Contributions to the ITU-T Ref. Manual (Part 3)

Networks Design, Dimensioning and Optimization (Chapter 7)

"Telecom Network Planning for evolving network architecture", (3rd draft of Ver.2.0)

by DINH Van Dzung, Vietnam, adlien@hn.vnn.vn

- *There are too many evolved network technologies as presented in Chapter 6, it is impossible to provide detail guidelines (of issues such as network design, dimensioning, and optimization) which are useful for network planning experts, policy makers, and tool developers and just in a chapter like in this manual. For this 1st edition of the handbook, the audience is expecting general frameworks, approaches, analysis, methodologies, and solutions to solve the network planning/design challenges when migrating to NGN, IP, MAN, xG, etc. Therefore, it is proposed to consider the following **outline of Chapter 7 of the manual**. Based on this manual, further studies for particular network planning problems including the "pure NGN" are expected to be appeared in next editions of the manual.*

Proposed outline of Chapter 7

- *It is proposed to add the proposed sections **7.1**, **7.2**, **7.3**, and **7.4** to Chapter 7 of the manual.*
- *the proposed section **7.9** is preferred to move to the Annex area.*
- *It is proposed to add **Annex 1** and the **References** list to the manual.*

7.1 General approach

This general network planning approach is mainly based on the framework for OTN (Optical Network) planning provided in [1]. It can be applied for implementing detail network planning processes and solutions for the *evolved networks* such as Ethernet, xDSL, RPR, PoS, MPLS, IP, … and NGN where it is stated for a further study in a next edition of the handbook.

As stated in Chapter 2, the network planning is commonly recognised as a very complex problem. The ***breakdown approach***, which divides the overall planning problem in smaller sub-problems decreasing the complexity of the planning activity, is recognised as a practical, feasible and widely-used solution for the evolved networks. There will be a need of analysing characteristics of the evolved networks that affect the planning processes (see Chapter 2). As a consequence, planning problems can be described in a similar way for different network technologies.

Working out the results obtained so far, it seems clear that in order to obtain the highest advantages from the breakdown approach it is important to make it technology-independent. The general approach for a ***technology-independent breakdown approach*** consists in identifying a set of planning sub-problem that cannot be considered further as splittable (called "atomic"). So doing an ***atomic sub-problem*** can be part either of planning problems related to different network layers (e.g. IP, Ethernet, RPR, ATM, MPLS, SDH, WDM) or of different planning problems in a single layer network (e.g. topological planning, network dimensioning). The result is a "vertical" and "horizontal" re-usability of algorithms and planning software. The atomic sub-problems can be selected on experience-basis looking for a good trade-off between complexity and allowed optimization.

The aim of this section is to give a clear distinction between the planning problems, the methodologies followed to solve them and the models and algorithms applied within the methodologies. The atomic sub-problems and the basic (atomic) relationships that can be established among them are described as well.

7.1.1 *Planning problems and processes*

A ***planning process*** is the technical approach or methodology to follow in order to solve an identified complex network-planning problem (see Chapter 2). Given a certain planning problem the planning process should define the sequence of steps to follow in order to solve it, namely:

- the identification of class of problems it belongs to, based on its generic characteristics;
- the partitioning of the problem into smaller, manageable tasks (the atomic sub-problems that can be solved with atomic sub-functions);
- the relationship among the atomic sub-functions.

In the case of the availability of different alternatives, a comparative analysis of these alternatives may be considered in view of the main objective of the planning exercise.

A global perspective of the original planning problem being handled should be kept (maintained), within a planning process, in order to assure coherence of the several intermediate results and allow the best choice solution to the given problem. This means that one of the most important aspects of a planning process is the control of inter-relation between different atomic sub-functions, and the simplifications and assumptions introduced on the different analysis being undertaken.

