Helsinki University of Technology
Systems Analysis Laboratory Research Reports E14, May 2003

MODELS AND ALGORITHMS FOR NETWORK PLANNING TOOLS - PRACTICAL EXPERIENCES

Jukka K. Nurminen

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Distribution:
Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100
FIN-02015 HUT, FINLAND
Tel. +358-9-451 3056
Fax +358-9-451 3096
systems.analysis@hut.fi

This report is downloadable at
www.e-reports.sal.hut.fi/pdf/E14.pdf

Series E – Electronic Reports
www.e-reports.sal.hut.fi

ISBN 951-22-6542-7
ISSN 1456-5218

Title: Models and Algorithms for Network Planning Tools - Practical Experiences

Author: Jukka K. Nurminen
Nokia Research Center
P.O.Box 407, 00045 Nokia Group, FINLAND
jukka.k.nurminen@nokia.com

Date: May 2003

Status: Systems Analysis Laboratory Research Reports E14 May 2003

Abstract: In this paper we consider different types of models and algorithms for network planning tools, in particular, from the point of view of the user needs. We discuss issues such as the significance of full optimality, adaptation to changes, usability of the models, and the model and tool development effort. Different approaches, mathematical programming, stochastic algorithms, functional modelling, and the use of special purpose algorithms are discussed from the perspective of how well they serve the various user needs. Based on our tool development experience, a practical incremental development scenario is proposed, and the role of the different modelling approaches in the scenario is discussed.

Keywords: practise of OR, telecommunications, planning, decision support systems, networks and graphs

Models and Algorithms for Network Planning Tools - Practical Experiences

Jukka K. Nurminen

May 6, 2003

Abstract:
In this paper we consider different types of models and algorithms for network planning tools, in particular, from the point of view of the user needs. We discuss issues such as the significance of full optimality, adaptation to changes, usability of the models, and the model and tool development effort. Different approaches, mathematical programming, stochastic algorithms, functional modelling, and the use of special purpose algorithms are discussed from the perspective of how well they serve the various user needs. Based on our tool development experience, a practical incremental development scenario is proposed, and the role of the different modelling approaches in the scenario is discussed.

Keywords:
practice of OR, telecommunications, planning, decision support systems, networks and graphs

1 Introduction

Planning tools and other decision support systems are one answer to the continuous need for more efficient and economic operations. Key components of the planning tools are quantitative models, which can both suggest planning decisions and assess their effect.

In this paper we focus on the operational and tactical planning level. Some characteristics of tool development in this area are:

- Instead of making a single decision of high economical or other value, the user of the planning tool is making a lot of small decisions
- It is not possible to tailor the models for each planning case. The users have to survive with the general models as they are in each release of the tool
- The goal is planning software for everyday use – not making a single difficult decision in a complex setting

The experiences discussed in this paper are derived mainly from the development of the NPS/10 telecom network-planning tool [1] that was started in 1992. In January 2000, NPS/10 had over 100 active users within Nokia and, as a commercial product, it also has a number of external users. Although our experiences are from the field of network planning, we think that several topics of the paper will also be applicable to planning tools in other areas as well.

There are several different sources of quantitative models and algorithms. Operations research (OR) (e.g. [2]) typically emphasises managerial decision making but provides a lot of useful tools and techniques with a wider use (e.g. [3], [4]). General computer algorithm research (e.g. [5]) is less concerned about the application area but focuses more on the speed and efficiency of the algorithms. Its subset, algorithmic graph-theory (e.g. [6]), is especially relevant for network planning. Finally, research areas like genetic algorithms (e.g. [7]),

simulated annealing (e.g. [8]), and other stochastic algorithms have lately opened new possibilities.

The goal of this paper is to look at the different kinds of models from the practical perspective. The starting point of our discussion is the user needs. The repeated finding clearly confirmed by our experiences is that understanding and respecting the users' needs is a key success factor in software development projects [9]. Another important aspect in a real development project is the limited development resources. Restricted development time and the number and competence of the developers do not always allow the best technical solution to be used.

In the next section we focus on user and, partly, the developer needs: the significance of full optimality, adaptation to changes, usability of the models, and the effect modelling has on tool development effort and schedule. In chapter 3 we will discuss some of the strengths and weaknesses of different types of models. Finally, in chapter 4 we try to provide some practical guidance about how to use the different types of models in a planning tool development project.

2 What are the needs for the network planning models?

Telecommunication network planning is a complex task. The network sizes are growing, cost efficiency is increasingly important, and capacity needs are expanding at a very fast pace. To help the network planners better accomplish their tasks a set of software systems, network planning tools, are available (e.g. [1], [10], [11], [12]). The tools typically provide interactive network editing, capacity calculation, traffic estimation, and many other features. Various algorithms are used to analyse and to plan the networks. The analysis algorithms calculate numeric measures that characterise the behaviour or capacity needs of a given network. The planning algorithms, on the other hand, handle the network synthesis by making planning decisions that affect the network topology or parameters.

