Global Optimization Algorithms and their Application to Distributed Systems

Thomas Weise, Hendrik Skubch, Michael Zapf, Kurt Geihs

2008-09-29

Abstract

In this report, we discuss the application of global optimization and Evolutionary Computation to distributed systems. We therefore selected and classified many publications, giving an insight into the wide variety of optimization problems which arise in distributed systems. Some interesting approaches from different areas will be discussed in greater detail with the use of illustrative examples.

1 Introduction

In this report, we will discuss different applications of global optimization for improving the features of distributed systems. Optimization algorithms are methods for finding optimal configurations of different features of their solution candidates. Many aspects of distributed systems are configurable or depend on parameter settings, such as the topology, security, and routing. Hence, there is a huge potential for using global optimization algorithms in order to improve them.

And indeed, this potential is widely utilized. The study by Sinclair [1] from 1999 reported that more than 120 papers had been published on work which employed Evolutionary Computation for optimizing network topology and dimension, node placement, trees, routing, and wavelength and frequency allocation. The incredibly comprehensive master's thesis by Kampstra from 2005 [2, 3] builds on this aforementioned study and classifies over 400 papers. According to Kampstra, communication networks was the field with the most researchers listed in EvoWeb, the European Network of Excellence in Evolutionary Computing, in 2005.[1] The first workshop on this topic, *Evolutionary Telecommunications* [4], took place in 1999. In the year 2000 alone, two books ([5] and [6]) have been published on the application of Evolutionary Computation to networking. Additional summary papers appeared around the same time [7, 8, 9, 10]. The recent studies from Alba and Chicano [11] and Cortés Achedad et al. [12] as well as the high number of papers published every year show that the interest in applying global optimization techniques in this problem domain has by no means decreased.

Most of the mentioned summaries concentrate on giving an overview in form of a more prosaic version of paper listings. We provide such a listing in a condensed form in Section 4, but focus on giving clear and detailed in-depth discussions of multiple example applications and also introduce the optimization algorithms utilized in them. This way, the subject becomes more tangible for audience which is rooted in only one the two involved subject areas.

We studied more than 130 papers from two decades of research in evolutionary telecommunication. Figure 1 illustrates how these papers distribute over the time from 1987 to 2008. The papers are classified according to the area of application, their optimization

[1] It still was when we checked their website http://evonet.lri.fr/evoweb/membership/bbsubcat.php [accessed 2008-09-07]

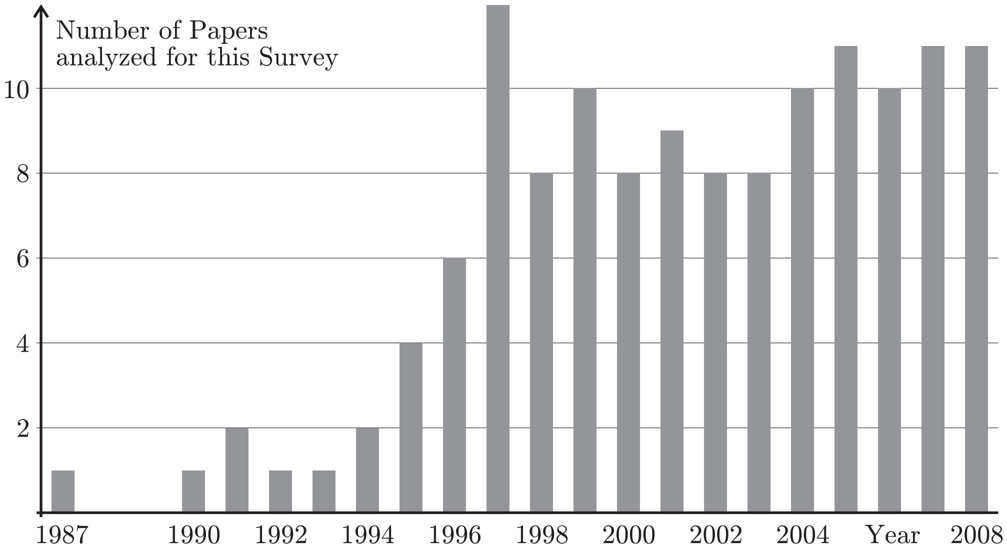


Figure 1: The Number of Papers studied for this Survey per Year.

goals, problem representations, and the optimization algorithms utilized. Figure 2 gives an overview of which areas were tackled by the researchers and which optimization algorithm they applied in the papers we studied. Here, it is important to notice that one paper may deal with multiple applications at once (like routing algorithms which also perform load balancing) and may thus occur in multiple columns. The complete subject catalog resulting from our survey can be found in Section 4. Such a list, however, gives only a limited idea about the actual approaches that have been developed. Therefore, we will use the following sections to first discuss the most important optimization algorithms (such as evolutionary algorithms, Ant Colony Optimization, Simulated Annealing, and Tabu Search) utilized by the selected research work in the next section. Afterwards, we will take a deeper look into some interesting optimization approaches from various areas of distributed systems which stand exemplary for the variety and the potential of this field of research. Different methods to synthesize or to improve network topologies are outlined in Section 3.1, adaptive or evolved routing protocols will be discussed in Section 3.2, and different approaches to the generation of protocols with global optimization algorithms is summarized in Section 3.3. In Section 3.4, we illustrate some security aspects and how they were optimized by different research groups before ending our overview on applications with software configuration and parameter adaption approaches in Section 3.5. After a representative list of publications in from all these research areas (Section 4), we conclude this paper in Section 5.

2 Global Optimization Algorithms

Global optimization is the branch of applied mathematics and numerical analysis that focuses on finding the best possible elements from the space of possible solutions according a set of criteria expressed as mathematical functions. These so-called objective functions may have multiple *local* optima. Still, global optimization algorithms are supposed to be able to find the *globally* optimal results (or at least, to come close to them).

The goal of this paper is to discuss the application of several heuristic and meta-heuristic global optimization methods to distributed systems. The purpose of this section is to give a short overview on these algorithms and to introduce their basic principles.

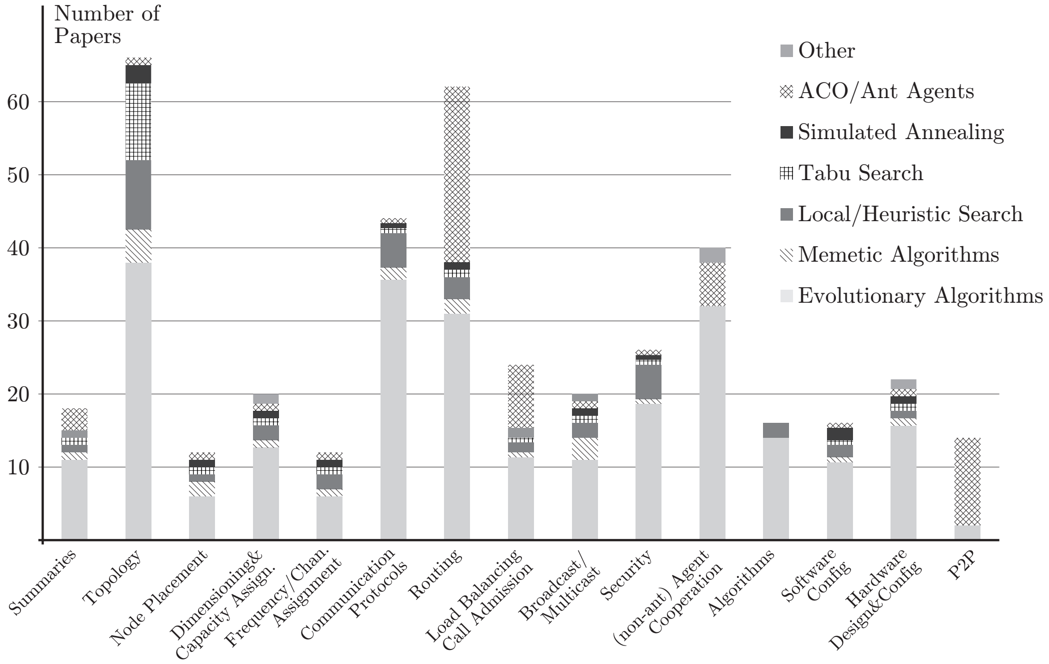


Figure 2: The Number of Papers analyzed, broken down to Area and Optimization Method.

2.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) are generic, population-based, meta-heuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and survival of the fittest. [13, 14, 15]

The advantage of evolutionary algorithms compared to other optimization methods is their "black box" character that makes only few assumptions about the underlying objective functions. Furthermore, the definition of objective functions usually requires lesser insight to the structure of the problem space than the manual construction of an admissible heuristic. EAs therefore perform consistently well in many different problem categories.

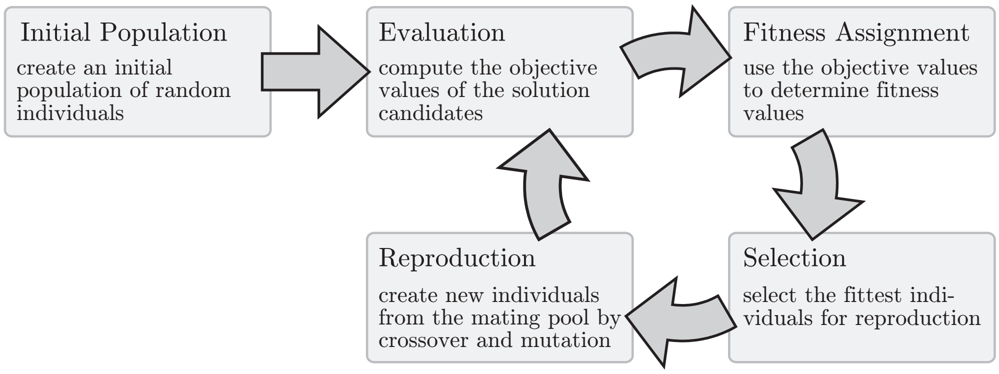

Initial Population
create an initial population of random individuals
Evaluation
compute the objective values of the solution candidates
Fitness Assignment
use the objective values to determine fitness values
Selection
select the fittest individuals for reproduction
Reproduction
create new individuals from the mating pool by crossover and mutation

Figure 3: The basic Cycle of Evolutionary Algorithms.

Similar to the evolutionary process in nature, we can distinguish the space of genotypes $\mathbb{G}$ (called search space) and the space of phenotypes $\mathbb{X}$ (the problem space). $\mathbb{G}$ corresponds to the set of all possible DNA sequences on which the reproductive mechanisms such as sexual recombination and mutation work. In EAs, the definition of $\mathbb{G}$ and the operations applied to its elements $g \in \mathbb{G}$ is often problem-dependent, but there also exists a set of well-researched and widely used representations.

The actual living organisms in nature emerge through a process called embryogenesis from the DNA. This process is copied in evolutionary algorithms in a simplified manner: a genotype-phenotype mapping (gpm) maps genotypes $g \in \mathbb{G}$ to solution candidates (phenotypes) $x \in \mathbb{X}$.

The phenotypes represent the possible solutions of an optimization problem and are rated by a set of objective functions $f \in F$. Based on its objective values in comparison to the objective values of the other individuals in the population, a fitness value can be assigned to each individual. Every living creature has features which influence its fitness positively or negatively. For a fish, such features could be the maximum swimming speed, its size and stamina, whether it is cunning or has sophisticated social behavior, its energy and food consumption and so on. From this point of view, the natural evolution is a multi-objective optimization process, and like in EAs, the objectives may conflict with each other: With the size of the fish, the number of potential predators decreases but its energy consumption will increase. The fitness in nature corresponds to the number of offspring that a creature will produce. The fitness of a fish depends on its "objective values" in relation to the other individuals of the population. A fish considered slow in one school could be the fastest one of another.

Again copying nature, evolutionary algorithms utilize sexual and asexual reproduction, which are implemented in form of crossover and mutation operators and create new genotypes from existing ones. Since the best solution candidates are selected with the highest probability, we expect the offspring resulting from the reproduction to also exhibit good characteristics and, hopefully, even better ones. The cycle of reproduction, genotype-phenotype mapping, objective function computation, fitness assignment, and selection sketched in Figure 3 will be repeated until the termination criterion is met.

2.1.1 Genetic Algorithms

Genetic algorithms (GAs) are a subclass of evolutionary algorithms where the elements of the search space $\mathbb{G}$ are binary strings ($\mathbb{G} = \mathbb{B}^*$) or arrays of other elementary types. [16, 17, 18, 19]

The roots of genetic algorithms go back to the mid-1950s, where biologists like Barricelli [20; 21; 22; 23] and the computer scientist Fraser [24] began to apply computer-aided simulations in order to gain more insight into genetic processes and the natural evolution and selection. Bremermann [25] and Bledsoe [26; 27; 28; 29] used evolutionary approaches based on binary string genomes for solving inequalities, for function optimization, and for determining the weights in neural networks (Bledsoe and Browning [30]) in the early 1960s. . At the end of that decade, important research on such search spaces was contributed by Bagley [31], Cavicchio, Jr. [32; 33], and Frantz [34] – all based on the ideas of Holland at the University of Michigan. As a result of Holland's work [35, 36, 17, 37] genetic algorithms as a new approach for problem solving could be formalized finally became widely recognized and popular.

2.1.2 Genetic Programming

The term *Genetic Programming* [38, 18] has two possible meanings. First, it is often used to subsume all evolutionary algorithms that produce tree data structures as phenotypes. Second, we can also define it as the set of all evolutionary algorithms that breed programs, algorithms, and similar constructs.

The history of Genetic Programming [39] goes back to the early days of computer science. In 1957, Friedberg [40] left the first footprints in this area by using a learning algorithm to stepwise improve a program. The program was represented as a sequence of instructions for a theoretical computer called *Herman* [40, 41]. Friedberg did not use an evolutionary, population-based approach for searching the programs.

The evolutionary programming approach for evolving finite state machines by Fogel et al. [42] dates back to 1966. In order to build predictive behaviors, different forms of mutation (but no crossover) were used for creating offspring from successful individuals.

Fourteen years later, the next generation of scientists began to look for ways to evolve programs. New results were reported by Smith [43] in his PhD thesis in 1980. Forsyth [44] evolved trees denoting fully bracketed Boolean expressions for classification problems in 1981 [44, 45, 46].

Genetic Programming became fully accepted at the end of this productive decade mainly because of the work of Koza [47; 48]. He also studied many benchmark applications of Genetic Programming, such as learning of Boolean functions [49, 50], the Artificial Ant problem [51, 52, 38], and symbolic regression [49, 38], a method for obtaining mathematical expressions that match given data samples.

2.2 Ant Colony Optimization

Inspired by the research done by Deneubourg et al. [53], [54, 55] on real ants and probably by the simulation experiments by Stickland et al. [56], Dorigo et al. [57] developed the Ant Colony Optimization (ACO) Algorithm for problems that can be reduced to finding optimal paths in graphs in 1996. [58, 59, 60, 61, 62] Ant Colony Optimization is based on the metaphor of ants seeking food. In order to do so, an ant will leave the anthill and begin to wander into a random direction. While the little insect paces around, it lays a trail of pheromone. Thus, after the ant has found some food, it can track its way back. By doing so, it distributes another layer of pheromone on the path. An ant that senses the pheromone will follow its trail with a certain probability. Each ant that finds the food will excrete some pheromone on the path. By time, the pheromone density of the path will increase and more and more ants will follow it to the food and back. The higher the pheromone density, the more likely will an ant stay on a trail. However, the pheromones vaporize after some time. If all the food is collected, they will no longer be renewed and the path will disappear after a while. Now, the ants will head to new, random locations.

This process of distributing and tracking pheromones is one form of stigmergy and was first described by Grassé [63]. Today, we subsume many different ways of communication by modifying the environment under this term, which can be divided into two groups: sematectonic and sign-based [64]. According to Wilson [65], we call modifications in the environment due to a task-related action which leads other entities involved in this task to change their behavior sematectonic stigmergy. If an ant drops a ball of mud somewhere, this may cause other ants to place mud balls at the same location. Step by step, these effects can cumulatively lead to the growth of complex structures. Sematectonic stigmergy has been simulated on computer systems by, for instance, Théraulaz and Bonabeau [66] and with robotic systems by Werfel and Nagpal [67], [68, 69].

The second form, sign-based stigmergy, is not directly task-related. It has been attained evolutionary by social insects which use a wide range of pheromones and hormones for communication.

The sign-based stigmergy is copied by Ant Colony Optimization [57], where optimization problems are visualized as (directed) graphs. First, a set of *ants* performs randomized walks through the graphs. Proportional to the goodness of the solutions denoted by the paths, pheromones are laid out, i. e., the probability to walk into the direction of the paths is shifted. The ants run again through the graph, following the previously distributed pheromone. However, they will not exactly follow these paths. Instead, they may deviate from these routes by taking other turns at junctions, since their walk is still randomized. The pheromones modify the probability distributions.

2.3 Simulated Annealing

In 1953, Metropolis et al. [70] developed a Monte Carlo method for "calculating the properties of any substance which may be considered as composed of interacting individual molecules". With this so-called "Metropolis" procedure stemming from statistical mechanics, the manner in which metal crystals reconfigure and reach equilibria in the process of annealing can be simulated. This inspired Kirkpatrick et al. [71] to develop the Simulated Annealing (SA) algorithm for global optimization in the early 1980s and to apply it to various combinatorial optimization problems. Independently, Černý [72] employed a similar approach to the travelling salesman problem [73, 74]. Simulated Annealing is an optimization method that can be applied to arbitrary search and problem spaces. Like simple hill climbing algorithms, Simulated Annealing only needs a single initial individual as starting point and a unary search operation.

In metallurgy and material science, annealing is a heat treatment of material with the goal of altering its properties such as hardness. Metal crystals have small defects, dislocations of ions which weaken the overall structure. By heating the metal, the energy of the ions and, thus, their diffusion rate is increased. Then, the dislocations can be destroyed and the structure of the crystal is reformed as the material cools down and approaches its equilibrium state. When annealing metal, the initial temperature must not be too low and the cooling must be done sufficiently slowly so as to avoid the system getting stuck in a meta-stable, non-crystalline, state representing a local minimum of energy.

Simple hill climbing algorithms iteratively create new solution candidates x_{i+1} from an existing one x_i and move on to this new offspring if it has better objective values only. Simulated Annealing enhance this scheme by also accepting worse solution candidates with a non-zero probability $P(\Delta f)$ which exponentially decreases with the number of iterations t. Here, the objective function is subject to minimization and corresponds to the energy level of annealing steel. k_B is the Boltzmann constant.

$$\Delta f = f(x_{i+1}) - f(x_i) \quad (1)$$

$$P(\Delta f) = \begin{cases} e^{-\frac{\Delta f}{k_B t}} & \text{if } \Delta f > 0 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

Nolte and Schrader [75] and van Laarhoven and Aarts [76] provide lists of the most important works showing that Simulated Annealing will converge to the global optimum if $t \to \infty$ iterations are performed, including the studies of Hajek [77]. Nolte and Schrader [75] further list research providing deterministic, non-infinite boundaries for the asymptotic convergence by Anily and Federgruen [78], Gidas [79], Nolte and Schrader [80], Mitra et al. [81].

2.4 Tabu Search

Tabu Search (TS) has been developed by Glover [82] in the mid 1980s [83]. Some of the basic ideas were introduced by Hansen [84] and further contributions in terms of formalizing this method have been made by Glover [85; 86], and de Werra and Hertz [87] (as summarized by Hertz et al. [88] in their tutorial on Tabu Search) as well as by Battiti and Tecchiolli [89] and Cvijović and Klinowski [90].

The word "tabu" stems from Polynesia and describes a sacred place or object. Things that are *tabu* must be left alone and may not be visited or touched. Tabu Search extends hill climbing by this concept – it declares solution candidates which have already been visited as tabu. Hence, they must not be visited again and the optimization process is less likely to get stuck on a local optimum. The simplest realization of this approach is to use a list which stores all solution candidates that have already been tested. If a newly created phenotype can be found in this list, it is not investigated but rejected right away. Of course, the list cannot grow infinitely but has a finite maximum length n. If the $n+1$st solution candidate is added, the first one must be removed. If some distance measure in the problem space $\mathbb{X}$ is available, a certain perimeter around the listed solution candidates can be declared as tabu. More complex approaches will store specific properties of the individuals instead of the phenotypes themselves in the list. This will not only lead to more complicated algorithms, but may also reject new solutions which actually are very good. Therefore, aspiration criteria can be defined which override the tabu list and allow certain individuals.

3 Examples

In the following sections, we will discuss a set of interesting applications from five different areas of network optimization. For the sake of simplicity, we use a coarser classification of these areas as done in the very fine-grained Figure 2. While the delivery of broadcast or multicast messages is considered as a distinct topic in Figure 2, for instance, we will subsume it here under the subject of *routing*.

3.1 Optimizing Network Topology

The topology of a network is a graph that defines which of the nodes are able to communicate with each other. The structure of any network, be it a distributed system, a social network, or a transportation system, can be described as such a graph. Especially for communication networks, the topology and the features of the nodes and links are very important since they define the maximum throughput, the latency and hops between the nodes, the robustness in terms of link failures, the installation costs, etc. These are also exactly the objectives which can be optimized when global optimization algorithms are used to design the networks, the network topologies, and the features of the hardware to be utilized.

The application of Evolutionary Computation in this area has a very long tradition, starting with the works of Coombs and Davis [91], Michalewicz [92], and Kumar et al. [93] between 1987 and 1992. Before the year 2000, more than 30 papers had been published in this area [94] and their number is still increasing. In this section, we will outline three interesting applications of topology optimization tasks: the Terminal Assignment Problem, the self-organized improvement of singular networks, and a European research initiative for planning a large-scale optical network.

3.1.1 Solving the Terminal Assignment Problem

The goal of the Terminal Assignment (TA) problem is to determine the minimum cost links to connect a given set of nodes (the terminals) to another (disjoint) set of nodes (the concentrators). The required capacity of each terminal is known and may vary from terminal to terminal. The capacity of the concentrators is also known and so are the costs for linking them to the terminals. Each terminal must be linked to exactly one concentrator in a way that the maximum capacity of no concentrators is exceeded. An assignment has to be found under the objective of minimizing the total costs. This problem is NP-hard [95], except for the special case where all terminals have the same capacity requirements and all concentrators have the same capacity.

Abuali et al. [96] were the first researchers who applied genetic algorithms to instances of this problem where the capacity of all concentrators was equal. Khuri and Chiu [97] investigated the utility of different heuristic algorithms, including greedy approaches and genetic algorithms for the general case where the capacities of the single concentrators differ. In their first genetic approach, Khuri and Chiu [97] use a terminal-centric representation in form of a string genome $\mathbb{G} = \mathbb{X} = \mathbb{N}^n$ consisting of natural numbers, where n is the number of terminals. The value of a gene s_i from the genotype/phenotype string $S = (s_1, s_2, .., s_n)$ stands for the concentrator to which the i^{th} terminal is connected. Strings that were infeasible, i. e., violate at least one of the constraints given above, are penalized with an offset separating them from all feasible solutions. Additionally, Khuri and Chiu [97] also applied their grouping genetic algorithm (GGA) to this problem. The results of their work indicated that the original genetic algorithm provided better results than the greedy approach which, in turn, outperformed the grouping GA.

Yao et al. [98] applied a hybrid evolutionary algorithm (i. e., a Memetic Algorithm) to a multi-objective variant of this problem where, besides the total cost, also the number of concentrators used is to be minimized. They furthermore abandon the terminal-centric representation in favor of a concentrator-centric genome consisting of m trees of depth 1, where m is the number of concentrators. Each concentrator is represented as root of a tree and terminals are leaves linked to these roots. If a root has no children, then the corresponding concentrator is not used. The experimental results of Yao et al. [98] showed that evolutionary algorithms hybridized with Lamarckian or Baldwin local search [99] both perform approximately equally well and are suitable for solving the terminal assignment problem. The concentrator-based representation also showed to advantageous in these experiments compared with the terminal-centric genome.

3.1.2 Singular, Selfish, and Self-Organized (S^3) Networks

Another interesting topology optimization approach is the iterative online method for self-organizing networks of selfish nodes defined by Zweig [100]. Her algorithm is not population-based – at any point in time there is exactly one (singular) network. In each time step, one node may modify the edges to its neighbors by either adding one connection (from itself) to another node or removing one edge attached to it.

If this modification seems to be advantageous from the perspective of the node (therefore *selfish*), it is committed, otherwise it is rejected. As a measure for the utility of a modification, a node may, for instance, compute the maximum distance from itself to any other node or the sum of the distances to all other nodes. In [101], it is shown that self-organization algorithms minimizing such objectives can decrease the diameter of a network down to a certain threshold in polynomial time.

Zweig [100] refers to this approach as evolutionary algorithm. Since its population has the size one and only a unary reproduction operator is used, one may rather consider it as a hill climbing with “evolutionary” aspects in the sense of steady improvements.

3.1.3 Optimization of an Optical Network for Europe

The last application of global optimization to topology optimization that we will summarize here is the research project 239 of the European Cooperation in the field of Scientific and Technical Research (COST). The partners in the COST Action 239 [102] studied the feasibility of plans for an European Optical Network (EON), a transparent optical overlay network capable of carrying all the international traffic between twenty of the main centers of Europe [103]. The initial design for this network is illustrated in Figure 4.

Sinclair [104] evolved a network topology for EON using a genetic algorithm with an objective function that took the estimated traffic on the links between the nodes and weights of the nodes themselves into account. This way, he was able to achieve significant improvements of both, costs and reliability, compared to the topology initially designed [105]. The same group of researchers later applied a similar GA to also optimize the route and wave length assignment for such a network. Again, they obtained results superior to initial expectations [106].

In [103] and [94], they tried out four different Genetic Programming approaches for the topology optimization problem and compared them with the initial genetic algorithm. We will outline the representations used in these approaches on the example from [104] concerning only the $N = 9$ central nodes (Amsterdam, Berlin, Brussels, Luxembourg, London,

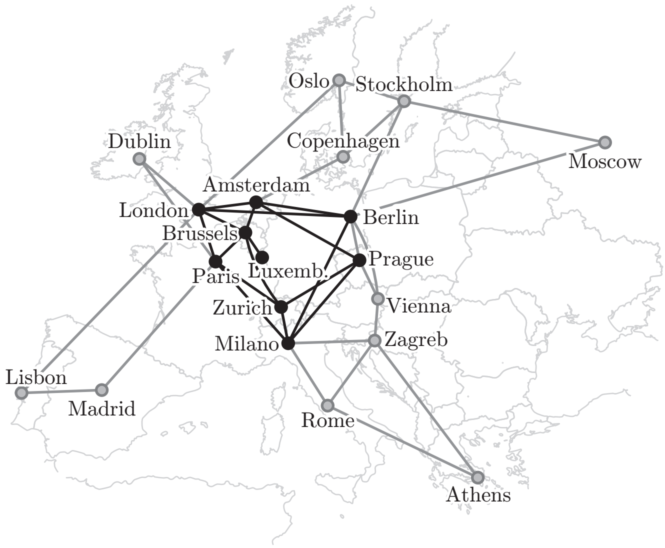
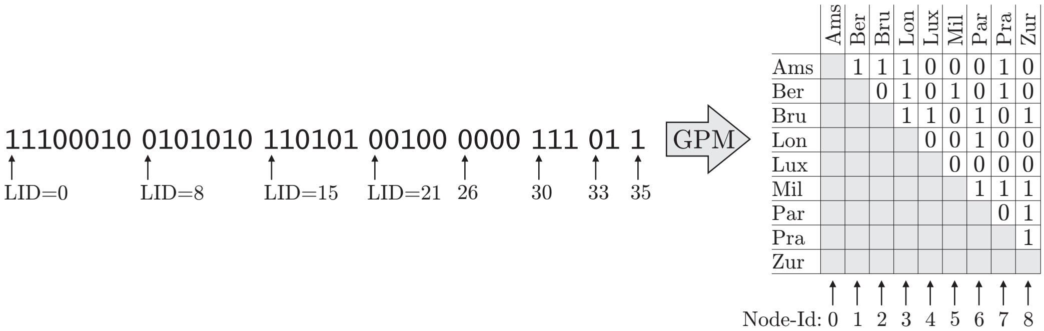


Figure 4: The initial network design of the EON network [104, 105].