7.1.1.1 Classes of planning processes

Classification of complex planning problems is essential as it speeds the achievement of a solution to a particular problem; this classification can be based on the similarities of planning problems, which allow similar formulation and modelling of identical problems and, finally, a similar process to solve them.

Each class of planning processes will therefore group similar planning problems, while each planning problem will be framed in a particular class of planning processes or may originate a new class of planning processes.

The following ***generic classes*** have been identified for network planning problems:

- Network clustering,
- Network structure optimization,
- Network dimensioning and optimization,
- Resource allocation.

For instance, two different problems could be identified when planning an optical transmission network (OTN):

- definition of rings (in terms of cluster of nodes) for an interconnected ring SDH network;
- selection of a set of nodes to deploy WDM functionality,

can be gathered in the "network clustering" class and can be solved with similar methodologies, models and algorithms.

It is important to note that the selected network architecture often strongly influences the planning process too. As a general consideration it is possible to state that similar planning process can solve different problems within the same class when the same architecture is taken into account. On the contrary, the considered network layer generally does not influence the planning process; for instance, the same planning problem can be solved with the same planning process in the SDH and in the WDM layer.

7.1.1.2 Phases of a planning process

The solution of each complex planning problem is obtained with the general methodology depicted in Figure 7. 1. The basic idea is to identify the suited process to solve the planning problem: if a similar problem has already been studied, a suited methodology should be already available, otherwise a new process must be identified. This procedure suggests a ***high degree of re-usability of the procedures*** (and, hopefully, of the algorithms, models, tools, experiences) identified to solve a planning problem.

The classification of each problem can be based on its formulation, and particularly on:

- Time frame,
- Set of inputs,
- Set of outputs,
- Variables,
- Constraints.

Examples of planning processes and their decomposition according the atomic breakdown approach can be found in Annex A1 (see more at Chapter 4 [3]).

7.1.2 *Atomic planning sub-problems and sub-functions*

On the basis of the experience that has been reached in planning a network, the following set of ***planning sub-problems*** has been identified as the most atomic problems a network planner has to accomplish with:

- routing of demands,
- demand grooming,
- stand-by network planning,
- switching/routing/transmission system allocation,

- equipment allocation and cost evaluation.

Each sub-problem can be described on the basis of the following information:

- target function,
- input data,
- constraints,
- required output.

