Patterns and Grammars for Transport Network Generation

Basil J. Vitins, IVT, ETH Zurich
Kay W. Axhausen, IVT, ETH Zurich

Conference paper STRC 2010

STRC | **10th Swiss Transport Research Conference**
Monte Verità / Ascona, September 1 - 3, 2010

Patterns and Grammars for Transport Network Generation

Basil J. Vitins
IVT
ETH Zürich
CH-8093 Zürich

Phone: +41 44 633 2702
Fax: +41 44 633 1057
Email: basil.vitins@ivt.baug.ethz.ch

Kay W. Axhausen
IVT
ETH Zürich
CH-8093 Zürich

Phone: +41 44 633 3943
Fax: +41 44 633 1057
Email: axhausen@ivt.baug.ethz.ch

August 2010

Abstract

Generation and optimization of transport networks remain major topics in transport planning, especially in emerging cities, but also in existing transport systems. This article addresses hierarchies, grammars and patterns for transport network generation. Network hierarchies define types of nodes and links. Network grammars describe rules how nodes and links of certain types can be added and joined to each other. Network patterns describe specific alignments of links and nodes which result from grammars and hierarchies. This paper emphasizes the importance of grammar rules for the generation of transport networks using grammar rules. So far, only little research has addressed network hierarchies, grammars and patterns.

This article focuses on optimal transport networks in a general way, based on the analysis of synthetically generated networks. The starting positions of the synthetic networks are featureless planes with given demand generating points. Possible historical network structures are currently ignored. Given a certain node alignment on a featureless plane, links are added or rearranged in order to built-up and optimize the transport network. The objective is to minimize generalized costs, comprising the most relevant components, e.g. travel time and infrastructure costs. A budget constraint limits the level of service and number of links.

Network grammars are compared through analyzing two network states. On one hand, synthetic networks are generated without use of any grammar rule. On the other hand, network grammars are included in the network generation process. Resulting networks are evaluated to compare different grammar rules. Network patterns directly derive from the resulting synthetic networks.

Keywords

Pattern – grammars – hierarchies – network – design – generation – optimization – artificial

1. Introduction

In a long-term perspective, transport networks emerge and develop before reaching the current network state with its specific network characteristics. This paper focuses on the process of network development. The development process ranges from the construction of an entire new district or town and its transport system to the restructuring of an already given transport infrastructure for performance improvements. The research will focus more on emerging networks (network generation) and less on optimization of existing networks by adding new links or improving existing links regarding network generation. Unknown interferences and secondary side-effects cause many difficulties when coping with complex transport networks. Therefore, optimal network generation and improvements of existing networks are a major challenge especially in highly saturated and complex urban areas. For that reason, methodologies and algorithms are very desirable to achieve best possible network performance. Beside network characteristics, other factors influence transport networks. Since network development is related to travel demand and land-use development (e.g. Zöllig and Axhausen, 2010), insights in interactions between network dynamics, land-use and travel behavior are crucial to fully understand transport network development.

Generally speaking, the major question arises on how a network must look like to perform best under given constraints. The aim is to reach a network state that is favorable for transport users and non- users and constructed and maintained with a feasible infrastructure budget. More precisely, generalized costs of network users, accessibility, infrastructure costs, and maintenance costs must be considered to reach an efficient network.

Assessment procedures and methodologies for planning of new transport infrastructures are necessary to meet this complex task. Most conveniently, network characteristics like network grammars, hierarchies and rules are applied through transport planners when building up a new network or complementing an existing network. The present work examines different grammar rules, which are in some parts built up of different hierarchies. Network grammars describe rules how nodes and links of certain types can be added and joined to each other. Network hierarchies define types of nodes and links, e.g. access and trough roads, with specific speed and capacity characteristics, and which can be applied in network grammars. It is crucial to analyze the networks with regard to the differences imposed by grammar rules. So far, only little research has addressed network hierarchies, grammars and patterns in network generation. No quantitative confirmation exists for network hierarchies, grammars or patterns. Research outcomes may contribute to further understanding of patterns and hierarchies and point in this new direction.

The literature review in Section 2 reflects the current state in network generation and artificial networks. Additionally, a brief reference is made to the relevant optimization algorithms. The proposed methodology to generate optimized networks and a network grammar description are described step by step in Section 3. The results are show in Section 4, and major findings are listed. An outlook is given on following-up research steps to improve and supplement the results in Section 5.

2. Literature review

2.1 Network generation

Transport network generation is a widely used expression for different applications. In this work, network generation is the creation of a transport network, without any links and nodes given in advance. In effect, all transport modes can be meant when talking about network generation. No methodology is implied when talking about network generation. In literature, network design is commonly used for similar tasks. The definition of network generation contrasts network improvement algorithms, which assume an existing transport network in advance and improve this network.

Alexander, Ishikawa and Silverstein (1977) and Alexander (1977) address a pattern language to describe an urban development in different scales. Their idea fits at least partially in the network generation context. The proposed pattern language promises an urban environment built-up by a language with different pattern rules. Pattern rules on how an urban area should develop to reach a favorable state are described in their work. The relation to computer science and object-oriented programming is of major relevance because of the language form and language rules. Further on, the rule-based approach can be seen in line also with the building block hypothesis in the genetic algorithm approach (Section 3.3). Building blocks also have to be joined together in a most favorable way to reach the optimum.

Graphical and modeling applications of urban spaces are implemented by Vanegas, Aliaga, Benes and Waddell (2009), inspired by UrbanSim (e.g. Waddell, 2002). They use computer simulations for the generation of transport networks. Transport networks rely partially on initial highway systems and preliminarily given geometric rules. Weber, Müller, Wonka and Gross (2009) simulate an artificial town with geometric rules and network growth but without utility measurement to assess transport system efficiency, similar to Vanegas et al. (2009).

LeBlanc (1975) applies an algorithm based on branch-and-bound technique with an extended calculation of the lower bound because of the Braess paradoxon (Braess, 1968), leading to an optimal network generation solution. Semboloni (2005) introduces a utility-based model for spatial growth, based on micro-economic analysis, and traffic assignment of the all-or-nothing type. Andersson, Lindgren, Rasmussen and White (2002) focus on urban sprawl effects. They establish a microsimulation to show growing urban space. Zöllig and Axhausen (2010) apply a utility maximization approach in an agent based simulation on a small transport network with different degrees of freedom and for a long-term perspective to analyze the shifting sources of user benefits.

Xie and Levinson (2007a,b) contribute to the analysis of road network structures and geometric classifications. Levinson, Xie and Zhu (2006) focus on land-use and network development. Historical approaches try to capture the development paths and decision making to understand the observed network development (Levinson and Karamalaputi, 2003a,b; Xie and Levinson, 2009).

2.2 Artificial networks

Artificial and simplified networks – in contrast to real-world networks – can serve as a tool to simulate and analyse certain network characteristics without relying on case studies. Levinson and Yerra (2006) show self-organizing hierarchical structures arising on road networks from simple rules, by applying a travel demand and assignment model. A grid network of variable size is used for the analysis. Van Nes (2002) established optimal grid sizes, intersections, and stop intervals for public and private transport, respectively. An objective function determines the cost and benefit of different variations of the variables. A theoretical network serves as an underlying basis. Yamins, Rasmussen and Fogel (2003) developed a network generation algorithm based on a grid network and constant land-use pattern. Schäffeler (2004) proposes grid size and network expansion rules for public transportation networks, depending on service characteristics like headway and speed, and service quality. Hierarchies are analyzed concerning travel time reduction accounting for transfer costs. Schweitzer, Ebelin, Rosé and Weiss (1998) optimize a sample network with predetermined intersections and apply a simulated annealing approach jointly with an evolutionary algorithm. Barthelemy and Flammini (2009) established dynamic population densities based on simple road network growth and cost. They simulate dependencies between the road network development and land-use for a population density and rent analysis, by applying different transport costs but without using a utility maximization approach.

Network grammars and network rules are developed based general and theoretical graph theory, e.g. Chomsky (1959), Rozenberg (1997) and Heckel (2006).

2.3 Optimization Algorithm

The generation of a transport network is a highly complex, discontinuous, and NP-hard problem, which means that the problems cannot be precisely solved within polynomially bounded computation time for arbitrarily large networks (e.g. Baaj and Mahmassani, 1991; Hsieh and Liu, 2004; Papadimitriou and Steiglitz, 1982). However, when manipulating the network, the outcome is very difficult to predict (Braess, 1986). For large network sizes and time consuming network evaluations, heuristics and meta-heuristics have gained major influence in network generation and optimization (e.g. Bianchi, Dorigo, Gambardella and

Gutjahr, 2009). Four major heuristics are discussed below. Additionally, two relevant optimization problems are added, which are similar to the network generation problem.

