Evolutionary Computing in Telecommunications

A likely EC success story

Peter Kampstra

August 2005
Foreword

In the final phase of my study Business Mathematics and Informatics a BMI thesis is obligatory. Not knowing exactly what I wanted, I contacted Rob van der Mei and Gusz Eiben for a subject. The subject ‘Evolutionary Telecommunications’ was interesting and should lead to novel results. However, from a time perspective it became a disaster….

During my study I have always been able to get good grades while not doing too much work. This thesis however, was totally not time efficient, and makes the single exception to this statement. The initial idea was to make an overview of the field. The main problem seemed to be finding enough references and scientific articles. During this research I found a thesis by Marc C. Sinclair that already contained more than 160 references to articles about ‘evolutionary telecommunications’ till 1998. I used this as a basis to find more than 400 references, which created a new problem. How could I read all these references? It would be a shame to throw them away.

Eventually I managed to solve the problem by speeding things up a little (discussing multiple articles at once, etc.). This is one of the reasons that the thesis contains multiple styles, but it had the advantage that no animals have been hurt during the making of this thesis.

For their help with this thesis, I would like to thank Marc C. Sinclair for his work in this area and my supervisors Rob van der Mei and Gusz Eiben.

Anyway, have pleasure reading this thesis! (Or just use it as a reference…)

Peter Kampstra
Summary

The world of telecommunications is booming, the telecommunication infrastructures are becoming more complex, and consequently, telecom operators need to deal with more complex problems to be solved. Evolutionary algorithms form a class of problem solvers that can effectively treat large and complex optimization problems, even in (1) the presence of constraints, (2) noise and (3) dynamic environments.

This thesis describes the use of evolutionary algorithms in the field of telecommunications to tackle these problems. It contains references to over 400 scientific papers in this field. Problems described are things like node location, topology design, routing and restoration, call admission, wavelength allocation, frequency assignment and dimensioning (capacity assignment). Some other, emerging problems seen are for ad hoc networks, node configuration, automated protocol & hardware design, satellites, distributed databases & distributed computing. Also, some possible future directions are described.

It appears that hybrid evolutionary computing is very well suited for solving combinatorial optimization problems in this field. Almost in every reported case, hybrid evolutionary algorithms (memetic algorithms) score best, compared to problem-specific heuristics, simulated annealing, tabu search, etc. An estimated operation cost reduction of 5-15% can be made by using evolutionary computing.
# Contents

Foreword..............................................................................................................2  
Summary..................................................................................................................3  
Contents ....................................................................................................................4  

1. Introduction .........................................................................................................6  
   1.1. Introduction ....................................................................................................6  
   1.2. Further content ............................................................................................7  

2. Evolutionary computing explained .....................................................................8  
   2.1. The basics of evolutionary computing ..........................................................8  
   2.2. Example: Antenna placement ......................................................................9  
   2.3. When to use EC ............................................................................................10  
   2.4. Multiple objectives ......................................................................................10  
   2.5. Alternatives to Evolutionary Computing ......................................................11  
       2.5.1. Tabu Search ..........................................................................................11  
       2.5.2. Simulated Annealing ............................................................................11  
       2.5.3. Other nature-inspired algorithms .........................................................11  
       2.5.4. Others ..................................................................................................11  
   2.6. Memetic algorithms .....................................................................................11  

3. Telecommunication network design ...................................................................12  
   3.1. Different kinds of networks .........................................................................12  
       3.1.1. Optical networks ..................................................................................12  
       3.1.2. Radio networks ....................................................................................12  
       3.1.3. Computer networks .............................................................................12  
       3.1.4. General networks ................................................................................12  
   3.2. Different steps in network design ..................................................................12  
   3.3. Articles on network design ..........................................................................13  
       3.3.1. Node Location ......................................................................................13  
       3.3.2. Topology design ....................................................................................14  
       3.3.3. Trees ...................................................................................................16  
       3.3.4. Routing, restoration and call admission .................................................18  
       3.3.5. Wavelength Allocation ........................................................................21  
       3.3.6. Frequency assignment .........................................................................22  
       3.3.7. Dimensioning .........................................................................................23  

4. Other telecommunication problems ...................................................................25  
   4.1. Node configuration & link tuning (power management, etc.) .....................25  
   4.2. Protocol validation & design .......................................................................25  
   4.3. Hardware design .........................................................................................25  
       4.3.1. Antenna design ......................................................................................25  
       4.3.2. Integrated circuits manufacturing .........................................................25  
       4.3.3. Evolutionary hardware ..........................................................................25  
   4.4. Provider selection/Least cost routing ............................................................25  
   4.5. Distributed databases & web caching ............................................................26  
   4.6. Distributed computing & Grid Computation Networks ............................26  
   4.7. Ad Hoc Networks .......................................................................................26  
   4.8. Registration area planning for mobile networks .........................................26  


4.9. Error correction codes, data equalization & CDMA multi-user detection ...............27
4.10. Satellites..............................................................................................................27
4.11. Others...................................................................................................................27
        4.11.1. Traffic mapping .........................................................................................28
        4.11.2. Electronic marketplaces .............................................................................28
        4.11.3. Intrusion detection ......................................................................................28
        4.11.4. Switch rollout planning ...............................................................................28
        4.11.5. Service Application Software design ..........................................................28
5. Future Research ........................................................................................................29
6. Conclusions...............................................................................................................31
7. References ..................................................................................................................32
        7.1. Available online (linked) ..................................................................................32
        7.2. Only abstract available at VU (IEEE Xplore) ....................................................52
        7.3. Only abstract available online .........................................................................57
        7.4. Not online ..........................................................................................................58
Appendix I. Network design papers overview .............................................................61
Appendix II. References not used (not online) ..............................................................63
1. Introduction

1.1. Introduction

Evolutionary Computation (EC) has many uses in the world of telecommunications. There are many problems in the area of telecommunications that can be solved by evolutionary computation. Evolutionary computation is used when locating antennas, designing networks, routing data in networks, and all kinds of other combinatorial optimization problems. Telecommunications is one of the most active application areas of EC. If we for example look to the number of individuals by subject from EvoWeb [6], we see telecommunications is at the top:

<table>
<thead>
<tr>
<th>Application Area</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telecommunications</td>
<td>58</td>
</tr>
<tr>
<td>Aerospace industry</td>
<td>37</td>
</tr>
<tr>
<td>Biology and chemistry</td>
<td>37</td>
</tr>
<tr>
<td>Business planning and operations research</td>
<td>34</td>
</tr>
<tr>
<td>Electronic and electrical engineering</td>
<td>32</td>
</tr>
<tr>
<td>Automobile industry</td>
<td>31</td>
</tr>
<tr>
<td>Finance</td>
<td>31</td>
</tr>
<tr>
<td>Art and music</td>
<td>30</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>30</td>
</tr>
<tr>
<td>Medicine</td>
<td>30</td>
</tr>
<tr>
<td>Transportation</td>
<td>25</td>
</tr>
<tr>
<td>Chemical engineering</td>
<td>22</td>
</tr>
<tr>
<td>Education</td>
<td>20</td>
</tr>
<tr>
<td>Energy and utilities</td>
<td>20</td>
</tr>
<tr>
<td>Mechanical engineering</td>
<td>18</td>
</tr>
<tr>
<td>Applications - general</td>
<td>16</td>
</tr>
<tr>
<td>Internet</td>
<td>16</td>
</tr>
<tr>
<td>Earth sciences and the environment</td>
<td>14</td>
</tr>
<tr>
<td>Agriculture, farming and food</td>
<td>9</td>
</tr>
<tr>
<td>Civil engineering</td>
<td>9</td>
</tr>
<tr>
<td>Physics</td>
<td>9</td>
</tr>
<tr>
<td>Entertainment and media</td>
<td>8</td>
</tr>
<tr>
<td>Social science</td>
<td>8</td>
</tr>
<tr>
<td>Miscellaneous applications</td>
<td>8</td>
</tr>
<tr>
<td>Linguistics language and speech</td>
<td>5</td>
</tr>
<tr>
<td>Psychology</td>
<td>5</td>
</tr>
<tr>
<td>Architecture</td>
<td>4</td>
</tr>
</tbody>
</table>

The first papers on the use of evolutionary computation for telecommunication problems began to arise already in 1987. The number of papers on this subject grows steadily, as shown in the following graph:
1.2. Further content

In chapter 2, the working of an evolutionary algorithm is explained. In chapter 3 ‘classic’ design of telecommunication networks is described. In chapter 4 some other problem in telecommunications are tackled using evolutionary computing. In chapter 5 describes likely future research. Finally, in chapter 6 some conclusions are drawn. Chapter 7 contains all references to evolutionary telecommunication articles. There are over 400 references, which makes this chapter quite large. The references have been ordered by availability (from ‘available online’, ‘available online but the Vrije Universiteit has no subscription to it’, ‘only abstract available online’ to ‘not available’). Appendix I contains a table with references to the network design papers by category. Appendix II contains some articles that are not discussed for various reasons, but seem related to evolutionary telecommunications.
2. **Evolutionary computing explained**

This chapter includes an (short) explanation of what evolutionary algorithms are. For more information, the reader is referred to the book by Eiben & Smith [1].