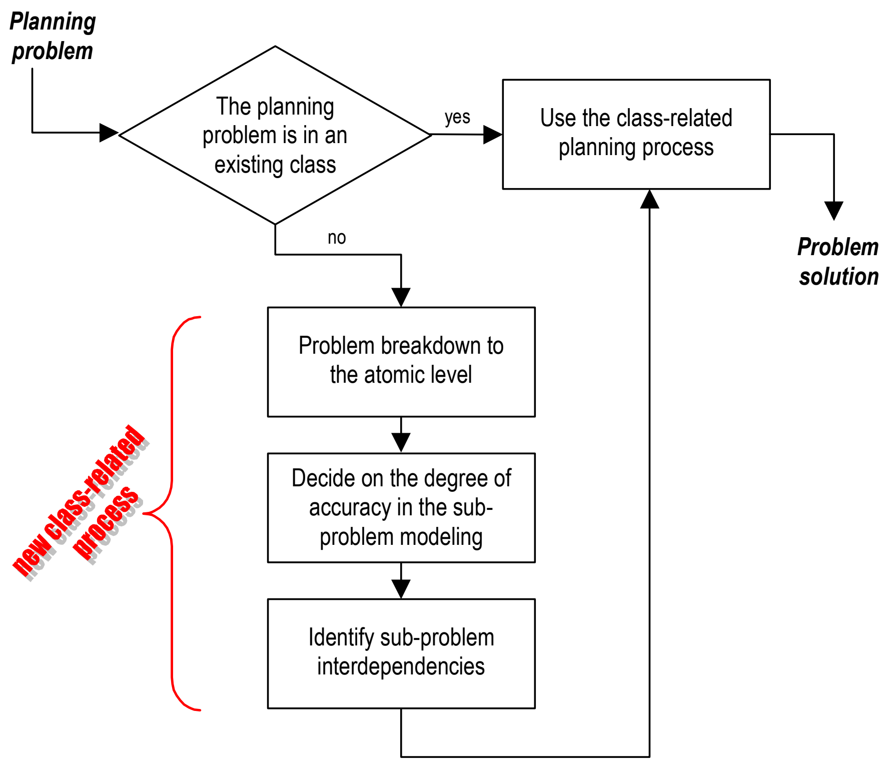


Figure 7. 1 - Class-related processes in planning problem solution

For example, the optical network layer inherits from SDH the above list of atomic sub-problems (eventually with some additional constraints) and adds to them the ***wavelength allocation*** one, that can be considered the only planning sub-problem specific for the optical layer[1]. The planning activity that allows solving an atomic sub-problem is called ***atomic sub-function***. As the overall planning problem can be decomposed into several sub-problems, the overall planning activity can be divided into several atomic sub-functions. Such an approach seems to be very good at exploiting the highest advantages from the decomposition approach, because it allows a high degree of re-usability of the atomic sub-functions and preserves all the experiences matured with SDH layer. A complete technical specification of the above mentioned atomic sub-problems can be found in Sub-chapter 3.1 [2].

[1] The wavelength allocation sub-problem arises only if wavelength conversion is not performed in the OTN's nodes.

7.1.2.1 Atomic relationships among sub-functions

The breakdown approach allows dividing a complex planning problem into several atomic planning sub-problems. However the overall results are obtained establishing relationships among atomic sub-problems where, for instance, the solution of an atomic sub-problem becomes input for another sub-problem. As a consequence the methodology that solves a planning problem can be viewed as a "re-composition" of atomic sub-functions and can be described with a flowchart. Each relationship between two atomic sub-functions is selected among a set of ***atomic relationship***. Table 7. 1 assigns a symbol to each atomic relationship and describes its results.

In order to understand better the relationships among sub-problems, it is helpful to provide some examples for the OTN case:

- the *routing* sub-problem is generally solved before the *grooming* one, so a *moving* relationship from *routing* to *grooming* is present. However it is possible to have a feedback from *grooming* to *routing* as well; for instance, when the application of grooming to a fixed routing map leads to an under-utilisation of some resources, a good policy could be to change the routing map of the client layer connections, which use these resources. In this case there is a backward *moving* relationship from *grooming* to *routing*. A feedback action between two sub-functions can be indicated with a two-sided arrow;
- a possible alternative is to solve the *routing* sub-problem together with the *grooming* one; as a consequence a single algorithm has to be developed for the two sub-functions and a *merging* relationship is used;

instead a good example of *interleaving* relationship is between *routing* and *stand-by* sub-functions in case of application of a protection scheme. In fact, in this case, the spare resources for protection can be found after that each single demand has been routed and the control moves from routing to stand-by algorithms demand-by-demand.

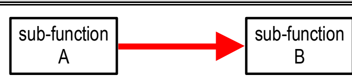
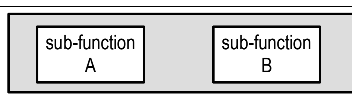
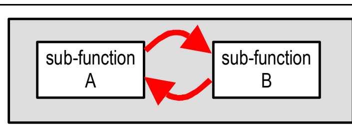
Name	***Symbol***	***Meaning***
moving		the output of process A becomes input for process B
merging		process A and B are solved at the same time with a single algorithm
interleaving		process A and B are solved at the same time with two different algorithms; the overall control passes from an atomic sub-function to the other each time that an iteration of the algorithm has finished; iteration by iteration the two atomic sub-functions exchange data

Table 7. 1 - Atomic relationship between atomic sub-functions

7.1.2.2 Handling the sub-problem interdependencies in the breakdown approach

As mentioned above due to the size and complexity of planning problems there is no feasible alternative to the problem breakdown into smaller more focused atomic sub-problems, to a size and detail that allows their solving. The main disadvantage is the loss of perception of the global problem when handling a particular atomic sub-function, as each one is solved as a stand alone problem (independent of the other tasks), and workarounds have to be used to introduce the atomic sub-function dependency relationship in the planning process.