Genetic algorithms belong to the class of evolutionary algorithms and are based on evolutionary mechanisms of genetic reproduction (Goldberg, 1989; Forrest, 1993; Sivanandam and Deepa, 2009). The first genetic algorithm was invented by Holland (1975) for binary solution representation, but meanwhile the concept can be extended to any alphabet. Genetic algorithms are purely stochastic search heuristics. The algorithm optimizes iteratively a set of solutions with genetic operators, i.e. selection, crossover and mutation, for a given fitness function. The method seems promising for network generation purposes and was already applied in different research projects, e.g. Hsieh and Liu (2004); Zhang, Lu and Xiang (2008); Sharma, Ukkusuri and Mathew (2009).

Swarm intelligence techniques also belong to the class of evolutionary algorithms (Dorigo and Stützle, 2004; Merkle, Middendorf and Schmeck, 2002). The methodology is adopted from social insect societies and mirrors self-organizing principles. As an analogy, social behaviour can be used to successfully solve complex computational problems, among others network optimization (Poorzahedy and Abulghasemi, 2005; Vitins and Axhausen, 2009; Yang, Yu and Cheng, 2007). Very efficient transport systems exist in biological system, e.g. mycelia transport systems (Fricker, Lee, Bebber, Tlalka, Hynes, Darrah, Watkinson and Boddy, 2008), slime mold (Tero, Takagi, Saigusa, Ito, Bebber, Fricker, Yumiki, Kobayashi, Nakagaki, 2010). Swarm behavior analogies seem very promising approaches even though one has to distinguish between growing networks and optimal networks of constant size. However, swarm behavior and self-generating path networks can fail in details when implementing them in practice.

Simulated annealing and *Tabu search* are widely applied heuristic in the transportation field. Simulated annealing is a local search method with a non-zero selection probability even if the new combination is “worse” than the previous (Frick, Axhausen, Carle and Wokaun, 2007; Friesz, Cho, Mehta, Tobin and Anandalingam, 1992; Zhao and Zeng, 2008). Tabu search meta-heuristics (Glover, 1989; Fan and Machemehl, 2004; Zhao and Zeng, 2008) can be superimposed on other heuristics. Starting from an initial solution, the algorithm looks for the best solution in its neighbourhood, even under degradation. After moving to the new solution, the alternatives of the previously analysed neighbourhood are stored in a tabu-list for a certain number of iterations.

Pipe routing algorithms find optimal pipe routes in industrial facilities. Ito (1999) uses a potential energy function over the relevant area to simulate constraints like obstacles and to find an optimal pipe routing. Leung (1992) established a facility layout algorithm for optimizing industry processes. From a graph theory perspective, facility layout and pipe routing problems are similar to optimal network generation, because of a predetermined attraction between given “nodes”. The material allocation problem deals with the optimal placement of material in mechanical components under high physical stress (Kaveh, Hassani, Shojaee and Tavakkoli, 2008) and deal with objective functions with high computation time as well. Algorithms may be modified for optimal network generation.

Algorithms to solve the *Euclidean Steiner tree problem* (Robins and Zelikovsky, 2000; Zhou, 2003; Barthélemy and Flammini, 2006) allocates nodes in a network. The Steiner tree problem is related to a more theoretical framework in graph theory. Algorithms connect a set of starting nodes with the use of available free-floating intermediate nodes. The Steiner tree is a minimal spanning tree of a given set of nodes or vertices, using additional intermediate (Steiner) nodes. It represents the lower bound for the minimum infrastructure cost. The manipulation of the intermediate points can be stressed out in the context of network generation as well as the complexity of the Steiner tree problem.

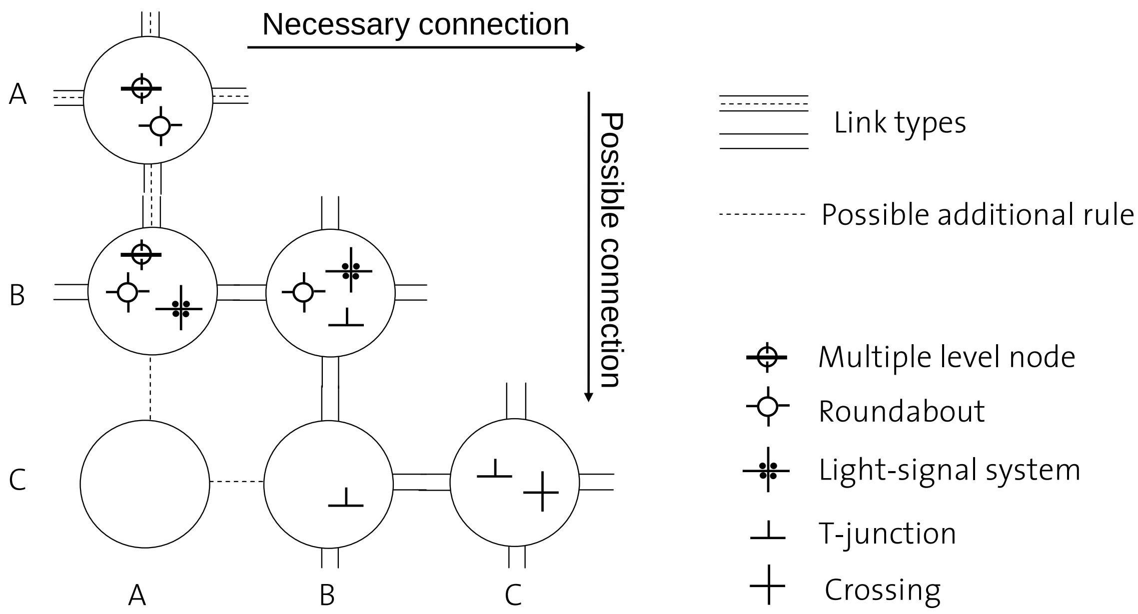
3. Methodology

3.1 Grammars

First, a short definition is given for the most relevant terms. Network hierarchies describe types of nodes and links, e.g. access and trough roads, with certain given characteristics, e.g. speed or capacity. E.g. in a residential areas, links normally have low speed limits and low capacities; as opposed to links in industrial zones or interurban connections with higher speed limits and capacities. Each link attribute, like speed or capacity, can be classified in different ranges. After that, hierarchies can assemble specific ranges of certain attributes. Network grammars describe in the form of rules how nodes and links of certain groups may be joined to each other; for example, if a four-lane road can be crossed by a local access road, or if a round-about can have five or more arms. Network patterns describe specific alignments of network links and nodes which result from grammars and hierarchies. Jacobs (1993) gives on overview of many different pattern types. Very well known pattern shapes are grids, ring or circle roads, centralized star or beam patterns.

This paragraph and Figure 1 refer to Marshall (2005) and the network grammar of adjacent link, which is are utilized throughout the article. Important network grammars are rules for adjacent links. Rules for necessary and possible adjacent link types are shown in Figure 1. In this example, the links can be classified in types A, B or C. Adjacent links have to follow certain rules. A link of type A must be connected at least to one other link of type A. Moreover, it can by connected to a link of type B. A possible additional rule could be that links of type A can also be connected to links of type C. A link of type B has to be connected with links of either type A or type B and may also be connected to links of type C. Additionally, different node types are possible for the connection of two links. The node types are indicated in the large circles.

Figure 1 Possible network grammars with three levels A, B and C.



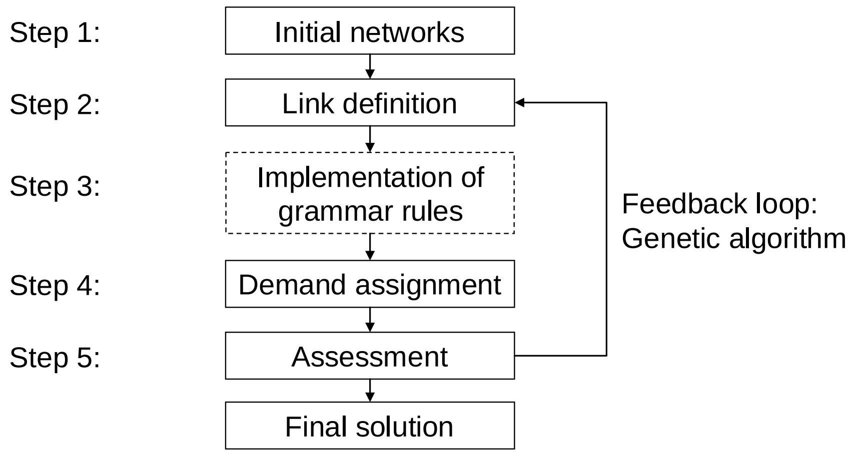
Source: Marshall (2005)

3.2 Overview of the applied methodology

The major goal of this paper is to evaluate network grammars in a quantitative way and independent from specific networks and case studies. When relying on case studies, results can have a bias because of historical development. The aim is to find a methodology not relying on already defined networks. In the proposed approach, optimal networks are generated with and without grammars to evaluate afterwards the difference between the resulting networks. Comparing optimal networks generated with and without grammars is a promising approach to evaluate network grammars. Featureless planes are employed to exclude specific spatial influence.