1.1 How important is optimality?

Planning is a goal directed activity and thus it is natural to aim for a solution that is optimal. Since optimality is one of the main criteria in the comparison of algorithms and it is heavily emphasised in the OR literature, it easily gets too much attention. In practical planning tasks optimality is seldom necessary, and, in most cases, it is not even possible to find or to define. Sometimes striving for an optimal solution can even be harmful.

2.1.1 Unclear and unquantifiable goals

The network engineering process starts with a set of requirements or planning goals. Typical requirements deal with issues like functionality, cost, reliability, maintainability, and expandability. Often there are case specific additional requirements such as location of the maintenance personnel, access to the sites, company policies etc.

In practice the requirements are often obscure. Cost is an obvious goal, but it can be rather complex, especially if the focus is not only on the initial cost but also on the lifetime cost of the network. The functionality can be defined fairly simply by specifying the traffic that needs to be carried on the network, but difficulties in estimating the growth and the effect of new services makes the traffic forecasts very unreliable. Some reliability measures can be calculated if the failure intervals of different equipment and transmission media are known. However, the necessary statistical data may not be available for new equipment and assessing the harm caused by different failures can be complicated.

The other requirements are still harder. Numerical measures for maintainability and expandability are hard to define. Typically an expert assessment is needed to be able to say something about the issues

The additional, case specific requirements, which often are very important or mandatory, are hard to model. Furthermore, it frequently happens that requirements change during the planning process as more information becomes available. The practical way to deal with unexpected requirements is to rely on the user. This leads to the requirement that the tool and the models must allow the user to interfere. Hermetic models, which try to automate the whole planning task, are not very useful.

A common pitfall is to build the models in such a way that only those attributes that are easy to measure are included. However, there is a risk that an attribute that is easy to quantify (e.g. equipment cost) gets too much emphasis. Attributes that are harder to calculate (e.g. the maintenance cost) are left out of the models. Because the limits of the tool tend to limit the focus of the user's attention, this easily results into a bias towards the "easy" attributes.

2.1.2 Multiple goals

The requirements are often conflicting. For instance, increasing reliability requires investment to spare capacity, which increases the network cost. Similarly a simple and clear network structure that facilitates maintenance is seldom the minimum cost network.

A typical approach in optimisation models is to optimise only one criterion, typically cost. The other goals are considered as constraints.

A simple mechanism is to assign weights to different criteria according to their importance. However, our experience is that dealing with the weights is difficult for the user. Out of two plans the expert is able to say which one is better by considering the different objectives of the planning. It is also possible to say that e.g. the reliability is emphasised too much in a plan. However, it is extremely difficult to know how much the corresponding weight should be reduced to get a more balanced plan. For the expert it is easier to think which concrete modifications should be done to the plan.

The obvious way to evaluate the network plans within the scope of multiple criteria is to rely on the expertise and judgement of the planner. The network planner is able to estimate and balance the different aspects and in a flexible way take into account any relevant issues. Of course doing it well requires a considerable set of skills from the user.

2.1.3 Optimality is typically impossible

Even if we are successful in formulating the planning problem as an optimisation task, its solution is difficult. Most of the interesting network planning problems are NP-complete [13]. Although a lot of research is available of exact algorithms, which work for small and medium size problems, the solutions of large scale problems are only approximations.

All models are approximations of reality. Solving the model to optimality does not necessary mean that the solution is optimal also to the real problem. The model developer makes a lot of important planning decisions by deciding which aspects to include in the model and which aspects to leave out.

As a result developing a very accurate model or trying to reach the exact optimum are not very useful goals. It can even be risky because when a network is optimised very tightly with one goal, typically money, in mind, it might be expensive to adjust the network to the changing needs. Rather than aiming at optimal solutions, the planning models should allow the user to understand the network behaviour and trade-offs better.

2.2 Models should adapt to changes

An important aspect of the models, that has not received much attention, is the adaptation to changes. Software engineering studies have shown that over half of software development

effort is spent on maintenance [14]. When the models are embedded in the software applications, it is likely that considerable effort is also needed to maintain the models.

2.2.1 Technological changes

The rapid technological development in the telecom sector is bringing new technologies to the forefront as well as changing the use of existing technologies. New equipment generations can change the requirements because new equipment can have different technical constraints.

For instance, a few years ago one of the cellular transmission network planning goals was to minimise the number of nodes that had more than two links connected to them. Later equipment generations with different hardware removed the importance of this goal. It simplified the topology generation because a minimum spanning tree is very close to the optimum if the branching cost is negligible. Later, when the emphasis shifted towards increased availability, ring topologies are becoming more important, and minimum spanning tree algorithms are supplemented with travelling salesman algorithms.

Relatively small technological changes can have very deep impacts on the models. The models should thus not depend too much on technical details that are subject to change. As this is not always possible to avoid, a strategy would be to avoid any complex models that would take a lot of effort to develop and update in case of change. Another approach would be to try to modularise the models so that this kind of change will affect only a part of the model. However, this may not work if the solution of the model requires some special application dependent heuristics. A change in the model, even a modular one, may cause the solution mechanism not to work anymore.