Three breakdown methods have been identified that differ in the handling of ***sub-problems interdependencies*** (cf. Figure 7. 2):

- ***decomposition***: the planning problem is modelled as a linear chain of sub-problems. Within the chain the most relevant interdependencies are kept. The optimality of the obtained solutions depends on the importance of the discarded atomic sub-function dependencies (in order to obtain the linear chain). The workarounds used to achieve more close to optimal solutions are the feedback, iterations and tolerance analysis.
- ***gradual simplification***: the problem and/or sub-problems are modelled and by successive gradual simplifications of the used models the size and complexity are reduced to a manageable level. The simplicity of each used particular model varies according to the importance given to each modelled problem, the more relevant problems being the ones with more complex and accurate models. The optimality of the final solution depends therefore strongly on the initial level of importance classification attributed to each sub-problem.
- ***mixed methodology***: a mixed approach of the above methodologies can also be used, i.e., some less important interdependencies will be discarded and the least relevant sub-problems will be simplified, therefore only the most important interdependencies will be considered and the accurate modelling will be only used for the most significant sub-problems.

As in the OTN case, further considerations on the interdependencies among sub-problems can be found in Sub-chapter 3.2 [2].

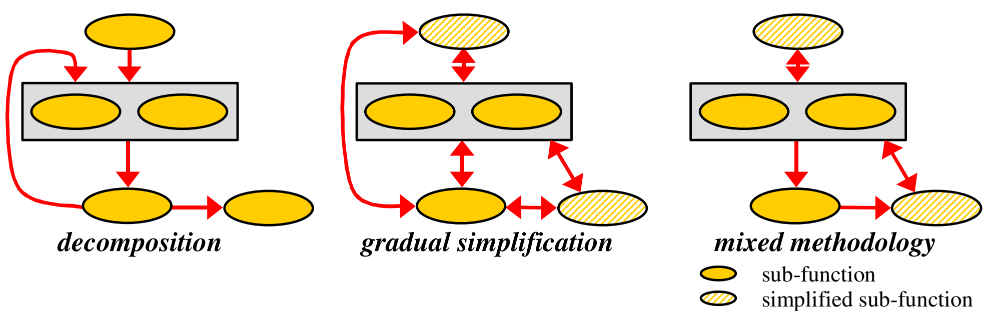


Figure 7. 2 - Simplified representation of the sub-problems interdependency policies

7.1.3 *Models and algorithms*

Problem solving, resulting in a solution that is desirably as close to optimum as possible, depends much on the mathematical formulation of the problem, i.e. the used models, simplifications and assumptions. The applicability of a particular solving technique is in close relation with the size of the problem, which depends on the type and number of variables and constraints, and on the type of the used cost function.

Thanks to the breakdown approach, models and algorithms must be applied to atomic sub-problems: as a consequence an atomic planning sub-function can be described as a combination of a model and an algorithm. In particular a model is a mathematical description of a problem, while an algorithm is a finite sequence of steps to obtain solutions to the problem. A survey of algorithms and models to be applied on the evolved networks planning problems can be found in Section 7.9. A non exhaustive list is proposed in Table 7. 2.