Figure 2 sketches the overall network generation procedure. The genetic algorithm starts with a population of randomly generated networks, which are improved during the optimization process. The genome of each individual encodes the link allocation between the nodes (step 2). Each bit in the genome represents one undirected link. If a specific bit is on, the corresponding link exists, and vice versa. In step 2, all links are defined in the network. Based on the link definition, possible grammar rules are implemented in the network in step 3. When generating networks without grammar rules, this step is left out. When implementing grammar rules, the relevant grammar rules are implemented here. In step 4, the travel demand is assigned to the network imposed by the genetic algorithm. At the first half of the iterations, only the shortest paths without demand assignment are considered. This is done to reduce calculation time and because the networks are at this point of time not yet efficient (see Figure 9). In step 5, the fitness functions of the individuals are calculated. The genetic algorithm stops when the stop criterion is reached. Otherwise, it returns to the second step, including modification of the individuals through recombination processes of the genetic algorithm (Section 3.3).

Figure 2 Outline of the network generation procedure.



The feedback loop optimizes the link definitions, which encompasses an enormous search space already. Based on the allocation of the links, the grammar rules are applied within the feedback loop. They are applied in an optimal way, so maximum benefit result from the grammar rules. For example, when distributing different hierarchies, higher levels are allocated to links with higher volume.

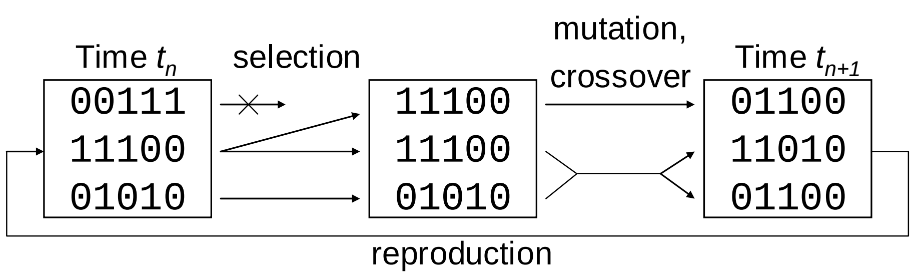
The whole algorithm is implemented in the programming language Java. Class-files from EVA2 (Kronfeld, Planatscher and Zell, 2010) are used for the genetic algorithm.

3.3 Genetic algorithms

This section sketches the idea of genetic algorithms in more details and discusses their major advantages and disadvantages. Genetic algorithms mimic the principle of natural evolution mentioned first by Darwin. A population of individuals improves over generations by selecting the individuals with the highest fitness for each reproduction cycle. Individuals with low fitness values die after the selection process. For that reason, only genomes from individuals with high fitness values are available for the following generation. Reproduction processes recombine the genomes for the sake of better individuals.

An overview of the idea of genetic algorithms is given in Figure 3. Genetic algorithms start with a given randomly generated population. The characteristics of an individual are called phenotype and is derived from its genome, also called genotype. Through recombination of the genomes of different individuals, their genome is manipulated and exchanged so that the genome of the offspring may differ from its parents. The recombination consist mostly of mutation and crossover processes. Crossover assembles the genome of the parents according to different rules. N-Point crossovers, which is widely used, randomly defines n bits in the genome and exchange the strings in-between the defined bits according to a certain probability p_c. Mutation processes mutate randomly chosen bits or areas within the genome according to a certain probability p_m. A selection mechanism chooses the best individuals for reproduction and thus the offspring. The fitness function is responsible for the evaluation of each individual. Repeating the process of reproduction may subsequently lead to the most successful population with individuals with the best fitness values. In our context, each individual represents a network and the genome is responsible for the link allocation. Specifications regarding the parameter settings are listed in Section 4.2. Due to the large amount of literature, the explanation of genetic algorithm is kept scarce, the reader is referred to Goldberg (1989), Forrest (1993), Sivanandam and Deepa (2009) for further information.

Figure 3 Overview genetic algorithm.



Source: Bäck and Hoffmeister (1994) p.872

Like in other evolutionary algorithms, parameter settings are crucial for a successful convergence rate and a high chance to reach the global optimum. Goldberg (2002) summarizes in his book major experiences and suggestions, which some of them are summarized in the following paragraph.

Genetic algorithms follow rather simple statistic rules. Selection, mutation and crossover are processes, which can be statistically analyzed with respect to their chance to succeed when applying them in a genetic algorithm. It is possible to predict convergence time and population size according to the statistics. Selection, mutation and crossover are processes which contain certain parameters like selection rate, mutation rate and number of crossovers. With statistical methods, calibration can be reduced to the most relevant parameters. When calibration is done correctly, the chance rises for successful convergence behavior and successful reduction of the variances in the results.

A driver for convergence behavior is assumed to be the principle of building blocks. Building blocks are units within the genome that are different from the rest and contribute significantly to the fitness value. Successful building blocks can take over in the population when the number of the building blocks is large enough. So starting with a large population guarantees the existence of the correct building blocks. On the other side, mutation likely destroys building blocks. There is a very low chance that mutation is generating a new and improved solution during an optimization process. Therefore, mutation is skipped in this work. Elitism is a method to always keep the best *n* individuals as part of the offspring. Elitism is also left out due to the fact that it is not necessary when parameters are set well at the beginning. Thus, crossover and selection remain the major processes. Both processes must be adjusted to each other in order to enlarge building block populations and reduce the danger of genetic shift. The results of the calibration process for crossover rate (p_c) and selection pressure (s) is shown in Section 4.2.

Especially the reliability in different network settings and the convergence behavior can be underpinned in this work. According to Goldberg (2002), the genetic algorithm is mostly reliable when applying the algorithm in similar settings, mainly because of its statistical nature.

3.4 Objective function

In order to apply the genetic algorithm as the main optimization methodology the objective function has to be calculated automatically based on the network outcome. In this work, this is done either based on the sum of the demand-weighted travel times ((F 1) or on the accessibility (F 2) (Fröhlich, Tschopp and Axhausen, 2005).

(F 1) $$\text{Travel time} = \sum_{o=1}^{O}\sum_{d=1}^{D} t_{od} \cdot d_{od} + P$$

with o: Origin node

d: Destination node

t_{od}: Travel time between o and d.

d: Demand between o and d.

P: Penalty for infrastructure cost exceeding

(F 2) $$\text{Accessibility} = \sum_{o=1}^{O} I_o \cdot \ln\left(\sum_{d=1}^{D} A_d \cdot \exp(-\beta \cdot t_{od})\right) - P$$

with o: Origin node

d: Destination node

I_o: Inhabitants at node o.

A_d: Attractiveness of node d, approximated by number of inhabitants and jobs.
t_{od}: Travel time between o and d.
β: Distance parameter (set to 0.2 according to Fröhlich, Tschopp and Axhausen, 2005).
P: Penalty for infrastructure cost exceeding

A budget penalty is introduced (P) when the preliminarily given length of the infrastructure is exceeded. The penalty is proportional to the length of exceedance. In accessibility measure in equation F 2, demand is not taken into account. However, network loading is indirectly reflected by travel time, since higher link loadings lead to longer travel times. Both objective functions were applied for reason of comparison.

3.5 Travel time and demand assignment

For the evaluation of the objective function, travel times need to be calculated for every OD-pair. Different procedures are possible, each with different computational time requirements.

A micro simulation of the traffic flow is not possible here because of the long calculation times. As another alternative, travel times on links can be estimated according to spatial data like population, job and network density in the vicinity. Hackney, Bernard, Bindra and Axhausen (2007) estimated parameters for travel speed, derived from floating car data in Zurich. Parameters may have to be adopted for considerably different environments, e.g. other land use, densities or road network characteristics. Using parameters from Hackney et al. (2007) requires two assignment procedures, which is reducing computation time only modestly and simultaneously relay on an approximation.

In this work, the static demand assignment (Ortuzar and Willumsen, 2001) is applied. It also needs some computation time scaling with the network size. The static assignment implemented in this work applies the Dijkstra algorithm (Dijkstra, 1959) and the MSA algorithm (Ortuzar and Willumsen, 2001). The Frank-and-Wolfe algorithm (Frank and Wolfe, 1956) or the A*-algorithm (Hart, Nilsson and Rafael, 1968) could be used to additionally reduce computation time. For calculating the travel time on each link, the CPR – function is used with $t_{act} = t_{freeflow} \cdot \left(1 + \left(l/c\right)^4\right)$ where l is the actual loading, and c is the link capacity.