Milano, Paris, Prague, and Zurich). In Figure 4, the black dots stand for these central nodes and the black lines for the connections between them.

The Genetic Algorithm In the genetic algorithm, Sinclair used a genome where each of the $L = \frac{1}{2}N(N-1) = \frac{1}{2} * 9 * 8 = 36$ possible links between these nine metropolises was represented by one bit. When a bit in the genotype is set, the corresponding link is present in the network plan. The central topology from Figure 4 thus translates to the genotype in Figure 5 (which effectively encodes the connection matrix).



	Ams	Ber	Bru	Lon	Lux	Mil	Par	Pra	Zur
Ams		1	1	1	0	0	0	1	0
Ber			0	1	0	1	0	1	0
Bru				1	1	0	1	0	1
Lon					0	0	1	0	0
Lux						0	0	0	0
Mil							1	1	1
Par								0	1
Pra									1
Zur									
Node-Id:	0	1	2	3	4	5	6	7	8

Figure 5: The genotype representing the connection matrix of the nine central nodes of EON.

Genetic Programming with a Relational Function Set In the Genetic Programming approach with a relational function set (RFS), a function `isLinkThere(LID)` was evolved which evaluates to `1` if the link `LID` is present and `0` otherwise. Therefore, each of the $L = 36$ possible links is associated with a unique number from the interval $[0..L-1]$. The terminal symbols available for GP are the link ID `LID` (i. e., the parameter of `isLinkThere`) and constants. As function set, addition (`+`) and subtraction (`-`) *modulo* L are provided together with greater than (`>`) and less than (`<`) which return `1` if their first argument is greater or lesser than their second one. The decision function `if` returns the value of its second argument if the first one equals to `1` and the value of the third one otherwise. Since the function evolved could return values from 0 to $L - 1$, a wrapper function was used which mapped these values to either `0` or `1`. Fig. 6.1 sketches how such a symbolic regression-like GP representation of the bit string from Figure 5 may look like, using the `LID`s as defined there.

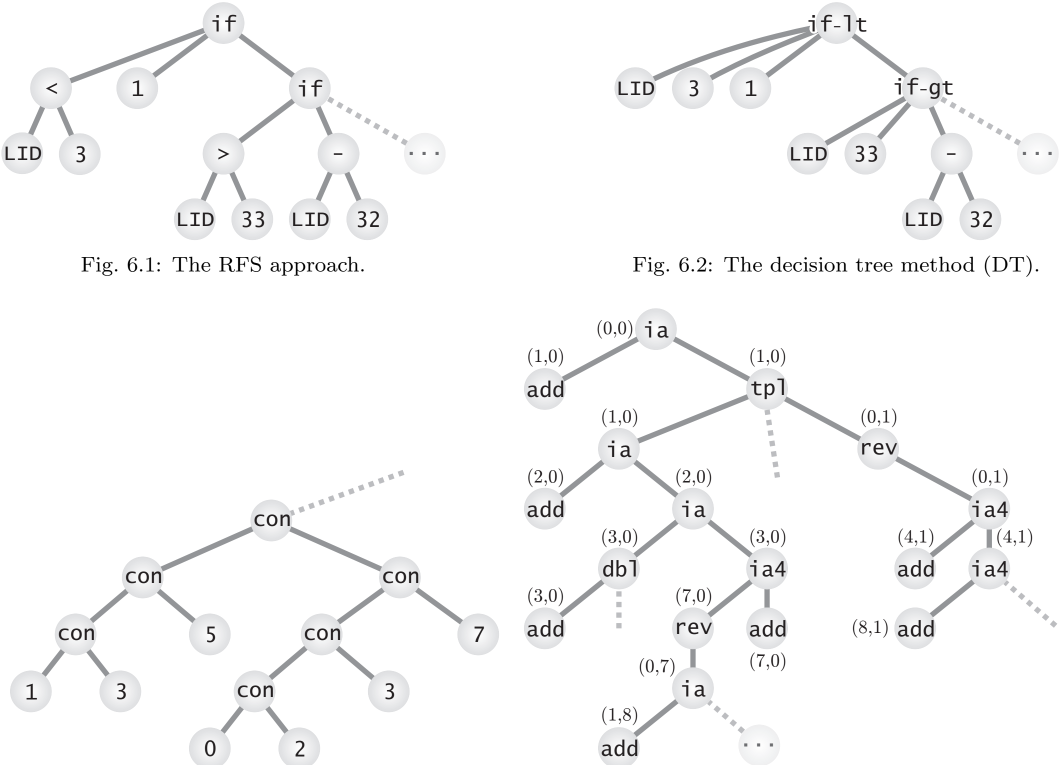


Fig. 6.1: The RFS approach.

Fig. 6.2: The decision tree method (DT).

Fig. 6.3: The connective approach (CN).

Fig. 6.4: The node-pair encoding GP (NP2).

Figure 6: Snippets from the GP representations of the bit string from Figure 5

Genetic Programming for Decision Trees With their DT-Genetic Programming approach, Aiyarak, Saket, and Sinclair [103] extended this method by combing the `<` and `>` operators with the `if` statement, yielding the new `if-lt` and `if-gt` expression. With this modification, they wanted to evolve decision trees and hoped to reduce the computational effort for finding solutions. For illustration purposes, we translated the program from Fig. 6.1 to this representation and depict the result in Fig. 6.2.

Genetic Programming – the Connective Approach The third Genetic Programming approach (called *connected nodes*, or CN for short) differs very much from the other two. As terminal symbols, it only uses node identifiers and it only has one function, `con` [94]. `con` simply adds a link between the two nodes identified by its parameters and returns its first argument. The tree in Fig. 6.3 describes how the network defined in Figure 5 can be derived.

Node-Pair encoding GP Inspired by the *edge encoding* method of Luke and Spector [107], Sinclair designed another Genetic Programming approach to the problem. His *node-pair encoding* (NP) involves a tree-to-graph genotype-phenotype mapping which explicitly

Abbreviation	NP1	NP2	Description
`rev`	1	1	reverse: $(a, b) \to (b, a)$
`da`	1	2	modulo-decrement a: $(a, b) \to (a - 1 \bmod N, a)$
`ia`	1	2	modulo-increment a: $(a, b) \to (a + 1 \bmod N, a)$
`ia2`	1	2	modulo-increment a by 2: $(a, b) \to (a + 2 \bmod N, a)$
`ia4`	1	2	modulo-increment a by 4: $(a, b) \to (a + 4 \bmod N, a)$
`ia8`	1	2	modulo-increment a by 8: $(a, b) \to (a + 8 \bmod N, a)$
`dbl`	2	2	plain doubling: pass node pair to both children
`tpl`	3	3	plain tripling: pass node pair to all three children
`add`	1	0	if $a \neq b$ create link between a and b
`cut`	1	0	if a and b are linked, remove the link
`nop`	0	0	do nothing (terminal)

Table 1: The Node-Types of the NP-approaches.

constructs the network [94, 108]. Table 1 illustrates the function and terminal sets of two NP variants, which only differ in the number of children of the function nodes (noted in the columns NP1 and NP2). NP programs work with tuples (a, b) of node IDs with $a, b \in [0..N-1]$ ($N = 9$ for our example). The program tree is processed from top to bottom and begins with the input $(0, 0)$ into the root node. Each node can either manipulate the first element of the tuple (with the functions `da`, `ia`, `ia2`, . . .), rotate the tuple (`rev`), create or cut a connection between the two nodes in the tuple (`add` or `cut`), or simply does nothing (`nop`). If it has children, it will pass the (resulting) node tuple to all its children. Fig. 6.4 illustrates how a program snippet in NP2 representation is executed and the values of the node tuples at the different levels.

Summary In [103], Aiyarak, Saket, and Sinclair tested the first three GP approaches (RFS, DT, and CN) and found that none of them produced solutions with better fitness than the original genetic algorithm method. RFS-GP performed worst. The DT method was better than CN on a small test set containing only the nine central nodes and scored even with the GA. On the complete EON network, CN in turn outperformed DT but was still worse than the solution provided by the genetic algorithm. All Genetic Programming approaches needed much (at least five times) more individual evaluations to find their solution than the genetic algorithm. These results were confirmed in [94], where CN and the two versions of NP were compared with the GA. Again, the GA was the best approach, but NP2 found the best result for one of the five test problems and came close to the GA in all other runs. Still, the experiments with Genetic Programming needed much longer than those with the genetic algorithm.

The lesson from these experiments is that the representation issue, i. e., the way in which solution candidates are encoded, is very important in optimization. Sometimes a simple, fixed-length bit string can outperform more complex approaches. And even if one class of algorithms (like GP) is applied, modifications in the representation used can still lead to vast differences in performance.

3.2 Optimizing Routing

Routing is the way in which messages are relayed over multiple nodes in a network in order to reach their destination. Naturally, there often exist multiple different routes from a source to a destination. The objective of a routing algorithm is to guide the messages along the shortest possible routes. A path in the network gets blocked if any of its nodes is congested, i. e., has to process too many message at once. On the other hand, additional paths may become available as new nodes are connected. Hence, routing algorithms have to avoid congestion and should incorporate new links as soon as possible in order to increase the network utility. In this section, we will discuss six approaches for improving routing with global optimization algorithms, spanning from the evolution of routing rules, adaptive routing with ant agents, to the synthesis of broadcast algorithms and the enhancement of search request distribution in peer-to-peer networks.

3.2.1 Evolving Fault-tolerant Routing Rules

Routing is tightly coupled with the problem of finding the shortest paths from one vertex in a graph to all others. This problem has been solved in 1959 by Dijkstra [109]. About forty years later, Kirkwood et al. [110], [111] use Genetic Programming to obtain such paths by evolution. Their approach is not in competition with Dijkstra's algorithm, since it aims to breed robust routing rules for networks where links may fail.

The Genetic Programming system of Kirkwood et al. [110] takes the ID of a target node and a graph (containing this node) as input. Each node in the graph has a unique ID and each connection has a certain length. The system then breeds routing rules in form of LISP S-expressions. In the context of this application, these expressions are trees consisting of nested instances of the domain-specific `IF-CUR-GO` instruction. The four parameters `W`, `X`, `Y`, and `Z` of this instruction as well as its return value each represent a node ID. On each node reached by a message which is travelling through the network described by the graph, the routing rule is evaluated and it is sent to the node whose ID is returned. `IF-CUR-GO W X Y Z` evaluates to `X` if the current node is `W`, `X` and `W` are directly connected, and `X` has not yet been visited twice. It returns `Y` if one of the latter two conditions is violated and `Z` if the current node is not `W`.

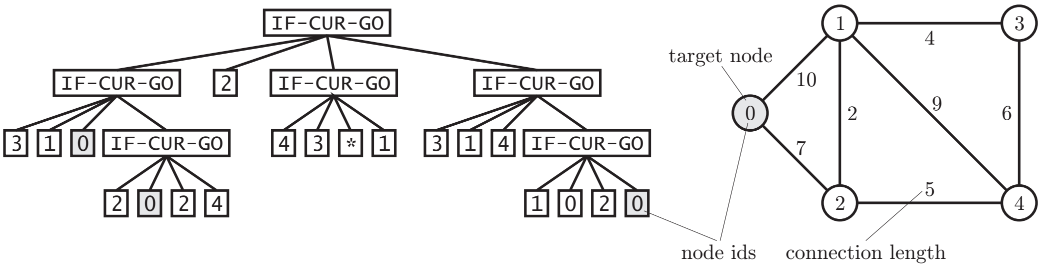


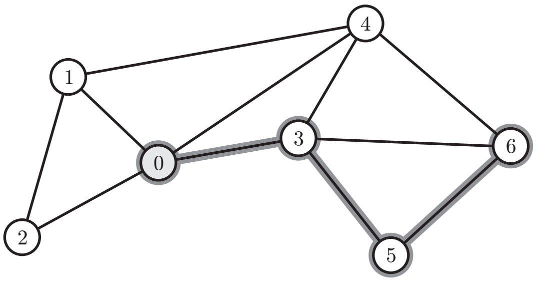
Figure 7: An evolved routing rule set (left) with target node 0 for the specified network (right).

As objective function, Kirkwood et al. [110] set the difference in length between the optimal route (computed with Dijkstra's algorithm) and the evolved route. As result, they obtain rules that are very robust in terms of link failures that, for instance, find correct routes in two third of all cases even if 47% of the connections have broken down. In Figure 7, a routing rule set resulting from one of the experiments conducted by Kirkwood et al. [110] is displayed on the left. It routes messages through the graph sketched on the right-hand side to Node 0.

3.2.2 Genetic Adaptive Routing Algorithm

In the internet, messages are forwarded to their destinations on the basis of routing tables. These tables can be filled using, for example, Dijkstra's algorithm. They contain either the address of the next hop, i. e., the next computer to send a packet to, or the complete routes to each possible destination (source routing). Protocols like RIP [112] and OSPF [113] build and distribute such tables, but disregard dynamic aspects like communication latency.

Munetomo et al. [114; 115; 116], [117] contributed GARA, the genetic adaptive routing algorithm, for building and maintaining routing tables for source routing. GARA runs independently on every node in the network. Each node maintains a list of multiple complete routes to the most frequently contacted communication partners. After a specific number of packets have been sent along a route, an additional packet is sent for observing the communication latency. The fitness of the routes and their likeliness for being chosen then corresponds to this delay. In Figure 8, the table with the destinations most frequently addressed by node 0 is sketched in the context of an example network.



Dest.	Routes	Delay	Weight
1	(0,1)	10	1.0
2	(0,1,2)	100	0.4
	(0,2)	80	0.6
5	(0,3,5)	270	0.7
	(0,4,6,5)	420	0.3
6	(0,1,4,3,6)	210	0.4
	(0,3,6)	250	0.5
	(0,3,5,6)	490	0.1

Figure 8: The table with the most frequently used routes of node 0.

From time to time, *path genetic operators* such as path mutation and path crossover, both illustrated in Figure 9, are applied to routes with the same destination. With these operators, new paths are discovered and made available for message propagation. Path mutation simply exchanges a node within a route with another one (Fig. 9.1). Path crossover (Fig. 9.2) is applied to two routes that pass through one common point (other than their source and destination). The sub-paths before and after that point are exchanged. Selection is performed when the routing table exceeds a specific size. By utilizing these evolutionary primitives, GARA becomes a self-organizing system able to adapt to changing network situations and to discover new routes challenging established protocols on equal footing.

3.2.3 Ant-based Routing

The routing of network messages or phone calls through a telecommunication system remotely resembles the transportation of food by ants to their nest [9].

First Steps by Schoonderwoerd et al. Schoonderwoerd et al. [64; 118], [119] have developed an routing method based on this simile in form of an online Ant Colony Optimization approach. In their systems, the nodes exchange control messages (the ants) from time to time in order to build and maintain "pheromone tables" used for routing. In principle, a pheromone table of a node with m neighbors is a $n{\times}m$ matrix where n is the total number of nodes in the network. For any of the n possible destinations, it contains a probability that a message is routed to any of the m neighbors. At each time step during Schoonderwoerd's simulations, any node can launch an ant with a random destination. Whenever an ant created by node i arrives at node j coming from node k, it will update the pheromone table P_j of j according to

$$\begin{aligned} P_{j\,[i,k]} &= \frac{P_{j\,[i,k]}+\Delta}{1+\Delta} && (3) \\ \forall l \in [1..n], l \neq k \;\; P_{j\,[i,l]} &= \frac{P_{j\,[i,l]}}{1+\Delta} && (4) \end{aligned}$$

In other words, the probability of selecting node k as next hop for future messages with destination i is increased while decreasing the probabilities of the other possible next hops of

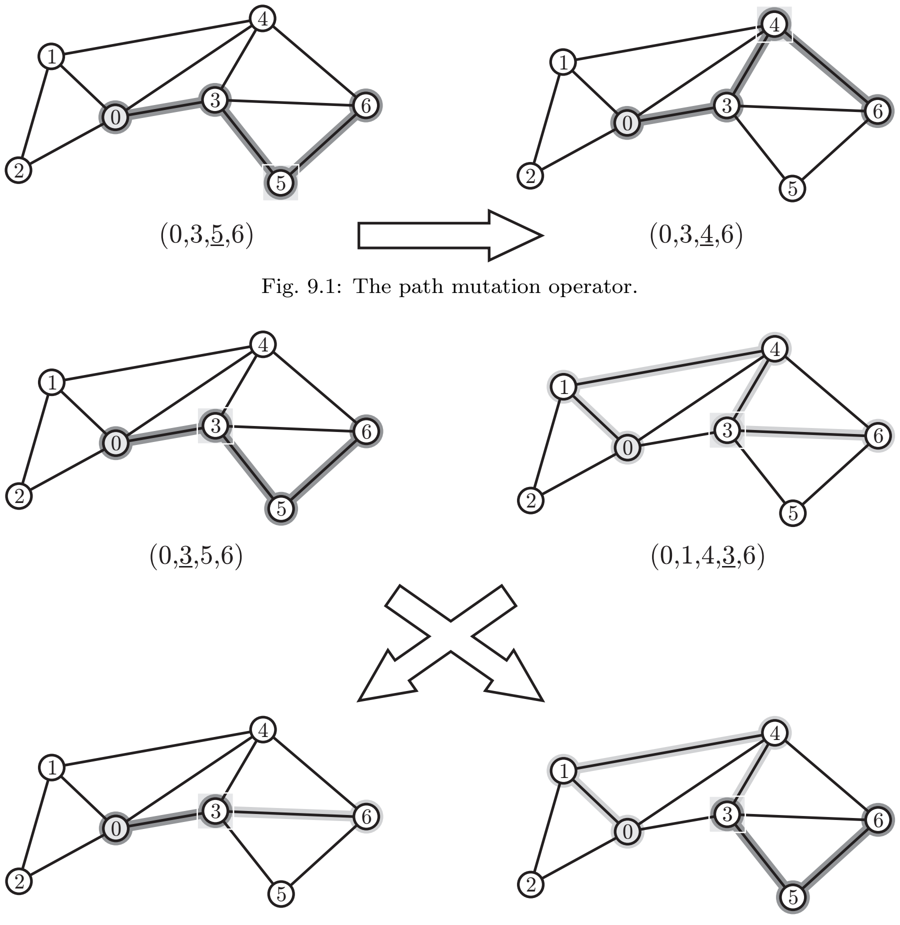


Fig. 9.1: The path mutation operator.

Fig. 9.2: The path mutation operator.

Figure 9: The path genetic operations in GARA.

such a message. The total cumulative probability of all hops together stays 1. This update method modifies the probabilities for messages travelling into the opposite direction of the ants. It indirectly influences the forward probability of the path taken by the ant since subsequent ants moving to node i along it will again have backward influence which then points into the original forward direction.

Schoonderwoerd adds an "age" to the ants which is increased in every hop in order to foster the usage of short pathways. Δ becomes a function which decreases with the age of the ant. Thus, the older an ant, i. e., the longer the path it has travelled, the lower will its influence on guiding future ants along this path be. Congested relay stations can artificially increase the age of the ants passing them and may also delay the ants on their way. This will decrease their traffic.

New routes can be discovered by introducing a factor $f : 0 < f < 1$ which denotes a probability that the next hop of an ant is chosen totally randomly whereas the pheromone table is used for the decision with probability $1 - f$.

In summary, Schoonderwoerd et al. [64] presented a decentralized and self-organized online optimization process for communication networks which can be generalized to arbitrary routing problems. The simple behavior of their ants leads to interesting, emergent overall configurations of the network it is applied to.

AntNet Di Caro and Dorigo [120; 121], [122] develop this concept further in their AntNet algorithm. For each node j, it builds a routing table P_j which has basically the same structure as the ones used by Schoonderwoerd et al.[2]. What changed is the way that this matrix is updated.

Regularly, every node s sends an agent (*forward ant*) $F_{s\to d}$ to a node $d \neq s$ in order to discover feasible and efficient paths. The selection of d is randomized but biased into the direction of nodes which exchange high traffic with s. Forward ants share the same queues as data packets, so they will experience the same delays [120]. These ant agents keep a history of their route and of the times needed for its sections.

At each node j, the travelling ant $F_{s\to d}$ selects the next node to visit according to:

$$P(j \to i) = \frac{P_{j\,[d,i]} + \alpha l(i)}{1 + \alpha\,(m-1)} \tag{5}$$

Where $P(j \to i)$ is the probability that i is chosen as next node and $P_{j\,[d,i]}$ is the routing table entry which describes the expected utility of i as next hop for messages going to d.[3] m is the number of neighbors of node j and $l(i)$ is a value proportional to the queue length for destination i normalized to the unit interval. The influence of $l(i)$ makes the system more reactive to the current traffic situation. It is weighted by the factor α which usually takes on values between 0.2 and 0.5.

If the forward ant $F_{s\to d}$ detects that it somehow travelled in a loop, it dies. Otherwise, when it arrives at its destination d, it is converted into a backward agent $B_{d\to s}$ which travels back to s along the same route that also $F_{s\to d}$ took. When arriving at a node j coming from a node i, the entry for choosing i as next hop for messages going to d in the routing table of j is updated as follows:

$$P_{j\,[i,d]} = P_{j\,[i,d]} + r\,(1 - P_{j\,[i,d]}) \tag{6}$$

where $r \in [0, 1]$ is a reinforcement factor, a measure of goodness of the observed trip time, based on a statistical model for the local traffic M (also updated by the agents) and the time. By multiplying r with $(1 - P_{j\,[i,d]})$, small probabilities are increased proportionally more, which allows faster exploration of new, potentially good routes. After this update, the corresponding column of P_j is normalized again.

[2] Well, except for naming changes and that the rows and columns are switched. We will ignore these minor differences in here.

[3] Again, notice that we have switched the columns and rows in Di Caro and Dorigo's approach in order to relate it to the one of Schoonderwoerd et al..

Global Information? The simulations of Di Caro and Dorigo [120] show that AntNet is an efficient and robust routing algorithm. However, they make the assumption that global information is available. Every node holds a routing table with entries defining how to send messages to any other node. Such global information is normally not available and furthermore would also be subject to constant changes if routers are added to or removed from the network.

Liang et al. [123; 124] point out this weakness and test a LocalAnt algorithm restricted to routing tables with only two entries for each neighboring node (for the neighbor itself and the probability to route over this node for other destinations). They show that, under these more realistic conditions, the efficiency of the algorithm is much lower than in the case where global information is available (the original approach is called GlobalAnt in this context). Liang et al. [124] propose a distributed algorithm where each node holds a population of individuals encoding routes. These individuals travel over the network as genetic agents and, subsequently, as backward agents similar to the AntNet approach and are subject recombination, mutation, and aging. According to their study [124], the performance of this method falls in between those of LocalAnt and GlobalAnt.

3.2.4 Genetic Programming of Broadcasting Algorithms

Routing does not necessarily concern the delivery of messages to single destinations only, but may also involve $1 : n$ communications, i. e., broadcasts and multicast. Broadcasting in a graph is the process of spreading information, which is initially known to only one node, to all other nodes. Comellas and Giménez [125] formulated an optimization problem with the following constraints:

- Only a node which already knows the information can spread it further.
- A node can only send one message per time step.
- A node can only send a message over vertexes which are connected to it.

The goal of their work was to find the broadcasting scheme which disseminates the information to all nodes in the shortest possible time. They used Genetic Programming for growing such broadcasting algorithms for directed 2-grids, toroidal grids, hypercubes, cube-connected cycles, and butterfly graphs. For the butterfly graph, a solution even better than the best known upper bound at that time (by Klasing et al. [126]) was found, whereas in all other cases, (known) optimal algorithms evolved.

Example: Directed 2-Grid In their experiments, Comellas and Giménez [125] used Standard Genetic Programming for evolving Lisp S-expressions. The function set for the directed 2-grid example was contained the following expressions:

- `IfOri` executes its parameter action if the node the program runs on was the one who initially knew the information to be disseminated.
- `IfProcHor` and `IfProcVer` execute their parameter if they received the information from a horizontally or vertically adjacent node, respectively.
- `Proc1`, `Proc2`, `Proc3`, and `Proc4` concatenate 1, 2, 3, or 4 instructions to a sequential block.
- `IfTurn0`/`IfTurn1` execute their actions only in the first/second time step after the node received the information.

As terminal symbols, `MoveHor` and `MoveVer` are available, which send the information to the next horizontally or vertically adjacent node, as well as `NULL` which does nothing. For computing the objective values, the algorithms evolved were applied to a graph. The number of nodes reached was counted, multiplied with a factor T. From this value, 2 are subtracted for each

```
1 (Prog3 (IfOri      (Prog2 (IfTurn0 MoveHor)
2                           (IfTurn1 MoveVer)))
3        (IfProcHor (Prog3 (IfTurn0 MoveHor)
4                           (IfTurn0 MoveVer)
5                           (IfTurn1 MoveHor)))
6        (IfProcVer (IfTurn0 MoveVer)))
```

Listing 1: The optimal broadcasting scheme evolved for 2-dimensional, directed grids.

condition value and `NULL` action and 4 for each other action in the tree. The connective `ProgN` functions had no influence on fitness. Listing 1 illustrates the result delivered by Genetic Programming for the 2-dimensional directed grid.