### 2.1. The basics of evolutionary computing

The basic idea of evolution is that you always have some sort of population of solutions to a problem, and then try to make that population better and better. At some point you have a solution that is good enough for you, or you have already done lots of attempts. At that point you select the best individual solution, which will be your end solution.

At each point in time during the solving of a problem, there is therefore a solution available. This is called the anytime behavior of an evolutionary algorithm. Normally the starting population with the starting solutions is generated at random. An evolutionary algorithm usually soon gives pretty good solutions. Therefore, most of the time, using an advanced algorithm for the initialization is not worth it.

The still open question is of course how to make the population better. If we zoom in, in pseudo code, an evolutionary algorithm usually works as follows:
INITIALIZE population with random candidate solutions
EVALUATE each candidate
Repeat until (TERMINATION CONDITION) is satisfied
  SELECT parents
  RECOMBINE pairs of parents
  MUTATE the resulting offspring
  EVALUATE new candidates
  SELECT individuals for new population
Loop

The words in CAPITALS are things that can change from algorithm to algorithm. Next to that a representation has to be chosen. This representation can for example be a string of bits (bitstring), an array of integers, a floating-point number, etc. By using the representation, each individual in the population can be mapped to a certain solution to the problem.

INITIALIZE is usually done by generating random individuals (solutions). At EVALUATE, the fitness of the individuals (=quality of the solutions) is determined. The TERMINATION CONDITION can be anything like doing a certain number of generations (iterations) or stopping if there are no improvements seen for some time. The SELECTing of pairs of parents is usually done at random or at random proportional to the fitness of the individuals. RECOMBINATION is usually done by selecting values for the children between the parents. MUTATATION can be done by randomly changing some variables of a child. Finally, SELECTing individuals for the new population can for example simply be done by throwing away all parents and keeping only the best offspring. To avoid degradation, the best parent is may be kept.

2.2. Example: Antenna placement
Let us look at the problem of selecting antenna sites for a new GSM network. Lets say we have 10 possible locations and want to select some sites to give us the coverage needed. A good representation can be 10 Booleans, a Boolean for each site, that determines whether the site is used or not. So there are \(2^{10}\) possible solutions.

A population, of size 2, can look like:

<table>
<thead>
<tr>
<th>Individual</th>
<th>Site 1</th>
<th>Site 2</th>
<th>Site 3</th>
<th>Site 4</th>
<th>Site 5</th>
<th>Site 6</th>
<th>Site 7</th>
<th>Site 8</th>
<th>Site 9</th>
<th>Site 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Each individual will be given a certain fitness value, for example individual 1 might have value 321.95. The fitness function is probably based on the radio coverage of the antenna’s and the costs (rental, placing). This fitness will be used to determine whether an individual is better or worse than another individual.

Initially, we simply generate a random population. We randomly assign the Booleans. Then we may recombine individuals. For example, we can take two individuals, and randomly take values from the first individual or from the second individual. We can make a second child at the same time that just has the other values. Based on the two individuals above, we might generate the following children (the gray values are values from the first parent):
Then we might also use some mutation on the children, for example by randomly switching two values or by randomly switching the value of a site. Finally, some selection will have to take place, to give better individuals an advantage.

Let's say we have a population of 10 individuals. We might generate 60 children, and then throw away all parents. Of these 60 children we select the best 10 and have a new, and hopefully better, population. We can repeat this process until we get a very good antenna placement (or until we give up).

### 2.3. When to use EC

Evolutionary computing is usually used for really hard optimization problems. It is usually well suited under the following conditions:

- **Large solution space**
  Evolutionary Computing is able to tackle really large problems and it is simple to use in combination with parallel computing.

- **No need to find the best solution**
  Evolutionary computing usually finds a good solution, but it is not guaranteed that this solution is the best solution possible. It also does not provide information on how far away the found solution is from the global optimum.

- **No very fast results needed**
  Evolutionary computing algorithms may take up some time.

- **No exact heuristics available**
  Evolutionary computing is well suited for little understood problems, as no problem-specific knowledge is required to use them. If good, fast and exact heuristics that lead to an optimum value are available, they should be used instead.

- **Constraints or multiple objectives are present**
  Evolutionary Computing is well able to handle difficult problems with constraints and multiple objectives.

- **Robust solutions are needed**
  Evolutionary Computing can be used in combination with noise and dynamic environments. The algorithm usually adapts well to a new environment.

### 2.4. Multiple objectives

Many problems in telecommunications are fundamentally multi-objective problems. For example, we want to:

- Minimize the costs
- Maximize the reliability
- Maximize the throughput
- Minimize the delay
We usually want to know the relation between the different objectives. We usually want to find solutions on a Pareto front, the solutions that are the best based on multiple objectives (see figure).

### 2.5. Alternatives to Evolutionary Computing

Evolutionary computing strategies are not the only global search algorithms. Especially tabu search and simulated annealing are well known alternatives.

#### 2.5.1. Tabu Search

Tabu search is another general combinatorial optimization technique. The idea is basically the same as with local search, but when a local optimum is found, the algorithm does not stop. Instead the scheduling might continue, by disallowing points previously visited (hence ‘tabu’).

#### 2.5.2. Simulated Annealing

Simulated annealing is also quite similar to local search. It is a very well known algorithm and is described in the literature. Technically, simulated annealing can be seen as an evolutionary algorithm with population size 1, but it is generally seen as another class of algorithms. The main difference between local search is that neighbors are inspected in a random fashion and that sometimes worse neighbor solutions are accepted. We want to find a global optimum instead of a local optimum. By accepting worse solutions, we are hoping that we leave a local optimum so we can head for a global one. Over the time, the chance of accepting a worse solution goes down to zero. The chance is also related to the difference in the value of the new solution. The chance of picking a slightly worse solution is higher than the chance of picking a much worse solution. As the last solution seen certainly does not have to be the best solution seen, the best solution is always kept.

#### 2.5.3. Other nature-inspired algorithms

A well-known algorithm is Ant colony optimization, which is derived from the structured behavior of ants. It works especially well for routing. Other nature-inspired algorithms are for example based on the behavior of bees and the working of the immune system.

#### 2.5.4. Others

There are many other alternatives, like Greedy Algorithms, Hill climbers, scatter search, path relinking methods, grasp algorithms, problem-specific algorithms, and so on.

### 2.6. Memetic algorithms

Mixing evolutionary computing with other heuristics leads to memetic algorithms (or hybrid genetic algorithms). For example, initialization can be done by using well-known solutions. During evolution, local search might be used first to find better solutions. According to ‘no free lunch’-theorems, no non-revisiting algorithm can be best for all possible problems. The idea is that only by incorporating problem-specific information, better algorithms can be created.
3. Telecommunication network design

Most, if not all, of the early work in the field of evolutionary telecommunications, falls in to this category. An exhaustive 1998 survey by Marc C. Sinclair in [2] forms the basis of this chapter.

3.1. Different kinds of networks

3.1.1. Optical networks

Optical networks have the special property that the connections between the nodes consist of optical fibers. The capacity of an optical fiber depends on the number of wavebands that go over the fiber. Of course, the higher the number of wavebands, the more the connection will cost. Another interesting aspect of optical networks is that routing a stream of data, without changing its waveband, is usually cheaper than with changing wavebands.