3.6 Further assumptions

Currently, the nodes can not move on the featureless plane. This assumption comes from the fact that streets are defined as links between given nodes. Variable nodes and links at the same time would tremendously increase the degrees of freedom and the computation time, respectively. The optimal arrangement of the nodes is, therefore, considered and will be addressed in future work.

Given that all nodes n are fixed, the size of the search room applying bidirectional links is $2^{0.5 \cdot |n||n-1|}$. A middle sized Swiss city, e.g. Winterthur with 100'000 inhabitants and about 400 nodes in the major road network would encompass above $10^{24'022}$ possibilities. One possibility to reduce the search space is the limitation of over- and underpasses. In reality, except for highway systems, networks operate on one level and the number of over- or underpasses is small. Therefore, it is an acceptable simplification to consider in the following only planar graphs. The new search space then has approximately the size of 10^{446}.

3.7 Current network settings

The current network setting, which is used in this work, is displayed in Figure 4; further specifications are listed in Table 1. The algorithm was successfully tested on other networks with different amount of nodes as well. Due to lack of space, this paper only refers to the setting in Figure 4. The nodes or intersections are allocated in a square. Each node can be connected only to the surrounding nodes, i.e. each node has maximum 8 neighbors if it does not lie on the periphery of the network.

Figure 4 Current preliminarily give network settings.

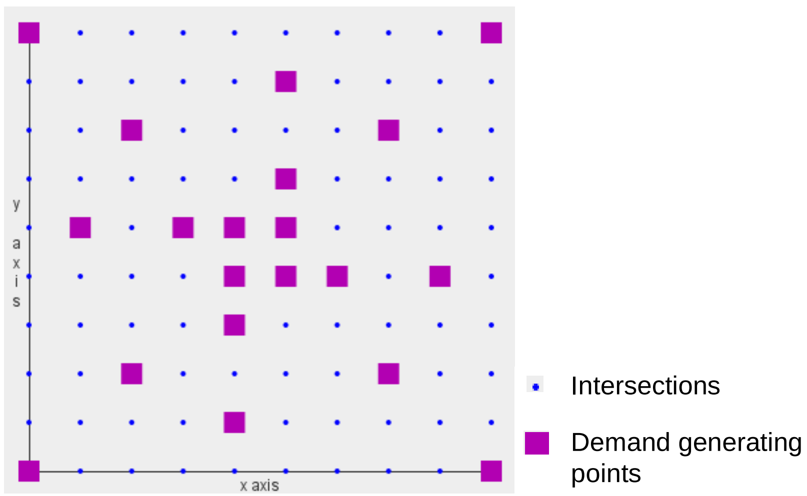


Table 1 The outcome of the objective function applying different grammar rules.

Specifications	
Number of nodes	100
Number of possible links	342
Number of centroids	20
Genome size [bits]	342
Number of possible link alignments	10^{103}
Total demand to be assigned [vehicles]	2’400

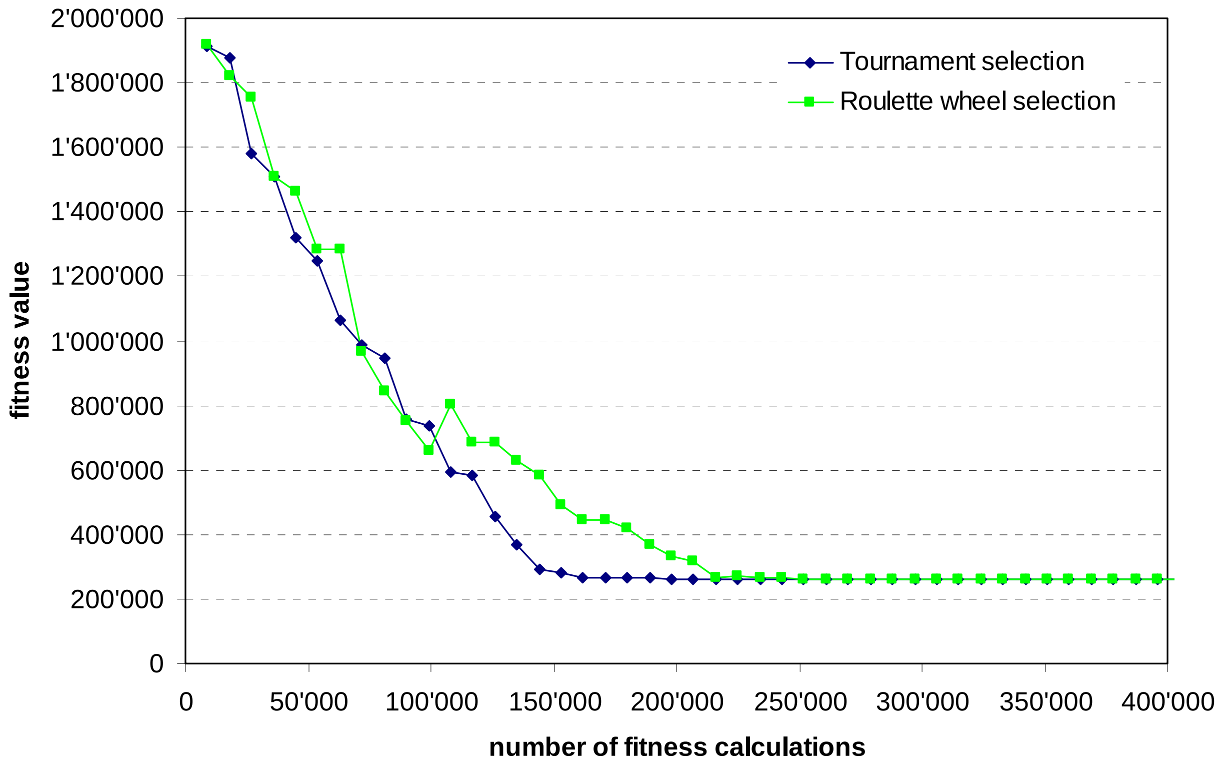
4. Results

This section presents the results of the scenario runs. It first focuses on the algorithm and its calibration (Section 4.1 and 4.2) and afterwards on the grammar results (Section 4.3 and 4.4). The simulation results show that the genetic algorithm is very reliable regarding the network generation. This is shown especially in the convergence behavior, the visualization and analysis of the resulting networks. This cannot be known in advance since there is a large variance in success between different kinds of problems when applying genetic algorithms (Goldberg, 2002). Success of the algorithm can have various reasons. It can be simply because the basically discrete genetic algorithm is very well suited for discrete network generation problems or due to the calibration of the different parameters and processes of the genetic algorithm.

4.1 Selection settings

Many different methods for crossover and selection processes are available in literature. Some of them are most likely not suitable for network generation purposes. In this section, two selection procedures, which are most promising, are compared for reliability and convergence behavior. Tournament selection chooses the best individual from a randomly selected subset of the current population. In roulette-wheel selection, each individual is assigned to a segment of a virtual roulette wheel. The segment size is proportional to the fitness value of the individual. The individuals are then chosen according to the probability, which is proportional to the segment size. The results of the convergence processes are shown in Figure 5. The results show that the tournament selection has a more straightforward convergence behavior, compared to the roulette-wheel selection. Additionally, the tournament selection uses less fitness function evaluations. That is the reason why tournament selection is used in the following.

Figure 5 Best fitness values of tournament and roulette-wheel selection.



4.2 Parameter settings

Like in other evolutionary algorithms, parameter settings are crucial for a successful convergence rate of genetic algorithms and a low risk to reach a local optimum. The following parameter calibration mainly follows the idea of Goldberg (2002). This methodology is comparable with fractional factorial design approaches for independent variables, but better suited for genetic algorithms.

Crossover and selection processes must be adjusted to each other in order to enlarge building block populations and reduce the danger of genetic shift (see Section 3.3 for more details). Figure 6 and Figure 7 show the $p_c - s$ plots, as proposed in Section 3.3. The x- and x- axes show the crossover rate p_c and selection pressure s or tournament size, respectively. The vertical z-axis shows the fitness function of the corresponding networks. The tested values are for p_c = {0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8} and for s = {2, 3, 5, 10, 20}. A simple 5 x 5 grid is taken as a basis. Three optimization processes were conducted for each parameter setting.

The result shows that depending on the crossover rate and the selection pressure, the algorithm performs significantly better. Remarkably, different parameter settings are tested in larger networks, but the parameter values remain stable. Parameters for crossover (= 0.9) and for selection pressure (possible range between 2 and 5) are successfully integrated in other network sizes and specifications and are used in the following.

Figure 6 p_c - s control map with 1-point crossover.