Yet another approach to keep abreast with the technological development is to use as general models as possible. With the general models it can be possible to get enough information to make the critical planning decisions, especially at the initial planning stages. The closer to the details and actual implementation the planning is heading, the more technology and equipment specific data is needed. Ideally the detailed data, e.g. equipment properties, could be defined in a separate input file so that changes could be implemented by just editing the equipment definition file.

Unfortunately there is another trade-off. Implementing something generic is often much more difficult than implementing something specific. A general model may have to support a lot of features that are seldom or never needed in practice. Moreover, the performance of a general model usually suffers in comparison with a special model. Finally, the use of a general model may be more difficult since the user has to tailor it to fit the particular case.

2.2.2 Freedom of the planning process

A planning tool and its algorithms support and force the user to follow a certain sequence of planning steps. The positive side of a fixed process is that the resulting networks are similar in structure and thus easier to plan and install. Especially for inexperienced planners following a straightforward planning process is simple.

The potential risk with a very rigid process is that all networks are based on a single planning philosophy. As [15] points out the initial planning goals and constraints limit the alternatives that are generated. If the important criteria are included in the initial problem formulation this is acceptable, but it is important to notice that planning is a learning process where planner's understanding of the problem, the needs, and the alternatives is constantly changing. Thus the tools available for the planners should not restrict the planner to a set of predefined solutions but would allow room for exploration.

We have observed that the users are good at inventing new ways to use the tools. They have been able to use models which were intended for completely different purposes to plan new kind of networks based on different technologies that were not supported by the planning tool. This is possible if the tool allows the user freedom to combine different elements in a flexible way.

2.3 Usability of the models

Usability has become an increasingly important factor in software and other products. Although the models are not directly visible to the user, they indirectly affect the usability.

2.3.1 Understanding the models

The usability and user interface literature frequently speaks about conceptual gaps. A conceptual gap arises because of some difference between the user's mental model of the application and how the application actually works [16], [17]. Their ability to bridge this gap has been one of the reasons for the success of graphical user-interfaces.

In a similar way there is a conceptual gap between the real world and its mathematical model. The smaller the gap, the more intuitive is the model to the users and, also, to the developers.

In the end the planner, not the planning tool, is responsible for the validity of the network plans. To be able to rely on the results, it is important that the user understands how the tool and models work. Since the users are typically telecommunication experts, complex mathematical models and algebraic formulations are not very appealing to them [18]. Simple models as well as straightforward and concrete telecommunication interpretation of the models are a lot easier for them to understand.

A model that is close to the application domain is also good for the tool developers. For network problems the algebraic formulations tend to be problematical because they are constructed explicitly in terms of variables and constraints, while the nodes and arcs are merely implicit in the way that the constraints are structured [19]. For network problems a more natural representation is a formulation that explicitly deals with nodes, arcs, and network flows between nodes.

2.3.2 Solution parameters

An important factor for the ease of use is the number of parameters that have to be tuned for a model and a solution algorithm to work efficiently. Many algorithms require parameters, like step size and penalty weights, that are critical for the correct and efficient operation of the algorithm. To the users finding proper values for this kind of parameters is difficult. Since the parameters do not have any concrete interpretation in the application domain it is hard for application area experts to understand why the parameters are needed and what issues should be considered in setting their values.

The best alternative for the user would be to hide these parameters completely or to develop a set of pretested parameter value combinations for particular cases. Unfortunately, finding the proper values can be a difficult task also for the developer. The need for parameter tuning and the effort needed for it, is thus a factor when considering the suitability of different modelling alternatives.

2.3.3 Execution speed

Execution speed has a lot of effect on the use of a tool. Operations with fast response times can be used interactively to perform what-if type of analyses. Also they can be used as building blocks for more complex algorithms.

Slower operations are not very suitable for experimental work. Therefore it is important that the more complex models generate the wanted result immediately or with a few executions at most.

2.4 Model development speed and effort

There is increasing pressure in all industries to reduce the cost and time required to develop increasingly sophisticated products. As the models are embedded in a planning tool, the pressure to reduce the development time applies equally well to the models in the tool.

This has several implications to the modelling tasks. Model development is competing for the scarce development resources with other activities. Experience within our company has been that no person can completely be dedicated to model development, but must also participate on other development activities. Finding developers who can both model the problems and develop software solutions to them is not easy.

Another implication is that it is not desirable to take big risks in the model development. A fairly trivial model with moderate accuracy might be preferable over a highly sophisticated one if the development of the advanced solution is risky. This can be the case with mathematical programming models which might work well with small test networks but which might be extremely difficult to solve with bigger, realistic size networks. The ideal way from the risk management point of view would be the incremental development of models. In this way some solution would always be available and the development of more advanced models could be terminated if the pressure to release the product requires it.

Incremental development, which is gaining more and more importance in software engineering [20], has the additional benefit that it is a good way to refine the needs. The users are not very good at describing what they need in advance, but they are good at pointing out omissions, mistakes, and additional needs when they play with a development release of the software. Incremental development with concrete, working intermediate releases tends to keep the users interested and motivated, since they can clearly see how their requests are shaping the software.