3.1.2. Radio networks

Radio or wireless networks do not need wires. Examples are the well known GSM, UMTS and WLAN networks. Krishnamachari & Wicker [4] made an interesting book chapter on the use of global search techniques for radio networks. They identified the following problems:

- Design of a fixed network topology (discussed in 3.3.2)
- Channel allocation (frequency assignment in 3.3.6)
- Optimal base station location (discussed in 3.3.1)
- Mobility management (discussed in 4.8)
- Call management (see call admission in 3.3.4)
- Optimal multi-user detection in CDMA networks (see 4.9)
- TDMA Frame Pattern Design (see traffic scheduling in 3.3.4)

3.1.3. Computer networks

Computer networks differ from ‘general networks’, by the fact that the data in the network is from communication between computers. Normally certain known protocols for communication between computers, like TCP or UDP on the Internet, are used. These protocols have certain properties, which could be used for solving a certain problem. For example, TCP communication from computer A to computer B is impossible if there is no connection from B to A. UDP communications require only a connection from A to B to work.

3.1.4. General networks

If there are no special properties of a network used for a certain method, we say it applies to general networks. A general network usually consists of some nodes, and some connections between these nodes with a certain capacity (bandwidth) for each connection.

3.2. Different steps in network design

Traditionally, the problem of designing a network is divided into different problems, like:

1. Designing the network topology and/or locating network nodes
2. Allocating network capacities for the links (dimensioning)
3. Developing the routing used and/or assigning wavelengths and frequencies.
Only a few papers consider all aspect at once, like the article by White and others [271], to avoid local optimization but global sub-optimization. Because many times there already is some existing technology and/or infrastructure, global optimization may not always be possible.

Articles on these different steps are described into different subchapters. An Exception to this are articles on tree structures, which have their own section, because these articles usually tackle the same problem (encoding trees).

3.3. Articles on network design
This part describes all articles found on network design. As there are lots of articles, it is mainly meant as a reference. Appendix I contains a table with all references by subject.

3.3.1. Node Location
Node location is the problem of finding locations for certain nodes in a network. For example, in radio networks it is evident that antennas must be positioned. Some of the papers covering node location for optical networks [24][402][387] and wireless networks [289] also cover topology and are discussed there. One paper [346] tackles both base station placement and frequency assignment and is discussed at frequency assignment.

In 1992, Potter and others [326] studied the design of military networks, using a hybrid genetic algorithm with several representations and problem-specific operators. Their work included battlefield location and network element selection. Later they [398][427][400] improved this work further with enhancements to the objective function, efficiency gains and additional heuristics. Already in 1994, Routen [321] showed that genetic algorithms could be used to place concentrators in local access network design. He used some problem-specific operators and an integer representation. In 1995, among other problems, Chardaire and others in [15] also covered this problem, this time for concentrators for computer terminals, among others.

Celli and others [322] also positioned concentrators with a genetic algorithm, this time for metropolitan area networks, to demonstrate that parallelization and a proper choice of variables speed up the algorithm used. A metropolitan area network (MAN) typically is a network, which lies qua size between a local area network (LAN, with local computers), and a world area network (WAN, like the internet backbone).

Webb and others [327][401] employed a genetic algorithm with heuristic repair for the selection of backbone nodes in a ring/star transport network. They showed the topologies resulting from true economic cost models, instead of distance-cost models. Levitin [264] positioned retransmitters in a transmission network with vulnerable nodes using genetic algorithms. In 2004, Livramento and others [108] studied the partitioning of a city network into service sections, which can be controlled by a single standard communication switch. They also positioned the switches. Tests of their genetic algorithm with real instances showed promising results.

For radio networks, the most natural problem is of course selecting the sites for antennas/receivers. In 1997, Calégari and others [36][238][279] showed that their parallel genetic algorithm with multiple populations outperformed a single population algorithm for this problem. They [37] also found some other heuristics to be less performing.
Gondim [328] tackled the problem of associating cells to switching centers with a genetic algorithm. Zimmermann and others [188][189] used evolutionary algorithms for the antenna placement problem. Tang and others [388][282] used a multi-objective genetic algorithm for determining the number of locations and their places in a wireless local area network (WLAN). Lieska and others [347] used three different approaches to locate base stations to show the behavior of genetic algorithms differs quite a bit depending on the approach. Krzanowski & Raper [237] used hybrid genetic algorithms to select base stations.

In 2004, Jedidi and others [107] tackled node location not using radiographic criteria, but using geometric criteria instead to enable theories on these geometric structures to be used by network designers. They used an evolutionary algorithm to accomplish this. Raisanen and others [172][173], Meunier and others [174], Watanabe and others [175] and Ozugur and others [176] used multi-objective genetic algorithms for base station location. Multiple good solutions (on a Pareto front) were produced. Reininger and others [362] investigated base station placement taking into account multiple periods of use. Chan and others [193] and Li and others [194] investigated base station placement in CDMA personal communication networks. Alba and others [217] used a parallel multi-objective evolutionary algorithm to select base stations for third generation networks. Hei and others [170], Siregar and others [180], Chan and others [255] and Vijayanand and others [223] studied wavelength converter placement and routing in optical networks using genetic algorithms.

### 3.3.2. Topology design

Some of the papers studied discuss the problem of the placement of concentrators [321][108] or network expansion [327][401][330][399] and have been discussed already at Node location. Some others [331][403][404] discuss topology based on routing, and are discussed in that paragraph.

Kumar and others [52][53] used a genetic algorithm, with some problem-specific repair and crossover functions, to develop topologies for computer networks. They mainly aimed at reliability and the paper was published already in 1991. Michalewicz [390] also developed tree topologies for computer networks, also using a problem-specific recombination and mutation operator. Later, Kumar and others [330] developed a genetic algorithm to tackle computer network expansion. Their genetic algorithm selects new nodes to be added and determines their link to the existing network. Srivastava and others [399] revised that work to demonstrate their distributed genetic algorithm.

In 1992, Sobotka [405] studied survivable military communication networks, assuming a fixed number of satellite links. He also used a problem-specific recombination and tried to reduce the impact of damages.

In 1997, Deeter & Smith [39] studied reliable topology design with genetic algorithms for small networks, while minimizing costs. Later, Deeter, Smith & Konak [220][221][11][17] designed (backbone) network topologies with capacities taking into account both economics and reliability. Dengiz and others [40] in the same period also studied this subject with a genetic algorithm, but their links all had the same reliability and fixed, known costs. Later they refined their fitness function to give a more accurate assessment for the fitter individuals [41][42], in order to study (even) larger networks. With a hybrid genetic algorithm, they [43][44] obtained much better results by using local search optimization and repair, among
other things. Bayan and others [34] used a team of solvers including genetic algorithms to create reliable networks. Sayoud and others [265] also found that genetic algorithms were able to create better network designs faster for small test networks. In 2001, Reichelt and others [250] created topologies under a reliability constraint using a genetic heuristic with intelligent repair operator. Liu & Iwamura [262] solved network reliability models by a simulation-based genetic algorithm. Ghosh and others [27] used genetic algorithms for backbone network design under a costs constraint.

From 2000 on, Ljubic, Raidl and others [303][304][305][306][307] used evolutionary computing to create biconnectivity graphs, where there are at least two connections between two nodes.

In 1999, Kim and others [364] used genetic algorithms to generate network topologies, using bicriteria optimization, considering both cost and reliability to generate a Pareto front. In 2004, Gen and others [276][396][397] also did bicriteria optimization for network topologies using genetic algorithms with fuzzy logic. In 2000, Kumar and others [234][235] extended their work, using genetic algorithms considering multiple objectives. They generated solutions on a Pareto front. Later they [241][242] extended their work even further, including to distributed evolutionary algorithms.

In 2001, Duarte and others [248] selected links using parallel genetic algorithms to generate a Pareto front considering multiple objectives for network design.

In 1998, Tang and others [270] showed that asynchronous transfer mode (ATM) network design was solved better using genetic algorithms. Thompson [166] compared the performance of genetic algorithms and simulated annealing for topology design of ATM networks. On average, the genetic algorithm solution was better.


Already in 1994, Abuali and others [57] used a genetic algorithm to assign terminals to concentrators with a permutation encoding. They found the genetic algorithm to work better than a greedy algorithm. Later they [38] solved the subset interconnection problem and produced superior results to some previous known algorithms. The subset interconnection problem is a topology design problem where all nodes in a subset have to be internally connected, as well as the subsets themselves. In 2004, Salcedo & Yao [239] combined genetic algorithms and neural networks to assign terminals to switches.

In 1996, Tanaka & Berlage [47] studied video-on-demand network design with genetic algorithms, by designing a topology and specifying storage nodes for the videos.