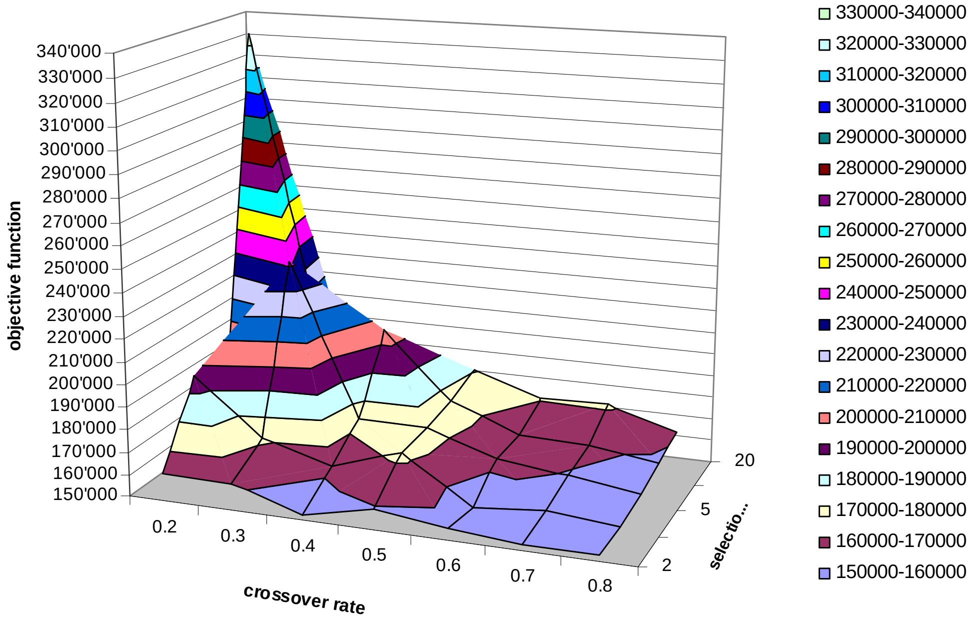
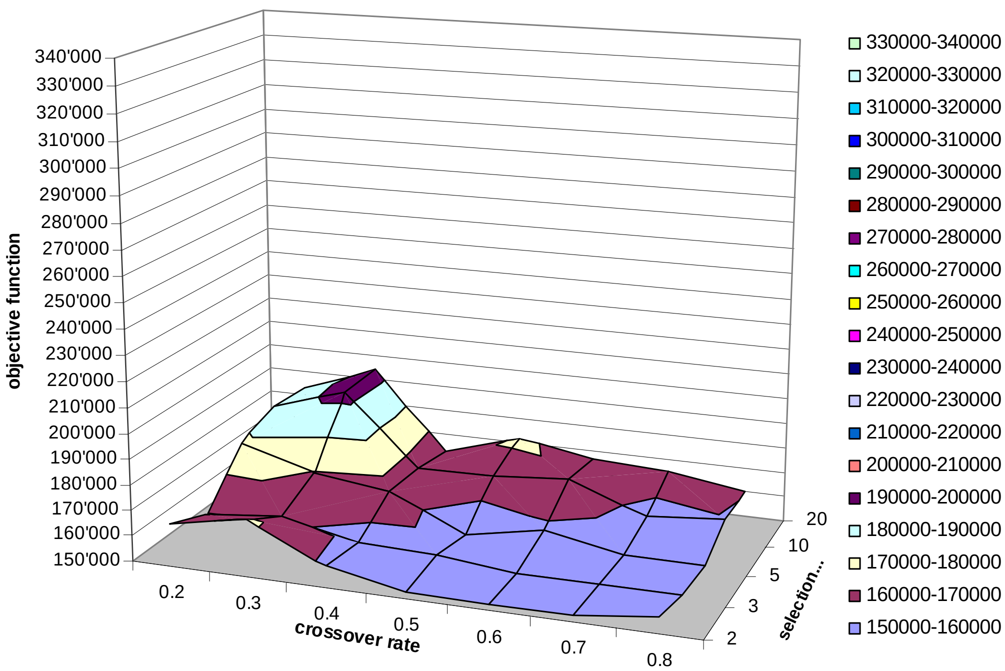


Figure 7 p_c - s control map with 2-point crossover.



4.3 Optimization procedure and outcome

In this section, the optimization process and outcome are explained to make the reader more familiar with the artificial planes, the settings chosen and the generated outcome. In a standard outcome plot presented here, nodes or intersections are shown as small dots and demand generating points or centroids are shown as larger squares (see Section 3.7 for network specifications). The amount of transport demand is given in such a way congestion effects occur in the network. The demand is distributed equally between the centroids. The size of each centroid is proportional to the transport demand.

Figure 8 shows a network of 100 nodes and 20 centroids. Links are shown as green lines. The width of the links is either proportional to the capacity or to the loading, as indicated in the title of the figures. Figure 9 shows the best fitness values during the optimization process, which is conducted with the grammar rule “links joined to links with the same or any higher hierarchy”. The best fitness value is 78’643, similar to the result in Table 3. The optimization process is divided in two parts. During the first part, the genetic algorithm only calculates the shortest path without travel demand to decrease computation time. In the first periode, the networks are not optimized yet and detailed demand assignment is not as relevant as the network structures. The full equilibrium is calculated during the second half of the iterations. More details about the current networks are listed in Section 3.7.

Figure 8a, b: Network with capacity (left) and network loading (middle).

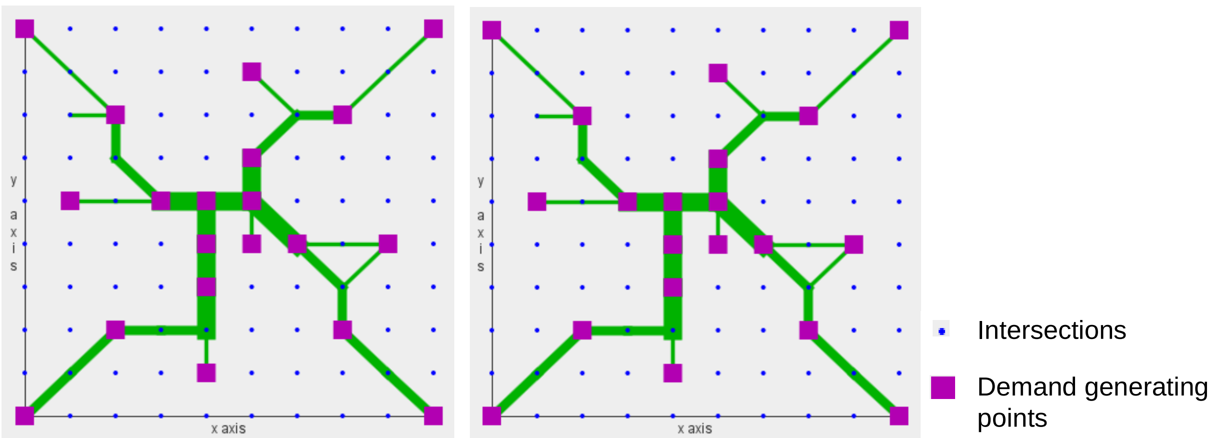
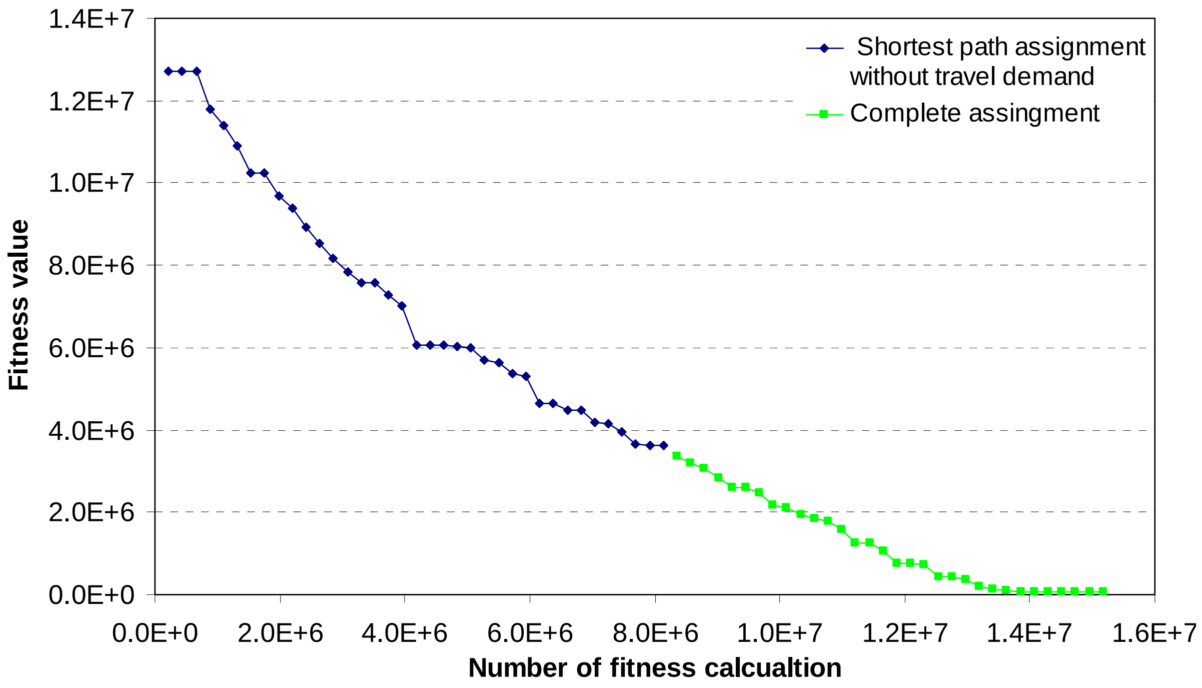


Figure 9: The best fitness value for each generation during an optimization process.