Since the user input is very important for modelling, we think that developing the models in an incremental way is highly useful. First start with a very simple model and then gradually improve it to handle more and more complex cases. The evolution might lead into a situation that the model has to be replaced with a new more powerful one but at that stage a lot is known about the needs so that the selection of the model can be based on well known requirements. More importantly the simple models may already have been in use in early releases of the tool and thus they have served a useful purpose.

The ability of the models to adapt to changes is another aspect to reduce the time to market. If an old model can be reused with small modifications it will save a lot of development time. In that respect, the reusability of the models is an issue much in the same way as the reuse of software.

3 What kind of models to use?

In this section we characterise four different modelling approaches. In practise, the classification is seldom so clear-cut, often a mixture of the approaches is used. However, in most tools the dominant modelling approach is relatively easy to recognise.

3.1 Mathematical programming models

The classical mathematical programming approach typically formulates the network-planning problem as an optimisation model where a given cost function is minimised (maximised) under a set of constraints. Normally a single model tries to capture many relevant aspects of the problem. When an optimal solution to the model is found the values of the decision variables can be used to decide the optimal action to be taken.

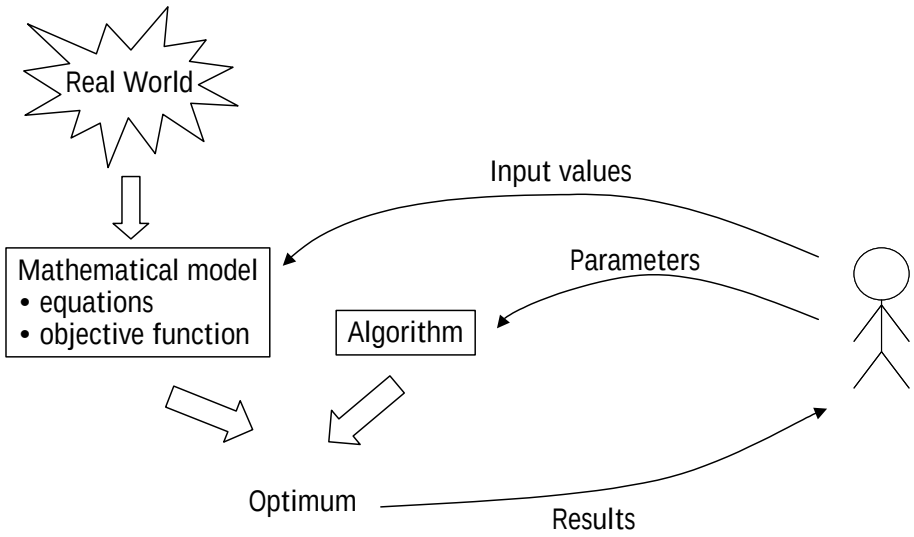

Figure 1 Problem solving with classical mathematical programming

Figure 1 shows an example of the classical mathematical programming approach where a single relatively large model is the core of the tool. The rest of the functionality (e.g. data input, parameterisation, model generation, and display of the results) are built around the model. It results into a model driven approach where the needs and capabilities of the model define what functionality the tool is providing.

Examples of the use of mathematical models for telecommunications network design are e.g. the backbone design problem [21], telecom network configuration [22], and spare capacity allocation [23].

Ideally mathematical programming is a nice way to approach a problem. The problem is formulated rigorously. When the model captures the relevant aspects of the problem the solution will give the best possible course of action. Finding the global optimum can save a lot since with big investments a few percents saving can have a large economic value.

The models arising from network planning problems are typically non-linear, combinatorial, and NP-complete. Following the ideal steps of an OR study as described in [2] the model creation is the real work for the OR practitioner while its solution is an easier step especially if any of the standard OR algorithms can be used. Unfortunately the network planning models are normally so complex that solving problems of realistic size is hard. As a result the model formulation and its solution cannot be decoupled. The practitioner has to consider both the

model and its solution: how to create a model that is a good enough abstraction of the problem and how to find and implement the right kind of solution heuristic for a given problem.

Finding the right balance between precision and tractability is a key issue in model development. The model is always an approximation of the real problem and it cannot represent all relevant aspects of the problem. Some aspects are ignored to keep the model simpler, other aspects are unquantifiable, and each planning case can have its own set of unanticipated goals and constraints. Dealing with unquantifiable criteria is problematical since the models need numeric evaluation functions for different aspects of the problems. Furthermore, finding the proper balance between the different objectives in a multicriteria situation is a major difficulty.

Even though the application specific heuristics are an effective way to speed up the solution time and find solution to problems that otherwise would be intractable, their negative side effect is that the system can become more vulnerable to changing needs. Technology changes can lead to small changes in the model formulation. The changes may turn a carefully tailored and fine-tuned heuristic solution algorithm useless.

Another complication is that the solution heuristics often need parameters (e.g. step size) that have to be tuned for good performance. In the worst-case these values need to be adjusted for each new planning case.