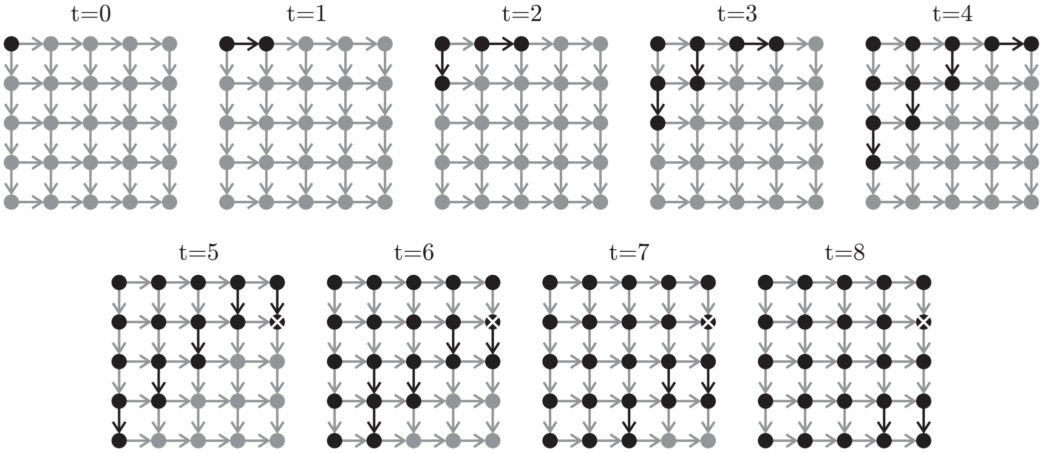


Figure 10: The application of Listing 1 to a 5 × 5 grid.

In Figure 10, we applied this algorithm to a 5×5 grid. The gray nodes have no knowledge about the information and turn black once they receive it. It is interesting to know that the `IfTurn0 MoveVer` in line 4 of Listing 1, although seemingly useless, is important. Without it, the last node in the second line of Figure 10 would not be reached in time step 5 but in step 6, since the `MoveHor` action cannot be executed by the last node in the first line. With the `MoveVert` instruction with the same preconditions, an unnecessary pause is avoided. For the other topologies mentioned, Comellas and Giménez [125] used modified function and terminal sets and obtained similarly optimal results.

3.2.5 Optimizing Collective Communication

Recently, Jaroš and Dvořák [127] developed a scheduling technique for similar collective communication (CC) tasks on basis of evolutionary algorithms. Their work addresses high-performance multi-processor systems where the single CPUs are connected via NoCs (networks on chip). Jaroš and Dvořák experiment with one-to-all broadcast (OAB), all-to-all broadcast (AAB), one-to-all scatter (OAS, a private message to each partner), and all-to-all scatter (AAS) [127].

Each collective communication process is considered as a set of point-to-point communications, and the CC scheduling problem is defined as partitioning this set into as few as possible subsets. The communications in each subset can be executed in parallel and the subsets themselves are executed one after the other in a sequence of synchronized steps. The objective is to find a minimal partition without causing conflicts.

Jaroš and Dvořák [127] introduce different encodings for broadcast and scatter communication as well as special reproduction and repairing operators. The objective function corresponds to the number of conflicts that a schedule introduces, i. e., the number of times that two point-to-point communications share the same link in the same time step. The experimental results showed that this approach is very efficient and even led to an improvement of the theoretical lower bounds of the number of steps needed for AAS communication.

3.2.6 Improving Routing in Gnutella

The Gnutella network [128, 129] is one of the oldest and simplest peer-to-peer file sharing systems. In Gnutella, each node only interacts with neighboring nodes which it chooses at startup. Normally, a node picks seven such neighbors and tries to maintain connections to them.

As last application of optimization algorithms to routing in this paper, we summarize how Iles and Deugo [130] use Genetic Programming for improving routing in such networks. They evolve rules for dynamically setting the number of neighbor nodes and for selecting (and possibly exchanging) them. These rules use information the peer-to-peer node can aggregate over time, like the (current) number of neighbors, the bandwidth of the node, the bandwidth of the neighbor nodes, and the duration of the connections to them. The expressions produced by Genetic Programming led to significant performance improvements in simulations. The results of the experiments of Iles and Deugo [130] still use a fixed number of neighbors, but tend to prefer five or six partner nodes instead of Gnutella's default of seven. The way these neighbors are picked, however, becomes a dynamic process depending on multiple aggregated parameters such as the distance in hops and number of search hits to the candidate nodes.

3.3 Synthesizing Protocols

Protocols like IP [131] and TCP [132] are the rules for message and information exchange in a distributed system. Depending on the application, protocols can become arbitrarily complex and strongly influence the efficiency and robustness of a distributed system.

3.3.1 Transformation of Petri Net-based Specifications

In [133], Yamaguchi et al. define the problem of transforming a *service* specification in form of a Petri Net with registers (PNR) to a *protocol* specification in the same format. Later, El-Fakihy et al. [134] show how to solve this problem efficiently with genetic algorithms under the objective of minimizing communication costs.

3.3.2 Transformation of Finite State Machine-based Specifications

A similar approach for synthesizing protocol specifications from service specifications has been contributed by de Araújo et al. [135; 136]. We will use the communication protocol example given in [135] to summarize this method in which all specifications are represented as finite state machines.

In Figure 11, we sketch the starting from [135]. At the left side, a global service specification is defined in form of a finite state machine which describes the interactions that take place in a communication protocol between a sender and a receiver. It contains only the interaction *primitives* between a user (or a user application) and the peer entities, where `?xyz` represents input message `xyz` from and `!xyz` stands output of `xyz` to the user. Events on the sender side are marked with `S` and those occurring at the receiver are annotated with `R`.

The finite state machine starts in state `1`. On the sender side, the user may then issue a connection request `CReq` causing a transition to state `2`. A window indicating an incoming connection `CInd` may then pop up at the receiver side which then can either be confirmed (3 → 4) or declined (3 → 7). In the latter case, a disconnection indication will ensue on the

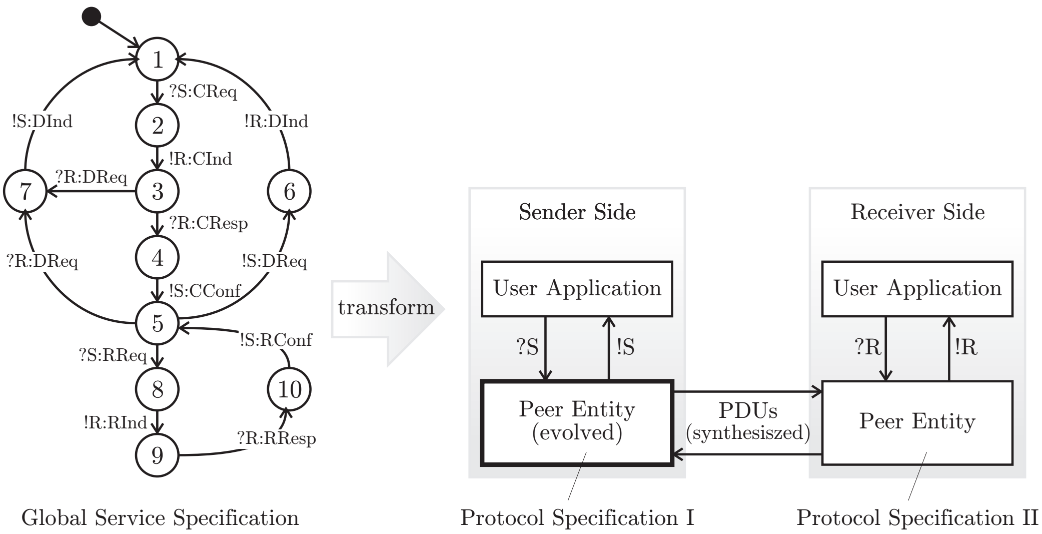


Figure 11: The scenario for protocol synthesis as used in [135].

sender side whereas the former transition leads to a connection confirmation. A connection may be reconnected (states 5, 8, 9, 10) or being closed by either the sender (5 → 6) or the receiver (5 → 7) with a disconnect request DReq, causing a disconnect indication DInd (maybe again in form of a window) to pop up on the other end.

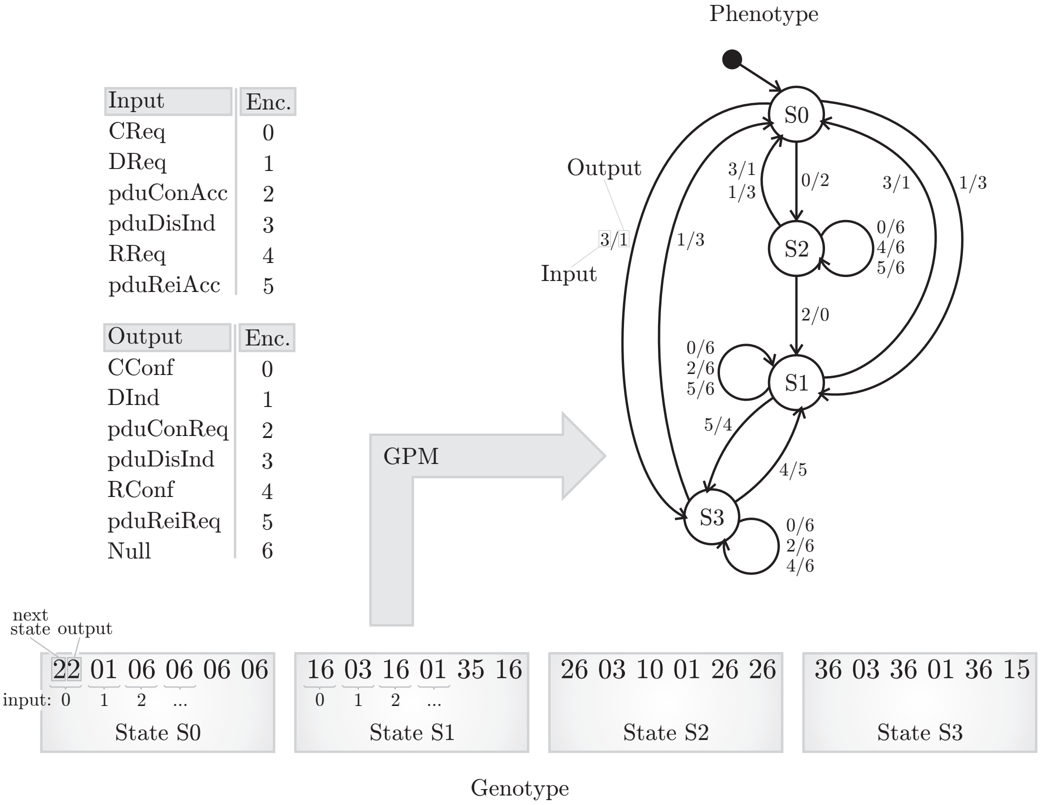
The goal is to evolve a finite state machine that governs the behavior of one of the two protocol partners and to synthesize the messages PDUs[4] that have to be exchanged between the sender and the receiver.

De Araújo et al. [135] first create a sequence of interaction primitives by traversing the service specification FSM. In this traversal, all state transitions need to be visited. One such sequence could be (?S:CReq, !R:CInd, ?R:CResp, !S:CConf, ?S:RReq, !R:RInd, ?R:RResp, !S:RConf,!S:DReq, !R:DInd, ?S:CReq, !R:CInd, ?R:DReq, !S:DInd, ?S:CReq, !R:CInd, ?R:CResp, !S:CConf, ?R:DReq, !S:DInd). It is obvious that if one event on the sender side is followed by an event on the receiver side (or vice versa) and the second event is not triggered by the user (?xyz), a message (i.e., a PDU) has to be exchanged between the two sides. These PDUs can either be automatically generated or predefined and are then inserted into the sequence, resulting in the list, (?S:CReq, pduConReq, !R:CInd, ?R:CResp, pduConAcc, !S:CConf, ?S:RReq, pduReiReq, !R:RInd, ?R:RResp, pduReiAcc, !S:RConf, !S:DReq, pduDisInd, !R:DInd, ?S:CReq, pduConReq, !R:CInd, ?R:DReq, pduDisInd, !S:DInd, ?S:CReq, pduConReq, !R:CInd, ?R:CResp, pduConAcc, !S:CConf, ?R:DReq, pduDisInd, !S:DInd), for instance.

From this list, the message exchanges are extracted which concern the protocol peer to be evolved. [135] was focused on the sender side protocol, which here results in the following tuples: ((?S:CReq, pduConAcc), (pduConAcc, !S:CConf), (?S:RReq, pduReiReq), (pduReiAcc, !S:RConf), (!S:DReq, pduDisInd), (?S:CReq, pduConReq), (pduDisInd, !S:DInd), (?S:CReq, pduConReq), (pduConAcc, !S:CConf), and (pduDisInd, !S:DInd)).

In order to allow the synthesized protocol to react to all possible messages in all states, random non-expected inputs are added to the list like ((CReq, pduConAcc), *(CReq, NULL)*, (pduConAcc, CConf), . . .). Such a list is then encoded and used as test set during the individual evaluation. For the evolution, de Araújo et al. [135] use the grammar-guided Genetic Programming Kernel (GPK) of Hörner [137] which allows for variable-length genomes. The objective function compares the behavior of a solution candidate with the expectations and counts the number of correct outputs.

[4] http://en.wikipedia.org/wiki/Protocol_data_unit [accessed 2008-06-21]



Input	Enc.
CReq	0
DReq	1
pduConAcc	2
pduDisInd	3
RReq	4
pduReiAcc	5

Output	Enc.
CConf	0
DInd	1
pduConReq	2
pduDisInd	3
RConf	4
pduReiReq	5
Null	6

Figure 12: The best individual from the experiment of de Araújo et al. [135].

Figure 12 illustrates the state-based representation (SBR) used by de Araújo et al. [135] to encode finite state machines. The genotypes are divided into genes, each representing one state. The total number of states is variable. For each possible input, every gene contains one tuple of two numbers. The first number in this tuple represents the next state and the second one stands for the output of the corresponding transition.

The receiver side of the protocol was evolved in the same manner during the experiments of de Araújo et al. [135] and the results were able to pass an automated equivalence check. The good performance reported further indicates the high utility of this approach for protocol synthesis.

3.3.3 Evolving Finite State Machine-based Protocol Definitions

Another approach for evolving finite state machine-based protocol specifications with genetic algorithms was contributed by Sharples and Wakeman [138]. In [139], for instance, protocols for unreliable media were evolved. In his PhD thesis, Sharples [140] shows that complex protocols can be evolved that may even outperform protocols designed by hand. In most cases though, the fitness of the evolved protocols is lower than the reference protocol [141].

3.3.4 Evolving Fraglet-based Protocols

In his seminal work, Tschudin [142] introduced Fraglets[5], a new artificial chemistry suitable for the development and even evolution of network protocols. Fraglets represent an execution model for communication protocols which resembles chemical reactions.

From the theoretical point of view, the Fraglet approach is an instance of Post's string rewriting systems [143]. Fraglets are symbolic strings of the form $[s_1 : s_2 : \dots : s_n]$. The symbols s_i either represent control information or payload. Each node in the network has a Fraglet store which corresponds to a reaction vessel in chemistry. Naturally, reaction vessels may contain equal molecules multiple times. Chemical reaction vessels usually contain equal molecules multiple times and the same goes for Fraglet stores which can be implemented as multisets keeping track on the multiplicity of the Fraglets they contain.

Tschudin [142] defines a simple prefix programming language for Fraglets. It has a fixed instruction set comprising transformation and reaction rules. Transformations like *dup* and *nop* modify a single Fraglet whereas the reactions *match* and *matchP* combine two Fraglets. For the definition of these rules in Table 2, we will use the syntax $[s_1 : s_2 : \dots : \mathit{tail}]$ where s_i is a symbol and *tail* is a possibly empty sequence of symbols.[6]

As example for the structure of Fraglets, we illustrate a quine Fraglet as introduced by Yamamoto et al. [144] in Figure 13. *Quines* are computer programs which produce copies of themselves (or their source code) as output. Yamamoto et al. [144] define them as vehicle for self-replicating and self-modifying programs.

According to Tschudin [142], this form of protocol representation is predestined for automated synthesis via evolutionary algorithms: Fraglets have almost no syntactical constraints and can represent complicated protocols in a compact manner. Tschudin [142] focused on the offline evolution of protocols using a genetic algorithm. A complete communication system was simulated for a given number of time steps during the evaluation of each individual. The objective values denote the correlation of the behavior observed during the simulation and the target behavior. While Tschudin's results substantiate that the evolutionary methods are suitable to optimize existing Fraglet protocols, they also indicated that the evolution of new distributed algorithms is difficult because of a strong tendency to overfitting. Furthermore, it is hard to define objective functions which can reward "partially correct" behavior,

[5] See `http://en.wikipedia.org/wiki/Fraglets` [accessed 2008-05-02] and `http://www.fraglets.net/` [accessed 2008-05-02] for more information.

[6] See `http://www.fraglets.net/frag-instrset-20070924.txt` [accessed 2008-05-02] for the full instruction set as of 2007-09-24.

tag	**transformation/reaction**
basic transformations	
dup	$[dup : t : a : tail] \longrightarrow [t : a : a : tail]$ duplicate a single symbol
exch	$[exch : t : a : b : tail] \longrightarrow [t : b : a : tail]$ swap two tags
fork	$[fork : a : b : tail] \longrightarrow [a : tail]\,,\ [b : tail]$ copy fraglet and prepend different header symbols
nop	$[nop : tail] \longrightarrow [tail]$ does nothing (except consuming the instruction tag)
null	$[null : tail] \longrightarrow \emptyset$ destroy a fraglet
pop2	$[pop2 : h : t : a : tail] \longrightarrow [h : a]\,,\ [t : tail]$ pop head element a out of a list $[a : b : tail]$
split	$[split : tail_1 : * : tail_2] \longrightarrow [tail_1]\,,\ [tail_2]$ break a fraglet into two at the first occurence of $*$
arithmetic transformations	
sum	$[sum : t : m : n : tail] \longrightarrow [t : m+n : tail]$ an operation comparing two numbers
lt	$[lt : yes : no : a : b : tail] \longrightarrow \begin{cases} [yes : \{a\} : \{b\} : tail] & \text{if } a < b \\ [no : \{a\} : \{b\} : tail] & \text{otherwise} \end{cases}$ a logic operation comparing two numbers a and b
communication primitives	
broadcast	$[broadcast : tail] \longrightarrow {}_{\mathrm{n}}[tail]$ broadcast *tail* to all nodes nin the network N
send	$[send : dest : tail] \longrightarrow {}_{dest}[tail]$ send *tail* to a single node *dest*
node	${}_{n}[node : t : tail] \longrightarrow {}_{n}[t : \{\mathrm{id(n)}\} : tail]$ obtain the ID id(n) of the current node n
reactions	
match	$[match : a : tail_1]\,,\ [a : tail_2] \longrightarrow [tail_1 : tail_2]$ two fraglets react, their tails are concatenated
matchP	$[matchP : a : tail_1]\,,\ [a : tail_2] \longrightarrow [matchP : a : tail_1]\,,\ [tail_1 : tail_2]$ “catalytic match”, i. e., the matchp rule persists

Table 2: Some Fraglet Instructions (from [142] and `http://www.fraglets.net/` (2008-05-02)).

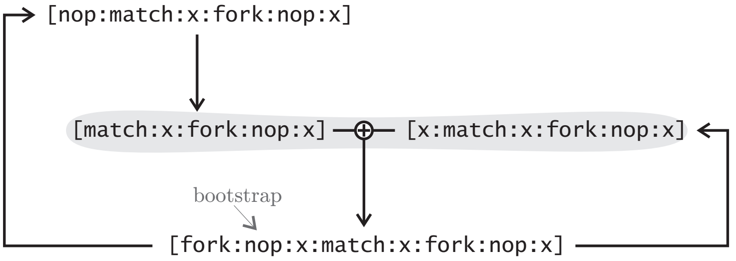


Figure 13: A simple quine Fraglet (borrowed from [144])

since this term is a contradiction in itself. This is known as the all-or-nothing property in Genetic Programming.

3.3.5 Online Protocol Adaptation and Evolution with Fraglets

Autonomic networks are networks where manual management is not desired or hard to realize, such as systems of hundreds of gadgets in an e-home, sensor networks, or arbitrary mesh networks with wireless and wired links. Yamamoto and Tschudin [141] propose that software in such networks needs to be self-modifying so as to be able to react to unforeseen network situations. They distinguish two forms of reaction – adaptation and evolution. Adaptation is the short-term reconfiguration of existing modules whereas evolution stands for the modification of old and the discovery of new functionality and happens at a larger timescale. Software with these abilities probably cannot predict whether the effects of a modification are positive or negative in advance and therefore needs to be resilient in order to mitigate faulty code that could evolve. In [145], Tschudin and Yamamoto show that such resilience can be achieved by introducing redundancy into Fraglet protocols.

Complementing Tschudin's work [142] on offline optimization summarized in Section 3.3.4, Yamamoto and Tschudin [141] describe online protocol evolution as a continuously ongoing, decentralized and asynchronous process of constant competition and selection of the most feasible modules. Genetic Programming with mutation and homologous crossover is chosen for accomplishing these features. The fitness measure (subject to maximization) is the performance of the protocols as perceived by the applications running in the network. The score of a solution candidate (i. e., a protocol) is incremented if it behaves correctly and decremented whenever an error is detected. The resource consumption in terms of the memory consumed by the protocols is penalized proportionally.

Yamamoto and Tschudin [146; 147] create populations containing a mix of different confirmed delivery and reliable delivery protocols for messages. These populations were then confronted with either reliable or unreliable transmission challenges and were able to adapt to these conditions quickly. If the environment changes afterwards, e.g., a formerly reliable channel becomes unreliable, the degree of re-adaptation is unsatisfying. The loss of diversity due to the selection of only highly fit protocols during the adaptation phase could not yet be compensated by mutation in these first experiments of Yamamoto and Tschudin [146].

Further information on approaches for evolutionary online optimization of communication protocols can be found in the report *Framework for Distributed On-line Evolution of Protocols and Services, 2nd Edition* from the EU-sponsored project BIONETS [148].

3.3.6 Learning Communication Patterns for Agents

The term *protocol* not only stands for the way data is exchanged over a network connection but has a much wider meaning which also subsumes the patterns of communication between humans or animals. Agents are a metaphor for all kinds of autonomous systems in computer science and the last issue which we will summarize in this section is the evolution of communication protocols for such systems.

Some early experiments in evolving communication patterns for autonomous agents were conducted by Werner and Dyer [149] on the example of artificial creatures. In their simulation, blind male animals capable of hearing and moving need to be guided by immobile females which can see and are able to produce sounds. Werner and Dyer [149] used a genetic algorithm with an integer string genome encoding the weights and biases of recurrent neural networks for both, the females and the males. The evolution, however, was not performed by a regular genetic algorithm. Instead, it took place in the simulated environment where the females lead the males to their location. When they meet, two offspring will be produced by applying standard crossover and mutation to their genome.

Based on their co-evolutionary strategy [150], Iba et al. [151] evolve communication protocols used by agents for cooperative predator-prey pursuit and robot navigation tasks [152]. Mackin and Tazaki [153; 154; 155] focus on evolving communication patterns for multi-agent systems modeling a market place. They use a divide-and-conquer approach for evolving the code for sending and receiving the messages exchanged amongst the agents in independent ADFs (automatically defined functions) with Genetic Programming. While starting with a scenario where each trading agent has only a single objective in [153], the evolutionary approach is extended to multi-objective Genetic Programming. By building separate communication trees for each objective, Mackin and Tazaki [154] allow the agents to weight three objectives in their experiments: to buy at the highest volume at lowest price and to collect the best performing services for buying agents and to sell the highest volume at the highest price while collecting the highest value return for selling agents.

3.4 Optimizing Network Security

Network security is one of the most vital aspects when developing distributed systems. The detection of ongoing malicious activities in a network as well as the generation of protocols which are safe against attacks are only two facets of this area. Here we will discuss how global optimization methods can be applied to both of them.

3.4.1 Intrusion Detection

According to Heady et al. [156], an intrusion is "any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource". These attacks most often come from the outside through network connections to the internet. Intrusion detection systems (IDS) identify such unwanted manipulation of computer systems. Generally, a network connection can be described by a broad set of features, concerning its content, time- or traffic-based details, and host-related data. Analysis of these features boils down to data mining. Global optimization methods can be utilized for improving the efficiency of classifiers that decide whether a dataset containing the features of a connection belongs to normal or abnormal (possible intrusive) traffic.

Lu and Traore [157], for instance, evolve intrusion detection rules with tree-based Genetic Programming. A similar approach is followed by Folino et al. [158], who apply the AdaBoost ensemble method of Freund and Schapire [159] in conjunction with the distributed evolution of decision trees. Heady et al. [156], on the other hand, use classical Learning Classifier Systems. Linear Genetic Programming also seems to be very efficient in growing such intrusion detectors, as shown by Mukkamala et al. [160], for example. Song, Heywood, and Zincir-Heywood [161], [162] use their page-based LGP variant [163] for this purpose. In [164], they extend this approach for identifying attacks on 802.11 WLAN networks. Hansen et al. [165] apply a homologous crossover operation to the evolution of linear machine-code structures for intrusion detection for the prevention of cyber-terrorism.

3.4.2 Security Protocol Generation

Perrig and Song [166] focus their work on the automatic generation of security protocols such as mutual authentication protocols. In the first step of this process, they use an iterative deepening depth-first search (IDDFS) algorithm to list *all* protocols that can be composed using a predefined set of primitives. The limit for this search is not the depth in the search space, but a value of a specified cost function which sums up user-specified costs of the single primitives of a protocol. Since the space of all protocols is very large, simplifying assumptions like the absence of redundant message fields and commutativity of the fields are made. Perrig and Song [166] conduct experiments with the goal to synthesize mutual authentication protocols. In these experiments, the IDDFS still generates tens of thousands of individuals even with the restrictions defined above. Therefore, a pruning algorithm is added which removes obviously flawed protocols such as those which openly disclose private keys in a messages. In this pruning process, also vulnerabilities to impersonation attempts or replay attacks are checked. Only a few hundreds of candidate protocols survive this pruning routine in the experiments mentioned. These candidates may still be incorrect and are therefore screened with Song's Athena security protocol checker [167]. As a result, two correct symmetric and one correct asymmetric protocol where synthesized, whereas the symmetric one was even simpler than the *Symmetric-Key Three-Pass Mutual Authentication Protocol* standardized 1993 by the ISO [168]. In 2001, Song et al. [169] added a code generator that transforms the generated protocol specifications to Java to their system.

3.5 Optimizing Parameters and Configurations

Protocols, distributed algorithms, and topologies can be optimized in order to create robust and efficient distributed applications. The efficiency of such application also strongly depends on its parameter settings and configuration. Today's web servers, for instance, have a variety of tuning parameters such as the thread pool size, the maximum number of connections queued, and the number of requests which can be served simultaneously. It is hard to find optimal settings, especially when applications consist of web frontends, application servers hosting the business logic, and data base servers in the backend. The configurations of these subsystems are not independent. Instead, there are epistatic and synergetic effects which are often highly complicated and hard to simulate offline.

3.5.1 Optimal Application Server Configuration

Xi et al. [170] developed a smart hill climbing algorithm and applied it to application server configuration. This optimization process only needs about ten online performance samples of a running WebSphere brokerage system for fine-tuning its parameters in order to achieve near-optimal results.