Saha & Chakraborty [46] studied ways to add new links to a network with a genetic algorithm. They assumed that both the cost and the profit for a link were known in advance. Nakamura & Oda [45] also studied this problem, but then over multiple planning periods. They used a genetic algorithm combined with a heuristic routing and restoration algorithm. Rothlauf & Grassr [251] used a genetic approach to support network topology planning over multiple use periods. Montana and others [252] studied adaptive reconfiguration of data networks.
For optical networks, Sinclair and others tried to figure out which links to include in the network topology, minimizing costs. They applied a simple genetic algorithm [49], a hybrid algorithm [50] and other genetic programming approaches [48][246] to this problem. Mikac & Inkret [406] extended Sinclair’s work with genetic algorithms by also taking minimizing unavailability into account. They used two different fitness functions. In odd generations, the fitness focused on minimum costs, while in even generations it focused on availability. White and others [271] not only determined ring structures, but also determined dimensioning and routing to avoid global sub-optimization. Armory and others [201] developed reliable ring structures for optical networks using genetic algorithms. Karunanithi & Carpenter [227] used a genetic algorithm for determining the size of (SONET) ring structures. Pickavet & Demeester [231] determined SONET topologies and capacities too, using a special zoom-in heuristic. Cortes [244] did global optical topology design using genetic algorithms that explored optimality conditions. Liu and others [367] also optimized logical topologies in optical networks. He and others [272] determined optical ring structures and showed better results with an evolutionary algorithm than a genetic one. Xin and others [22] used genetic algorithms combined with heuristics to design large optical networks.

In 1996, Paul and others [402] studied tree topologies for optical networks. They minimized costs for local access networks and used some problem-specific operators. Brittain and others [387] also did this, however they provided full details and also considered non-tree topologies.

For computer networks, Hewitt and others [332] studied minimum cost network design, using connectivity and delay constraints. They used a hybrid genetic algorithm. Later, Ko and others [51] used hybrid genetic algorithms in three stages, to create after each other a topology, routing and link capacities. Pierre & Legault [54][56] also studied minimum cost network design with genetic algorithms, using connectivity and delay constraints. They select network links, but the fitness function used also allocates link capacities. Qin and others [55] designed ISDN networks using a genetic algorithm. Their fitness function used also allocates link capacities, but network delay is not calculated. Habib and others [260] created a computer program for designing hierarchical intranets using genetic algorithms. Berryman and others [268][269] used evolutionary computing to explore the tradeoffs between network redundancy and pleiotropy (‘server duplication’) in computer networks. Combining both created robust networks. Gen & Cheng [165] studied network design on various topics including LAN design, and found genetic algorithms quite effective. Altiparmak and others [28] compared various metaheuristics for computer network design and found that memetic algorithms did best.


### 3.3.3. Trees

The optical networks papers [24][402] have already been discussed at topology, in order to keep them at one place. The work by Routen [321], about placing concentrators in local access network design, has been discussed at Node Location. Some papers that cover routing in tree networks [66][67] are discussed there.
The main problem with trees is probably how to choose a correct representation for the tree. In the papers, different solutions are described to this problem.

Already in 1989, Hesser and others [408] applied a simple genetic algorithm with some heuristics for decoding to determine arbitrary Steiner points. Steiner points are points that are added to a graph, to make the total length of a network covering all points smaller (see figure). Later, Kapsalis and others [14][15] used a simple genetic algorithm to select Steiner points from a given set of nodes. Julstrom [391] covered the rectilinear Steiner tree problem, where only horizontal and vertical lines are allowed. He used a hybrid genetic algorithm, but previous heuristics seemed to give better results. Later, Esbensen & Mazumder [68] returned to the selection of Steiner points from a given set of nodes. Their hybrid genetic algorithm uses a bit-string encoding and some repair routines. Later they [69] demonstrated that the algorithm performed better than the one Kapsalis made. In 1995 [409] the algorithm was improved further. In 2000, Chu and others [280] found that for the Steiner tree problem genetic algorithms performed comparable to their tabu search algorithm. Kulturel and others [16], Saltouros and others [196], Panadero & Fernández [198] and Presti and others [197] also looked at the Steiner problem and found good results. Galiasso & Wainwright [219] tackled the Steiner routing tree problem with single split paths.

Abuali and others [58] also looked at trees. They investigated minimal spanning trees using several genetic algorithms. Their permutation-encoded genetic algorithms performed better than some heuristics for this problem, especially for large networks. Routen [321] also studied this problem in 1994, but had disappointing results with an integer-based encoding. In 1995, Zhao and others [63] had more success with a hybrid genetic algorithm. They used an object-oriented representation. In 1994 Abuali and others [59] moved to the stochastic version of the problem, where nodes only need to be connected with a certain probability. They used a so-called Prüfer encoding. The same year Palmer & Kershenbaum [70][71] argued against that encoding. They obtained better results than some existing algorithms using a genetic algorithm with a new encoding called node-cost bias encoding. In 2001, Gottlieb and others [311] argued again against Prüfer numbers. In 1995, Abuali and others [60] later designed the so-called determinant-factorization encoding for the stochastic minimum spanning tree problem. They [61] compared Prüfer encoding, node-cost bias encoding and determinant-factorization. They found that determinant-factorization and node-cost bias encoding performed well, but node-cost bias encoding performed better for larger networks. Abuali and others [62] later looked at the more constrained three-star tree problem, where all branches have a chain of three nodes to the root. They concluded the same (determinant-factorization and node-cost bias encoding performed well, but node-cost bias encoding performed better for larger networks).

In 1995, Berry and others [407] also studied the minimum spanning tree problem. They used predecessor-vector encoding (also deprecated by Palemer & Kershenbaum [70]) with some heuristic mutation and recombination. Walter & Smith [410] also independently developed a hybrid genetic algorithm for (directed) minimum spanning trees. Unfortunately there seems to be no comparison of these papers with the work of others. In 1997, Soper & McKenzie [365] and Chou and others [366] also studied minimal spanning trees. In 2000, Li & Bouchebaba [218] solved the problem of finding an optimal spanning tree.
by working directly on the tree itself without an intermediate encoding. Hsinghua and others [369] tested the effect of various properties of a genetic algorithm for the constrained minimal spanning tree problem. Despite being argued against, in 2002 Haghighat and others [199] still used Prüfer numbers for constrained tree optimization. In 2003, Zhou & Gen [162] showed that their genetic algorithm for generating tree-like networks was highly effective compared to other heuristics.


Elbaum & Sidi [64][65] studied the design of local area networks (LANs). The LAN is seen as a tree, and encoded with a hybrid chromosome encoding.

### 3.3.4. Routing, restoration and call admission

Some papers [412][51][271] have already been discussed at dimensioning/topology. The papers on optical routing and wavelength allocation [18][20][323][324][325][21][93][94][95][109][113][23][266] are discussed at wavelength allocation. The papers on wavelength converter placement [170] are placed at node location.


In 1993, Munakata & Hashier [73] used a hybrid genetic algorithm for the maximum flow problem in a capacitated graph and got good results. However, they were unable to get better results than existing conventional heuristics. Sinclair [74] used a bit-string encoding for the design of static-routing tables for an unreliable circuit-switched network. However, his work also performed worse than some existing heuristics. His genetic algorithm in a 1999 paper [263] for reliable military communication networks also had worse results than another heuristic. Shimamoto and others [75] studied dynamic routing in switched networks. They used a steady-state genetic algorithm to exploit the relatively slow changes in traffic distribution, and obtained a grade-of-service that compared well to a fixed routing algorithm.

In 1995, Mann and others [413] tackled static routing to minimize costs while balancing the traffic load. Later, they [414] made a detailed comparison with different algorithms, both genetic algorithms and simulated annealing. The results where similar, but simulated annealing took less computation effort. Mann & Smith [415] later studied routing in optic ring networks, again minimizing costs while balancing the load. This time, they found a genetic algorithm to be more robust than some simulated annealing algorithms.

In 2000, Knowles and others [89] researched static routing for backbone networks. They introduced a new operator that can be used for incremental evaluation to improve the performance of a genetic algorithm. They [179] also researched offline-routing and generated a Pareto front using cost and reliability criteria. In 2001, Luckschandl [132] made a framework in Java to use genetic algorithms for routing optimization. In 2002, Kwong and others [257] used genetic algorithms for use in networks using the virtual path routing concept. Their approach worked for both normal and broadcast traffic. Liang and others [88] investigated a distributed genetic algorithm for dynamic routing. It made good decisions without the need of global information (such as the number of nodes in the network). They improved on an ant colony based algorithm for this problem called AntNet.