The mathematical programming models do not adapt very easily into user interaction. The user should understand how the model is working and how the parameters can be used to control the model. This can be quite hard since typical users are application area experts, not mathematicians. Parameterizing the models, for instance setting the weights for multiple optimisation criteria, is difficult. Also the speed of the algorithm is important to allow fast experimentation with different values.

3.2 Stochastic algorithms

The stochastic algorithms, such as simulated annealing [8] and genetic algorithms [7], are an alternative way to solve optimisation type of problems. They solve the optimisation problems via the use of random steps, and are, in particular, in discontinuous, ill-shaped search spaces more likely to find global optima than the classical mathematical programming algorithms.

For example for simulated annealing it can be shown under fairly general conditions that the global minimum will be eventually visited, and the search process will be confined with high likelihood to solutions that are globally optimal [24]. On the other hand solving a problem with simulated annealing is very slow. For problems where there exists accumulated insight and experience simulated annealing is usually inferior to other local search approaches [24]. Empirical comparison of different types of stochastic algorithms [25] also shows, that the more application specific heuristics are utilised, the better the results are.

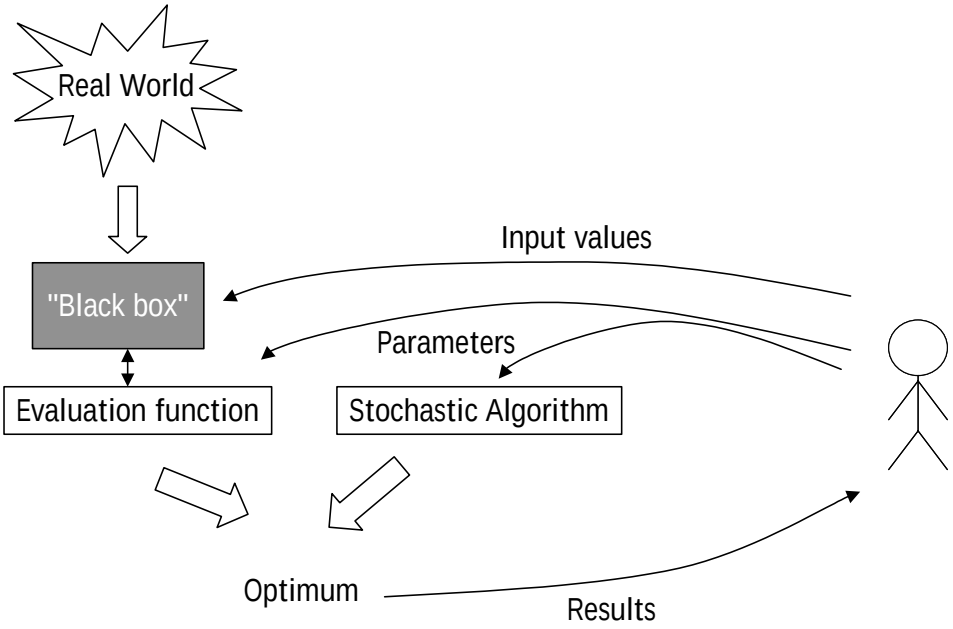

Figure 2 Problem solving with stochastic algorithms

Figure 2 shows an example of the use of stochastic algorithms to solve a problem. An attractive aspect of stochastic algorithms is that they can be used without having an algebraic model of the system behaviour. The system can be treated as a black box as long as there is an evaluation function available that allows solution candidates to be compared. This makes them easy to implement, and since they only need the evaluation function they are very adaptable to changes.

3.3 Functional modelling

Another way to provide planning support is to develop a functional model that defines how the system behaves. Contrary to mathematical programming models, a functional model (e.g. a simulation model) does not have an explicit goal; it just models the behaviour of the system in different circumstances. With the functional model users can try out different alternatives and analyse the outcomes of different scenarios.

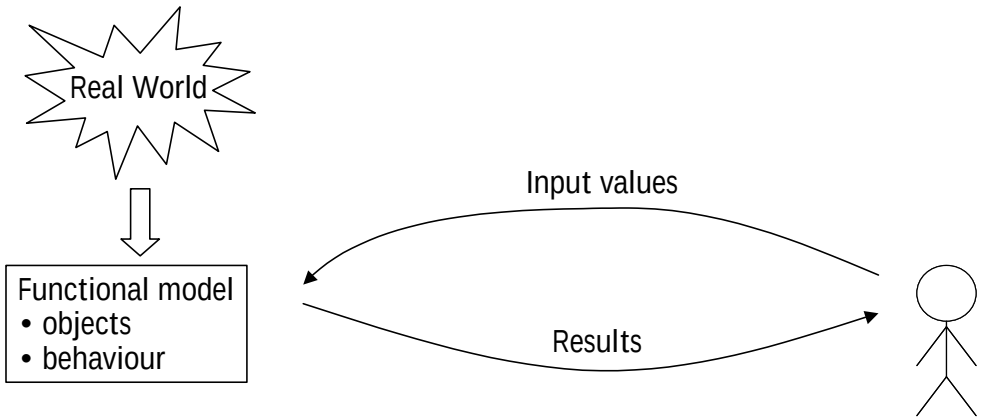

Figure 3 Problem solving with the functional modelling

Figure 3 shows an example of the use of functional modelling approach. In the functional modelling approach the user has a central role. The functional model both records the

planning decisions and simulates their effect. The model can be supplemented with a set of "add-on" type of analysis algorithms of varying complexity that can be used to study different aspects. By trying out various alternatives and studying their consequences the user should be able to gain insight to the problem at hand and thus be able to make good decisions, and, very importantly, potentially come up with new innovative solutions that where not foreseen when the tool and the models were developed.