3.5.2 Distributed Agent Evolution for Service Management

In the work of Nakano and Suda [171; 172; 173], network services are represented by mobile agents. An agent capable of behaviors like replication, migration, and termination could, for instance, stand for a HTTP server. A behavior i is triggered when a corresponding weighted sum surpasses a certain threshold θ_i [171]. This sum incorporates environmental information such as the request rate and the resource costs as well as the internal state of the agent like its age and energy level. The weights of the factors in this sum are subject to optimization.

The agents are equipped with an energy value. They obtain energy for performing services for the user and spend energy for resource access and replication. If the energy of an agent is depleted, the agent dies, i. e., its service is terminated. Agents may create new agents by replication or reproduction. Replication corresponds to mutation in evolutionary algorithms, leading to a child with slightly changed weight vectors. Reproduction is basically a crossover operation where a partner is selected within n hops in the network based on three measures: the time a user has to wait between requesting and receiving a service, the number of hops a service request travels from the user to the executing agent, and the energy efficiency.

With their experiments, Nakano and Suda [171] showed that their approach allows the services, i. e., the agents, to adapt to changing conditions such as network failures, varying workload, and changing platform costs (which is reflected in energy consumption).

3.5.3 Protocol Configuration

In his master's thesis, Grace [174] uses Genetic Programming to evolve configurations for the JavaGroups [175] reliable group communication framework. The behavior of a JavaGroups instance depends on a protocol stack specification in form of a string which is passed in when it is created. This specification determines which protocol layers will be present in the stack. Grace [174] showed that the evolution of communication protocol configurations for predefined scenarios, such as reliable, ordered multicast, is feasible with his approach.

3.5.4 Evolving Protocol Adapters

In distributed systems with components manufactured by different organizations, interprocess communication (IPC) can become problematic. Therefore, protocol adapters are applied, which are components that mediate between different protocols. In the worst case, the number of required adaptors rises quadratically with the number of processes in the system. Hence, creating them by hand is infeasible and automated solutions are requires. Van Belle et al. [176] evolve such protocol adapters in form of classifier systems representing Petri nets with a genetic algorithm. They show the viability of this method on the example of an adapter for a locking protocol for process synchronization.

Similar to our own experiences, they found that evolving components of distributed systems always requires special means [176] of overfitting prevention in order to stop the system from generating useless implementations that just respond with the wanted messages but do not perform the work which they are supposed to do. A discussion of the use of Learning Classifier System for the purpose of protocol adapter synthesis can be found in Van Belle's PhD thesis [177].

4 Paper List

In this section, we list the papers concerning the optimization of distributed systems. This concise list groups the papers according to the area of application, the optimization goals, the problem representations, and the optimization algorithms utilized. This collection lists a wide variety of approaches developed by a large number of authors (nearly 200 authors are involved in the papers listed). In our opinion, this heterogeneity and distribution should be interpreted as a clear indicator that the optimization of distributed systems lends itself to heuristic and meta-heuristic approaches. Many papers provide engineering-level solutions which often deliver excellent results.

4.1 Topology Optimization and Terminal Assignment

4.1.1 General Networks or Theory

1. Abuali et al. [178, 96] (1994) *aims*: synthesis, costs; *representation*: integer strings; *optimization methods*: evolutionary algorithms and local search, see also Section 3.1.1

2. Khuri and Chiu [97] (1997) *aims*: synthesis, costs; *representations*: bit strings and integer strings; *optimization methods*: evolutionary algorithms and local search, see also Section 3.1.1

3. Salcedo-sanz and Yao [179] (2004) *aims*: synthesis, costs; *representations*: bit strings and integer strings; *optimization method*: evolutionary algorithms

4. Lehmann and Kaufmann [101, 100] (2005–2007) *aims*: synthesis, self-organization, QoS features, dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization method*: evolutionary algorithms, see also Section 3.1.2

4.1.2 Computer Networks in General

5. Michalewicz [92] (1991) *aims*: synthesis, robustness; *optimization method*: evolutionary algorithms, see also Section 3.1

6. Kumar et al. [93] (1993) *aims*: synthesis, robustness, QoS features; *representation*: bit strings; *optimization method*: evolutionary algorithms, see also Section 3.1

7. Pierre and Legault [180, 181] (1996–1998) *aims*: synthesis, QoS features, costs; *representation*: bit strings; *optimization method*: evolutionary algorithms

8. Ko et al. [182] (1997) *aims*: synthesis, QoS features, costs; *representation*: bit strings; *optimization methods*: evolutionary algorithms and local search

9. Dengiz et al. [183] (1997) *aims*: synthesis, robustness, costs; *representation*: integer strings; *optimization methods*: evolutionary algorithms and Memetic Algorithms

10. Yao et al. [98] (2005) *aims*: synthesis, costs; *representation*: trees; *optimization methods*: evolutionary algorithms and local search, see also Section 3.1.1

4.1.3 Telecommunication Networks in General

11. Dengiz et al., see entry 9.

12. Pierre and Elgibaoui [184] (1997) *aims*: synthesis, robustness, QoS features, costs; *optimization method*: Tabu Search

4.1.4 Wireless or Mobile Telecommunication Networks

13. Pierre and Houéto [185, 186] (2002) *aims*: synthesis, costs; *representation*: bit strings; *optimization method*: Tabu Search

14. Quintero and Pierre [187, 188, 189] (2002–2003) *aim*: costs; *representation*: integer strings; *optimization methods*: evolutionary algorithms, local search, Memetic Algorithms, Tabu Search, and Simulated Annealing

15. St-Hilaire et al. [190, 191] (2006) *aims*: synthesis, costs; *optimization methods*: local search and Tabu Search

16. Salcedo-Sanz et al. [193, 192] (2008) *aims*: synthesis, QoS features, costs; *representation*: bit strings; *optimization methods*: evolutionary algorithms and Memetic Algorithms

4.1.5 Optical Networks in General

17. Sinclair [104, 103, 94, 108] (1995–2000) *aims*: synthesis, robustness, costs; *representations*: bit strings and trees plus genotype-phenotype mappings; *optimization method*: evolutionary algorithms, see also Section 3.1.3

18. Brittain et al. [194] (1997) *aims*: synthesis, costs; *representations*: bit strings and integer strings; *optimization methods*: evolutionary algorithms and local search

4.2 Node Placement

19. Alba et al. [195] (2002) *aims*: synthesis, robustness, costs; *representation*: bit strings; *optimization method*: evolutionary algorithms

20. Salcedo-Sanz et al., see entry 16.

4.3 Dimensioning and Capacity Assignment

4.3.1 Computer Networks in General

21. Coombs and Davis [91] (1987) *aim*: QoS features; *optimization method*: evolutionary algorithms, see also Section 3.1

22. Ko et al., see entry 8.

23. Martin et al. [196] (2008) *aim*: synthesis; *representation*: integer strings; *optimization methods*: evolutionary algorithms, Extremal Optimization, and Particle Swarm Optimization

4.3.2 Telecommunication Networks in General

24. Martin et al., see entry 23.

4.4 Frequency and Channel Assignment

25. Tan and Sinclair [106] (1995) *aims*: synthesis, costs; *representation*: bit strings; *optimization methods*: evolutionary algorithms and local search

4.5 Protocol Generation and Optimization

4.5.1 General Networks or Theory

26. Mackin and Tazaki [153, 154, 155] (1999–2002) *aim*: synthesis; *representation*: trees; *optimization method*: evolutionary algorithms, see also Section 3.3.6

4.5.2 Computer Networks in General

27. El-Fakihy et al. [133, 134] (1995–1999) *aims*: synthesis, QoS features; *representation*: bit strings; *optimization methods*: evolutionary algorithms and Memetic Algorithms, see also Section 3.3.1

28. Sharples and Wakeman [139, 138, 140] (1999–2001) *aims*: synthesis, robustness, QoS features; *representation*: bit strings plus genotype-phenotype mappings; *optimization method*: evolutionary algorithms, see also Section 3.3.3

29. Song et al. [166, 169] (2000–2001) *aims*: synthesis, QoS features; *representation*: trees; *optimization method*: local search, see also Section 3.4.2

30. Grace [174] (2000) *aims*: synthesis, robustness, QoS features; *representation*: trees; *optimization method*: evolutionary algorithms, see also Section 3.5.3

31. Van Belle et al. [177, 176] (2001–2003) *aims*: synthesis, robustness, QoS features; *representation*: bit strings; *optimization method*: evolutionary algorithms, see also Section 3.5.4

32. de Araújo et al. [135, 136] (2003) *aim*: synthesis; *representation*: integer strings; *optimization method*: evolutionary algorithms, see also Section 3.3.2

33. Tschudin [142] (2003) *aims*: synthesis, robustness; *representation*: linear programs; *optimization method*: evolutionary algorithms, see also Section 3.3.4

34. Yamamoto and Tschudin [147, 141, 146] (2005) *aims*: synthesis, robustness, dynamic or adaptive behavior; *representations*: information distributed over the network and linear programs; *optimization method*: evolutionary algorithms, see also Section 3.3.5

4.5.3 Wireless or Mobile Computer Networks

35. Weise et al. [197, 198] (2007–2008) *aims*: synthesis, dynamic or adaptive behavior; *optimization method*: evolutionary algorithms

4.6 Routing

4.6.1 General Networks or Theory

36. Christensen et al. [199] (1997) *aims*: QoS features, costs; *representation*: integer strings; *optimization method*: evolutionary algorithms

37. Kirkwood et al. [110] (1997) *aims*: synthesis, robustness; *representation*: trees; *optimization method*: evolutionary algorithms, see also Section 3.2.1

38. Zhu et al. [200] (1998) *aim*: synthesis; *representation*: integer strings; *optimization methods*: evolutionary algorithms and local search

4.6.2 Computer Networks in General

39. Kirkwood et al., see entry 37.

40. Munetomo et al. [114, 115, 116, 117] (1997–1999) *aims*: self-organization, robustness, dynamic or adaptive behavior; *representations*: integer strings and information distributed over the network; *optimization method*: evolutionary algorithms, see also Section 3.2.2

41. Ko et al., see entry 8.

42. Di Caro and Dorigo [121, 120, 122] (1998–2004) *aims*: self-organization, robustness, dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization method*: ACO/ant agents, see also Section 3.2.3

43. Bonabeau et al. [201] (1999) *aim*: dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization method*: ACO/ant agents

44. Fei et al. [202] (1999) *aims*: robustness, QoS features; *representation*: bit strings

45. Liang et al. [123, 124] (2002–2006) *aims*: robustness, dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization methods*: evolutionary algorithms and ACO/ant agents, see also Section 3.2.3

46. Sim and Sun [203] (2002) *representation*: information distributed over the network; *optimization method*: ACO/ant agents

4.6.3 Telecommunication Networks in General

47. Cox, Jr. et al. [204] (1991) *aims*: QoS features, costs, dynamic or adaptive behavior; *representation*: integer strings; *optimization method*: evolutionary algorithms

48. Schoonderwoerd et al. [119, 64, 118] (1996–1997) *aims*: synthesis, self-organization, robustness, QoS features, dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization method*: ACO/ant agents, see also Section 3.2.3

49. Christensen et al., see entry 36.

50. Zhu et al., see entry 38.

51. Lukschandl et al. [205, 206, 207] (1999–2000) *aims*: robustness, costs; *representation*: linear programs; *optimization method*: evolutionary algorithms

52. Galiasso and Wainwright [208] (2001) *aims*: synthesis, costs; *representation*: integer strings; *optimization methods*: evolutionary algorithms and Memetic Algorithms

53. Sandalidis et al. [209] (2001) *aim*: dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization method*: ACO/ant agents

4.6.4 Optical Networks in General

54. Wang et al. [210] (2004) *aim*: QoS features; *optimization method*: evolutionary algorithms

4.7 Load Balancing and Call Admission

4.7.1 Computer Networks in General

55. Munetomo et al., see entry 40.

56. Oates and Corne [211] (2001) *aim*: QoS features; *representation*: integer strings; *optimization methods*: evolutionary algorithms, local search, and Simulated Annealing

57. Zapf and Weise [212, 213] (2007) *aims*: synthesis, self-organization; *representation*: bit strings; *optimization method*: evolutionary algorithms

4.7.2 Telecommunication Networks in General

58. Schoonderwoerd et al., see entry 48.

4.8 Peer-To-Peer Systems

59. Iles and Deugo [130] (2002) *aims*: robustness, dynamic or adaptive behavior; *representation*: trees; *optimization method*: evolutionary algorithms, see also Section 3.2.6

60. Forestiero et al. [214, 215, 216, 217, 218, 219] (2005–2008) *aims*: self-organization, QoS features, dynamic or adaptive behavior; *representation*: information distributed over the network; *optimization method*: ACO/ant agents

4.9 Broadcast and Multicast

4.9.1 General Networks or Theory

61. Christensen et al., see entry 36.

62. Zhu et al., see entry 38.

63. Comellas and Giménez [125] (1998) *aims*: synthesis, QoS features; *representation*: trees; *optimization method*: evolutionary algorithms, see also Section 3.2.4

4.9.2 Computer Networks in General

64. Fei et al., see entry 44.

65. Grace, see entry 30.

4.9.3 Telecommunication Networks in General

66. Christensen et al., see entry 36.

67. Zhu et al., see entry 38.

68. Galiasso and Wainwright, see entry 52.

4.9.4 Other

69. Jaroš and Dvořák [127] (2008) *aims*: synthesis, QoS features; *representation*: integer strings; *optimization methods*: Memetic Algorithms and estimation of distribution algorithms, see also Section 3.2.5

4.10 Security and Intrusion Detection

4.10.1 Computer Networks in General

70. Heady et al. [156] (1990) *aim*: synthesis; *representation*: bit strings; *optimization method*: evolutionary algorithms, see also Section 3.4.1

71. Song et al., see entry 29.

72. Song et al. [162, 161] (2003) *aim*: synthesis; *representation*: linear programs; *optimization method*: evolutionary algorithms, see also Section 3.4.1

73. Liu et al. [220] (2004) *aim*: synthesis; *optimization method*: evolutionary algorithms

74. Mukkamala et al. [160] (2004) *aims*: synthesis, robustness; *representation*: linear programs; *optimization method*: evolutionary algorithms, see also Section 3.4.1

75. Lu and Traore [157] (2004) *aim*: synthesis; *representation*: integer strings plus genotype-phenotype mappings; *optimization method*: evolutionary algorithms, see also Section 3.4.1

76. Folino et al. [158] (2005) *aim*: synthesis; *representations*: trees and linear programs; *optimization method*: evolutionary algorithms, see also Section 3.4.1

77. Hansen et al. [165] (2007) *aim*: synthesis; *representation*: linear programs; *optimization method*: evolutionary algorithms, see also Section 3.4.1

4.10.2 Wireless or Mobile Computer Networks

78. LaRoche and Zincir-Heywood [164] (2005) *aim*: synthesis; *representation*: linear programs; *optimization method*: evolutionary algorithms, see also Section 3.4.1

4.11 Agent Cooperation (non-ant)

4.11.1 General Networks or Theory

79. Werner and Dyer [149] (1992) *aim*: synthesis; *representation*: integer strings plus genotype-phenotype mappings; *optimization method*: artificial life, see also Section 3.3.6

80. Andre [221] (1995) *aim*: synthesis; *representation*: trees; *optimization method*: evolutionary algorithms

81. Qureshi [222, 223, 224] (1996–2001) *aim*: synthesis; *representation*: trees; *optimization method*: evolutionary algorithms

82. Iba et al. [150, 151, 152, 225] (1996–1999) *aims*: synthesis, robustness; *representation*: trees; *optimization method*: evolutionary algorithms, see also Section 3.3.6

83. Mackin and Tazaki, see entry 26.

4.11.2 Computer Networks in General

84. Nakano and Suda [172, 171, 173] (2004–2007) *aims*: self-organization, QoS features, dynamic or adaptive behavior; *representations*: real vectors and information distributed over the network; *optimization method*: evolutionary algorithms, see also Section 3.5.2

85. Zapf and Weise, see entry 57.

4.11.3 Telecommunication Networks in General

86. Schoonderwoerd et al., see entry 48.

4.12 Software Configuration

87. Grace, see entry 30.

88. Iles and Deugo, see entry 59.

89. Xi et al. [170] (2004) *aim*: QoS features; *representation*: integer strings; *optimization methods*: local search and Simulated Annealing, see also Section 3.5.1

90. Nakano and Suda, see entry 84.

4.13 Hardware Design and Configuration

4.13.1 Computer Networks in General

91. Martin et al., see entry 23.

4.13.2 Wireless or Mobile Networks in General

92. Choo et al. [226] (2000) *aim*: synthesis; *representation*: bit strings; *optimization method*: evolutionary algorithms

93. Lohn et al. [227] (2004) *aim*: synthesis; *representation*: real vectors; *optimization method*: evolutionary algorithms

94. Villegas et al. [228] (2004) *aim*: synthesis; *optimization method*: evolutionary algorithms

95. Koza et al. [229] (2005) *aim*: synthesis; *representation*: trees; *optimization method*: evolutionary algorithms

96. John and Ammann [230, 231] (2006) *aim*: synthesis; *representation*: bit strings; *optimization method*: evolutionary algorithms

97. Chattoraj and Roy [232] (2006) *aim*: synthesis; *representation*: bit strings; *optimization method*: evolutionary algorithms

4.14 Algorithm Synthesis

4.14.1 Computer Networks in General

98. Weise et al. [233, 234] (2007–2008) *aim*: synthesis; *representation*: integer strings; *optimization methods*: evolutionary algorithms and local search

99. Weise et al. [235] (2007) *aim*: synthesis; *representation*: bit strings; *optimization method*: evolutionary algorithms

4.14.2 Wireless or Mobile Computer Networks

100. Weise and Geihs [236, 237, 238] (2001–2006) *aims*: synthesis, robustness; *representation*: linear programs; *optimization method*: evolutionary algorithms

101. Weise et al. [239, 240] (2007) *aim*: synthesis; *representation*: linear programs; *optimization method*: evolutionary algorithms

5 Conclusions

In this study, we gave a short overview on the wide variety of applications of global optimization to distributed systems. For the last ten years, this has been one of the most active research areas in Evolutionary Computation, with many researchers steadily contributing new and enhanced approaches.

We not only provided a representative list and classification of publications, but also introduced many interesting approaches in a detailed. Yet, we can only provide a small glimpse on the real amount of work available. The master's thesis of Kampstra [2] is now already three years old and referenced over 400 papers. From the related work section of the papers that we have summarized we know that there should exist at least another 200 contributions not mentioned in his list or not yet published when it was compiled.

Practitioners in the area of networking or telecommunication tend to feel skeptical when it comes to the utilization of randomized or bio-inspired approaches for optimizing, managing, or controlling their systems. One argument against them is that the worse case may be unpredictable although they may provide good results in average.

Nevertheless, certain problems (like the Terminal Assignment Problem, see Section 3.1.1) are NP-hard and therefore can only be solved efficiently with such approaches, which, of course, goes hand in hand with a certain trade-off in terms of optimality, for instance. In static design cases, the worse case scenario in which an EA would create inferior solutions, can be ruled out by checking its results before the actual deployment or realization.

Furthermore, in practice additional application-specific constraints are often imposed on standard problems. The influence of these constraints on the problem hardness and the applicability of the well-known solutions is not always easy to comprehend. Thus, incorporating the constraints into a global optimization procedure tends to be much easier than customizing a problem-specific heuristic algorithm. Assume that we want to find fast routes in a network which are also robust against a certain fraction of failed links. If we have an EA with an objective function that measures the time a message travels in a fully functional network, it is intuitively clear that we can extend this approach by simply applying this function to a couple of scenarios with randomly created link failures, too. Creating a corresponding extension of Dijkstra's algorithm, however, is less straightforward.

Nature-inspired approaches have not only shown their efficiency in static optimization problems, but were proven to be especially robust in dynamic applications, too. This is especially interesting in the looming age of networks of larger scale. Wireless networks [241, 242, 243], sensor networks [244], wireless sensor networks [245]), Smart Home networks [246, 247], ubiquitous computing [248, 249], and more require self-organization, efficient routing, optimal parameter settings, and power management. We are sure that nature and physics-inspired global optimization methods will provide viable answers to many of these questions which become more and more eminent in the near future.

When condensing the essence of this summary down to a single sentence, “Evolutionary Computing in Telecommunications – A likely EC success story”, the title of Kampstra’s thesis maybe fits best. We believe that the *likely* is no longer required, since many of the methods developed already reached engineering-level applicability.

References

[1] Mark C. Sinclair from the Electronic Systems Engineering, University of Essex, Colchester CO4 3SQ, UK. Evolutionary Telecommunications: A Summary. In *Proceedings of Evolutionary Telecommunications: Past, Present and Future – A Bird-of-a-feather Workshop at GECCO 99*, 1999. In proceedings [4]. Online available at http://uk.geocities.com/markcsinclair/ps/etppf_sin_sum.ps.gz and http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.6797 [accessed 2008-08-01], presentation online available at http://uk.geocities.com/markcsinclair/ps/etppf_sin_slides.ps.gz [accessed 2008-08-01].

[2] Peter Kampstra. Evolutionary Computing in Telecommunications – A likely EC success story. Master’s thesis, Vrije Universiteit, Amsterdam, August 2005. BMI Paper, Supervisors: Rob D. van der Mei and Gusz Eiben (≡Ágoston E. Eiben). Online available at http://www.few.vu.nl/stagebureau/werkstuk/werkstukken/werkstuk-kampstra.pdf [accessed 2008-09-04].

[3] Peter Kampstra, Rob D. van der Mei, and Ágoston E. Eiben from the Vrije Universiteit, Faculty of Exact Sciences, Amsterdam, Netherlands. Evolutionary Computing in Telecommunication Network Design: A Survey, 2006. Online available at http://www.few.vu.nl/~mei/articles/2006/kampstra/art.pdf [accessed 2008-09-04].

[4] Mark C. Sinclair, David Corne, and George D. Smith, editors. *Proceedings of Evolutionary Telecommunications: Past, Present and Future – A Bird-of-a-feather Workshop at GECCO 99*, July 13, 1999, Orlando, Florida, USA. . Online available at http://uk.geocities.com/markcsinclair/etppf.html [accessed 2008-06-25].

[5] David Wolfe Corne, Martin J. Oates, and George D. Smith, editors. *Telecommunications Optimization: Heuristic and Adaptive Techniques*. John Wiley & Sons, Ltd., September 2000. ISBN: 0-4719-8855-3, 978-0-47198-855-7, 978-0-47084-163-1. doi:10.1002/047084163X.

[6] Witold Pedrycz and Athanasios V. Vasilakos, editors. *Computational Intelligence in Telecommunications Networks*. CRC Press, September 15, 2000. ISBN: 0-8493-1075-X, 978-0-84931-075-1. Partly online available at http://books.google.de/books?id=dE6d-jPoDEEC [accessed 2008-09-03].

[7] Y. Ahmet Şekercioğlu, Andreas Pitsilides, and Athanasios V. Vasilakos. Computational intelligence in management of ATM networks: a survey of current state of research. In *ESIT’99, European Symposium on Intelligent Techniques*, June 3–4, 1999, Orthodox Academy of Crete, Chania-Crete, Greece. Online available at http://www.erudit.de/erudit/events/esit99/12525_P.pdf and http://www.cs.ucy.ac.cy/networksgroup/pubs/published/1999/CI-ATM-London99.pdf [accessed 2008-09-03].

[8] Witold Pedrycz and Athanasios V. Vasilakos. Computational Intelligence: A Development Environment for Telecommunications Networks. In *Computational Intelligence in Telecommunications Networks*, chapter 1, pages 1–27. CRC Press, 2000. In collection [6].

[9] Guy Théraulaz and Eric Bonabeau. Swarm Smarts. *Scientific American*, pages 72–79, March 2000. Online available at http://www.santafe.edu/~vince/press/SciAm-SwarmSmarts.pdf [accessed 2008-06-12].

[10] Athanasios V. Vasilakos, Kostas G. Anagnostakis, and Witold Pedrycz. Application of computational intelligence techniques in active networks. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 5(4):264–271, August 2001. ISSN: 1432-7643 (Print) 1433-7479 (Online). doi:10.1007/s005000100100. Online available at http://www.springerlink.com/content/xxuyux0wgmnt73gy/fulltext.pdf [accessed 2008-09-03].

[11] Enrique Alba and J. Francisco Chicano from the Department Lenguajes y Ciencias de la Computación, ETSI Informática, University of Málaga, Campus de Teatinos, 29071 - Málaga - Spain. Evolutionary algorithms in telecommunications. In *IEEE Mediterranean Electrotechnical Conference, MELECON 2006*, pages 795–798, May 16-19, 2006. doi:10.1109/MELCON.2006.1653218. Online available at http://dea.brunel.ac.uk/ugprojects/docs/melecon06.pdf [accessed 2008-08-01].

[12] Pablo Cortés Achedad, Luis Onieva Giménez, Jesús Muñuzuri Sanz, and José Guadix Martín from the Ingeniería de Organización. Escuela Técnica Superior de Ingenieros, University of Seville, 41092, Seville, Spain. A Revision of Evolutionary Computation Techniques in Telecommunications and An Application for The Network Global Planning Problem. In *Success in Evolutionary Computation*, pages 239–262. Springer Berlin / Heidelberg, 2008. doi:10.1007/978-3-540-76286-7_11. In collection [250].

[13] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, January 1996. ISBN: 0-1950-9971-0.

[14] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation.* Computational Intelligence Library. Oxford University Press in cooperation with the Institute of Physics Publishing, Bristol, New York, ringbound edition, April 1997. ISBN: 0-7503-0392-1.

[15] Thomas Bäck, Ulrich Hammel, and Hans-Paul Schwefel. Evolutionary computation: comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, April 1997. Online available at http://sci2s.ugr.es/docencia/doctobio/EC-History-IEEETEC-1-1-1997.pdf and http://citeseer.ist.psu.edu/601414.html [accessed 2007-08-24].

[16] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, first edition, January 1989. ISBN: 0-2011-5767-5.

[17] John Henry Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, 1975. ISBN: 978-0-26258-111-. Reprinted by MIT Press, April 1992.

[18] Jörg Heitkötter and David Beasley, editors. *Hitch-Hiker's Guide to Evolutionary Computation: A List of Frequently Asked Questions (FAQ).* ENCORE (The EvolutioNary Computation REpository Network), 1998. USENET: comp.ai.genetic. Online available at http://www.cse.dmu.ac.uk/~rij/gafaq/top.htm and http://alife.santafe.edu/~joke/encore/www/ [accessed 2007-07-03].

[19] L. Darell Whitley from the Computer Science Department, Colorado State University, Fort Collins. A Genetic Algorithm Tutorial. *Statistics and Computing*, 4(2):65–85, June 1994. ISSN: ISSN 0960-3174 (Print) 1573-1375 (Online). doi:10.1007/BF00175354. Online available at http://samizdat.mines.

edu/ga_tutorial/ga_tutorial.ps and http://www.citeulike.org/user/Bc91/article/1449453 [accessed 2007-08-12]. Also published as technical report [251].