In 1996, Hoelting and others [76] studied the problem of finding an investigator tour in a network. This is the problem of finding a tour for fault detecting in a point-to-point telecommunication network. Their genetic algorithm outperformed a deterministic heuristic. Hoelting and others [77] studied broadcasting a message through a network to all nodes in minimum time. Their genetic algorithm with a permutation based encoding outperformed a recent heuristic found in the literature. Almost the same group went on [66] studying point-to-multipoint routing. The work of Cox and others [411] was extended and a permutation-based genetic algorithm was combined with a heuristic Steiner tree algorithm. Compared to previous results, the results appeared to be superior. In 1998, Zhu and others [67] extended that work to also consider subsets of requests handled, if handling all requests is unfeasible.

Servet and others [416] used genetic algorithms to estimate the size of traffic streams between nodes, starting from the offered traffic on the links.

In 1996, Huang and others [403] evolved two routes between (pairs) of nodes in a network, guiding topology selection and link dimensioning. They used heuristic recombination and mutation. Later they [331][404] investigated parallel computing to speed up the process.

In 1997, Bentall and others [333] used a genetic algorithm to find restoration paths for a heavily loaded network (where not all connections could be restored). They used the results as a benchmark, not for real-time computation.

Kozdrowski and others [392] used a hybrid genetic algorithm to assign link capacities and traffic streams in a network with link failures.

Kirkwood and others [78][79] used genetic programming to obtain primary and restoration paths in a network with single link failures. Their approach however did not scale well to larger networks. In 2000, He & Mort [87] developed backup-routing tables (=restoration paths) with a hybrid genetic algorithm, while developing primary tables using shortest path routing. They found solutions that led to less congested nodes and links and a better utilization of network resources.

Call admission, for radio networks, was studied by Yener & Rose. In 1994, they [424] studied a bit-string genetic algorithm to create admission policies for a small network. This led to results comparable to a simple ‘admit if possible’ heuristic. Later they [84][85] studied admission policies based on local instead of global information, to be able to tackle larger networks. By examining the results, a novel heuristic local admission policy was suggested.

Sherif and others [358] studied call admission in wireless networks, considering the Quality of Service(QoS) of various multimedia services. Karabudak and others [295] studied call admission for next generation wireless networks, including for multimedia QoS ‘calls’.
Abedi & Vadgama [171], Chakraborty [356], Huang & Cheng [177] and Ngo & Li [357] studied the scheduling of wireless broadcasting.

For computer networks, Carse and others [417][418] evolved multi-agent systems for adaptive routing. They had good results with a small network. In 1997, Munetomo and others [425] studied approximately the same problem, but for larger networks. They employed path-based genetic operators to manipulate the routing tables directly. In comparison with existing Internet routing algorithms, their algorithm showed better performance (especially with heavy traffic). He [337] researched static routing and link capacity allocation in large computer networks. He used a genetic algorithm that accidentally used the same representation as Pan & Wang [72]. In 1996, Tantertid and others [338][426] applied Shimamoto’s [75] genetic algorithm for reliable circuit-switched networks to the problem of virtual routing in ATM networks. They were able to find some optimal solution to this problem. Later, Pitsillides and others [91][92] compared a genetic algorithm to a classical algorithm for this problem, and concluded that the genetic algorithm produced better results. Swaminathan and others [236] also had good results with genetic algorithms. In 2000, Shazli and others [97] used fuzzy logic with a genetic algorithm to tackle this problem and got reasonable results. Medhi & Tipper [192] selected virtual paths for broadband routing. For larger networks, their hybrid approach worked better.

Lin and others [222] determined link capacities and routing for large-scale computer networks. Markaki and others [339] researched a two-dimensional binary genetic algorithm combined with a hill-climbing algorithm for connection-oriented channel selection in a LAN/MAN. They got better results than a graph-coloring based solution.


OSPF is a routing protocol frequently used for Internet connections between Internet Service Providers (ISPs). It requires weights to be set for the connections. Since 2002, a group of overlapping authors consisting of Resende and others [116][117][118][119] examined a (hybrid) genetic algorithm for this task. They showed good or better results than other algorithms in terms of efficiency and robustness. Riedl [139] investigated different service metrics (bandwidth and delay) on this subject. Mulyana & Killat [168] also studied setting OSPF weights using hybrid genetic algorithms, but they minimized utilization as an objective function. In 2004, Kang and others [120] evaluated the Quality of Service (QoS) of their algorithm using simulation.

For optical networks, Pan and others [245] investigated genetic algorithms for message scheduling. They reduced the number of passes required to sent messages.


In 2004, Gelenbe and others [100] discussed the use of a genetic algorithm for path finding and maintaining as an extension for their QoS routing protocol called CPN. Cornellas & Dalfo [110] investigated broadcast algorithms for Manhattan street networks using a genetic algorithm. Barolli and others [136][137][138] also did research using genetic algorithms to do QoS routing. Zhengying and others [142] and Haghighat and others [143] investigated least-cost multicast routing with bandwidth constraints using genetic algorithms and multicast trees. Li and others [144], Araujo and others [145], Tsai and others [146], Pan and others [157], Cui and others [147] and Layuan and others [149] also investigated QoS multicast routing with genetic algorithms under various circumstances. Siregar and others [156] and Din and others [155] investigated multicast routing in optical (WDM) networks.

Another interesting development is the use of intelligent agents for routing. In 1999, Nonas & Poulouvassilis [102] investigated network routing adaptation by intelligent agents using genetic algorithms. They showed that link failures could be recovered in a dynamic situation.

3.3.5. Wavelength Allocation
If wavelength allocation is done, often the algorithm also takes care of routing.

The earliest paper, by Tan & Sinclair [18] in 1995, uses a genetic algorithm for route selection between nodes in the network. The number of wavelengths needed to route traffic from node A to B is assumed to be an integer. A gene describes how this traffic is routed. The number of wavelength required between neighboring nodes C and D is simply the sum of all routes that go through it and the maximum wavelength count is minimized. Sinclair [20][323][324] later extended his research with hybridization and considering cost models. At the same time, Abed & Ghanta [19] used a genetic algorithm for determining the topology of an optical network or Lightwave Network Architecture (LNA), using wavelength allocation on single optical fiber.

In 2000, Ali & Ramamurthy [113] used genetic algorithms to tackle the wavelength assignment and routing problem taking into account power considerations. Additional time taken by the genetic algorithm seemed to pay off. Saha and others [230] also determined routing and wavelengths for optical networks taking into account reliability. Qin and others [23][266] solved the routing and wavelength assignment problem with limited range wavelength conversions. In 2003, Banjeree and others [93][94][95] investigated the routing and wavelength assignment problem using multi-objective genetic algorithms, for presenting a number of combinations to network operators. The objectives used were things like average flow, average delay and expected blocking. Better results than with simulated annealing were seen. In 2004, Cagatay Talay & Oktug [109] independently tackled this problem using a hybrid genetic algorithm to minimize costs. Their results looked promising compared to recent heuristics.
Recently, the area of **traffic grooming** is starting to get more attention. Traffic grooming is the science of combining multiple low-bandwidth traffic streams into one waveband, in order to minimize the number of waveband stoppers needed. Lee & Park [325] and Xu and others [21][229] used genetic algorithms on this problem for some specific topologies.

### 3.3.6. Frequency assignment

Almost all papers here can be found in the database on frequency assignment problems at [http://fap.zib.de/biblio/](http://fap.zib.de/biblio/).

A lot of work in this area has been done by Crompton. In 1993, Crompton and others [419] used a parallel genetic algorithm to minimize interference for frequency assignment for air to ground to air problems using an integer representation. Later [420] they improved this algorithm using heuristics and another representation. In [393] they compared their results with a backtracking heuristic. The genetic algorithm turned out to be better when using the same execution time. However, Hurley & Smith [334] demonstrated that simulated annealing, using incremental fitness evaluation, showed even better results. Tabu search [80] was also better than the genetic algorithm (but worse than simulated annealing).

Valenzuela and others [394] studied the same problem, but minimized the number of frequencies required if no interference is allowed. They [81] however showed that their genetic algorithm was better than simulated annealing or tabu search.