The main advantages and disadvantages of the functional modelling approach are to some extent complementary to the mathematical programming approach. Modelling is usually easier since the model is only used to simulate the behaviour of the network. The user has more responsibility since he has to make the planning decisions himself. The model can only show the consequences of the decisions – not suggest any good solution by itself. Additionally the user has flexibility in controlling the planning process, to decide the steps from an initial plan to the final one.

There are several aspects that make developing the functional model easier than a mathematical programming model. There is no need to specify the objective function, which avoids the problems with multiple-criteria and unquantifiable objectives. The model can be represented in a procedural way which, especially when dealing with complex interactions, is often easier to develop and understand than an algebraic presentation. The object-oriented techniques, that have been successful in software engineering, are well suited to the development of the functional model.

According to our experience users find the functional models easy to understand. The conceptual gap between the real world and the functional model is not very wide. A user-friendly graphical user-interface is another key component to reduce the gap.

3.4 Special purpose algorithms

Special purpose algorithms are intended to solve problems of a particular type. Some of the algorithms, such as Kruskal's algorithm for the minimum spanning tree [5], are able to find optimal solutions. Other algorithms, such as algorithms for the travelling salesman problem, are heuristics that find solutions close to optimal [26].

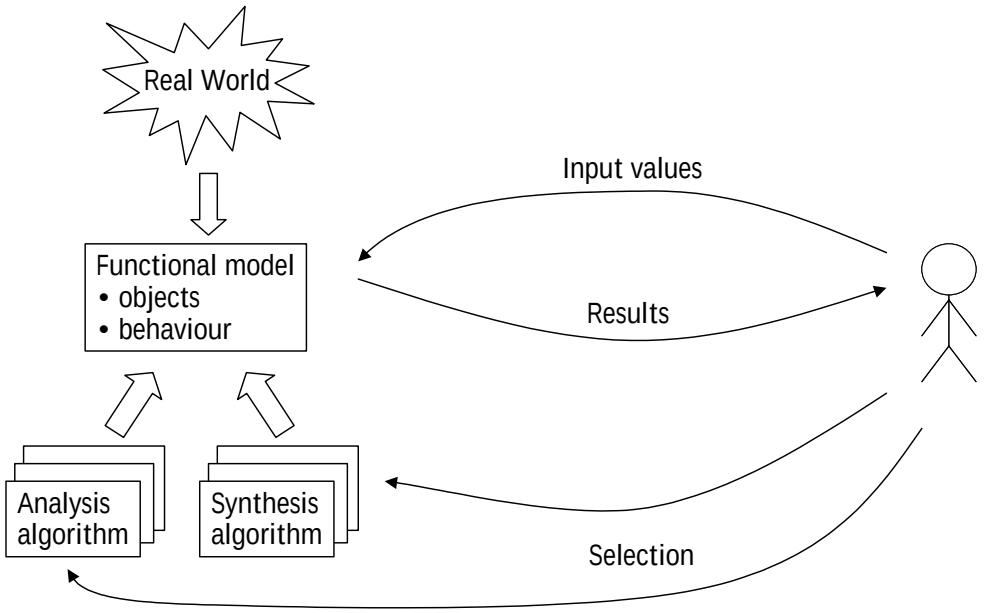

Figure 4 Problem solving using functional modelling with special purpose algorithms

The special purpose algorithms are often used together with the functional models (see Figure 4). The functional model represents the data that is manipulated by the algorithms. Since an algorithm typically handles only one issue, a set of co-operating algorithms are needed. The role of the algorithms is to act as "tools" that are applied to the data under the control of the user or under the control of some upper level algorithm.

When controlling the use of the algorithms, the user can select which algorithm to execute, which subpart of the problem it is applied to, and, potentially, perform some manual modifications in between. For instance, a travelling salesman type of algorithm can be used to create a ring topology for the core network region requiring alternative routes for protection, and a minimum spanning tree type of algorithms can be used to create tree topologies for each of the connected access regions. The algorithms speed up the planning by taking care of routine tasks that can be tedious and error-prone. This allows the user to study more alternatives, understand the behaviour of the network better, and thus develop better plans.

The adverse side of the user centeredness is that the user has to know how to plan networks. The planning algorithms help the users in computationally straightforward subproblems but they do not give enough assistance for the more complex network planning tasks. Especially the inexperienced users appreciate a system that makes more network planning decisions for them.