[20] Nils Aaall Barricelli. Esempi numerici di processi di evoluzione. *Methodos*, pages 45–68, 1954.

[21] Nils Aaall Barricelli. Symbiogenetic Evolution Processes Realized by Artificial Methods. *Methodos*, 9:35–36, 1957.

[22] Nils Aaall Barricelli from the Department of Biology, Division of Molecular Biology, Vanderbilt University, Nashville, Tennessee. Numerical Testing of Evolution Theories. Part I. Theroetical Introduction and Basic Tests. *Acta Biotheoretica*, 16(1/2):69–98, March 1962. ISSN: 0001-5342 (Print) 1572-8358 (Online). doi:10.1007/BF01556771. Received: 27 November 1961, . Online available at http://www.springerlink.com/content/y502315688024453/fulltext.pdf [accessed 2007-10-31].

[23] Nils Aaall Barricelli from the Department of Biology, Division of Molecular Biology, Vanderbilt University, Nashville, Tennessee. Numerical Testing of Evolution Theories. Part II. Preliminary Tests of Performance. Symbiogenesis and Terrestrial Life. *Acta Biotheoretica*, 16(3/4):99–126, September 1963. ISSN: 0001-5342 (Print) 1572-8358 (Online). doi:10.1007/BF01556602. Received: 27 November 1961, . Online available at http://www.springerlink.com/content/h85817217u25w6q7/fulltext.pdf [accessed 2007-10-31].

[24] Alex S. Fraser. Simulation of Genetic Systems by Automatic Digital Computers. *Australian Journal of Biological Science*, 10:484–491, 1957. ISSN: 0004-9417.

[25] Hans J. Bremermann from the Department of Mathematics University of California Berkeley, California. Optimization Through Evolution and Recombination. *Self-Organizing systems*, pages 93–10, 1962. Online available at http://holtz.org/Library/Natural%20Science/Physics/ [accessed 2007-10-31].

[26] Woodrow "Woody" Wilson Bledsoe. Lethally Dependent Genes Using Instant Selection. Technical Report PRI 1, Panoramic Research Inc., Palo Alto, California, USA, 1961.

[27] Woodrow "Woody" Wilson Bledsoe. The Use of Biological Concepts in the Analytical Study of Systems. Technical Report PRI 2, Panoramic Research, Inc., Palo Alto, California, USA, 1961. Presented at ORSA-TIMS National Meeting, San Francisco, California, November 10, 1961.

[28] Woodrow "Woody" Wilson Bledsoe. An Analysis of Genetic Populations. Technical Report, Panoramic Research Inc., Palo Alto, California, USA, 1962.

[29] Woodrow "Woody" Wilson Bledsoe. The Evolutionary Method in Hill Climbing: Convergence Rates. Technical Report, Panoramic Research Inc., Palo Alto, California, USA, 1962.

[30] Woodrow "Woody" Wilson Bledsoe and I. Browning. Pattern Recognition and Reading by Machine. In *Proceedings of the Eastern Joint Computer Conference (EJCC)*, pages 225–232, 1959.

[31] John Daniel Bagley. *The Behavior of Adaptive Systems which employ Genetic and Correlation Algorithms.* PhD thesis, The University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department, Ann Arbor, MI, USA, December 1967. doi:bab0779.0001.001. Order No. AAI6807556. Published as Technical Report and in Dissertations International 28(12), 5106B, University Microfilms No. 68-7556. Online available at http://hdl.handle.net/2027.42/3354 [accessed 2007-10-31].

[32] Daniel Joseph Cavicchio, Jr. *Adaptive Search using Simulated Evolution.* PhD thesis, The University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department, Ann Arbor, Michigan, USA, August 1970. doi:bab9712.0001.001. Published as Technical Report. Chairman: John Henry Holland. Online available at `http://hdl.handle.net/2027.42/4042` [accessed 2007-10-31].

[33] Daniel Joseph Cavicchio, Jr. Reproductive adaptive plans. In *ACM'72: Proceedings of the ACM annual conference*, pages 60–70, 1972, Boston, Massachusetts, United States. ACM, New York, NY, USA. doi:10.1145/800193.805822. Online available at `http://doi.acm.org/10.1145/800193.805822` [accessed 2008-09-25].

[34] Daniel Raymond Frantz. *Nonlinearities in Genetic Adaptive Search.* PhD thesis, The University of Michigan, Ann Arbor, MI, USA, September 1972. Order No. AAI7311116. Chairman: John Henry Holland. Published as Technical Report Nr. 138 and in Dissertations International 33(11), 5240B-5241B, University Microfilms No. 73-11116. Online available at `http://hdl.handle.net/2027.42/4950` [accessed 2007-10-31].

[35] John Henry Holland. Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*, 9(3):297–314, 1962. ISSN: 0004-5411. doi:10.1145/321127.321128. Online available at `http://portal.acm.org/citation.cfm?id=321128` [accessed 2007-07-28].

[36] John Henry Holland from the The University of Michigan, Ann Arbor, MI, USA. Adaptive Plans Optimal for Payoff-Only Environments. In *Proceedings of the Second Hawaii International Conference on System Sciences, Periodicals*, pages 917–920. North Holland, January 22–24, 1969, University of Hawaii, Honolulu, USA. DTIC Accession Number: AD0688839.

[37] John Henry Holland. *Nonlinear environments permitting efficient adaptation*, volume II of *Computer and Information Sciences.* Academic Press, New York, USA, 1967.

[38] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* A Bradford Book. The MIT Press, Cambridge, Massachusetts, USA, 1992 first edition, 1993 second edition, 1992. ISBN: 0-2621-1170-5, 978-0-26211-170-6. Partly online available at `http://books.google.de/books?id=Bhtxo60BV0EC` [accessed 2008-08-16].

[39] Peter John Angeline from the Natural Selection, Inc. A Historical Perspective on the Evolution of Executable Structures. *Fundamenta Informaticae*, 35(1–4):179–195, August 1998. ISSN: 0169-2968. Special volume: Evolutionary Computation. Online available at `http://citeseer.ist.psu.edu/angeline98historical.html` and `http://www.natural-selection.com/Library/1998/gphist.pdf` [accessed 2008-05-29]. .

[40] Richard M. Friedberg. A learning machine: Part I. *IBM Journal of Research and Development*, 2:2–13, November 1958. Online available at `http://www.research.ibm.com/journal/rd/021/ibmrd0201B.pdf` [accessed 2007-09-06]. .

[41] Richard M. Friedberg, B. Dunham, and J. H. North. A Learning Machine: Part II. *IBM Journal of Research and Development*, 3(3):282–287, March 1959. Online available at `http://www.research.ibm.com/journal/rd/033/ibmrd0303H.pdf` [accessed 2007-09-06]. .

[42] Lawrence Jerome Fogel, Alvin J. Owens, and Michael J. Walsh. *Artificial Intelligence through Simulated Evolution.* John Wiley & Sons, New York, USA, October 1966. ISBN: 978-0-47126-516-0.

[43] Stephen Frederick Smith. *A Learning System based on Genetic Adaptive Algorithms.* PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1980. Order No. AAI8112638, University Microfilms No. 81-12638.

[44] Richard Forsyth from the Department of Mathematics, Polytechnic of North London, Holloway Road, London N78DB, UK. BEAGLE A Darwinian Approach to Pattern Recognition. *Kybernetes*, 10:159–166, 1981. Received December 17, 1980. Online available at http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/kybernetes_forsyth.pdf [accessed 2007-11-01] (copy from British Library May 1994).

[45] Richard Forsyth and Roy Rada. *Machine Learning applications in Expert Systems and Information Retrieval.* Ellis Horwood series in Artificial Intelligence. Ellis Horwood, Chichester, UK, 1986. ISBN: 0-7458-0045-9, 0-4702-0309-9. Contains chapters on BEAGLE, see [44]. Also published by John Wiley & Sons Australia (May 28, 1986) and Halsted Press, New York, NY, USA.

[46] Richard Forsyth. The Evolution of Intelligence. In Richard Forsyth, editor, *Machine Learning, Priciples and Techniques*, chapter 4, pages 65–82. Chapman and Hall, 1989. ISBN: 0-4123-0570-4, 0-4123-0580-1. Refers also to PC/BEAGLE, see [44].

[47] John R. Koza. *Non-Linear Genetic Algorithms for Solving Problems.* United States Patent and Trademark Office, 1988. United States Patent 4,935,877. Filed May 20, 1988. Issued June 19, 1990. Australian patent 611,350 issued september 21, 1991. Canadian patent 1,311,561 issued december 15, 1992.

[48] John R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI-89*, pages 768–774, 1989. In proceedings [252]. Partly online available at http://dli.iiit.ac.in/ijcai/IJCAI-89-VOL1/PDF/123.pdf [accessed 2008-05-29].

[49] John R. Koza. The genetic programming paradigm: Genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Computer Science Department, Stanford University, Margaret Jacks Hall, Stanford, CA 94305, June 1990. Online available at http://citeseer.ist.psu.edu/koza90genetic.html [accessed 2008-05-29]. .

[50] John R. Koza from the Computer Science Department, Stanford University, Stanford CA 94305, USA. A Hierarchical Approach to Learning the Boolean Multiplexer Function. In *Proceedings of the First Workshop on Foundations of Genetic Algorithms*, pages 171–191, 1990. In proceedings [253]. Online available at http://citeseer.ist.psu.edu/koza91hierarchical.html and http://books.google.de/books?id=Df12yLrlUZYC [accessed 2008-05-29].

[51] John R. Koza from the Computer Science Department, Margaret Jacks Hall, Stanford University, Stanford, CA 94305 USA. Genetic evolution and co-evolution of computer programs. In *Artificial Life II*, pages 603–629, 1990. Revised November 29, 1990. In proceedings [254]. Online available at http://citeseer.ist.psu.edu/177879.html [accessed 2008-05-29].

[52] John R. Koza. Evolution and co-evolution of computer programs to control independent-acting agents. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 366–375, 1990. In proceedings [255]. Online available at http://www.genetic-programming.com/jkpdf/sab1990.pdf [accessed 2008-05-29].

[53] Jean-Louis Deneubourg, Jacques M. Pasteels, and J. C. Verhaeghe. Probabilistic behaviour in ants: a strategy of errors? *Journal of Theoretical Biology*, 105(2):259–271, 1983. doi:10.1016/S0022-5193(83)80007-1. Online available at `http://dx.doi.org/10.1016/S0022-5193(83)80007-1` [accessed 2008-03-27].

[54] Jean-Louis Deneubourg and Simon Goss. Collective patterns and decision-making. *Ethology, Ecology & Evolution*, 1(4):295–311, December 1989.

[55] Simon Goss, R. Beckers, Jean-Louis Deneubourg, S. Aron, and Jacques M. Pasteels. How trail laying and trail following can solve foraging problems for ant colonies. In R. N. Hughes, editor, *Behavioural Mechanisms of Food Selection*, NATO ASI Series, G 20, pages 661–678. Springer-Verlag, Berlin, April 1990. ISBN: 0-3875-1762-6, 978-0-38751-762-9.

[56] T. R. Stickland, C. M. N. Tofts, and N. R. Franks from the School of Biological Sciences, University of Bath, Claverton Down, Bath, Avon, UK. A path choice algorithm for ants. *Naturwissenschaften*, 79(12):567–572, December 1992. ISSN: 0028-1042 (Print) 1432-1904 (Online). doi:10.1007/BF01131415. Online available at `http://www.springerlink.com/content/v2t8328844843156/fulltext.pdf` [accessed 2008-06-12].

[57] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996. Online available at `ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.10-SMC96.pdf` and `http://citeseer.ist.psu.edu/dorigo96ant.html` [accessed 2007-08-05].

[58] Marco Dorigo and Christian Blum. Ant colony optimization theory: a survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005. ISSN: 0304-3975. Online available at `http://code.ulb.ac.be/dbfiles/DorBlu2005tcs.pdf` [accessed 2007-08-05].

[59] Marco Dorigo, Gianni Di Caro, and Luca Maria Gambardella. Ant Algorithms for Discrete Optimization. Technical Report IRIDIA/98-10, Universite Libre de Bruxelles, Belgium, 1998. Online available at `http://citeseer.ist.psu.edu/dorigo98ant.html` [accessed 2007-08-05]. See [256].

[60] Bernard Manderick and Frans Moyson. The collective behaviour of Ants: an Example of Self-Organization in Massive Parallelism. In *Proceedings of AAAI Spring Symposium on Parallel Models of Intelligence*, 1988, Stanford, California.

[61] Vittorio Maniezzo, Luca Maria Gambardella, and Fabio de Luigi. Ant Colony Optimization. In *New Optimization Techniques in Engineering*, chapter 5, pages 101–117. Springer-Verlag, 2004. In collection [257]. Online available at `http://www.idsia.ch/~luca/aco2004.pdf` and `http://citeseer.ist.psu.edu/maniezzo04ant.html` [accessed 2007-09-13].

[62] Marco Dorigo, Mauro Birattari, and Thomas Stützle from the Univ. Libre de Bruxelle, Brussels, Belgium. Ant colony optimization. *Computational Intelligence Magazine*, 1 (4):28–39, November 2006. ISSN: 1556-603X. doi:10.1109/MCI.2006.329691. INSPEC Accession Number: 9184238.

[63] Pierre-Paul Grassé. La reconstruction du nid et les coordinations inter-individuelles chez bellicositermes natalensis et cubitermes sp. la théorie de la stigmergie: Essai d'interprétation des termites constructeurs. *Insectes Sociaux*, 6:41–48, 1959, Paris.

[64] Ruud Schoonderwoerd, Owen E. Holland, Janet L. Bruten, and Leon J. M. Rothkrantz. Ant-Based Load Balancing in Telecommunications Networks. *Adaptive Behavior*, 5(2):169–207, Fall 1996. Online available at `http://citeseer.ist.psu.edu/schoonderwoerd96antbased.html` and `http://www.ifi.unizh.ch/ailab/teaching/FG06/` [accessed 2007-08-19].

[65] Edward Osborne Wilson. *Sociobiology: The New Synthesis*. Belknap Press / Harvard University Press, Cambridge, Massachusets, USA / Cumbreland, Rhode Island, USA, 1975. .

[66] Guy Théraulaz and Eric Bonabeau. Coordination in Distributed Building. *Science*, 269(5224):686–688, August 4, 1995. doi:10.1126/science.269.5224.686.

[67] Justin Werfel and Radhika Nagpal. Extended Stigmergy in Collective Construction. *IEEE Intelligent Systems*, 21(2):20–28, March/April 2006. doi:10.1109/MIS.2006.25. Online available at `http://hebb.mit.edu/people/jkwerfel/ieeeis06.pdf` [accessed 2008-06-12].

[68] Crystal Schuil, Matthew Valente, Justin Werfel, and Radhika Nagpal. Collective Construction Using Lego Robots. In *Annual National Conference on Artificial Intelligence (AAAI), Robot Exhibition*, 2006. In proceedings [258]. Online available at `http://www.eecs.harvard.edu/~rad/ssr/papers/aaai06-schuil.pdf` [accessed 2008-06-12].

[69] Justin Werfel, Yaneer Bar-Yam, Daniela Rus, and Radhika Nagpal. Distributed Construction by Mobile Robots with Enhanced Building Blocks. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 15–19, 2006, Hilton in the Walt Disney World Resort hotel, Walt Disney World Resort, Orlando, Florida (Lake Buena Vista), USA. Online available at `http://www.eecs.harvard.edu/~rad/ssr/papers/icra06-werfel.pdf` [accessed 2008-06-12].

[70] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953. doi:10.1063/1.1699114. Online available at `http://link.aip.org/link/?JCPSA6/21/1087/1` [accessed 2008-03-26].

[71] Scott Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 13, 1983. Online available at `http://fezzik.ucd.ie/msc/cscs/ga/kirkpatrick83optimization.pdf` and `http://citeseer.ist.psu.edu/kirkpatrick83optimization.html` [accessed 2008-03-26].

[72] V. Černý from the Institute of Physics and Biophysics, Comenius University, Mlynska Dolina, Bratislava, Czechoslovakia. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, January 1985. ISSN: 0022-3239 (Print) 1573-2878 (Online). doi:10.1007/BF00940812. Communicated by S. E. Dreyfus. Online available at `http://www.springerlink.com/content/gt0743622913gg33/fulltext.pdf` [accessed 2008-03-26]. Also: Technical Report, Comenius University, Bratislava, Czechoslovakia, 1982.

[73] E. L. Lawler, Jan Karel Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley Series in Discrete Mathematics & Optimization. Wiley Interscience, Chichester, UK, September 1985. ISBN: 0-4719-0413-9, 978-0-47190-413-7.

[74] David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics. Princeton University Press, 41 William Street, Princeton, New Jersey, USA, 2006. ISBN: 978-0-69112-993-8, 0-6911-2993-2.

[75] Andreas Nolte and Rainer Schrader from the Mathematisches Institut, Universität zu Köln. A Note on the Finite Time Behaviour of Simulated Annealing. *Mathematics of OR*, 25(347):476–484, 2000. Online available at `http://citeseer.`

ist.psu.edu/nolte99note.html and http://www.zaik.de/~paper/unzip.html?file=zaik1999-347.ps [accessed 2008-03-26] (revised version from March 1999). .

[76] P. J. M. van Laarhoven and E. H. L. Aarts, editors. *Simulated Annealing: Theory and Applications*. Mathematics and Its Applications. Springer, Kluwer Academic Publishers, Norwell, MA, USA, June 30, 1987. ISBN: 9-0277-2513-6, 978-9-02772-513-4. Reviewed in [259]. Partly online available at http://books.google.de/books?id=-IgUab6Dp_IC [accessed 2008-03-26].

[77] B. Hajek. Cooling Schedules for Optimal Annealing. *Mathematics of Operations Research*, (13), 1988.

[78] S. Anily and Awi Federgruen. Ergodicity in parametric non-stationary Markov chains: An Application to Simulated Annealing Methods. *Operations Research*, (35):867–874, 1987.

[79] Basilis Gidas. Nonstationary Markov chains and convergence of the annealing algorithm. *Journal of Statistical Physics*, 39(1-2):73–131, April 1985. ISSN: 0022-4715 (Print) 1572-9613 (Online). doi:10.1007/BF01007975. Online available at http://www.springerlink.com/content/x3821j328gv33085/fulltext.pdf [accessed 2008-03-26].

[80] Andreas Nolte and Rainer Schrader. A Note on the Finite Time Behaviour of Simulated Annealing. In U. Zimmermann, U. Derigs, W. Gaul, R. Möhring, and K.-P. Schuster, editors, *Operations Research Proceedings 1996, Selected Papers of the Symposium on Operations Research (SOR 96)*, pages 175–180. Springer, 1996. Published 1997. .

[81] Debasis Mitra, Fabio Romeo, and Alberto Sangiovanni-Vincentelli from the AT & T Bell Laboratories, Murray Hill, NJ. Convergence and finite-time behavior of simulated annealing. In *Proceedings of the 24th IEEE Conference on Decision and Control*, volume 24, pages 761–767. IEEE Computer Society, December 1985. doi:10.1109/CDC.1985.268600.

[82] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986. ISSN: 0305-0548. doi:10.1016/0305-0548(86)90048-1. Online available at http://dx.doi.org/10.1016/0305-0548(86)90048-1 [accessed 2008-03-27].

[83] Fred Glover, Eric Taillard, and Dominique de Werra. A user's guide to tabu search. *Annals of Operations Research*, 41(1):3–28, 1993. ISSN: 0254-5330, 1572-9338. doi:10.1007/BF02078647. Special issue on Tabu search. Online available at http://www.springerlink.com/content/k607q47112622n45/fulltext.pdf [accessed 2008-08-29].

[84] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In *Talk presented at the Congress on Numerical Methods in Combinatorial Optimization*. Academic Press, 1986, Capri, Italy.

[85] Fred Glover. Tabu Search – Part I. *Operations Research Society of America (ORSA) Journal on Computing*, 1(3):190–206, Summer 1989. Online available at http://leeds-faculty.colorado.edu/glover/TS%20-%20Part%20I-ORSA.pdf [accessed 2008-03-27].

[86] Fred Glover. Tabu Search – Part II. *Operations Research Society of America (ORSA) Journal on Computing*, 2(1):190–206, Winter 1990. Online available at http://leeds-faculty.colorado.edu/glover/TS%20-%20Part%20II-ORSA.pdf [accessed 2008-03-27].

[87] Dominique de Werra and Alain Hertz. Tabu Search Techniques: A tutorial and an application to neural networks. *OR Spectrum*, 11:131–141, 1989. ISSN: 0171-6468 (Print) 1436-6304 (Online). Online available at http://www.springerlink.de/content/x25k97k0qx237553/fulltext.pdf [accessed 2008-03-27].

[88] Alain Hertz, Eric Taillard, and Dominique de Werra. A Tutorial on Tabu Search. In *Proceedings of Giornate di Lavoro AIRO'95 (Enterprise Systems: Management of Technological and Organizational Changes)*, pages 13–24, 1995, Italy. Online available at http://citeseer.ist.psu.edu/73006.html [accessed 2007-09-15].

[89] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6 (2):126–140, 1994. Online available at http://citeseer.ist.psu.edu/141556.html and http://rtm.science.unitn.it/~battiti/archive/reactive-tabu-search.ps.gz [accessed 2007-09-15].

[90] Djurdje Cvijović and Jacek Klinowski from the Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK. Taboo search: an approach to the multiple minima problem. *Science*, 267(5198):664–666, February 3, 1995. ISSN: 0036-8075. doi:10.1126/science.267.5198.664.

[91] Susan Coombs and Lawrence Davis. Genetic algorithms and communication link speed design: constraints and operators. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 257–260, 1987. In proceedings [260]. Partly online available at http://www.questia.com/PM.qst?a=o&d=97597557 [accessed 2008-08-29].

[92] Zbigniew Michalewicz. A Step Towards Optimal Topology of Communication Networks. In Vibeke Libby, editor, *Proceedings of Data Structures and Target Classification, the SPIE's International Symposium on Optical Engineering and Photonics in Aerospace Sensing*, volume 1470 of *SPIE*, pages 112–122. SPIE – The International Society for Optical Engineering., April 1–5 1991, Orlando, Florida. ISBN: 0-8194-0579-5, 978-0-81940-579-1.

[93] Anup Kumar, Rakesh M. Pathak, M. C. Gupta, and Yash P. Gupta. Genetic Algorithm based Approach for Designing Computer Network Topolgies. In *CSC'93: Proceedings of the 1993 ACM conference on Computer science*, pages 358–365, February 16–18, 1993, Indianapolis, Indiana, United States. ACM, New York, NY, USA. ISBN: 0-8979-1558-5. doi:10.1145/170791.170871. Online available at http://doi.acm.org/10.1145/170791.170871 [accessed 2008-08-01].

[94] Mark C. Sinclair from the Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester, Essex, CO4 3SQ, UK. Optical Mesh Network Topology Design using Node-Pair Encoding Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1192–1197, 1999. In proceedings [261]. Online available at http://citeseer.ist.psu.edu/177545.html and http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/sinclair_1999_OMNTDNEGP.html [accessed 2008-07-29].

[95] Aaron Kershenbaum. *Telecommunications Network Design Algorithms*. McGraw-Hill Companies, January 15, 1993. ISBN: 0-0703-4228-8, 978-0-07034-228-6.

[96] Faris N. Abuali, Dale A. Schoenefeld, and Roger L. Wainwright from the Mathematical and Computer Science Department, The University of Tulsa, 600 South College Avenue, Tulsa, OK 74104-3189, USA. Terminal assignment in a communications network using genetic algorithms. In *CSC'94: Proceedings of the 22nd annual ACM computer science conference on Scaling up: meeting the challenge of complexity in real-world computing applications*, pages 74–81, March 8–10, 1994,

Phoenix, Arizona, United States. ACM, New York, NY, USA. ISBN: 0-8979-1634-4. doi:10.1145/197530.197559. Online available at http://euler.mcs.utulsa.edu/~rogerw/papers/Abuali-Terminal-CSC-94.pdf [accessed 2008-08-28].

[97] Sami Khuri and Teresa Chiu from the Department of Mathematics and Computer Science, San José State University, One Washington Square, San José, CA. Heuristic algorithms for the terminal assignment problem. In *SAC'97: Proceedings of the 1997 ACM symposium on Applied computing*, pages 247–251, February 28–March 2, 1997, San Jose, California, United States. ACM, New York, NY, USA. ISBN: 0-8979-1850-9. doi:10.1145/331697.331748. Online available at http://doi.acm.org/10.1145/331697.331748 and http://citeseer.ist.psu.edu/khuri97heuristic.html [accessed 2008-06-15].

[98] Xin Yao, F. Wang, K. Padmanabhan, and Sancho Salcedo-Sanz from the Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), the University of Birmingham, U.K. Hybrid Evolutionary Approaches to Terminal Assignment in Communications Networks. In *Recent Advances in Memetic Algorithms*, pages 129–159. Springer, 2005. doi:10.1007/3-540-32363-5_7. In collection [262].

[99] L. Darrell Whitley, V. Scott Gordon, and Keith E. Mathias from the Computer Science Department, Colorado State University, Fort Collins, CO 80523, USA. Lamarckian Evolution, The Baldwin Effect and Function Optimization. In *PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*, pages 6–15, 1994. In proceedings [263]. Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2428 [accessed 2008-09-11].

[100] Katharina Anna Zweig from the Fakultät für Informations- und Kognitionswissenschaften der Eberhard-Karls-Universität Tübingen. *On Local Behavior and Global Structures in the Evolution of Complex Networks*. PhD thesis, Fakultät für Informations- und Kognitionswissenschaften der Eberhard-Karls-Universität Tübingen, July 2007. Online available at http://www-pr.informatik.uni-tuebingen.de/mitarbeiter/katharinazweig/downloads/Diss.pdf [accessed 2008-06-12]. Zweig was formerly known as Lehmann.

[101] Katharina Anna Lehmann and Michael Kaufmann. Evolutionary algorithms for the self-organized evolution of networks. In *GECCO'05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 563–570, 2005. doi:10.1145/1068009.1068105. In proceedings [264]. Online available at http://doi.acm.org/10.1145/1068009.1068105 and http://www-pr.informatik.uni-tuebingen.de/mitarbeiter/katharinazweig/downloads/299-lehmann.pdf [accessed 2008-05-28]. Lehmann is now known as Zweig.

[102] General Secretariat of the Council of the European Union; COST Secretariat. *COST 1991–1992*, volume 7 of *Collected Agreements Concluded within the Framework of European Cooperation in the Field of Scientific and Technical Research*. Office for Official Publications of the European Communities, Luxembourg, 1994. ISBN: 9-2824-1119-2, 9-2824-1120-6.

[103] P. Aiyarak, A. S. Saket, and Mark C. Sinclair from the Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester, Essex, CO4 3SQ, UK. Genetic Programming Approaches for Minimum Cost Topology Optimisation of Optical Telecommunication Networks. In *Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA*, 1997. In proceedings [265]. Online available at http://citeseer.ist.psu.edu/30631.html and http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/aiyarak_1997_GPtootn.html [accessed 2008-07-29].