In 1994, Cuppini [421] used a bit-string genetic algorithm with no crossover for small frequency assignment problems. Kapsalis and others [15][423] also used a bit-string genetic algorithm. They tested a wide range of operators and fitness functions and tried to minimize both interference and the number of frequencies required. Later, Kapsalis and Smith [422] used a meta-genetic algorithm to select an even larger number of operators and tweak the parameters for their algorithm. In 1997, Ngo and Li [335] used a bit-string genetic algorithm with heuristic operators and local search to minimize interference.

Kaminsky [83] studied the hourly assignment of military frequencies using a genetic algorithm. Sandalidis and others [336] used bit-string evolutionary strategies for dynamic frequency assignment for cellular mobile radio. Their algorithm showed better results than four other (problem-specific) algorithms, but was not tested for real-time operations.

In 1995, Hao & Dorne [121][122] tackled the frequency assignment problem (FAP) with a hybrid genetic algorithm and showed that their hybrid approach is promising compared to other global search methods. In 1997, Renaud & Caminada [128] tested various methods and operators for the frequency assignment problem. In 1998, Crisan & Mühlenbein [82] used a so-called breeder genetic algorithm for the frequency assignment in digital cellular networks, to minimize interference. Their problem formulation is more complex than the others. Later they [124] study the fitness landscape of the function that is optimized to explain the performance of previous evolutionary algorithms. Valenzuela [125] improved on the previous methods by exploring different operators and hybridization with a greedy algorithm. Weinberg and others [123] combined multiple local search methods and introduced two new operators to tackle larger problems. Also in 2001, Cotta & Troya [126] compared different evolutionary algorithms and operators. Mabed and others [127] dynamically allocated frequencies. In 2004, Matsui and others [130] studied fixed frequency assignment with a bandwidth constraint.
In a 2002 article, Aardal and others [129] describe the CALMA project. In this project, a big evaluation of global search methods for frequency assignment was undertaken. The algorithm with the best results is best described as a hybrid genetic algorithm.

In 2003, Weicker and others [346] studied station placement and frequency assignment together. A multi-objective genetic algorithm was used.

Beckmann and others [349] Ghosh and others [350], Funabiki and others [167], Li and others [182], Yoshino & Ohtomo [183], Lau & Coghill [359], Jaimes-Romero and others [360], Zomaya [361] and Sandalidis and others [186][187] tested various evolutionary algorithms for the channel assignment problem. Kwok [351] used a Linux cluster to solve channel assignment problems using genetic algorithms. Kassotakis and others [352] studied channel re-allocation using genetic algorithms. Matsui and others [184] [185] researched channel assignment in case of limited bandwidth. Mabed and others [181] tackled the channel assignment problem considering multiple periods.

3.3.7. Dimensioning

Many papers on dimensioning handle dimensioning among other subjects. Most of the optical papers [50][48][415][406][24][402][49][231][271] have been dealt with already. Many general network papers [39][403][404][331][392][73][70][71][47][410][220][221][265] and computer network papers [337][332][51][54][55][56][72][222] also meanly consider other aspects. The papers of Potter and others [326][398][427][400] on military radio networks are discussed at node location.

Already in 1987, Davis & Coombs [428][429][430] chose link capacities for packet-switched wide-area networks using an integer genetic algorithm with problem-specific operators. They also tackled the problem of incorporating constraints into their model.

In 1993, Davis and others [412] minimized cost under a reliability constraint in a network. They used an integer genetic algorithm to select link capacities and traffic routes in a network, using problem-specific operators and local repair. In 1997, Ahuja [368] dimensioned computer networks to optimize reliability. In 1998, Garcia and others [86] tackled network expansions over multiple periods. They tried to determine when to increase link sizes to meet forecasted demand. They used a steady-state integer genetic algorithm combined with local search. Heegaard and others [261] used genetic algorithms for dimensioning full service access networks, including both unicast and multicast traffic. In 2000, Mostafa & Eid [345] used a genetic algorithm combined shortest path routing to determine link capacities in a packet-switched network. A 3.17% reduction in costs was made over a previous method. Al-Rumaih and others [214] determined spare capacities for survivable mesh networks. In 2001, Arabas and Kozdrowski [371] used an evolutionary algorithm to find good backbone capacities. In 2003, Runggeratigul [133] showed a memetic algorithm for the link capacity problem in packet-switched networks, while taking into consideration the existing network facilities. Podnar & Skorin [370] used genetic algorithms to solve a problem of minimizing link costs, for the case that the costs for link usage are discounted if the usage exceeds a certain threshold. Atzori and others [258] used genetic algorithms to determine network capacities for multicast traffic.
For optical networks, Chen & Zheng [228] studied the capacity allocation for optical ring structures. Chong & Kwong [256] had very favorable results with using genetic algorithms to allocate spare capacities. Mutafungwa [298] designed link redundancy enhancements for optical cross-connected nodes.
4. Other telecommunication problems

4.1. Node configuration & link tuning (power management, etc.)

In 2001, Moustafa and others [96] optimized the amount of power and bit rate used by mobile phones using genetic algorithms. A significant enhancement in signal quality and power level was noticed throughout several experiments. In 2004, Hu & Goodman [372] used genetic algorithms to configure wireless access points. Zhou and others [169][191] studied setting transmission power in wireless (CDMA) networks. Genetic algorithms generated optimal results.

In 2003, Withall and others [90] investigated a genetic algorithm for optimizing the parameters for sending data that a single computer on the Internet can change. For some situations, they reported better performance than gained with using a fixed package size (standard). Yang and others [274][275] used genetic algorithms to generate a Pareto front between the criteria throughput and delay for an optical link.

4.2. Protocol validation & design

Evolutionary Computing can be used to design and test telecommunication protocols. In 1999, Corno and others [103][104] used genetic algorithms to verify the implementation of protocols. The genetic algorithm with a simulator was able to provide reliable testing results. Alba & Troya [224] and Baldi and others [225][226] also showed the benefits of using genetic algorithms for protocol validation.


4.3. Hardware design

This part contains some examples, but might not be exhaustive as hardware design is not really unique to telecommunications.

4.3.1. Antenna design

Edwards and others [385] designed a spiral antenna using multi-objective genetic algorithms. Edwards & Cook [178] used genetic algorithms to design an antenna for triple band devices (for the current European GSM bands and the new UMTS band). For three different criteria, a Pareto front for is generated. Himdi & Daniel [386] used genetic algorithms to design multi band antennas.

4.3.2. Integrated circuits manufacturing

For the design of integrated circuits, evolutionary methods can be used. This is for example the design of integrated circuitry in switches and other network components. The design of hardware by itself is not really unique to telecommunications, so it won’t be covered here in detail.

4.3.3. Evolutionary hardware

An interesting area is also evolutionary hardware. The outcome of evolving hardware is hard to determine, and could therefore be better for security. The design of hardware by itself is not really unique to telecommunications, so it won’t be covered here in detail.
4.4. Provider selection/Least cost routing

The telecommunication industry is subject to all kinds of agreements and regulations, which lead to new and interesting problems. One example of this is regulation that allows callback facilities. This means that a call from A to B, may be immediately terminated, and followed by a call from B to A. The reason for this is usually that there are cheaper rates from B to A, than from A to B. According to the ECTELNET Report [3], some work has already been done by Alain Sutter, but by using tabu search (see 2.5.1) instead of an evolutionary algorithm.

Pressmar [164] studied the optimization of a rented network topology using genetic algorithms. Ahuja [431] probably already used genetic algorithms to select the cheapest Internet service providers. As I could not locate any other public papers, this area might be open to future research.

4.5. Distributed databases & web caching

Deciding the distribution and duplication of the data in distributed databases is a problem that can be tackled with evolutionary methods.

Knowles and others [179][296][12][13] researched this problem and generated a Pareto front considering cost and reliability criteria. Ahmad and others [215] also created a genetic algorithm for allocating data in distributed databases. Oates [253] showed how genetic algorithms can be used for optimizing distributed database performance.

Yanxiang and others [259] used genetic algorithms to support distributed information retrieval systems built upon mobile agent technology.


4.6. Distributed computing & Grid Computation Networks


4.7. Ad Hoc Networks

Ad Hoc Networks are a hot topic in telecommunications research. The basic idea is that normal nodes are also forwarders and routers in the network. Barolli and others [134][135] did some research on using genetic algorithms to incorporation quality of service requirements into ad hoc routing. Mao and others [148][292] investigated multicast routing for video. Turgut and others [281] optimized the clustering of nodes in an ad hoc network. Marwaha and others [373] and Liu and others [300] did routing in ad hoc networks using fuzzy logic considering multiple objectives. Montana & Redi [301] optimized parameters for an ad hoc routing protocol using genetic algorithms.