Another shortcoming is that the use of a set of algorithms to locally optimise one part of the network and one criterion at a time obviously does not lead to the global optimum. If the whole network could be optimised in a single step, the result would most likely be better in terms of the optimisation function used. However, as discussed earlier, this may not be computationally possible and the definition of the optimum is not unambiguous.

The special purpose algorithms can also be used as building blocks for more complex algorithms. For instance the shortest path algorithm that routes the traffic between a single node pair can be called iteratively for all node pairs to route the traffic in the whole network. A simple heuristic of ordering the node pairs based on the traffic between them results into a useful algorithm for traffic routing in a capacity limited network [27]. For this kind of use the fast execution speed of the algorithm is essential.

Many of the special purpose algorithms are easy to implement. The most common algorithms are available in textbooks e.g. [5] and algorithm libraries e.g. [28]. Since the algorithms are often quite general, they are not very vulnerable to changes in network technology and the same algorithms can be used with minor modifications to solve a set of related problems [27]. Since many of the special purpose algorithms perform straightforward tasks, it is easy for the user to learn what they are doing and to control their execution.

4 One way to develop a planning tool

Each of the modelling approaches of the previous section has its own merits. In this section we represent a way to use a mix of different models in the development of a planning tool. We have found this incremental development approach useful in the development of the Nokia NPS/10 planning tool [1].

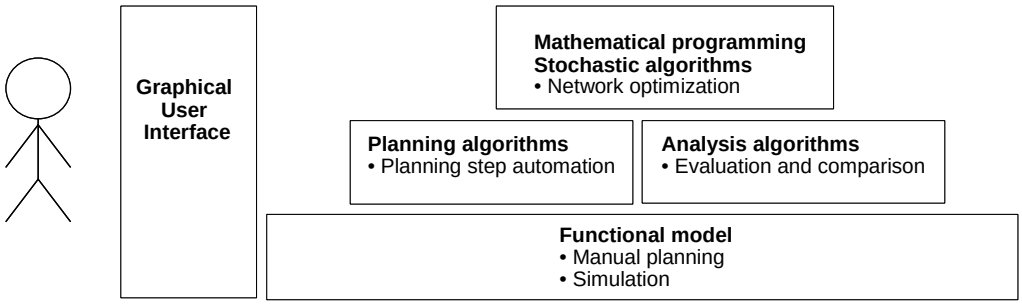

Figure 5 Model and algorithm layers in a planning tool

One way to look at the role of different kind of models and algorithms is the hierarchy in Figure 5. The foundation is the functional model that allows manual planning and analysis of the network behaviour. This layer alone would be enough for many planning tasks.

At the second layer the functional model is supplemented with analysis algorithms for more thorough evaluation of the network performance and with planning algorithms for automating straightforward planning steps. The analysis algorithms save the planning time by providing feedback in a condensed form to the user, for example evaluating the cost of the network. The planning algorithms automate tedious or error prone tasks, for example the traffic routing. The role of the second layer algorithms is to reduce the need for manual planning and thus speed up the planning process and reduce errors. This allows more options to be studied.

The optimization models at the third layer generate optimal or close to optimal results. They supplement the other algorithms of the planning tool and may use some of them as submodules. Since these models address many aspects of the problem simultaneously they are, assuming the models are appropriate and they can be solved in a reasonable time, likely to produce results that are better than those found by human planners.

From the tool development point of view the different layers pose different challenges. The layers can also be used to structure the development project so that only the most important layers are implemented in early releases. From our experience the functional model and a graphical user interface are the only obligatory parts of a planning system. The very first version of the tool must have these parts to be usable.

These should be supplemented in future releases by simple analysis and planning algorithms. Simple algorithms are typically good when the benefits and costs are compared. They are easy to implement and there is not much risk about their success and usefulness.

Mathematical programming models are a more high-risk investment. In many cases their development requires comparisons of different modelling and solution alternatives. The existing models in the literature may not be adequate and new kinds of models have to be developed. The solution times can be too long making the models useless for real size networks. From the development project point of view it is unfortunate if relatively late, after the model has been developed and several solution techniques have been tried, it is realised that the solution is not possible. A look at the development risks suggests that it should be very carefully considered when the mathematical programming models are worth the development effort.

5 Conclusions

Deciding what kind of algorithms and models to use is fundamental for the development and use of a planning tool. The needs of the users should be considered when making the selection between different modelling approaches.

Network planning is a complex task. It has multiple goals that are often hard to formulate, define, or compare with each other. The planning data is not very accurate since it is based on forecasts with very high error margins. The network technologies are developing on a fast pace so that generality and easy maintenance of the models are important. The development speed is often a critical issue, which can be controlled by using as simple models as possible, developing the models incrementally, and aiming for reuse of the models.

The functional modelling approach with simple and fast algorithms structured around a network analysis model and a user interface that allows the planner to control the planning steps and study alternative what-if scenarios seems to be the most straightforward and promising approach.

The relatively simple special purpose algorithms can effectively augment the functional models and take care of many routine planning tasks. This speeds up the planning process and allows more alternatives to be considered.