[104] Mark C. Sinclair from the Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester, C04 3SQ, UK. Minimum cost topology optimisation of the COST 239 European optical network. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 26–29, 1995. In proceedings [266]. Online available at http://citeseer.ist.psu.edu/63056.html [accessed 2008-07-29].

[105] M. O'Mahony, Mark C. Sinclair, and B. Mikac. Ultra-High Capacity Optical Transmission Network: European Research Project COST 239. In Mladen Kos, editor, *International Conference on Telecommunications, ConTEL'93*. Croatian Telecommunications Society, June 1993, Zagreb, Croatia. OCLC: 77843406. Informacija – Telecomunikacije – Automati Journal, 12(1–3):33–45, 1993.

[106] L. G. Tan and Mark C. Sinclair from the Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester, C04 3SQ, UK. Wavelength Assignment between the Central Nodes of the COST 239 European Optical Network. In *Proceedings of the 11th UK Performance Engineering Workshop*, pages 235–247, September 1995, Liverpool, UK. Online available at http://citeseer.ist.psu.edu/99015.html and http://uk.geocities.com/markcsinclair/ps/ukpew11_tan.ps.gz [accessed 2008-07-29].

[107] Sean Luke and Lee Spector. Evolving Graphs and Networks with Edge Encoding: A Preliminary Report. In *Late Breaking Papers at the First Annual Conference Genetic Programming (GP-96)*, 1996. In proceedings [267]. Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.9484 and http://cs.gmu.edu/~sean/papers/graph-paper.pdf [accessed 2008-08-02].

[108] Mark C. Sinclair. Node-Pair Encoding Genetic Programming for Optical Mesh Network Topology Design. In *Telecommunications Optimization: Heuristic and Adaptive Techniques*, chapter 6, pages 99–114. John Wiley & Sons, Ltd., 2000. In collection [5].

[109] Edsger Wybe Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. Online available at http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf [accessed 2008-06-24].

[110] I. M. A. Kirkwood, S. H. Shami, and Mark C. Sinclair from the Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester, Essex CO4 3SQ, UK. Discovering Simple Fault-Tolerant Routing Rules using Genetic Programming. In *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, 1997. ISBN: 3-2118-3087-1. In proceedings [268]. Online available at http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/kirkam_1997_dsftrr.html and http://citeseer.ist.psu.edu/120285.html [accessed 2008-06-24]. .

[111] S. H. Shami, I. M. A. Kirkwood, and Mark C. Sinclair. Evolving simple fault-tolerant routing rules using genetic programming. *Electronics Letters*, 33(17):1440–1441, August 14, 1997. ISSN: Print: 0013-5194, Online: 1350-911X. doi:10.1049/el:19970996.
.

[112] C. Hedrick from the Rutgers University. Routing Information Protocol. Request for Comments (RFC) 1058, Internet Engineering Task Force (IETF), Internet Society (ISOC), Network Working Group, June 1988. Online available at http://tools.ietf.org/html/rfc1058 and http://www.faqs.org/rfcs/rfc1058.html [accessed 2008-06-27].

[113] John Moy from the Proteon, Inc., 2 Technology Drive, Westborough, MA 01581-5008. The OSPF Specification. Request for Comments (RFC) 1131, Internet Engineering Task Force (IETF), Internet Society (ISOC), October 1989. Online available at http://www.packetizer.com/rfc/rfc1131/pdf [accessed 2008-06-27].

[114] Masaharu Munetomo, Yoshiaki Takai, and Yoshiharu Sato. An Adaptive Network Routing Algorithm Employing Path Genetic Operators. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 643–649, 1997. In proceedings [269].

[115] Masaharu Munetomo, Yoshiaki Takai, and Yoshiharu Sato. An adaptive routing algorithm with load balancing by a genetic algorithm. *Transactions of the Information Processing Society of Japan (IPSJ)*, 39(2):219–227, 1998. in Japanese.

[116] Masaharu Munetomo, Yoshiaki Takai, and Yoshiharu Sato. A Migration Scheme for the Genetic Adaptive Routing Algorithm. In *Proceedings of the 1998 IEEE Conference on Systems, Man, and Cybernetics*, volume 3, pages 2774–2779. IEEE Press, October 11–14, 1998, San Diego, U.S.A. doi:10.1109/ICSMC.1998.725081.

[117] Masaharu Munetomo from the Graduate School of Engineering, Hokkaido University, North 13, West 8, Sapporo 060-8628, Jappan. Designing genetic algorithms for adaptive routing algorithms in the internet. In *Evolutionary Telecommunications: Past, Present, and Future*, pages 215–216, 1999. Online available at http://citeseer.ist.psu.edu/198441.html and http://uk.geocities.com/markcsinclair/ps/etppf_mun.ps.gz [accessed 2008-06-25]. Presentation online available at http://uk.geocities.com/markcsinclair/ps/etppf_mun_slides.ps.gz and http://assam.cims.hokudai.ac.jp/~munetomo/documents/gecco99-ws.ppt [accessed 2008-06-25].

[118] Ruud Schoonderwoerd, Owen E. Holland, and Janet L. Bruten. Ant-like agents for load balancing in telecommunications networks. In Jörg Mller, editor, *AGENTS'97: Proceedings of the first international conference on Autonomous agents*, pages 209–216, February 5–8 1997, Marina del Rey, California, United States. ACM, New York, NY, USA. ISBN: 0-8979-1877-0. doi:10.1145/267658.267718. Online available at http://staff.washington.edu/paymana/swarm/schoon97-icaa.pdf and http://doi.acm.org/10.1145/267658.267718 [accessed 2008-07-26]. Slides online available at [accessed 2008-07-26]. See also http://sigart.acm.org/proceedings/agents97/ [accessed 2008-07-26].

[119] Ruud Schoonderwoerd. Collective intelligence for network control. Master's thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, May 30, 1996. Online available at http://staff.washington.edu/paymana/swarm/schoon96-thesis.pdf [accessed 2008-06-12]. Graduation committee: Prof. dr. H. Koppelaar, Prof. dr. ir. E.J.H. Kerckhoffs, Drs. dr. L.J.M. Rothkrantz, and J.L. Bruten M.Sc.

[120] Gianni Di Caro and Marco Dorigo from the IRIDIA, Université Libre de Bruxelles, 50, av. F. Roosevelt, CP 194/6, 1050, Brussels, Belgium. AntNet: Distributed Stigmergetic Control for Communications Networks. *Journal of Artificial Intelligence Research (JAIR)*, 9:317–365, December 1998. ISSN: 11076-9757. Submitted 5/98; published 12/98. Online available at http://citeseer.ist.psu.edu/dicaro98antnet.html and http://www.jair.org/media/544/live-544-1748-jair.pdf [accessed 2008-07-26].

[121] Gianni Di Caro and Marco Dorigo from the IRIDIA, Université Libre de Bruxelles 50, av. F. Roosevelt, CP 194/6 1050, Brussels, Belgium. Two ant colony algorithms for best-effort routing in datagram networks. In *Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, pages 541–546. IASTED/ACTA Press, October 1–4, 1998, Las Vegas, USA. Online available at http://citeseer.ist.psu.edu/dicaro98two.html and http://www.idsia.ch/~gianni/Papers/PDCS98.ps.gz [accessed 2008-07-26].

[122] Gianni Di Caro from the IRIDIA, Université Libre de Bruxelles, 50, av. F. Roosevelt, CP 194/6, 1050, Brussels, Belgium. *Ant Colony Optimization and its application to adaptive routing in telecommunication networks*. PhD thesis, Applied Sciences, Polytechnic School, Université Libre de Bruxelles, Brussels, Belgium, September 2004. Advisor: Marco Dorigo. Partly online available at http://www.idsia.ch/~gianni/Papers/thesis-abstract.pdf [accessed 2008-07-26] and http://www.idsia.ch/~gianni/Papers/routing-chapters.pdf [accessed 2008-07-26].

[123] Suiliong Liang, A. Nur Zincir-Heywood, and Malcom I. Heywood from the Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada. The effect of routing under local information using a social insect metaphor. In *CEC'02: Proceedings of the Congress on Evolutionary Computation*, pages 1438–1443, 2002. IEEE Computer Society, Washington, DC, USA. In proceedings [270]. Online available at http://users.cs.dal.ca/~mheywood/X-files/Publications/01004454.pdf [accessed 2008-07-26].

[124] Suiliong Liang, A. Nur Zincir-Heywood, and Malcolm I. Heywood from the NIMS Group, Faculty of Computer Science, Dalhousie University, 6050 University Avenue, Halifax, NS, Canada B3H 1W5. Adding more intelligence to the network routing problem: AntNet and Ga-agents. *Applied Soft Computing*, 6(3):244–257, March 2006. ISSN: 1568-4946. doi:10.1016/j.asoc.2005.01.005.

[125] F. Comellas and G. Giménez from the Department de Matemàtica Aplicata i Telemàtica, Universitat Politècnica de Catalunya, Campus Nord, Edifici C3, 08034 Barcelona, Catalonia, Spain. Genetic Programming to Design Communication Algorithms for Parallel Architectures. *Parallel Processing Letters*, 8(4):549–560, 1998. ISSN: 0129-6264. Online available at http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/Comellas_1998_GPD.html and http://citeseer.ist.psu.edu/comellas98genetic.html [accessed 2007-09-14].

[126] Ralf Klasing, Burkhard Monien, R. Peine, and Elena Stöhr. Broadcasting in butterfly and de Bruijn networks. In Alain Finkel and Matthias Jantzen, editors, *Proceedings of 9th Annual Symposium on Theoretical Aspects of Computer Science (STACS'92)*, volume 577 of *Lecture Notes in Computer Science (LNCS)*, pages 351–362, February 13–15, 1992, Cachan, France. Springer, Berlin, Heidelberg, New York. ISBN: 978-3-54055-210-9.

[127] Jiri Jaroš and Vaclav Dvořák from the Brno University of Technology, Bozetechova 2, 612 66, Brno, CZ. An Evolutionary Design Technique for Collective Communications on Optimal Diameter-Degree Networks. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2008. In proceedings [271].

[128] *The gnutella protocol specification v0.4*. Clip2 Distributed Search Services, August 10, 2000. Online available at http://www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf [accessed 2008-06-24]. Document Revision 1.2, June 15, 2001 online available at http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf [accessed 2008-06-24].

[129] Matei Ripeanu, Ian Foster, and Adriana Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1):50–57, February 2002. Online available at http://people.cs.uchicago.edu/~matei/papers/ic.pdf and http://citeseer.ist.psu.edu/ripeanu02mapping.html [accessed 2008-06-24].

[130] Michael Iles and Dwight Lorne Deugo. A Search for Routing Strategies in a Peer-to-Peer Network Using Genetic Programming. In *SRDS'02: Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems (SRDS'02)*, pages 341–346, October 13–16, 2002, Osaka, Japan. IEEE Computer Society, Washington, DC, USA.

ISBN: 0-7695-1659-9. doi:10.1109/RELDIS.2002.1180207. Inspec Accession Number: 7516795.

[131] Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Marina del Rey, California 90291, USA. Internet Protocol, Darpa Internet Program Protocol Specification. RFC 791, Defense Advanced Research Projects Agency, Information Processing Techniques Office, 1400 Wilson Boulevard, Arlington, Virginia 22209, USA, September 1981. Online available at `http://tools.ietf.org/html/rfc791` [accessed 2008-06-13].

[132] Vinton Cerf, Yogen Dalal, and Carl Sunshine. Specification of Internet Transmission Control Program (December 1974 Version). Request for Comments (RFC) 675, Network Working Group, December 1974. Online available at `http://tools.ietf.org/html/rfc675` [accessed 2008-06-13].

[133] Hirozumi Yamaguchi, Kozo Okano, Teruo Higashino, and Kenichi Taniguchi from the Department of Information and Computer Sciences, Osaka University, Machikaneyama 1-3, Toyonaka, Osaka 560, Japan. Synthesis of Protocol Entities' Specifications from Service Specifications in a Petri Net Model with Registers. In *ICDCS'95: Proceedings of the 15th International Conference on Distributed Computing Systems*, pages 510–517, May 30–June 2, 1995, Vancouver, British Columbia, Canada. IEEE Computer Society, Washington, DC, USA. ISBN: 0-8186-7025-8. Online available at `http://citeseer.ist.psu.edu/yamaguchi95synthesis.html` [accessed 2008-06-15].

[134] Khaled El-Fakihy, Hirozumi Yamaguchi, and Gregor von Bochmann. A Method and a Genetic Algorithm for Deriving Protocols for Distributed Applications with Minimum Communication Cost. In *Proceedings of Eleventh IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'99)*, November 3–6, 1999, Cambridge, Boston, MA, USA. Online available at `http://citeseer.ist.psu.edu/430349.html` and `http://www-higashi.ist.osaka-u.ac.jp/~h-yamagu/resource/pdcs99.pdf` [accessed 2007-09-14].

[135] Sérgio Granato de Araújo, Aloysio de Castro Pinto Pedroza, and Antônio Carneiro de Mesquita Filho from the Electrical Engineering Deptartment, Universidade Federal do Rio de Janeiro (UFRJ), Caixa Postal 68504, CEP 21945-970, Rio de Janeiro, RJ, Brazil. Evolutionary Synthesis of Communication Protocols. In *Proceedings of the 10th International Conference on Telecommunications (ICT'03)*, volume 2, pages 986–993, February 23–March 1, 2003, Tahiti, French Polynesia. doi:10.1109/ICTEL.2003.1191573. Online available at `http://www.gta.ufrj.br/ftp/gta/TechReports/AMP03a.pdf` [accessed 2008-06-21]. for Portuguese versions.

[136] Sérgio Granato de Araújo, Aloysio de Castro Pinto Pedroza, and Antônio Carneiro de Mesquita Filho from the Electrical Engineering Deptartment, Universidade Federal do Rio de Janeiro (UFRJ), Caixa Postal 68504, CEP 21945-970, Rio de Janeiro, RJ, Brazil. Uma Metodologia de Projeto de Protocolos de Comunicação Baseada em Técnicas Evolutivas. In *XX Simpósio Brasileiro de Telecomunicações (SBT'03)*, October 5–8, 2003, Rio de Janeiro, Brasil. Online available at `http://www.gta.ufrj.br/ftp/gta/TechReports/AMP03e.pdf` [accessed 2008-06-21]. and the English version [135].

[137] Helmut Hörner. A C++ Class Library for GP: Vienna University of Economics Genetic Programming Kernel (Release 1.0, operating instructions). Technical Report, Vienna University of Economics, May 29, 1996. Online available at `http://citeseer.ist.psu.edu/253491.html` [accessed 2007-09-09].

[138] Nicholas Peter Sharples and Ian Wakeman from the University of Sussex Falmer, Brighton, BN1 9QH. Protocol Construction Using Genetic Search Techniques. In

Real-World Applications of Evolutionary Computing, EvoWorkshops 2000: EvoIASP, EvoSCONDI, EvoTel, EvoSTIM, EvoROB, and EvoFlight, pages 73–95, 2000. doi:10.1007/3-540-45561-2_23. In proceedings [272].

[139] Nicholas Peter Sharples. Evolutionary Approaches to Adaptive Protocol Design. In Darren Pearce, editor, *CSRP 512: The 12th White House Papers Graduate Research in Cognitive and Computing Sciences at Sussex*, Cognitive Science Research Papers, pages 60–62. University of Sussex at Brighton, November 1999. Online available at `ftp://ftp.informatics.sussex.ac.uk/pub/reports/csrp/csrp512.ps.Z` [accessed 2008-06-23].

[140] Nicholas Peter Sharples from the University of Sussex Falmer, Brighton, BN1 9QH. *Evolutionary Approaches to Adaptive Protocol Design*. PhD thesis, School of Cognitive & Computing Sciences of the University of Sussex, August 2001. ASIN: B001ABO1DK.

[141] Lidia A. R. Yamamoto and Christian F. Tschudin from the Computer Science Department, University of Basel, Bernoullistrasse 16, CH-4056 Basel, Switzerland. Genetic Evolution of Protocol Implementations and Configurations. In *IFIP/IEEE International workshop on Self-Managed Systems and Services (SelfMan 2005)*, May 19, 2005, Nice, France. Online available at `http://cn.cs.unibas.ch/pub/doc/2005-selfman.pdf` [accessed 2007-09-17].

[142] Christian F. Tschudin from the Computer Science Department, University of Basel, Bernoullistrasse 16, CH-4056 Basel, Switzerland. Fraglets – a Metabolistic Execution Model for Communication Protocols. In *Proceedings of 2nd Annual Symposium on Autonomous Intelligent Networks and Systems (AINS)*, June 30–July 3, 2003, SRI International, Menlo Park, CA, USA. Online available at `http://cn.cs.unibas.ch/pub/doc/2003-ains.pdf` and `http://path.berkeley.edu/ains/final/007%20-%2022-tschudin.pdf` [accessed 2008-05-02]. See also `http://www.fraglets.net/` [accessed 2007-09-17].

[143] Emil Leon Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65(2):197–215, 1943, Baltimore, MD, USA. ISSN: 0002-9327, 1080-6377. doi:10.2307/2371809.

[144] Lidia A. R. Yamamoto, Daniel Schreckling, and Thomas Meyer. Self-Replicating and Self-Modifying Programs in Fraglets. In *Proceedings of BIONETICS 2007, 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, 2007. In proceedings [273]. Online available at `http://cn.cs.unibas.ch/people/ly/doc/bionetics2007-ysm.pdf` [accessed 2008-05-04].

[145] Christian F. Tschudin and Lidia A. R. Yamamoto. A Metabolic Approach to Protocol Resilience. In *Autonomic Communication – Revised Selected Papers from the First International IFIP Workshop, WAC 2004*, volume 3457/2005 of *Lecture Notes in Computer Science (LNCS)*, pages 191–206. Springer Berlin / Heidelberg, October 18–19, 2004, Berlin, Germany. ISBN: 978-3-54027-417-9. doi:10.1007/11520184_15. Online available at `http://cn.cs.unibas.ch/pub/doc/2004-wac-lncs.pdf` [accessed 2008-06-20], presentation online available at `http://www.autonomic-communication.org/wac/wac2004/program/presentations/wac2004-tschudin-yamamoto-slides.pdf` [accessed 2008-06-20].

[146] Lidia A. R. Yamamoto and Christian F. Tschudin from the Computer Science Department, University of Basel, Bernoullistrasse 16, CH-4056 Basel, Switzerland. Experiments on the Automatic Evolution of Protocols Using Genetic Programming. In *Autonomic Communication – Revised Selected Papers from the Second International IFIP Workshop, WAC 2005*, volume 3854/2006 of *Lecture Notes in Computer Science*

(LNCS), pages 13–28. Springer Berlin / Heidelberg, October 2–5 2005, Vouliagmeni, Athens, Greece. ISBN: 978-3-54032-992-3. doi:10.1007/11687818_2. Online available at http://cn.cs.unibas.ch/people/ly/doc/wac2005-lyct.pdf [accessed 2008-06-20]. .

[147] Lidia A. R. Yamamoto and Christian F. Tschudin from the Computer Science Department, University of Basel, Bernoullistrasse 16, CH-4056 Basel, Switzerland. Experiments on the Automatic Evolution of Protocols Using Genetic Programming. Technical Report CS-2005-002, University of Basel, April 21, 2005. Online available at http://cn.cs.unibas.ch/people/ly/doc/wac2005tr-lyct.pdf [accessed 2008-06-20]. .

[148] Daniele Miorandi, Paolo Dini, Eitan Altman, and Hisao Kameda. *WP 2.2 – Paradigm Applications and Mapping, D2.2.2 Framework for Distributed On-line Evolution of Protocols and Services*. BIOlogically inspired NETwork and Services (BIONETS) and Future and Emerging Technologies (FET) project of the EU, 2^{nd} edition, June 14, 2007. BIONETS/CN/wp2.2/v2.1. Editor: Lidia A. R. Yamamoto. Verification David Linner and Fran coise Baude. Status: Final, Public. Online available at http://www.bionets.eu/docs/BIONETS_D2_2_2.pdf [accessed 2008-06-23].

[149] Gregory M. Werner and Michael G. Dyer. Evolution of Communication in Artificial Organisms. In *Artificial Life II*, pages 659–687, 1992. Redwood City, CA. In proceedings [254]. Online available at http://www.isrl.uiuc.edu/~amag/langev/paper/werner92evolutionOf.html [accessed 2008-07-28].

[150] Hitoshi Iba. Emergent Cooperation for Multiple Agents Using Genetic Programming. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 32–41, 1996. doi:10.1007/3-540-61723-X_967. In proceedings [274].

[151] Hitoshi Iba, Tishihide Nozoe, and Kanji Ueda. Evolving Communicating Agents based on Genetic Programming. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pages 297–302, 1997. In proceedings [275]. Online available at http://lis.epfl.ch/~markus/References/Iba97.pdf [accessed 2008-07-28].

[152] Hitoshi Iba from the Machine Inference Section, Electrotechnical Laboratory (ETL), 1-1-4 Umezono, Tsukuba Science City, Ibaraki 305, Japan. Evolutionary learning of communicating agents. *Information Sciences – Informatics and Computer Science: An International Journal*, 108(1–4):181–205, July 1998. ISSN: 0020-0255. doi:10.1016/S0020-0255(97)10055-X. Online available at http://dx.doi.org/10.1016/S0020-0255(97)10055-X [accessed 2008-07-28].

[153] Kenneth J. Mackin and Eiichiro Tazaki from the Department of Control and System Engineering, Toin University of Yokohama, Japan. Emergent agent communication in multi-agent systems using automatically defined function genetic programming (ADF-GP). In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC'99*, volume 5, pages 138–142, October 12–15, 1999, Tokyo, Japan. ISBN: 0-7803-5731-0, 978-0-78035-731-0. doi:10.1109/ICSMC.1999.815536.

[154] Kenneth J. Mackin and Eiichiro Tazaki from the Department of Control and System Engineering, Toin University of Yokohama, Japan. Unsupervised training of multiobjective agent communication using genetic programming. In *KES*, pages 738–741, 2000. doi:10.1109/KES.2000.884152. In proceedings [276]. Online available at http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/Mackin00.html and http://www.lania.mx/~ccoello/EMOO/mackin00.pdf.gz [accessed 2008-07-28].

[155] Kenneth J. Mackin and Eiichiro Tazaki from the Department of Control and System Engineering, Toin University of Yokohama, Japan. Multiagent communication combining genetic programming and pheromone communication. *Kybernetes*, 31(6):

827–843, 2002. doi:10.1108/03684920210432808. Online available at http://www.emeraldinsight.com/10.1108/03684920210432808 [accessed 2008-07-28].

[156] Richard Heady, George Luger, Arthur Maccabe, and Mark Servilla. The architecture of a network level intrusion detection system. Technical Report CS90-20, LA-SUB–93-219, W-7405-ENG-36, DE97002400, AHC29703%%44, Department of Computer Science, University of New Mexico, Albuquerque, NM, USA, August 15, 1990. doi:10.2172/425295. Online available at http://www.osti.gov/energycitations/product.biblio.jsp?osti_id=425295 [accessed 2008-09-07].

[157] Wei Lu and Issa Traore from the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada. Detecting New Forms of Network Intrusions Using Genetic Programming. *Computational Intelligence*, 20(3):475–494, August 2004. Online available at http://www.isot.ece.uvic.ca/publications/journals/coi-2004.pdf [accessed 2008-06-11].

[158] Gianluigi Folino, Clara Pizzuti, and Giandomenico Spezzano from the ICAR-CNR, Via P.Bucci 41/C, Univ. della Calabria, 87036 Rende (CS), Italy. GP Ensemble for Distributed Intrusion Detection Systems. In *Pattern Recognition and Data Mining, Proceedings of the Third International Conference on Advances in Pattern Recognition, ICAPR 2005*, volume 3686/2005 of *Lecture Notes in Computer Science (LNCS)*, pages 54–62. Springer Berlin / Heidelberg, August 22–25, 2005, Bath, UK. ISBN: 978-3-54028-757-5. doi:10.1007/11551188_6. Online available at http://dns2.icar.cnr.it/pizzuti/icapr05.pdf [accessed 2008-06-23].

[159] Yoav Freund and Robert E. Schapire. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning, ICML'96*, pages 148–156, July 3–6, 1996, Bari, Italy. Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?cid=45368 and http://www.cs.ucsb.edu/~yfwang/courses/cs290i/papers/freund-adaboost96.pdf [accessed 2008-06-23].

[160] Srinivas Mukkamala, Andrew H. Sung, and Ajith Abraham. Modeling intrusion detection systems using linear genetic programming approach. In Robert Orchard, Chunsheng Yang, and Moonis Ali, editors, *Proceedings of the 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2004*, volume 3029/2004 of *Lecture Notes in Computer Science (LNCS)*, pages 633–642. Springer Springer Verlag Inc, May 17–20, 2004, Ottawa, Canada. ISBN: 3-5402-2007-0, 978-3-54022-007-7. doi:10.1007/b97304. Online available at http://www.rmltech.com/LGP%20Based%20IDS.pdf [accessed 2008-06-17].

[161] Dong Song, Malcom I. Heywood, and A. Nur Zincir-Heywood. A Linear Genetic Programming Approach to Intrusion Detection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2003. In proceedings [277]. Online available at http://users.cs.dal.ca/~mheywood/X-files/Publications/27242325.pdf [accessed 2008-06-16].

[162] Dong Song. A Linear Genetic Programming Approach to Intrusion Detection. Master's thesis, Dalhousie University, Halifax, Nova Scotia, Canada, March 2003. Advisors: Malcolm Heywood and Nur Zincir-Heywood. Online available at http://users.cs.dal.ca/~mheywood/Thesis/DSong.pdf [accessed 2008-06-16].

[163] Malcom I. Heywood and A. Nur Zincir-Heywood. Dynamic Page-Based Linear Genetic Programming. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, 32(3):380–388, 2002. ISSN: 1083-4419.

[164] Patrick LaRoche and A. Nur Zincir-Heywood from the Dalhousie University, Halifax, Nova Scotia, Canada. 802.11 Network Intrusion Detection

Using Genetic Programming. In *GECCO'05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, pages 170–171, 2005. doi:10.1145/1102256.1102295. In proceedings [278]. Online available at http://larocheonline.ca/~plaroche/papers/gecco-laroche-heywood.pdf and http://doi.acm.org/10.1145/1102256.1102295 [accessed 2008-06-16].

[165] James V. Hansen, Paul Benjamin Lowry, Rayman Meservy, and Dan McDonald. Genetic programming for prevention of cyberterrorism through previous dynamic and evolving intrusion detection. *Decision Support Systems*, 43(4):1362–1374, August 2007. doi:10.1016/j.dss.2006.04.004. Special Issue Clusters. Online available at http://dx.doi.org/10.1016/j.dss.2006.04.004 and http://ssrn.com/abstract=877981 [accessed 2008-06-17].

