp. 386–392. DOI: [10.1109/RAMS.2005.1408393](https://doi.org/10.1109/RAMS.2005.1408393).
- [126] Liang Zhao et al. “Onset of traffic congestion in complex networks”. In: *Phys. Rev. E* 71 (2 Feb. 2005), p. 026125. DOI: [10.1103/PhysRevE.71.026125](https://doi.org/10.1103/PhysRevE.71.026125). URL: <https://link.aps.org/doi/10.1103/PhysRevE.71.026125>.

- [127] J. F. Ziegler. “Terrestrial cosmic rays”. In: *IBM Journal of Research and Development* 40.1 (1996), pp. 19–39. DOI: [10.1147/rd.401.0019](https://doi.org/10.1147/rd.401.0019).