The mathematical programming models or the use of stochastic algorithms can be feasible alternatives in cases when the planning problem is stable, when finding a close-to-optimal result is of high financial or other value, when the planning operation is performed frequently, or when the users do not have a very high level of expertise.

Understanding the strengths and weaknesses of the different modelling approaches makes it possible to choose the appropriate approach for different tasks. The selection should not only be based on the immediate technical needs, but also on a broader view covering the user and the development project perspectives. A tool can combine different modelling approaches that can even be different in the different evolution phases of the tool.

The results of this paper are derived from the field of network planning tool development. It would be interesting to compare how the results of this study apply to other application fields. Another direction would be to consider tool development issues and study the relationship between modelling approaches and key development project facets, such as duration, effort, and risk. Since incremental development seems to be a practical way to develop models, further studies about models and algorithms that are most suitable for incremental development and reuse would also be useful.

References

[1] Nokia Networks, 1999. NPS/10. Computer program.

[2] Hillier, F. S., Lieberman, G. J., 1995. *Introduction to Operations Research*, McGraw-Hill, New York.

[3] Fortuin, L., van Beek, L., van Wassenhove, L., 1996. *OR at wORk: Practical experiences of operational research.* Taylor & Francis, London.

[4] Bortolon, S., Tavares, H., Ribeiro, R. V., 1996. Sonet planning using optimization tools in large networks: A practical case study. *Proceedings of the 4th International Conference on Telecommunication Systems* (pp. 494-502).

[5] Sedgewick, R., 1995. *Algorithms in C++.* Addison-Wesley, Reading, MA.

[6] McHugh, A. J., 1990. *Algorithmic Graph Theory.* Prentice-Hall, Englewood Cliffs.

[7] Goldberg, D. E., 1989. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading.

[8] Otten, R. H. J. M. van Ginneken, L. P. P. P., 1989. *The annealing algorithm.* Kluwer Academic Publishers, Norwell, MA.

[9] The Standish Group, 1994. *Charting the Seas of Information Technology*. The Standish Group, Dennis, MA.

[10] Nokia Networks, 1998. NPS/X. Computer Program.

[11] Telcordia Technologies, 1998.The Planning Workbench. Computer Program.

[12] Technical University of Budapest, 1998. PLANET. Computer Program.

[13] Johnson, D. S., Lenstra, J. K., Rinnooy Kan, A. H. G., 1978. The complexity of the network design problem. *Networks* 8, 279-285.

[14] Kain, J. B. Provost, W., 1999. The corporate benefits of a distributed component architecture. *Distributed Computing Architecture/e-Business Advisory Service Executive Report* 2(11), 1-29.

[15] Simon, H. A., 1995. Problem forming, problem finding, and problem solving in design. In Collen, A., Gasparski, W. W. (Eds.) *Design and systems: general applications of methodology* (pp. 245-257). Transaction Publishers, New Brunswick.

[16] Nielsen, J., 1994. *Usability Engineering.* AP Professional, Boston, MA.

[17] User Interface Engineering, 1999. *Bridging Conceptual Gaps*. http://world.std.com/~uieweb/gaps.htm.

[18] Grossman Jr, T. A., 1999. Why Spreadsheets Should Be in OR/MS Practitioners' Tool Kits. *OR/MS Today* 26(2), 20-21.

[19] Fourer, R., Gay, D. M., Kernighan, B. W., 1993. *AMPL: A Modeling Language for Mathematical Programming*. boyd & frazer publishing company, Danvers, MA.

[20] Jacobson, I., Booch, G., Rumbaugh, J., 1999. *The Unified Software Development Process*. Addison Wesley, Reading, MA.

[21] Gavish, B., 1991. Topological Design of Telecommunication Networks - Local Access Design Methods. *Annals of Operations Research,* J. C. Balzer AG, 17-71.

[22] Dahl, G., Martin, A., Stoer, M., 1999. Routing through virtual paths in layered telecommunication networks. *Operations Research* 47(5), 693-702.

[23] Kennington, J. L., Whitler, J. E., 1999. An efficient decomposition algorithm to optimize spare capacity in a telecommunication network. *INFORMS Journal on Computing 11(2),* 149-160.

[24] Bertsekas, D., 1998. *Network optimization: continuous and discrete models.* Athena Scientific, Belmont, MA.

[25] Lahtinen, J., Myllymäki, P., Silander, T., Tirri, H. 1996. Empirical comparison of stochastic algorithms in a graph optimization problem. *Proceedings of the Second Nordic Workshop on Genetic Algorithms and their Applications*. Vaasa, Finland.

[26] Golden, B. L. and Stewart, W. R., 1985. Empirical analysis of heuristics. In Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B. (eds.) *The Travelling Salesman Problem* (pp. 207-249). John Wiley & Sons, Chichester.

[27] Akkanen, J., Nurminen, J. K., 2001. Case Study of the Evolution of Routing Algorithms in a Network Planning Tool. *The Journal of Systems and Software* 58(3), 181-198.

[28] Max Plank Institut, 1998. LEDA. Computer Program.