[166] Adrian Perrig and Dawn Xiaodong Song from the Computer Science Department of Carnegie Mellon University and University of California, Berkeley, USA. A First Step towards the Automatic Generation of Security Protocols. In *In Proccedings of the Symposium on Network and Distributed Systems Security (NDSS)*. The Internet Society (ISOC), February 2–4, 2000, Catamaran Resort Hotel, San Diego, California, USA. ISBN: 1-8915-6207-X, 1-8915-6208-8. Online available at http://www.cs.berkeley.edu/~dawnsong/papers/apg.pdf and http://www.isoc.org/isoc/conferences/ndss/2000/proceedings/031.pdf [accessed 2008-06-23].

[167] Dawn Xiaodong Song from the Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. Athena: an Automatic Checker for Security Protocol Analysis. In *Proceedings of 12th IEEE Computer Security Foundation Workshop*. IEEE press, June 1999. Online available at http://www.ece.cmu.edu/~dawnsong/papers/Athena.pdf and http://www.ece.cmu.edu/~dawnsong/papers/Athena.ps [accessed 2008-06-23].

[168] ISO. Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques. ISO/IEC 9798-3, International Standards Organization (ISO), 1993. JTC 1 Information technology. TC/SC: JTC 1/SC 27. Status: withdrawn.

[169] Dawn Xiaodong Song, Adrian Perrig, and Doantam Phan from the University of California, Berkeley, USA. AGVI – Automatic Generation, Verification, and Implementation of Security Protocols. In *Computer Aided Verification – CAV'01: Proceedings of the 13th International Conference on Computer Aided Verification*, volume 2102/2001 of *Lecture Notes in Computer Science (LNCS)*, pages 241–245. Springer Berlin / Heidelberg, July 18–22, 2001, Paris, France. ISBN: 3-5404-2345-1, 978-3-54042-345-4. doi:10.1007/3-540-44585-4_21.

[170] Bowei Xi, Zhen Liu, Mukund Raghavachari, Cathy H. Xia, and Li Zhang. A smart hill-climbing algorithm for application server configuration. In *WWW'04: Proceedings of the 13th international conference on World Wide Web*, pages 287–296, May 17–20, 2004, New York, NY, USA. ACM Press, New York, NY, USA. ISBN: 1-5811-3844-X. doi:10.1145/988672.988711. Session: Server performance and scalability. Online available at http://citeseer.ist.psu.edu/xi04smart.html [accessed 2007-09-11].

[171] Tadashi Nakano and Tatsuya Suda. Self-organizing network services with evolutionary adaptation. *IEEE Transactions on Neural Networks*, 16(5):1269–1278, September 2005. ISSN: 1045-9227. doi:10.1109/TNN.2005.853421. Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.74.9149 [accessed 2008-06-23].

[172] Tadashi Nakano and Tatsuya Suda from the Department of Information and Computer Science, University of California, Irvine, Irvine, CA 92697-3425,

USA. Adaptive and Evolvable Network Services. In *Genetic and Evolutionary Computation – GECCO 2004*, pages 151–162, 2004. In proceedings [279]. Online available at http://www.cs.york.ac.uk/rts/docs/GECCO_2004/Conferenceroceedings/papers/3102/31020151.pdf [accessed 2008-08-29].

[173] Tadashi Nakano and Tatsuya Suda. Applying biological principles to designs of network services. *Applied Soft Computing*, 7(3):870–878, June 2007. ISSN: 1568-4946. doi:10.1016/j.asoc.2006.04.006.

[174] Paul Grace. Genetic Programming and Protocol Configuration. Master's thesis, Computing Department, Lancaster University, Lancaster, LA1 4YW, Lancashire, UK, September 2000. Supervisor: Professor Gordon Blair. Online available at http://www.lancs.ac.uk/postgrad/gracep/msc.pdf [accessed 2008-06-23].

[175] *JavaGroups – A Reliable Multicast Communication Toolkit for Java*, Cornell University and SourceForge, 1999–2008. See http://www.cs.cornell.edu/Info/Projects/JavaGroupsNew/ [accessed 2008-06-23] and http://sourceforge.net/projects/javagroups [accessed 2008-06-23].

[176] Werner Van Belle, Tom Mens, and Theo D'Hondt from the Programming Technology Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium. Using Genetic Programming to Generate Protocol Adaptors for Interprocess Communication. In Andy M. Tyrrell, Pauline C. Haddow, and Jim Torresen, editors, *Proceedings of the 5th International Conference on Evolvable Systems: From Biology to Hardware. ICES 2003*, volume 2606/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 67–73. Springer Berlin / Heidelberg, March 17–20, 2003, Trondheim, Norway. ISBN: 978-3-54000-730-2. Online available at http://werner.yellowcouch.org/Papers/genadap03/ICES-online.pdf and http://prog.vub.ac.be/Publications/2003/vub-prog-tr-03-33.pdf [accessed 2008-06-23].

[177] Werner Van Belle from the Programming Technology Lab (PROG), Department Informatica (DINF), Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium. *Automatic Generation of Concurrency Adaptors by Means of Learning Algorithms.* PhD thesis, Vrije Universiteit Brussel, Faculteit van de Wetenschappen, Departement Informatica, August 2001. Promotor: Theo D'Hondt, co-Promotor: Tom Tourwé. Online available at http://borg.rave.org/adaptors/Thesis6.pdf [accessed 2008-06-23].

[178] Faris N. Abuali, Dale A. Schoenefeld, and Roger L. Wainwright from the Mathematical and Computer Science Department, The University of Tulsa, 600 South College Avenue, Tulsa, OK 74104-3189, USA. Designing telecommunications networks using genetic algorithms and probabilistic minimum spanning trees. In *SAC '94: Proceedings of the 1994 ACM Symposium on Applied computing*, pages 242–246, March 6–8, 1994, Phoenix, Arizona, United States. ACM, New York, NY, USA. ISBN: 0-8979-1647-6. doi:10.1145/326619.326733. Online available at http://doi.acm.org/10.1145/326619.326733 and http://euler.mcs.utulsa.edu/~rogerw/papers/Abuali-Prufer-CSC-94.pdf [accessed 2008-08-28].

[179] Sancho Salcedo-sanz and Xin Yao from the Center of Excellence for Research in Computational Intelligence and Applications (CERCIA), the University of Birmingham, U.K. A hybrid Hopfield network-genetic algorithm approach for the terminal assignment problem. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(6):2343–2353, December 2004. ISSN: 1083-4419. doi:10.1109/TSMCB.2004.836471. INSPEC Accession Number: 8207612. Online available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.7879 [accessed 2008-09-07].

[180] Samuel Pierre and Gisèle Legault from the LICEF Laboratory, Tél'e-université, Université du Québec, 1001, rue Sherbrooke Est. 4 étage, Case Postale 5250, Succ. C,

Montréal, Quebéc, Canada, H2L 4L5. An evolutionary approach for configuring economical packet switched computer networks. *AI in Engineering*, 10(2):127–134, 1996. doi:10.1016/0954-1810(95)00022-4. Online available at `http://dx.doi.org/10.1016/0954-1810(95)00022-4` [accessed 2008-08-01].

[181] Samuel Pierre and Gisèle Legault. A genetic algorithm for designing distributed computer network topologies. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(2):249–258, April 1998. doi:10.1109/3477.662766.

[182] King-Tim Ko, Kit-Sang Tang, Cheung-Yau Chan, Kim-Fung Man, and Sam Kwong. Using Genetic Algorithms to Design Mesh Networks. *Computer*, 30(8):56–61, August 1997. ISSN: 0018-9162. doi:10.1109/2.607086.

[183] Berna Dengiz, Fulya Altiparmak, and Alice E. Smith. Local Search Genetic Algorithm for Optimal Design of Reliable Networks. *IEEE Transactions on Evolutionary Computation*, 1(3):179–188, September 1997. Online available at `http://citeseer.ist.psu.edu/dengiz97local.html` and `http://www.eng.auburn.edu/~aesmith/publications/journal/ieeeec.pdf` [accessed 2008-08-01].

[184] Samuel Pierre and Ali Elgibaoui. Improving communication network topologies using tabu search. In *LCN: Proceedings of the 22nd IEEE Conference on Local Computer Networks (LCN '97)*, pages 44–53. IEEE Computer Society, November 2–5 1997, Minneapolis, Minnesota, USA. ISBN: 0-8186-8141-1.

[185] Samuel Pierre and Fabien Houéto from the Department of Computer Engineering, Mobile Computing, and Networking Research (LARIM), École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, Canada H3C 3A7. A tabu search approach for assigning cells to switches in cellular mobile networks. *Computer Communications*, 25(5):464–477, March 2002. doi:10.1016/S0140-3664(01)00371-1. Online available at `http://dx.doi.org/10.1016/S0140-3664(01)00371-1` [accessed 2008-08-01].

[186] Samuel Pierre and Fabien Houéto from the Department of Computer Engineering, Mobile Computing, and Networking Research (LARIM), École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, Canada H3C 3A7. Assigning cells to switches in cellular mobile networks using taboo search. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32(3):351–356, June 2002. doi:10.1109/TSMCB.2002.999810.

[187] Alejandro Quintero and Samuel Pierre from the Department of Computer Engineering, Mobile Computing, and Networking Research (LARIM), École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, Canada H3C 3A7. Sequential and multi-population memetic algorithms for assigning cells to switches in mobile networks. *Computer Networks*, 43(3):247–261, October 22, 2003. doi:10.1016/S1389-1286(03)00270-6. Online available at `http://dx.doi.org/10.1016/S1389-1286(03)00270-6` [accessed 2008-08-01].

[188] Alejandro Quintero and Samuel Pierre from the Department of Computer Engineering, Mobile Computing, and Networking Research (LARIM), École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, Canada H3C 3A7. Assigning cells to switches in cellular mobile networks: a comparative study. *Computer Communications*, 26(9):950–960, June 2003. doi:10.1016/S0140-3664(02)00224-4. Online available at `http://dx.doi.org/10.1016/S0140-3664(02)00224-4` [accessed 2008-08-01].

[189] Alejandro Quintero and Samuel Pierre from the Department of Computer Engineering, Mobile Computing, and Networking Research (LARIM), École Polytechnique de Montréal, C.P. 6079, Succ. Centre-Ville, Montréal, Québec, Canada H3C

3A7. Evolutionary approach to optimize the assignment of cells to switches in personal communication networks. *Computer Communications*, 26(9):927–938, June 2, 2003. doi:10.1016/S0140-3664(02)00238-4. Online available at http://dx.doi.org/10.1016/S0140-3664(02)00238-4 [accessed 2008-08-01].

[190] Marc St-Hilaire, Steven Chamberland, and Samuel Pierre. A Tabu Search Heuristic for the Global Planning of UMTS Networks. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2006)*, pages 148–151, June 19–21, 2006, Montréal, Québec, Canada. ISBN: 1-4244-0494-0. doi:10.1109/WIMOB.2006.1696339. INSPEC Accession Number: 9132969.

[191] Marc St-Hilaire, Steven Chamberland, and Samuel Pierre from the École Polytechnique de Montréal, Montréal Québec, Canada. A local search heuristic for the global planning of UMTS networks. In Seizo Onoe, Mohsen Guizani, Hsiao-Hwa Chen, and Mamoru Sawahashi, editors, *Proceedings of the International Conference on Wireless Communications and Mobile Computing, IWCMC 2006*, pages 1405–1410, July 3–6 2006, Vancouver, British Columbia, Canada. ACM, New York, NY, USA. ISBN: 1-5959-3306-9. doi:10.1145/1143549.1143831. SESSION: R2-D: general symposium. Online available at http://doi.acm.org/10.1145/1143549.1143831 [accessed 2008-08-01].

[192] Sancho Salcedo-Sanz, Jose A. Portilla-Figueras, Emilio G. Ortiz-García, Angel M. Pérez-Bellido, Christopher Thraves, Antonio Fernández-Anta, and Xin Yao. Optimal switch location in mobile communication networks using hybrid genetic algorithms. *Applied Soft Computing*, 8(4):1486–1497, September 2008. ISSN: 1568-4946. doi:10.1016/j.asoc.2007.10.021. Soft Computing for Dynamic Data Mining. Online available at http://nical.ustc.edu.cn/papers/asc2007b.pdf and http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.8731 [accessed 2008-09-01].

[193] Sancho Salcedo-Sanz and Xin Yao. Assignment of cells to switches in a cellular mobile network using a hybrid Hopfield network-genetic algorithm approach. *Applied Soft Computing*, 8(1):216–224, January 2008. ISSN: 1568-4946. doi:10.1016/j.asoc.2007.01.002.

[194] David Brittain, Jon Sims Williams, and Chris McMahon from the Department of Engineering Mathematics, University of Bristol, Bristol, UK. A Genetic Algorithm Approach To Planning The Telecommunications Access Network. In *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, pages 623–628, 1997. In proceedings [269]. Online available at http://citeseer.ist.psu.edu/brittain97genetic.html [accessed 2008-08-01].

[195] Enrique Alba, C. Cotta, J. Francisco Chicano, and A. J. Nebro from the Dept. Lenguajes y Ciencias de la Computacióon, ETSI Informática, University of Málaga, Campus de Teatinos, 29071 Málaga, Spain. Parallel Evolutionary Algorithms in Telecommunications: Two Case Studies. In *Proceedings of the Argentinean Conference of Computer Science (CACIC'02)*, October 2002, Buenos Aires, Argentina. Online available at http://www.lcc.uma.es/~eat/./pdf/cacic02.pdf and http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.4073 [accessed 2008-08-01].

[196] Max Martin, Eric Drucker, and Walter Don Potter from the Artificial Intelligence Center, University of Georgia, Athens, GA, USA. GA, EO, and PSO Applied to the Discrete Network Configuration Problem. In *International Conference on Genetic and Evolutionary Methods, GEM'08*, 2008. In proceedings [280]. Online available at http://www.cs.uga.edu/~potter/CompIntell/GEM_Martin-et-al.pdf [accessed 2008-08-23].

[197] Thomas Weise, Kurt Geihs, and Philipp Andreas Baer from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Genetic Programming for Proactive Aggregation Protocols. In *Proceedings of the 8th International Conference on Adaptive and Natural Computing Algorithms ICANNGA'07, Part 1*, pages 167–173, 2007. doi:10.1007/978-3-540-71618-1. In proceedings [281]. Online available at http://www.it-weise.de/documents/files/W2007DGPFb.pdf [accessed 2008-09-29].

[198] Thomas Weise, Michael Zapf, and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Evolving Proactive Aggregation Protocols. In *Genetic Programming – Proceedings of the 11th European Conference on Genetic Programming, EuroGP 2008*, pages 254–265, 2008. doi:10.1007/978-3-540-78671-9_22. In proceedings [282]. Online available at http://www.it-weise.de/documents/files/WZG2008DGPFa.pdf and http://dx.doi.org/10.1007/978-3-540-78671-9_22 [accessed 2008-09-29].

[199] Heather Christensen, Roger L. Wainwright, and Dale A. Schoenefeld from the Mathematical and Computer Science Department, The University of Tulsa, 600 South College Avenue, Tulsa, OK 74104-3189, USA. A Hybrid Algorithm for the Point to Multipoint Routing Problem. In *Proceedings of the 1997 ACM/SIGAPP Symposium on Applied Computing (SAC'97)*, pages 263–268, February 28–March 2, 1997, Hyatt St. Claire, 302 S. Market St., San Jose, CA 95113, USA. ACM Press, New York, NY, USA. ISBN: 0-8979-1850-9. doi:10.1145/331697.331751. Online available at http://euler.mcs.utulsa.edu/~rogerw/papers/Heather-PMP.pdf and http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.8559 [accessed 2008-08-28].

[200] Liming Zhu, Roger L. Wainwright, and Dale A. Schoenefeld from the Mathematical and Computer Sciences Department, The University of Tulsa, 600 South College Avenue, Tulsa, OK 74104-3189, USA. A genetic algorithm for the point to multipoint routing problem with varying number of requests. In *Proceedings of the IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on Evolutionary Computation*, pages 171–176, 1998. doi:10.1109/ICEC.1998.699496. In proceedings [283]. Online available at http://euler.mcs.utulsa.edu/~rogerw/papers/Zhu-ICEC98.pdf and http://citeseer.ist.psu.edu/382506.html [accessed 2008-07-21].

[201] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press USA, Oxford University Press, 198 Madison Avenue, New York, NY 10016, U.S.A., August 1999. ISBN: 0-1951-3159-2, 978-0-19513-159-8. Partly online available at http://books.google.de/books?id=mhLj70l8HfAC [accessed 2008-08-23].

[202] Xiang Fei, Junzhou Luo, Jieyi Wu, and Guanqun Gu from the Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China. QoS routing based on genetic algorithms. *Computer Commnunications*, 22(15-16):1392–1399, September 25, 1999. doi:10.1016/S0140-3664(99)00113-9. Online available at http://neo.lcc.uma.es/opticomm/main.html and http://dx.doi.org/10.1016/S0140-3664(99)00113-9 [accessed 2008-08-01].

[203] Kwang Mon Sim and Weng Hong Sun. Multiple Ant-Colony Optimization for Network Routing. In *First International Symposium on Cyber Worlds: Theory and Practices (CW'02)*, pages 277–281, November 6–8, 2002. IEEE Computer Society, Los Alamitos, CA, USA. ISBN: 0-7695-1862-1. doi:10.1109/CW.2002.1180890.

[204] Louis Anthony Cox, Jr., Lawrence Davis, and Yuping Qiu. Dynamic Anticipatory Routing In Circuit-Switched Telecommunications Networks. In *Handbook of Genetic Algorithms*, pages 124–143. Van Nostrand Reinhold, 1991. In collection [284].

[205] Eduard Lukschandl, Henrik Borgvall, Lars Nohle, Mats G. Nordahl, and Peter Nordin. Evolving Routing Algorithms with the JBGP-System. In *EvoIASP'99/EuroEcTel'99: Proceedings of the First European Workshops on Evolutionary Image Analysis, Signal Processing and Telecommunications*, pages 193–202, 1999. doi:10.1007/10704703_16. In proceedings [285].

[206] Eduard Lukschandl, Henrik Borgvall, Lars Nohle, Mats Nordahl, and Peter Nordin. Distributed Java Bytecode Genetic Programming with Telecom Applications. In *European Conference on Genetic Programming, EuroGP*, 2000. doi:10.1007/b75085. In proceedings [286].

[207] Eduard Lukschandl, Peter Nordin, and Mats Nordahl. Using the Java Method Evolver for Load Balancing in Communication Networks. In *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*, pages 236–239, 2000. In proceedings [287].

[208] Pablo Galiasso and Roger L. Wainwright from the Mathematical and Computer Sciences Department, The University of Tulsa, 600 South College Avenue, Tulsa, OK 74104-3189, USA. A hybrid genetic algorithm for the point to multipoint routing problem with single split paths. In *SAC'01: Proceedings of the 2001 ACM symposium on Applied computing*, pages 327–332, March 11–14, 2001, Las Vegas, Nevada, United States. ACM, New York, NY, USA. ISBN: 1-5811-3287-5. doi:10.1145/372202.372354. Online available at http://doi.acm.org/10.1145/372202.372354 and http://citeseer.ist.psu.edu/451433.html [accessed 2008-07-21].

[209] Harilaos G. Sandalidis, Constandinos X. Mavromoustakis, and Peter P. Stavroulakis from the Telecommunication Systems Institute of Crete (T.S.I.), Iroon Polytehniou 37 str., 73133 Chania, Crete, Greece. Performance measures of an ant based decentralised routing scheme for circuit switching communication networks. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 5(4):313–317, August 2001. ISSN: 1432-7643 (Print) 1433-7479 (Online). doi:10.1007/s005000100104. Online available at http://www.springerlink.com/content/g1u7lygrugwk3nl5/fulltext.pdf [accessed 2008-09-03].

[210] Zhao-Xia Wang, Zeng-Qiang Chen, and Zhu-Zhi Yuan from the Department of Automation, Nankai University, Tianjin 300071, P.R. China. QoS routing optimization strategy using genetic algorithm in optical fiber communication networks. *Journal of Computer Science and Technology*, 19(2):213–217, March 2004. ISSN: 1000-9000. CODEN: JCTEEM.

[211] Martin J. Oates and David W. Corne. Global web server load balancing using evolutionary computational techniques. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 5(4):297–312, August 2001. ISSN: 1432-7643 (Print) 1433-7479 (Online). doi:10.1007/s005000100103. Online available at http://www.springerlink.com/content/g1u7lygrugwk3nl5/fulltext.pdf [accessed 2008-09-03].

[212] Michael Zapf and Thomas Weise from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Offline Emergence Engineering For Agent Societies. Kasseler Informatikschriften (KIS) 2007, 8, University of Kassel, FB16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany, December 7, 2007. Persistent Identifier: urn:nbn:de:hebis:34-2007120719844. Online available at https://kobra.bibliothek.uni-kassel.de/handle/urn:nbn:de:hebis:34-2007120719844 and http://www.it-weise.de/documents/files/ZW2007EUMASTR.pdf [accessed 2007-11-20], .

[213] Michael Zapf and Thomas Weise from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Offline Emergence

Engineering For Agent Societies. In *Proceedings of the Fifth European Workshop on Multi-Agent Systems EUMAS'07*, December 14, 2007, Elmouradi Hotel, Hammamet, Tunesia. Online available at http://www.it-weise.de/documents/files/ZW2007EUMASTR.pdf [accessed 2008-09-29], .

[214] Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano. Construction of a Peer-to-Peer Information System in Grids. In Hans Czap, Rainer Unland, Cherif Branki, and Huaglory Tianfield, editors, *Self-Organization and Autonomic Informatics (I), Proceedings of the International Conference on Self-Organization and Adaptation of Multi-agent and Grid Systems (SOAS 2005)*, volume 135 of *Frontiers in Artificial Intelligence and Applications*, pages 220–236, December 11–13, 2005, Glasgow, UK. IOS Press, Nieuwe Hemweg 6B, 1013 BG Amsterdam, The Netherlands. ISBN: 1-5860-3577-0, 978-1-58603-577-8. Online available at http://grid.deis.unical.it/papers/pdf/SOAS05-ForestieroMastroianniSpezzano.pdf [accessed 2008-09-05].

[215] Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano from the Institute for High Performance Computing and Networks (ICAR-CNR), 87036 Rende (CS) Italy. Antares: an Ant-Inspired P2P Information System for a Self-Structured Grid. In *Proceedings of BIONETICS 2007*, 2007. doi:10.1109/BIMNICS.2007.4610103. In proceedings [273]. Online available at http://grid.deis.unical.it/papers/pdf/ForestieroMastroianniSpezzano-Antares.pdf [accessed 2008-06-13].

[216] Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano from the Institute for High Performance Computing and Networks (ICAR-CNR), 87036 Rende (CS) Italy. QoS-based Dissemination of Content in Grids. *Future Generation Computer Systems*, 24(3):235–244, March 2008. ISSN: 0167-739X. doi:10.1016/j.future.2007.05.003. Online available at http://grid.deis.unical.it/papers/pdf/ForestieroMastroianniSpezzano-FGCS-QoS.pdf [accessed 2008-09-05].

[217] Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano from the Institute for High Performance Computing and Networks (ICAR-CNR), 87036 Rende (CS) Italy. So-Grid: A Self-organizing Grid Featuring Bio-inspired Algorithms. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(2):5:1–5:37, May 2008. ISSN: 1556-4665. doi:10.1145/1352789.1352790. Article 5. Online available at http://grid.deis.unical.it/papers/pdf/Article.pdf and http://doi.acm.org/10.1145/1352789.1352790 [accessed 2008-09-05].

[218] Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano from the Institute for High Performance Computing and Networks (ICAR-CNR), 87036 Rende (CS) Italy. Building a Peer-to-peer Information System in Grids via Self-organizing Agents. *Journal of Grid Computing*, 6(2):125–140, June 2008. ISSN: 1570-7873 (Print) 1572-9814 (Online). doi:10.1007/s10723-007-9062-z. Online available at http://grid.deis.unical.it/papers/pdf/JGC2008-ForestieroMastroianniSpezzano.pdf [accessed 2008-09-05].

[219] Agostino Forestiero, Carlo Mastroianni, and Giandomenico Spezzano from the Institute for High Performance Computing and Networks (ICAR-CNR), 87036 Rende (CS) Italy. Reorganization and Discovery of Grid Information with Epidemic Tuning. *FGCS – The International Journal of Grid Computing: Theory, Methods and Applications*, 24 (8):788–797, October 2008. ISSN: 0167-739X. doi:10.1016/j.future.2008.04.001. Online available at http://grid.deis.unical.it/papers/pdf/FGCS-epidemic.pdf [accessed 2008-09-05].

[220] Yongguo Liu, Kefei Chen, Xiaofeng Liao, and Wei Zhang. A genetic clustering method for intrusion detection. *Pattern Recognition*, 37(5):927–942, May 2004. ISSN: 0031-3203. doi:10.1016/j.patcog.2003.09.011.

[221] David Andre. The Automatic Programming of Agents that Learn Mental Models and Create Simple Plans of Action. In *IJCAI-95 Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 741–747, 1995. MAPMAKER searches for gold. Online available at http://citeseer.ist.psu.edu/188996.html and http://dli.iiit.ac.in/ijcai/IJCAI-95-VOL%201/pdf/097.pdf [accessed 2007-10-03]. In proceedings [288].

[222] Mohammad Adil Qureshi. Evolving Agents. Research Note RN/96/4, UCL, Gower Street, London, WC1E 6BT, UK, January 1996. Online available at http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/AQ.gp96.ps.gz and http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/qureshi_1996_eaRN.html [accessed 2008-09-02]. .

[223] Mohammad Adil Qureshi from the Genetic Programming Group, Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK. Evolving Agents. In *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 369–374, 1996. In proceedings [289] and also [222]. Online available at http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/AQ.gp96.ps.gz [accessed 2007-09-17].

[224] Mohammad Adil Qureshi from the Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK. *The Evolution of Agents*. PhD thesis, University College, London, London, UK, July 2001. Online available at http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/qureshi_thesis.html and http://citeseer.ist.psu.edu/759376.html [accessed 2007-09-14].

[225] Hitoshi Iba. Evolving Multiple Agents by Genetic Programming. In *Advances in genetic programming 3*, pages 447–466. MIT Press, Cambridge, MA, USA, 1999. In collection [290]. Online available at http://www.cs.bham.ac.uk/~wbl/aigp3/ch19.pdf [accessed 2007-10-03].

[226] Hosung Choo, Adrian Hutani, Luiz Cezar Trintinalia, and Hao Ling from the Dept. of Electr. & Comput. Eng., Texas Univ., Austin, TX, USA. Shape optimisation of broadband microstrip antennas using genetic algorithm. *Electronics Letters*, 36(25):2057–2058, December 7, 2000. ISSN: 0013-5194. CODEN: ELLEAK. doi:10.1049/el:20001452. INSPEC Accession Number: 6798475.