Remarkably, the optimization process seems to have a straightforward direction without any bumps and almost no asymptotic convergence behavior at the end, even though elitism is left out. This behavior is observed in all optimization processes of different network sizes. The results are in line with the statistic nature of the algorithm described in Section 3.3. Thus, the results also allow determining a clear cut-off criterion for the algorithm, which is not implemented yet.

Calculation time depends of course on the network size. An optimization process for a network of 100 nodes including the implementation of grammar rules takes about 3 hours. For the calculations, four quad-core processors are used with 8'380 CPUs. Implemented grammar rules currently increase the calculation time by a factor 2 to 3. Thus, besides the demand assignment, also grammar rules play a significant role in calculation time. Especially in larger networks (100 nodes or more), this is an important factor. Genetic algorithms are well suited for multi-threading, since calculation time is almost proportional to the number of threads. Due to the recent development in computer science that focuses on multiple cores instead of faster single cores, this will remain an advantage in the future.

4.4 Adjacent link rule

This work is reduced for link hierarchies and pattern analysis. Other grammars like node types will follow in further research work. The adjacent link rule is one of Marshall's rules to compose a network and is a remarkably strong rule (Marshall, 2005, or Section 3.1). For that, the total link length of each hierarchy, used in the network, has to be determined. Either a certain cost function for each link type is defined in advance to optimize the network according to this function, or, as a more straightforward approach, by defining the ratio of different link types. In the following, the latter approach is used for simplicity (Table 2, right column). The capacity and speed levels for the different hierarchies (Table 2) are directly determined in the MSA-algorithm and conducted after each iteration during the first four iterations. After that, it is assumed that the capacity settings do not significantly change anymore and the time-consuming capacity determination is not necessary.

Table 2 Grammar rule currently applied.

Hierarchy level	Capacity [vehicles / unit of time]	Speed [unit of length / unit of time]	fraction
Level 1	300	1	2
Level 2	700	2	2
Level 3	1’400	3	1

The comparison of the fitness values for the different grammar rules is shown in Table 3. The first column describes the grammar rule. The second column lists the fitness values without penalty factor, using the demand-weighted travel times (F 1). The third column lists the outcome when the network is optimized according to the accessibility measure (F 2). The penalty factor is left out to get an impression of the performance of the resulting network. The preliminary given maximum length is practically not exceeded in all networks.

Table 3 The outcome of the objective function applying different grammar rules.

Grammar rule	Weighted travel time:	Accessibility:
No grammar rule applied during optimization	78’003	34.7
Links joined to links with the same or any higher hierarchy	78’682	35.9
Links joined to links with the same or 1 higher hierarchy	79’174	35.9
Links joined to links with the same or 2 higher hierarchies	79’801	35.0

Table 3 shows that with respect to travel times, the best outcome is achieved when no grammar rule is applied during network generation. Applying accessibility optimization, the rule “links joined to links with the same or any higher hierarchy” performs well, and the rule “links joined to links with the same or 1 higher hierarchy” also performs well. Remarkably, optimization including demand-weighted travel time and accessibility do not lead to the same result. Nevertheless, the differences between the different grammar rules are not large. Therefore, three further tests are necessary: First, the adjacent link rules should be coupled with other rules, e.g. node differentiation, with fractional factorial design. Second, the variety of the outcome for each grammar rule should be revised as well. Third, testing on other network settings is necessary.

The runs with the demand-weighted travel times as an objective function are visualized in Figure 10 to Figure 13. The figures obtained with accessibility optimization are left out due to lack of space and similarity reasons.

Figure 10 Outcome without adjacent link rule, capacity (left) and loading (right).

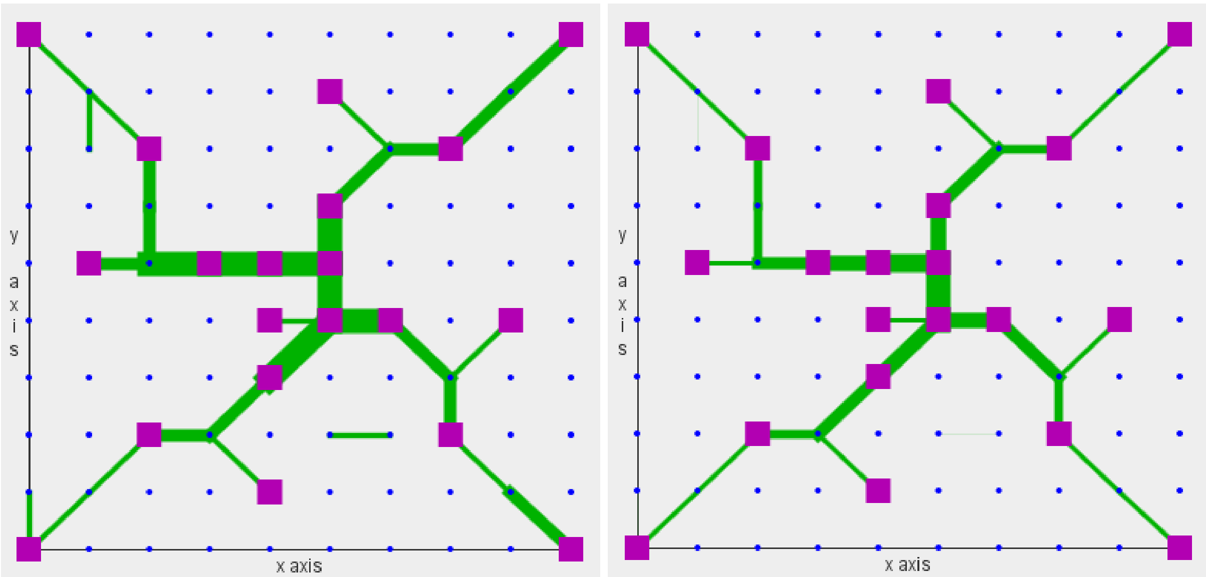


Figure 11 Outcome with adjacent link to links with same or any higher hierarchy, capacity (left) and loading (right).

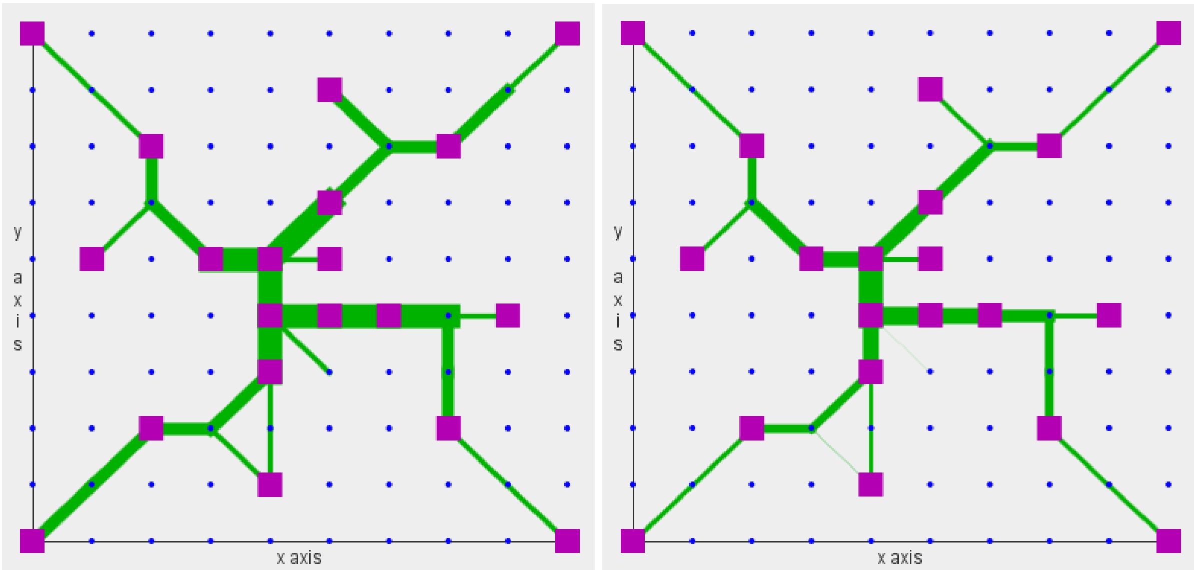


Figure 12 Outcome with adjacent link to links with same or 1 higher hierarchy, capacity (left) and loading (right).