4.8. Registration area planning for mobile networks

Subrata & Zomaya [33], Wang and others [381], Das & Sen [290] and Junping & Lee [291] used genetic algorithms for mobile registration area planning. As mobile phones travel, the phone is not always registered at the nearest base station or the current cell they are in. Instead, travel patterns are investigated. This approach saves costs, bandwidth and power, as less re-
registration has to take place. Demestichas and others [26] compared taboo search, simulated annealing and genetic algorithms for fixed location area planning and found similar results between the heuristics.

**4.9. Error correction codes, data equalization & CDMA multi-user detection**

Chen and others [240] used parallel genetic algorithms combined with simulated annealing to create error-correcting codes. Alba & Chicano [395][217] used hybrid genetic algorithms to tackle the problem of finding error correcting codes. Cotta [233] used a memetic approach to tackle the error correcting code problem.


Khuri and others [232] used genetic algorithms to hand out unique codes to nodes in a wireless (CDMA) network. Lim and others [374], Juntti and others [375], Wu and others [376], Abedi and others [377], Yen & Hanzo [283][284] and Ng and others [285] investigated the use of these codes to do multi-user detection using all kinds of evolutionary computation algorithms.

**4.10. Satellites**

In 1999, George D. Smith [302] pointed out the possible advantages of evolutionary computing for satellite communications. He identified the following (non-independent) configuration issues:

- The number of orbits
- The number of satellites per orbit
- The types of orbits (polar, rosette, equatorial, polyhedral, mixed)
- The trajectories (circle, elliptic)
- The power management systems
- The number and types of antennas used for direct user communication
- The routing methodology (ground-based, space-based or both)
- Maintenance and upgrade

In 2002, Confessore [277] and others used genetic algorithms and simulation to try to find Pareto-optimal satellite constellations for regional coverage, minimizing the number of satellites and maximizing the coverage. Barbulescu and others [278] compared genetic algorithms with heuristics for satellite range scheduling, and concluded that genetic algorithms yielded the best performance on difficult problems. Asvial and others [353][354][355] researched the design of satellite constellation using genetic algorithms. They also researched radio resource management for satellites. Genetic algorithms seemed to be able to provide good and robust results.

Hassan and others [383] and Linden [384] used evolutionary computing to process (GPS) satellite signals.

**4.11. Others**

This section contains some other papers found that do not really fall in the previous categories but seem related to evolutionary telecommunications.
4.11.1. **Traffic mapping**
Hoang & Zorn [190] were able to map traffic in real time with the help of genetic algorithms.

4.11.2. **Electronic marketplaces**

4.11.3. **Intrusion detection**
In 2000, Neri [105] used data mining with genetic algorithms to detect TCP/IP network intrusions.

4.11.4. **Switch rollout planning**
Davis and others [273] minimized the costs involved for adding switch modules to a switch station using genetic algorithms.

4.11.5. **Service Application Software design**
Martin [101] used a genetic algorithm to design service applications for telecommunication providers.
5. Future Research

Evolutionary telecommunications is an active area of research where much more research is to be expected. As no area of this field is researched exhaustively, new articles are to be expected in any area previously described. According to the following graph, there currently is no area with more than 55 papers for network design:

For example, this year there is a special issue on Nature-inspired approaches to networks and telecommunications. The list of subjects [5] gives a good view of expected upcoming research. It contains almost all subjects that have been discussed and some new technologies:

- General network design problems
  - Physical topology design problems
  - Survivability and reliability
  - Quality of service
  - Protection and restoration
  - Network management
  - Congestion control
  - Simulation and queuing models
- Optical networks
  - Routing and wavelength assignment
  - Traffic grooming
  - Placement of wavelength converters
  - Placement of optical amplifiers
  - Protection and restoration on optical layer
- Fixed networks design problems
- Cellular and wireless networks
- Satellite communications networks
- Other topics
  - Ad hoc networks
  - Bluetooth/Personal area networks
  - IP/WDM
  - GMPLS and MPLS
  - Internet applications

Following the current trend, future research will tackle more complicated problems. The algorithms used will probably be multi-objective orientated, so that the resulting solutions are on a Pareto front. The algorithms used will be hybrid genetic algorithms or memetic algorithms and able to run parallel on multiple computers. Papers will not tackle only topology for new networks, but also dimensioning and routing to avoid local sub-optimization. The evaluation function will be more complicated, but more accurate. On the other hand, as there is lots of existing infrastructure, there will also be a focus on adapting this infrastructure to new requirements. Also, the papers will be on newer techniques like mobile network papers on GMPLS and MPLS instead of GSM (or the newer UMTS). Papers on routing will consider Quality of Service and multicast traffic. For example, I would not be surprised to see a paper on multimedia content delivery server placement in 200 mbit power line networks.
6. Conclusions

Evolutionary computing is used in a wide variety of telecommunication problems. Over 400 scientific papers on this area have been identified, covering all sorts of problems. Problems tackled are network design things like node location, topology design, routing and restoration, call admission, wavelength allocation, frequency assignment and dimensioning (capacity assignment). Some other, emerging problems seen are for ad hoc networks, node configuration, automated protocol & hardware design, satellites, distributed databases & distributed computing.

It appears that hybrid evolutionary computing is very well suited for solving combinatorial optimization problems in telecommunications. When comparisons are made, almost for every problem, hybrid evolutionary algorithms (memetic algorithms) do best, compared to problem-specific heuristics, simulated annealing, tabu search, etc. A nice example is the CALMA project for frequency assignment.

As heuristics can be incorporated into evolutionary computing resulting in memetic algorithms, even if a heuristic produces better results than an evolutionary computing algorithm, combining both might be very worthwhile.

Note that evolutionary computing also has its disadvantages. For example, finding the global optimum is not guaranteed, the distance to the optimum remains unknown and the results may take some time to generate (see 2.3).

It is hard to estimate the exact added value of evolutionary computing in telecommunications. The 1999 ECTELNET report [3] mentions that an estimated operation cost reduction of 5-15% can be made by using evolutionary computing instead of classic heuristics.

For hard global optimization problems in telecommunications, I therefore think it is a very good idea to look to hybrid evolutionary computing.
7. References

7.1. Available online (linked)


[40] Dengiz, B., Altiparmak, F. & Smith, A.E., Efficient optimization of all-terminal reliable networks using an evolutionary approach, IEEE Transactions on Reliability v46 n1, March 1997, pp. 18-26


[42] Nakamura, H. & Oda, T., Optimization of facility planning and circuit routing for survivable transport networks—an approach based on genetic algorithm and
incremental assignment, IEICE Transactions on Communications vE80-B n2, February 1997, pp. 240-251


[51] Ko, K.-T., Tang, K.-S., Chan, C.-Y., Man, K.-F. & Kwong, S., Using genetic algorithms to design mesh networks, Computer v30 n8, August 1997, pp. 56-61


[64] Elbaum, R. & Sidi, M., Topological design of local area networks using genetic algorithms, Proc. IEEE INFOCOM'95, Boston, Massachusetts, USA, April 1995, v1, pp. 64-71


[73] Munakata, T. & Hashier, D.J., A genetic algorithm applied to the maximum flow problem, Proc. 5th Intl. Conf. on Genetic Algorithms (ICGA'93), University of Illinois at Urbana Champaign, USA, July 1993, pp. 488-493

[74] Sinclair, M.C., The application of a genetic algorithm to trunk network routing table optimisation, Proc. 10th UK Teletraffic Symposium, Martlesham Heath, UK, April 1993, pp. 2/1-2/6


[85] Yener, A. & Rose, C., Genetic algorithms applied to cellular call admission: Local policies, IEEE Transactions on Vehicular Technology v46 n1, February 1997, pp. 72-79 (also on IEEE Xplore)


[143] Li, Layuan & Li, Chunlin, Genetic Algorithm-Based QoS Multicast Routing for Uncertainty in Network Parameters, Lecture Notes in Computer Science, Volume 2642, Jan 2003, Pages 430 – 441

[144] Araujo, PT de & Oliveira, GMB de, Multicast Routing with Quality of Service and Traffic Engineering Requirements in the Internet, Based On Genetic Algorithm, VII Brazilian Symposium on Neural Networks (SBRN'02), November 11 - 14, 2002, Pernambuco, Brazil, Pages 194