[227] Jason D. Lohn, Gregory S. Hornby, and Derek Linden. An evolved antenna for deployment on NASA's space technology 5 mission. In *Genetic Programming Theory and Practice II*, 2004. In proceedings [291]. Online available at http://ic.arc.nasa.gov/people/hornby/papers/lohn_gptp04.ps.gz [accessed 2007-08-17].

[228] Frank J. Villegas, Tom Cwik, Yahya Rahmat-Samii, and Majid Manteghi from the Raytheon Electron. Syst., El Segundo, CA, USA. A parallel electromagnetic genetic-algorithm optimization (EGO) application for patch antenna design. *IEEE Transactions on Antennas and Propagation*, 52(9):2424–2435, September 3, 2004. ISSN: 0018-926X. doi:10.1109/TAP.2004.834071. INSPEC Accession Number: 8091160.

[229] John R. Koza, Martin A. Keane, Matthew J. Streeter, William Mydlowec, Jessen Yu, and Guido Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Genetic Programming. Springer, first edition, May 2005. ISBN: 978-0-38725-067-0. Partly online available at http://books.google.de/books?id=YQxWzAEnINIC [accessed 2008-09-03].

[230] Matthias John and Max J. Ammann from the Centre for Telecommunications Value-chain Driven Research, Dublin Institute of Technology, School of Electronic and Communications Engineering, Kevin Street, Dublin 8, Ireland. Design Of A Wide-Band

Printed Antenna Using A Genetic Algorithm On An Array Of Overlapping Sub-Patches. In *IEEE International Workshop on Antenna Technology: Small Antennas and Novel Metamaterials (IWAT2006)*, pages 92–95, March 6–8, 2006, White Plains, NY, USA. ISBN: 0-7803-9443-7. Online available at http://www.ctvr.ie/docs/RF%20Pubs/01608983.pdf [accessed 2008-09-03].

[231] Matthias John and Max J. Ammann from the Centre for Telecommunications Value-chain Driven Research, Dublin Institute of Technology, School of Electronic and Communications Engineering, Kevin Street, Dublin 8, Ireland. Optimisation of a Wide-Band Printed Monopole Antenna using a Genetic Algorithm. In *Loughborough Antennas & Propagation Conference (PAPC)*, pages 237–240. Loughborough University, April 11–12, 2006. Online available at http://www.ctvr.ie/docs/RF%20Pubs/LAPC_2006_MJ.pdf [accessed 2008-09-02].

[232] Neela Chattoraj and Jibendu Sekhar Roy. Application of Genetic Algorithm to the Optimization of Microstrip Antennas with and without Superstrate. *Mikrotalasna Revija (Microwave Review)*, 2(6), November 2006. ISSN: 1450-5835. Online available at http://www.mwr.medianis.net/pdf/Vol12No2-06-NChattoraj.pdf [accessed 2008-09-02].

[233] Thomas Weise, Steffen Bleul, and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Web Service Composition Systems for the Web Service Challenge – A Detailed Review. Kasseler Informatikschriften (KIS) 2007, 7, University of Kassel, FB16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany, University of Kassel, November 19, 2007. Persistent Identifier: urn:nbn:de:hebis:34-2007111919638. Online available at http://kobra.bibliothek.uni-kassel.de/handle/urn:nbn:de:hebis:34-2007111919638 and http://www.it-weise.de/documents/files/WBG2007WSCb.pdf [accessed 2007-11-20]. .

[234] Thomas Weise, Steffen Bleul, Diana Comes, and Kurt Geihs from the Distributed Systems Group, FB 16 – Elektrotechnik und Informatik, University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Different Approaches to Semantic Web Service Composition. In Abdelhamid Mellouk, Jun Bi, Guadalupe Ortiz, Dickson K. W. Chiu, and Manuela Popescu, editors, *Proceedings of The Third International Conference on Internet and Web Applications and Services, ICIW 2008*, pages 90–96, June 8–13, 2008, Athens, Greece. IEEE Computer Society Press, Los Alamitos, CA, USA. ISBN: 978-0-76953-163-2. Library of Congress Control Number: 2008922600. Product Number: E3163. BMS Part Number: CFP0816C-CDR. Online available at http://www.it-weise.de/documents/files/WBCG2008ICIW.pdf [accessed 2008-09-29].

[235] Thomas Weise, Michael Zapf, and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Rule-based Genetic Programming. In *Proceedings of BIONETICS 2007, 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, 2007. In proceedings [273]. Online available at http://www.it-weise.de/documents/files/WZG2007RBGP.pdf [accessed 2008-09-29].

[236] Thomas Weise and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Genetic Programming Techniques for Sensor Networks. In *Proceedings of 5. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze"*, pages 21–25, July 17–18, 2006, University of Stuttgart, Stuttgart, Germany. Online available at http://www.it-weise.de/documents/files/W2006DGPFb.pdf and http://elib.uni-stuttgart.de/opus/volltexte/2006/2838/ [accessed 2007-11-07].

[237] Thomas Weise from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Genetic Programming for Sensor Networks. Technical Report, University of Kassel, University of Kassel, January 2006. Online available at http://www.it-weise.de/documents/files/W2006DGPFa.pdf [accessed 2008-09-29].

[238] Thomas Weise and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. DGPF – An Adaptable Framework for Distributed Multi-Objective Search Algorithms Applied to the Genetic Programming of Sensor Networks. In *Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Application, BIOMA 2006*, pages 157–166, 2006. In proceedings [292]. Online available at http://www.it-weise.de/documents/files/W2006DGPFc.pdf [accessed 2008-09-29].

[239] Thomas Weise, Michael Zapf, Mohammad Ullah Khan, and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Genetic Programming meets Model-Driven Development. Kasseler Informatikschriften (KIS) 2007, 2, University of Kassel, July 2, 2007. Persistent Identifier: urn:nbn:de:hebis:34-2007070218786. Online available at http://www.it-weise.de/documents/files/WZKG2007DGPFd.pdf and http://kobra.bibliothek.uni-kassel.de/handle/urn:nbn:de:hebis:34-2007070218786 [accessed 2007-09-17].

[240] Thomas Weise, Michael Zapf, Mohammad Ullah Khan, and Kurt Geihs from the University of Kassel, FB-16, Distributed Systems Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany. Genetic Programming meets Model-Driven Development. In *Proceedings of Seventh International Conference on Hybrid Intelligent Systems*, 2007. In proceedings [293]. Online available at http://www.it-weise.de/documents/files/WZKG2007DGPFg.pdf [accessed 2008-09-29].

[241] Theodore S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall PTR, 2001, (2nd ed: January 2002). ISBN: 978-0-13042-232-3, 0-1304-2232-0. Partly online available at http://books.google.com/books?id=8ZXdHQAACAAJ [accessed 2008-07-27].

[242] Andrea Goldsmith from the Stanford University, California. *Wireless Communications.* Cambridge University Press, The Edinburgh Building, Shaftesbury Road, Cambridge, CB2 8RU, UK, August 2005. Partly online available at http://books.google.com/books?id=mF7X_8D7_BIC [accessed 2008-07-27].

[243] Andreas F. Molisch. *Wireless Communications.* John Wiley & Sons / IEEE Press, November 18, 2005. ISBN: 978-0-47084-887-6, 0-4708-4887-1. Partly online available at http://books.google.com/books?id=2NNCHgAACAAJ [accessed 2008-07-27].

[244] Mohammad Ilyas and Imad Mahgoub, editors. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems.* CRC, July 16, 2004. ISBN: 978-0-84931-968-6.

[245] Edgar H. Callaway, Jr. *Wireless Sensor Networks: Architectures and Protocols.* AUERBACH, August 26, 2003. ISBN: 978-0-84931-823-8.

[246] Richard Harper, editor. *Inside the smart home.* Springer, 2003. ISBN: 1-8523-3688-9, 978-1-85233-688-2. Partly online available at http://books.google.com/books?id=SvcHvHuv86gC [accessed 2008-07-27].

[247] Werner Harke. *Smart Home – Vernetzung von Haustechnik und Kommunikationssystemen im Wohnungsbau.* Hüthig Verlag / C. F. Müller, December 31, 2003. ISBN: 978-3-78807-713-6, 3-7880-7713-1.

[248] Adam Greenfield. *Everyware: The Dawning Age of Ubiquitous Computing*. New Riders Publishing, March 20, 2006. ISBN: 0-3213-8401-6, 978-0-32138-401-0.

[249] John Krumm, Gregory D. Abowd, Aruna Seneviratne, and Thomas Strang, editors. *UbiComp 2007: Ubiquitous Computing. Proceedings of the 9th International Conference*, volume 4717 of *Lecture Notes in Computer Science*, September 16–19, 2007, Innsbruck, Austria. Springer Berlin / Heidelberg. ISBN: 978-3-54074-852-6. doi:10.1007/978-3-540-74853-3.

[250] Ang Yang, Yin Shan, and Lam Thu Bui, editors. *Success in Evolutionary Computation*, volume 92/2008 of *Studies in Computational Intelligence*. Springer Berlin / Heidelberg, January 16, 2008. ISBN: 3-5407-6285-X, 978-3-54076-285-0, 978-3-54076-286-7. Library of Congress Control Number: 2007939404. doi:10.1007/978-3-540-76286-7.

[251] L. Darell Whitley from the Computer Science Department, Colorado State University, Fort Collins. A Genetic Algorithm Tutorial. Technical Report CS-93-103, Computer Science Department, Colorado State University, Fort Collins, March 10, 1993. Online available at `http://citeseer.ist.psu.edu/177719.html` [accessed 2007-11-29]. .

[252] N. S. Sridharan, editor. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, volume 1, August 1989, Detroit, MI, USA. Morgan Kaufmann, San Francisco, CA, USA. ISBN: 1-5586-0094-9. Online available at `http://dli.iiit.ac.in/ijcai/IJCAI-89-VOL1/CONTENT/content.htm` [accessed 2008-04-01]. .

[253] Bruce M. Spatz and Gregory J. E. Rawlins, editors. *Proceedings of the First Workshop on Foundations of Genetic Algorithms*, July 15–18, 1990, Indiana University, Bloomington Campus, Indiana, USA. Morgan Kaufmann Publishers, Inc., 2929 Campus Drive, Suite 260, San Mateo, CA 94403, USA. ISBN: 1-5586-0170-8. Published July 1, 1991. Partly online available at `http://books.google.de/books?id=Df12yLrlUZYC` [accessed 2008-05-29].

[254] Christopher G. Langton, C.E. Taylor, D.J. Farmer, and S. Rasmussen, editors. *Artificial Life II: Proceedings of the Workshop on Artificial Life*, volume X of *Santa Fe Institute Studies in the Science of Complexity*, February 1990, Santa Fe, New Mexico, USA. Addison-Wesley. ISBN: 0-2015-2571-2, 978-0-20152-571-7. Published 1992, republished by Westview Press (March 28, 2003).

[255] Jean-Arcady Meyer and Stewart W. Wilson, editors. *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, September 24–28, 1990, Paris, France. The MIT Press, Cambridge, MA, USA. ISBN: 0-2626-3138-5. Published in 1991.

[256] Marco Dorigo, Gianni Di Caro, and Luca Maria Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137–172, 1999. Online available at `http://www.idsia.ch/~luca/ij_23-alife99.pdf` [accessed 2007-09-15]. See [59].

[257] Godfrey C. Onwubolu and B. V. Babu, editors. *New Optimization Techniques in Engineering*, volume 141 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin/Heidelberg, Germany, March 5, 2004. ISBN: 3-5402-0167-X, 978-3-54020-167-0. Partly online available at `http://books.google.de/books?id=YRsq-Pz0kAAC` [accessed 2008-06-07].

[258] *Proceedings, The Twenty-First National Conference on Artificial Intelligence (AAAI) and the Eighteenth Innovative Applications of Artificial Intelligence Conference (IAAI)*, July 16–20, 2006, Boston, Massachusetts, USA. AAAI Press. See `http://www.aaai.org/Conferences/AAAI/aaai06.php` [accessed 2007-09-06] and `http://www.aaai.org/Conferences/IAAI/iaai06.php` [accessed 2007-09-06].

[259] Chii-Ruey Hwang. Simulated annealing: Theory and applications. *Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications*, 12(1):108–111, May 1988. ISSN: 0167-8019 (Print) 1572-9036 (Online). doi:10.1007/BF00047572. Academia Sinica. This is a review of [76]. Online available at `http://www.springerlink.com/content/t1n71m85j33556m4/` [accessed 2007-08-25].

[260] John J. Grefenstette, editor. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms and their Applications*, July 28–31, 1987, Massachusetts Institute ofTechnology, Cambridge, Massachusetts, USA. American Association for Artificial Intelligence, Lawrence Erlbaum Associates, Inc., Hillsdale/Mahwah, NJ, USA. ISBN: 0-8058-0158-8.

[261] Wolfgang Banzhaf, Jason M. Daida, Ágoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark J. Jakiela, and Robert E. Smith, editors. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, July 13–17, 1999, Orlando, Florida, USA. Morgan Kaufmann. ISBN: 1-5586-0611-4. .

[262] William E. Hart, Natalio Krasnogor, and J. E. Smith, editors. *Recent Advances in Memetic Algorithms*, volume 166/2005 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin, Germany, 2005. ISBN: 3-5402-2904-3, 978-3-54022-904-9. doi:10.1007/3-540-32363-5. Partly online available at `http://books.google.de/books?id=LYf7YW4DmkUC` [accessed 2008-04-01].

[263] Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors. *Parallel Problem Solving from Nature – PPSN III, International Conference on Evolutionary Computation. Proceedings of the Third Conference on Parallel Problem Solving from Nature*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, October 9–14, 1994, Jerusalem, Israel. Springer. ISBN: 3-5405-8484-6, 978-3-54058-484-1. doi:10.1007/3-540-58484-6. See `http://ls11-www.informatik.uni-dortmund.de/PPSN/ppsn3/ppsn3.html` [accessed 2007-09-05].

[264] Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Guenther R. Raidl, Terence Soule, Andy M. Tyrrell, Jean-Paul Watson, and Eckart Zitzler, editors. *Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2005*, June 25–29, 2005, Loews L'Enfant Plaza Hotel, 480 L'enfant Plaza Sw, Washington, D.C. 20024, USA. ACM Press, Washington DC, USA. ISBN: 1-5959-3010-8. Order Number: 910050. GECCO-2005 A joint meeting of the fourteenth international conference on genetic algorithms (ICGA-2005) and the tenth annual genetic programming conference (GP-2005). ACM Order Number 910052. .

[265] *Proceedings of the 2nd IEE/IEEE International GALESIA Conference*, September 2–4, 1997, University of Strathclyde, Glasgow, UK. IEE. ISBN: 0-8529-6693-8.

[266] David W., Nigel C. Steele, and F. Albrecht, editors. *Proceedings of the 2nd International Conference on Artificial Neural Nets and Genetic Algorithms, ICANNGA*, April 18–21, 1995, École des Mines d'Alès, Alès, France. Springer Verlag. ISBN: 3-2118-2692-0.

[267] John R. Koza, editor. *Late Breaking Papers at the First Annual Conference Genetic Programming (GP-96)*, July 28–31, 1996, Stanford University, CA, USA. Stanford Bookstore. ISBN: 0-1820-1031-7.

[268] George D. Smith, Nigel C. Steele, and Rudolf F. Albrecht, editors. *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, ICANNGA*, April 1–4, 1997, University of East Anglia, Norwich, England, U.K. Springer, Vienna, Austria. ISBN: 978-3-21183-087-1, 3-2118-3087-1. Published in 1998.

[269] Thomas Bäck, editor. *Proceedings of The Seventh International Conference on Genetic Algorithms ICGA'97*, July 19–23, 1997, Michigan State University, East Lansing, Michigan, USA. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN: 1-5586-0487-1. see `http://garage.cse.msu.edu/icga97/` [accessed 2007-09-01].

[270] David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors. *Proceedings of the IEEE Congress on Evolutionary Computation, CEC2002*, May 12–17, 2002, Honolulu, HI, USA. IEEE Press, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA. ISBN: 0-7803-7278-6. CEC 2002 – A joint meeting of the IEEE, the Evolutionary Programming Society, and the IEE. Held in connection with the World Congress on Computational Intelligence (WCCI 2002).

[271] Maarten Keijzer, Giuliano Antoniol, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Nikolaus Hansen, John H. Holmes, Gregory S. Hornby, Daniel Howard, James Kennedy, Sanjeev Kumar, Fernando G. Lobo, Julian Francis Miller, Jason Moore, Frank Neumann, Martin Pelikan, Jordan Pollack, Kumara Sastry, Kenneth Stanley, Adrian Stoica, El-Ghazali Talbi, and Ingo Wegener, editors. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*, July 12–16, 2008, Renaissance Atlanta Hotel Downtown, 590 West Peachtree Street NW, Atlanta, Georgia 30308 USA. ACM Press, New York, NY, 10286-1405, USA. ISBN: 978-1-60558-131-6. ACM Order Number: 910081. See `http://www.sigevo.org/gecco-2008/` [accessed 2008-07-16]. .

[272] Stefano Cagnoni, Riccardo Poli, Yun Li, George D. Smith, David Corne, Martin J. Oates, Emma Hart, Pier Luca Lanzi, Egbert J. W. Boers, Ben Paechter, and Terence C. Fogarty, editors. *Real-World Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2000: EvoIASP, EvoSCONDI, EvoTel, EvoSTIM, EvoROB, and EvoFlight*, volume 1803/200 of *Lecture Notes in Computer Science (LNCS)*, April 17, 2000, Edinburgh, Scotland, UK. Springer. ISBN: 3-5406-7353-9, 978-3-54067-353-8. doi:10.1007/3-540-45561-2.

[273] *Proceedings of BIONETICS 2007, 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, December 10–13, 2007, Radisson SAS Beke Hotel, 43. Terez krt., Budapest H-1067, Hungary. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST), IEEE, ACM. ISBN: 978-9-63979-905-9.

[274] Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberger, and Hans-Paul Schwefel, editors. *Parallel Problem Solving from Nature – PPSN IV, International Conference on Evolutionary Computation. Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, volume 1141/1996 of *Lecture Notes in Computer Science (LNCS)*, September 22–24, 1996, Berlin, Germany. Springer. ISBN: 3-5406-1723-X. doi:10.1007/3-540-61723-X. See `http://ls11-www.informatik.uni-dortmund.de/PPSN/ppsn4/ppsn4.html` [accessed 2007-09-05].

[275] Thomas Bäck, Zbigniew Michalewicz, and X. Yao, editors. *IEEE International Conference on Evolutionary Computation, CEC97*, April 13–16, 1997, Indianapolis, IN, USA. IEEE Press, Piscataway, NJ. ISBN: 0-7803-3949-5. See `http://ieeexplore.ieee.org/servlet/opac?punumber=4619` [accessed 2007-09-06].

[276] Robert J. Howlett and Lakhmi C. Jain, editors. *Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, KES 2000, Volume 2*, August 30–September 1, 2000, Brighton, UK. IEEE. ISBN: 0-7803-6400-7. .

[277] Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell K. Standish, Graham Kendall, Stewart W. Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitchell A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasa Jonoska, and Julian Francis Miller, editors. *Proceedings of Genetic and Evolutionary Computation - GECCO 2003, Genetic and Evolutionary Computation Conference, Part II*, volume 2724/2003 of *Lecture Notes in Computer Science (LNCS)*, July 12–16, 2003, The Holiday Inn Chicago – Mart Plaza, 350 N. Orleans St., Chicago, IL 60654, USA. Springer Berlin/Heidelberg. ISBN: 978-3-54040-603-7. doi:10.1007/3-540-45110-2. .

[278] Franz Rothlauf, Misty Blowers, Jürgen Branke, Stefano Cagnoni, Ivan I. Garibay, Ozlem Garibay, Jörn Grahl, Gregory Hornby, Edwin D. de Jong, Tim Kovacs, Sanjeev Kumar, Claudio F. Lima, Xavier Llorà, Fernando Lobo, Laurence D. Merkle, Julian Francis Miller, Jason H. Moore, Michael ONeill, Martin Pelikan, Terry P. Riopka, Marylyn D. Ritchie, Kumara Sastry, Stephen L. Smith, Hal Stringer, Keiki Takadama, Marc Toussaint, Stephen C. Upton, Alden H. Wright, and Shengxiang Yang, editors. *Proceedings of the 2005 workshops on Genetic and evolutionary computation*, June 25–26, 2005, Loews L'Enfant Plaza Hotel, 480 L'enfant Plaza Sw, Washington, D.C. 20024, USA. ACM, New York, NY, USA. .

[279] Kalyanmoy Deb, Riccardo Poli, Wolfgang Banzhaf, Hans-Georg Beyer, Edmund K. Burke, Paul J. Darwen, Dipankar Dasgupta, Dario Floreano, James A. Foster, Mark Harman, Owen Holland, Pier Luca Lanzi, Lee Spector, Andrea Tettamanzi, Dirk Thierens, and Andrew M. Tyrrell, editors. *Proceedings of Genetic and Evolutionary Computation - GECCO 2004, Genetic and Evolutionary Computation Conference, Part I*, volume 3102/2004 of *Lecture Notes in Computer Science (LNCS)*, June 26–30, 2004, Red Lion Hotel, 1415 5th Avenue, Seattle, WA 98101, USA. Springer Berlin/Heidelberg. ISBN: 978-3-54022-344-3. .

[280] Hamid R. Arabnia, Jack Y. Yang, and Mary Qu Yang, editors. *Proceedings of the 2008 International Conference on Genetic and Evolutionary Methods, GEM 2008*, June 14–17 2008, Monte Carlo Resort, Las Vegas Nevada, USA.

[281] Bartłomiej Beliczyński, Andrzej Dzieliński, Marcin Iwanowski, and Bernardete Ribeiro, editors. *Proceedings of Adaptive and Natural Computing Algorithms, 8th International Conference, ICANNGA 2007, Part I*, volume 4431/2007 of *Lecture Notes in Computer Science (LNCS)*, April 11–14, 2007, Warsaw University of Technology, Warsaw, Poland. Springer Berlin Heidelberg New York. ISBN: 978-3-54071-589-4. doi:10.1007/978-3-540-71618-1. , `http://icannga07.ee.pw.edu.pl/` [accessed 2007-08-31], and `http://www.springerlink.com/content/978-3-540-71590-0/` [accessed 2007-08-31].

[282] Michael O'Neill, Leonardo Vanneschi, Steven Matt Gustafson, Anna Isabel Esparcia Alcázar, Ivanoe De Falco, Antonio Della Cioppa, and Ernesto Tarantino, editors. *Genetic Programming – Proceedings of the 11th European Conference on Genetic Programming, EuroGP 2008*, volume 4971/2008 of *Lecture Notes in Computer Science (LNCS), Theoretical Computer Science and General Issues*, March 26–28 2008, Naples, Italy. Springer-Verlag GmbH, Berlin/Heidelberg, Germany. ISBN: 3-5407-8670-8, 978-3-54078-670-2. doi:10.1007/978-3-540-78671-9.

[283] *The 1998 IEEE International Conference on Evolutionary Computation, Proceedings of IEEE World Congress on Computational Intelligence, CEC98*, May 4–9, 1998, An-

chorage, Alaska, USA. IEEE Press. Part of WCCI, see http://ieeexplore.ieee.org/servlet/opac?punumber=5621 [accessed 2007-09-06].

[284] Lawrence Davis, editor. *Handbook of Genetic Algorithms*. Thomson Publishing Group / Van Nostrand Reinhold Company, New York, USA, January 1991. ISBN: 0-4420-0173-8, 978-0-44200-173-5.

[285] Riccardo Poli, Hans-Michael Voigt, Stefano Cagnoni, David Corne, George D. Smith, and Terence C. Fogarty, editors. *Proceedings of the First European Workshops on Evolutionary Image Analysis, Signal Processing and Telecommunications, EvoIASP'99 and EuroEcTel'99*, volume 1596/1999 of *Lecture Notes in Computer Science (LNCS)*, May 26–27, 1999, Göteborg, Sweden. Springer. ISBN: 3-5406-5837-8, 978-3-54065-837-5. doi:10.1007/10704703. Partly online available at http://books.google.de/books?id=CYDLxvi0-SYC [accessed 2008-09-16].

[286] Riccardo Poli, Wolfgang Banzhaf, William B. Langdon, Julian Francis Miller, Peter Nordin, and Terence C. Fogarty, editors. *Proceedings of the European Conference on Genetic Programming*, volume 1802/2000 of *Lecture Notes in Computer Science (LNCS)*, April 15-16, 2000, Edinburgh, Scotland, UK. Springer Berlin/Heidelberg. ISBN: 3-5406-7339-3. doi:10.1007/b75085.

[287] L. Darrell Whitley, editor. *Late Breaking Papers at Genetic and Evolutionary Computation Conference (GECCO'00)*, July 8–12, 2000, The Riviera Hotel and Casino, Las Vegas, Nevada, USA. .

[288] *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 1, August 20–25, 1995, Montréal, Québec, Canada. Morgan Kaufmann, San Francisco, CA, USA. ISBN: 1-5586-0363-8. Online available at http://dli.iiit.ac.in/ijcai/IJCAI-95-VOLt.htm [accessed 2008-04-01]. .

[289] John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors. *Proceedings of the First Annual Conference Genetic Programming (GP-96)*, July 28–31, 1996, Stanford University, CA, USA. MIT Press.

[290] Lee Spector, William B. Langdon, Una-May O'Reilly, and Peter John Angeline, editors. *Advances in Genetic Programming*, volume 3 of *Complex Adaptive Systems*. MIT Press, Cambridge, MA, USA, July 16, 1999. ISBN: 0-2621-9423-6, 978-0-26219-423-5. Partly online available at http://books.google.de/books?id=5Qwbal3AY6oC [accessed 2008-09-16].

[291] Una-May O'Reilly, Gwoing Tina Yu, Rick Riolo, and Bill Worzel, editors. *Genetic Programming Theory and Practice II, Proceedings of the Genetic Programming Theory Practice 2004 Workshop (GPTP-2004)*, volume 8 of *Genetic Programming Series*, May 13-15, 2004, The Center for the Study of Complex Systems (CSCS), University of Michigan, Ann Arbor, Michigan, USA. Springer, New York, USA. ISBN: 978-0-38723-253-9, 978-0-38723-254-6. doi:10.1007/b101112. See http://www.cscs.umich.edu/gptp-workshops/gptp2004/ [accessed 2007-09-28].

[292] Bogdan Filipič and Jurij Šilc, editors. *Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Application, BIOMA 2006*, October 9–10, 2006, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia. Jožef Stefan Institute. ISBN: 978-9-61630-381-1.

[293] Andreas König, Mario Köppen, Ajith Abraham, Christian Igel, and Nikola Kasabov, editors. *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*, September 17–19, 2007, Fraunhofer-Center, University of Kaiserslautern, Kaiserslautern, Germany. IEEE Computer Society. ISBN: 0-7695-2946-1. Library of

Congress Control Number: 2007936727. Product Number E2946. see http://his07.hybridsystem.com/ [accessed 2007-09-01].

Index

List of Figures

List of Tables