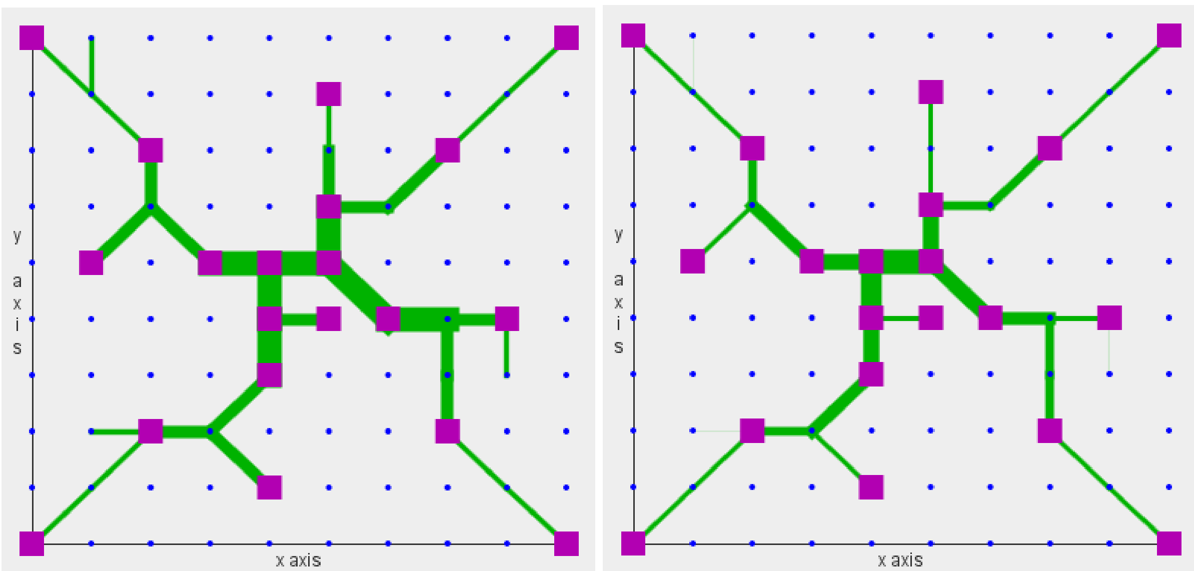
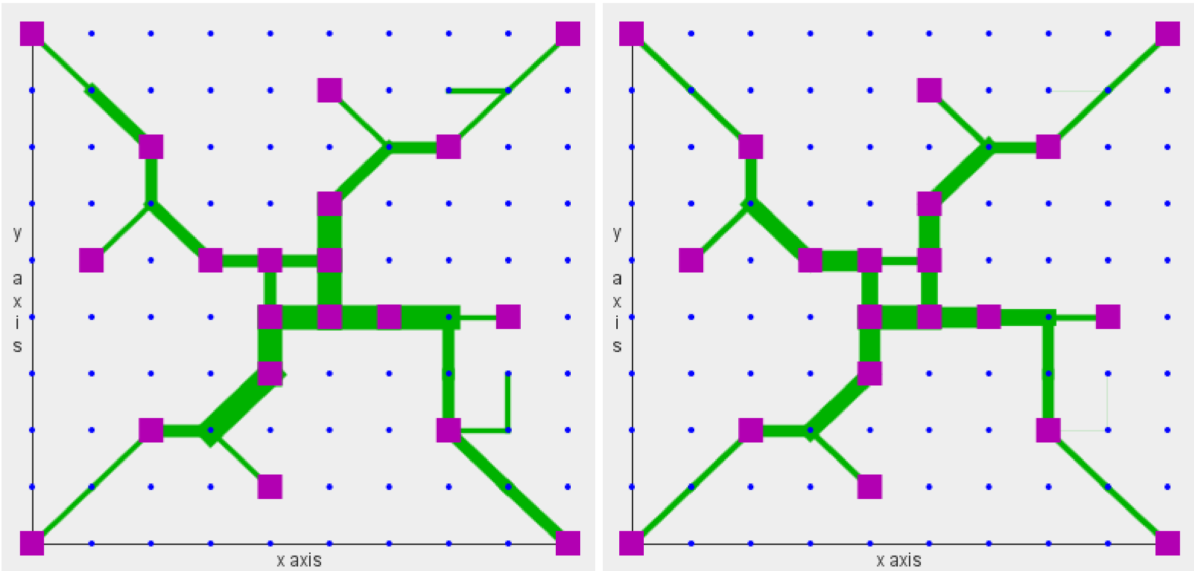


Figure 13 Outcome with adjacent link to links with same or 2 higher hierarchies, capacity (left) and loading (right).



5. Conclusion and outlook

The general idea of comparing networks generated with and without grammar rules seems to be a promising approach. The method of combining network generation with a genetic algorithm works well. There is a strong evidence for reliable convergence behavior when looking at the results and the convergence characteristics. Moreover, the implementation of hierarchies in the network works well.

Additional calculation outcome has to be generated for further evidence. The results should be compared with other network settings and demand matrices. In the near future, an implementation of grammar rules for different node types and their number of arms will follow. Link length grammar rules will be tested when variable node positions are included in the algorithm. Additionally, also floating nodes should be considered to relax the grid form.

A major remaining problem is the high number of iterations and the time consuming evaluation. Moreover, applying grammar rules requires a lot of calculation time especially in larger networks compared to the demand assignment. Acceleration only can be realized with less precision. Partitioning of the network in sub-network (Barreiros, 2003) can also be a solution for accelerating the algorithm, but it increases the chance of local minimum.

6. References

Alexander, C. (1977) *The Timeless Way of Building*, Oxford University Press, New York.

Alexander, C., S. Ishikawa and M. Silverstein (1977) *A Pattern Language*, Oxford University Press, New York.

Andersson, C., K. Lindgren, S. Rasmussen and R. White (2002) Urban growth simulation from „first principles“, *Physical Review*, **66** (2) 1-9.

Baaj, M.H. and H.S. Mahmassani (1995) Hybrid route generation heuristic algorithm for the design of transit networks, *Transportation Research Part C*, **3** (1) 31-50.

Bäck, T. and F. Hoffmeister (1994) Basic aspects of evolution strategies, *Statistics and Computing*, **4** (2) 51-63.

Barreiros, J. (2003) An Hierarchic Genetic Algorithm for Computing (near) Optimal Euclidean Steiner Tree, *Procceedings of the Genetic and Evolutionary Computation - GECCO*, Springer-Verlag, Chicago, July 2003.

Barthélemy, M. and A. Flammini (2006) Optimal traffic networks, *Journal of Statistical Mechanics: Theory and Experiment*, **3** (7) 1-8.

Barthélemy, M. and A. Flammini (2009) Co-evolution of density and topology in a simple model of city formation, *Networks and Spatial Economics*, **9** (3) 401-425.

Bianchi, L., M. Dorigo, L.M. Gambardella and W.J. Gutjahr (2009) A survey on metaheuristics for stochastic combinatorial optimization, *Natural Computing*, **8** (2) 239-287.

Braess, D. (1968) Über ein Paradoxon aus der Verkehrsplanung, *Unternehmensforschung*, **12**, 258-268.

Cantarella, G.E. and A. Vitetta (2006) The multi-criteria road network design problem in an urban area, *Transportation*, **33** (6) 567-588.

Chomsky, N. (1959) On certain formal properties of grammars, *Information and Control*, **2** (2) 137-167.

Dijkstra, E.W. (1959) A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik*, **1** (1) 269-271.

Dorigo, M. and T. Stützle (2004) *Ant Colony Optimization*, MIT Press, Boston (www.aco-metaheuristic.org).

Fan, W. and R.B. Machemehl (2004) A tabu search based heuristic method for the transit route network design problem, *Proceedings of the 2004 Conference on Computer-Aided Scheduling of Public Transport (CASPT)*, San Diego, August 2004.

Forrest, S. (1993) Genetic algorithms: Principles of natural selection applied to computation, *Science*, **261** (5123) 872-878.

Frank, M. and P. Wolfe (1956) An algorithm for quadratic programming, *Naval Research Logistics Quarterly,* **3** (1-2), 95-110.