[150] Tsuji, Miwako, Munetomo, Masaharu & Akama, Kiyoshi, Metropolitan Area Network Design Using GA Based on Hierarchical Linkage Identification, Lecture Notes in Computer Science, Volume 2724, Jan 2003, Pages 1616 – 1617


[177] Huang, J.-L. & Chen, M.-S., Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment, IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 9, pp. 1143-1156, September 2004 (also on IEEE Xplore)


[181] Mabed, Hakim, Caminada, Alexandre & Hao, Jin-Kao, Multi-period Channel Assignment, Lecture Notes in Computer Science, Volume 2775, Sep 2003, Pages 541-554


Matsui, Shouichi, Watanabe, Isamu & Tokoro, Ken-ichi, An Efficient Hybrid Genetic Algorithm for a Fixed Channel Assignment Problem with Limited Bandwidth, Lecture Notes in Computer Science, Volume 2724, Jan 2003, Pages 2240 – 2251

Matsui, Shouichi, Watanabe, Isamu & Tokoro, Ken-ichi, A Parameter-Free Genetic Algorithm for a Fixed Channel Assignment Problem with Limited Bandwidth, Lecture Notes in Computer Science, Volume 2439, Jan 2002, Pages 789 - 799


Hoang, T.T. Mai & Zorn, W., Simulation of Traffic Engineering in IP-Based Networks, Lecture Notes in Computer Science, Volume 2093, Jan 2001, Page 743


Medhi, D. & Tipper, D., Some approaches to solving a multihour broadband network capacity design problem with single-path routing, Telecommunication Systems, Volume 13, Issue 2 - 4, Jun 2000, Pages 269-291


Li, Juan, Guan, Yong Liang & Soong, Boon Hee, Effect of genetic algorithm parameters on PCS network planning, 25th Annual IEEE Conference on Local Computer Networks (LCN’00), November 08 - 10, 2000, p. 400


Haghighat, A.T., Faez, K., Dehghan, M., Mowlaei, A., Ghahremani, Y., A Genetic Algorithm for Steiner Tree Optimization with Multiple Constraints Using Prüfer
Number, Lecture Notes in Computer Science, Volume 2510, Jan 2002, Pages 272 – 280


[231] Pickavet, Mario & Demeester, Piet, A Zoom-In Approach to Design SDH Mesh Restorable Networks, Journal of Heuristics, Volume 6, Issue 1, Apr 2000, Pages 107 – 130


[242] Kumar, Rajeev, Multicriteria Network Design Using Distributed Evolutionary Algorithm, Lecture Notes in Computer Science, Volume 2913, Nov 2003, Pages 343-352


[256] Chong, H. W. & Kwong, Sam, Optimization of Spare Capacity in Survivable WDM Networks, GECCO 2003, pp. 2396-2397


[259] Yanxiang, He & Yifeng, Chen, A GA-based solution to the migration problem of mobile agents in distributed information retrieval systems, 23rd International Conference on Distributed Computing Systems Workshops, Pages 466-471, May 2003


[261] Heegaard, Poul E., Helvik, Bjarne E. & Stol, Norvald, Genetic algorithms for dimensioning of full service access networks, The 14th Nordic Teletraffic Seminar (NTS-14), Copenhagen, Denmark, 18-20 August 1998


Computer Science, Parallel Problem Solving from Nature 7 (PPSN), Volume 2439, Jan 2002, Pages 611 - 620


[312] Zeng, Yong & Wang, Yu-Ping, A new genetic algorithm with local search method for degree-constrained minimum spanning tree problem, Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCI'03), pp. 218-222, 2003


[317] Gaube, Thomas & Rothlauf, Franz, The Link and Node Biased Encoding Revisited: Bias and Adjustment of Parameters, Lecture Notes in Computer Science, Volume 2037, Jan 2001, Page 1


7.2. Only abstract available at VU (IEEE Xplore)


7.3. Only abstract available online


7.4. Not online


## Appendix I.  
Network design papers overview

<table>
<thead>
<tr>
<th>Problem</th>
<th>General</th>
<th>Optical</th>
<th>Radio</th>
<th>Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensioning</td>
<td>[39] [403] [404] [331] [392] [73] [70] [71] [47] [410] [326] [398] [427] [400] [412] [86] [345] [214] [220] [221] [258] [261] [265] [370] [371]</td>
<td>[50] [48] [415] [406] [24] [402] [49] [228] [231] [256] [271] [298]</td>
<td>[326] [398] [427] [400]</td>
<td>[337] [332] [51] [54] [55] [56] [428] [429] [430] [72] [90] [133] [222] [368]</td>
</tr>
<tr>
<td>Node location</td>
<td>[322] [321] [399] [327] [401] [326] [398] [427] [400] [108] [264]</td>
<td>[24] [402] [387] [170] [180] [223] [255]</td>
<td>[36] [37] [328] [388] [282] [326] [398] [427] [400] [107] [347] [172] [173] [174] [175] [176] [188] [189] [193] [194] [362] [217] [237] [238] [279] [289] [293] [346]</td>
<td>[15] [330]</td>
</tr>
<tr>
<td>Topology</td>
<td>[11] [17] [34] [27] [30] [31] [32] [35] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [57] [321] [327] [331] [399] [403] [404] [401] [405] [108] [150] [162] [165] [166] [220] [221] [234] [235] [241] [242] [364] [248] [250] [251] [262] [265] [270] [276] [396] [397] [303] [304] [305] [306] [307]</td>
<td>[48] [49] [50] [387] [406] [227] [231] [244] [246] [367] [271] [272] [22]</td>
<td>[202] [203] [204] [25] [205] [206] [207] [208] [209] [243] [287] [288] [289]</td>
<td>[51] [52] [53] [54] [55] [56] [390] [330] [332] [239] [252] [260] [268] [269] [28]</td>
</tr>
<tr>
<td>Trees</td>
<td>[408] [14] [15] [391] [68] [69] [409] [58] [321] [63] [59] [60] [61] [62] [407] [16] [70] [71] [410] [66] [67] [365] [366] [254] [395] [280] [308] [309] [310] [311] [312] [313] [314] [315] [316] [317] [318] [319] [320] [218] [219] [195] [196] [197] [198] [199]</td>
<td>[24] [402]</td>
<td>[64] [65]</td>
<td></td>
</tr>
<tr>
<td>Routing, restoration &amp; call admission</td>
<td>[411] [73] [74] [75] [413] [414] [76] [77] [66] [67] [416] [403] [331] [404] [333] [392] [78] [79] [412] [88] [89] [100] [102] [110] [343] [114] [344] [115] [132] [249] [136] [137] [138] [140] [141] [142] [143] [151] [152] [154] [157] [159] [160] [179] [192] [257] [263]</td>
<td>[18] [20] [323] [324] [325] [21] [415] [93] [94] [95] [109] [113] [155] [156] [170] [230] [245] [23] [266] [271]</td>
<td>[84] [85] [424] [99] [158] [348] [153] [356] [357] [358] [171] [177] [295]</td>
<td>[72] [417] [418] [425] [337] [338] [426] [339] [51] [88] [91] [92] [97] [116] [117] [118] [119] [120] [139] [168] [216] [222] [236]</td>
</tr>
<tr>
<td>Wavelength allocation &amp; frequency assignment (&amp; setting transmission power)</td>
<td>[18] [19] [20] [323] [324] [21] [229] [325] [93] [94] [95] [109] [113] [230] [23] [266]</td>
<td>[419] [393] [420] [334] [80] [394] [81] [421] [15] [423] [422] [82] [335] [83] [336] [121] [122] [123] [124] [125] [126] [167] [349] [350] [351] [352] [169] [181] [182] [183] [184] [185] [186] [187] [359] [360] [361] [127] [128] [129] [130] [346]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix II. References not used (not online)

This section contains some references that were not discussed in this paper, mainly because they could not be found online.

30. Saha, D., Virtual topology design in all-optical networks using independent sets and genetic algorithm, Proc. Int'l Conf. on Fibre Optics and Photonics (Photonics-98), IIT Delhi, India, Dec. 14-18, 1998 (note the misspelling)
40. Ho, Alex C. H. & Kwong, Sam, Optimization Of CDMA Based Wireless System. GECCO 2002, pp. 1266
46. Jörg Zimmermann , Robin Höns , Heinz Mühlenbein, From theory to practice: an evolutionary algorithm for the antenna placement problem, Advances in evolutionary