Frick, M., K.W. Axhausen, G. Carle and A. Wokaun (2007) Optimization of the distribution of compressed natural gas (CNG) refueling stations: Swiss case studies, *Transportation Research Part D*, **12** (1) 10-22.

Fricker, M.D., J.A. Lee, D.P. Bebber, M. Tlalka, J. Hynes, P.R. Darrah, S.C. Watkinson and L. Boddy (2008) Imaging complex nutrient dynamics in mycelial networks, *Journal of Microscopy*, **231** (2) 317-331.

Friez, T. L., H.-J. Cho, N.J. Mehta, R.L. Tobin and G. Anandalingam (1992) A simulated annealing approach to the network design problem with variational inequity constraints, *Transportation Science*, **26** (1) 18-26.

Fröhlich, Ph., M. Tschopp und K.W. Axhausen (2005): Entwicklung der Erreichbarkeit der Schweizer Gemeinden: 1950-2000, *Raumforschung und Raumordnung,* **64** (6) 385-399.

Glover, G. (1989) Tabu search – part 1, *Journal on Computing*, **1** (3) 190-206.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.

Goldberg, D.E. (2002) *The Design of Innovation*, Kluwer, Dordrecht.

Guihaire, V. and J.-K. Hao (2008) Transit network design and scheduling: A global review, *Transportation Research Part A*, **42** (10) 1251-1273.

Hackney, J.K., M. Bernard, S. Bindra and K.W. Axhausen (2007) Predicting road system speeds using spatial structure variables and network characteristics, *Journal of Geographical Systems*, **9** (4) 397-417.

Hart, P.E., N.J. Nilsson and B. Raphael (1968) A formal basis for the heuristic determination of minimum cost paths, *IEEE Transaction on System Science and Cybernetics*, **4** (2), 100-107.

Heckel, R. (2006) Graph transformation in a nutshell, *Electronic Notes in Theoretical Computer Science*, **148** (1) 187-198.

Holland, J. (1975) *Adaption in Natural and Articial Systems: An Introductory Analysis with Applications to Biology, Control and Articial Systems*, The University Press of Michigan Press, Ann Arbor.

Hsieh, T.Y., and H.L. Liu (2004) Genetic algorithm for optimization of infrastructure investment under time-resource constraints, *Computer-Aided Civil and Infrastructure Engineering*, **19** (3) 203-212.

Ito, T. (1999) A genetic algorithm approach to piping route path planning, *Journal of Intelligent Manufacturing*, **10** (1) 103-114.

Jacobs, A.B. (1993) *Great Streets*, The MIT Press, Cambridge.

Kaveh, A., B. Hassani, S. Shojae and S.M. Tavakkoli (2008) Structural topology optimization using ant colony methodology, *Engineering Structures,* **30** (9) 2559-2565.

Kronfeld, M., H. Planatscher and A. Zell (2010) The EvA2 Optimization Framework, *Proceedings of the Learning and Intelligent Optimization Conference,* Venice, January 2010.

LeBlanc, L. (1975) An algorithm for the discrete network design problem, *Transportation Science,* **9** (3) 183-199.

Leung, J. (1992) A new graph-theoretic heuristic for facility layout, *Management Science,* **38** (4) 594-605.

Levinson, D. and R. Karamalaputi (2003a) Induced supply, a model of highway network expansion at the microscopic level, *Journal of Transport Economics and Policy*, **37** (3) 297-318.

Levinson, D. and R. Karamalaputi (2003b) Predicting the construction of new highway links, *Journal of Transportation and Statistics*, **6** (2/3) 1-9.

Levinson, D., F. Xie and S. Zhu (2006) The co-evolution of land use and road networks, presented at the *53rd North American Conference of Regional Science Association International*, Toronto, November 2006.

Levinson, D. and B.M. Yerra (2006) Self-organization of surface transportation networks, *Transportation Science*, **40** (2) 179-188.

Marshall, S. (2005) *Streets & Patterns*, Spon Press, London.

Merkle, D., M. Middendorf and H. Schmeck (2002) Ant colony optimization for resource-constrained project scheduling, *Transactions on Evolutionary Computation*, **6** (4) 333-346.

Ortuzar, J.D. and L.G. Willumsen (2001) *Modelling Transport*, Wiley, New York.

Papadimitriou, C.H. and K. Steiglitz (1982) *Combinatorial Optimization*, Printice-Hall, Englewood Cliffs.

Poorzahedy, H. and F. Abulghasemi (2005) Application of ant system to network design problem, *Transportation*, **32** (3) 251-273.

Robins, G. and A. Zelikovsky (2000) Improved steiner tree approximaiton in graphs, Proceedings of the *11th annual ACM-SIAM symposium on Discrete Algorithms*, San Francisco, USA.

Rozenberg, G. (1997) *Handbook of Graph Grammars and Computing by Graph Transformation*, World Scientific Publishing, Singapore.

Schäffeler, U. (2004) Netzgestaltungsgrundsätze für den öffentlichen Personennahverkehr in Verdichtungsräumen, *Dissertation*, Institut für Verkehrsplanung und Transportsysteme ETH Zürich, Zürich.

Schweitzer, F., W. Ebeling, H. Rosé and O. Weiss (1998) Optimization of road networks using evolutionary strategies, *Evolutionary Computation*, **5** (4) 419-438.

Semboloni, F. (2005) Multi-agents simulation of urban dynamic, *Proceedings of the 10th Computers in Urban Planning and Urban Management Conference*, Iguassu Falls, Brazil.

Sharma, S., S.V. Ukkusuri and T.V. Mathew (2009) Pareto optimal multiobjective optimization for robust transportation network design problem, *Transportation Research Record*, **2090**, 95-104.

Sivanandam, S.N. and S.N. Deepa (2009) *Introduction to Genetic Algorithms*, Springer, Berlin.

Streichert, F. and H. Ulmer (2005) JavaEvA, A Java based framework for Evolutionary Algorithms, *Technical Report*, **WSI-2005-06**, ZBIT, Eberhard-Karls-University, Tübingen.

Tero, A., S. Takagi, T. Saigusa, K. Ito, D.P. Bebber, M.D. Fricker, K. Yumiki, R. Kobayashi, T. Nakagaki (2010) Rules for biologically inspired adaptive network design, *Science*, **327** (5964) 439-442.

Vanegas, C.A., D.G. Aliaga, B. Benes and P.A. Waddell (2009) Interactive design of urban spaces using geometrical and behavioral modeling, *ACM Transactions on Graphics*, **28** (5) 1-10.

van Nes, R. (2002) *Design of Multimodal Transport Networks*, DUP Science, Delft.

Vitins, B.J. and K.W. Axhausen (2009) Optimization of large transport networks using the ant colony heuristic, *Computer-Aided Civil and Infrastructure Engineering,* **24** (1) 1-14.

Waddell, P.A. (2002) UrbanSim: Modeling urban development for land ese, transportation and environmental planning, *Journal of the American Planning Association*, **68** (3) 297-314.

Weber, B., P. Müller, P. Wonka and M. Gross (2009) Interactive geometric simulation of 4D Cities, *Eurographics*, **28** (2) 481-492.

Xie, F. and D. Levinson (2007a) Measuring the structure of road networks, *Geographical Analysis*, **39** (3) 336-356.

Xie, F. and D. Levinson (2007b) The topological evolution of road networks, paper presented at the *87th Annual Meeting of the Transportation Research Board,* Washington D.C., January 2007.

Xie, F. and D. Levinson (2009) Modeling the growth of transportation networks: A comprehensive review, *Networks and Spatial Economics*, **9** (3) 291-307.

Yamins, D., S. Rasmussen and D. Fogel (2003) Growing urban roads, *Network and Spatial Economics*, **3** (1) 69-85.

Yang, Z., B. Yu and C. Cheng (2007) A parallel ant colony algorithm for bus network optimization, *Computer-Aided Civil and Infrastructure Engingeering*, **22** (1) 44-55.

Zhang, G., J. Lu and Q. Yiang (2008) Application of genetic algorithm to network design problem, *Proceedings of the International Conference on Intelligent Computation Technology and Automation*, Changsha, October 2008.

Zhao, F. and X. Zeng (2006) Simulated annealing-genetic algorithm for transit network optimization, *Journal of Computing in Civil Engineering*, **20** (1) 57-68.

Zhao, F. and X. Zeng (2008) Optimization of transit route network, vehicle headways and timetables for large-scale transit networks, *European Journal of Operational Research*, **186** (2) 841-855.

Zhou, H. (2003) Efficient Steiner Tree Construction Based on Spanning Graphs, *International Symposium on Physical Design*, Monterey, April 2003.

Zöllig, C, and K.W. Axhausen (2010) Calculating benefits of infrastructural investment, *Working Paper*, **612**, IVT, ETH Zurich, Zurich.