Models	***Algorithms***
∇ Multi commodity flow model	√ *Enumeration*
∇ Path-flow model	√ *Branch and bound*
∇ Path formulation model	√ *Relaxation techniques*
∇ Mathematical Programming	√ *Marginal improvement techniques*
∇ Analytical Model	√ *Simulated annealing*
∇ Artificial Intelligence	√ *Tabu search*
∇ Scalable Architecture	√ *Successive smooth approximation*
∇ Hybrid Combination	√ *Simulated assignment*
	√ *Genetic programming*
	√ *Heuristics*

Table 7. 2 - Algorithms and models are proposed for planning evolved networks

7.2 Multi-layer network planning

Incumbent network operators (NOs) are still deploying networks based on the TDM technology and they are likely to keep this technology alive in their network for years. At the same time, the increasing IP traffic is "pushing" toward the introduction of many broadband networks candidates in the existing networks. As a consequence the ***network planner has to plan multi-layer networks***: that represent a demanding new task for the network planner. In fact, although the more technology alternatives are available the more possibilities to plan a suitable and cost-effective network, practically the network planner has to manage the increased complexity of the planning activity.

A possible approach to multi-layer planning is presented in section. This represent an adaptation to the atomic breakdown planning approach of the multi-single layer planning methodology proposed in [4] and [5].

7.2.1 *Layering and the client/server relationship*

The layering concept (first introduced to describe SDH networks, but extended in a straightforward way to multi-technology networks) is very useful when describing complex transport networks. Broadband networks can then be viewed as consisting of multiple network layers, with a client/server relationship between adjacent layers, where each layer provides transport to a higher layer. That is, client link connections are provided by setting up trails in the server layer. Hereby, each layer acts as a client of its transport layer, and conversely each layer is a server layer to higher layers.

The most relevant aspect of layering to planning is that each link of a higher layer is in fact realised by establishing a trail in the layer below (its associated server layer). The facilities (i.e. working capacities) of the server layer are used to construct the topology of the client layer. This is done by routing of server layer paths, which carry the demand from the upper layer. As a consequence, ***the topology - including the link capacities - of each layer (excluding the lowest layer) is equivalent to a traffic matrix of its associated server layer***.

It is possible to describe multi-layer networks without using the layering concept, i.e. without a layered network model. In fact, physically, signals of the different technologies are mapped in each other. For instance the switching network traffic is mapped into a VC-4 containers, the

SDH electrical signal is converted via a transmitter in an optical channel, which is multiplexed with other channels (other wavelengths) on an optical fibre. So, in reality we only have physical links (cables, fibres). The links in layer networks higher than the physical layer could be considered as virtual.

However, if we would not use a layered network model, the planning problem would become extremely difficult to solve, as one is obliged to tackle the planning of multi-technology network as a whole. In contrast, layering permits to ***split up the complex network design problem into different 'semi-independent' single-layer design problems of the different layers***.

7.2.2 *Integrated approach*

Obviously, the best way (in terms of solution optimisation) to design a multi-layer network is to tackle the problem in its entirely, which means optimising all network layers at the same time. However, due to practical reasons, an integrated approach is not suited for the multi-layer planning problems. In fact doing so, we get confronted with a very complex optimisation problem, solving it would require an impractical amount of resources, both on human side (development of new algorithms, implementation effort,…) and on the computer side (running time issues, memory concerns,…). Moreover, little or no knowledge exists - in the literature - about tackling multi-layer planning problems. Finally, we have to take into account the enormous amount of existing single-layer planning knowledge and algorithms. It is a wise strategy to re-use as much as possible of this valuable expertise.

7.2.3 *Sequential approaches*

The multi-layer planning problem has somehow to be divided in a number of simpler sub-problems, which each can be solved in reasonable time. The result is a ***layered approach*** to the multi-layer planning problem, where the different sub-problems consist in the planning of only one layer. The planning of a multi-layer network then consists of sequential solving of different single-layer planning problems. Each of these single-layer planning problems can be viewed as an LTP or MTP problem, like the ones described in the previous chapters, and can be solved with the same approach. The overall multi-layer planning approach is a ***layered (or vertical) atomic breakdown approach***. Again, a maximum re-use of existing (single-layer) planning knowledge and tools is assured.

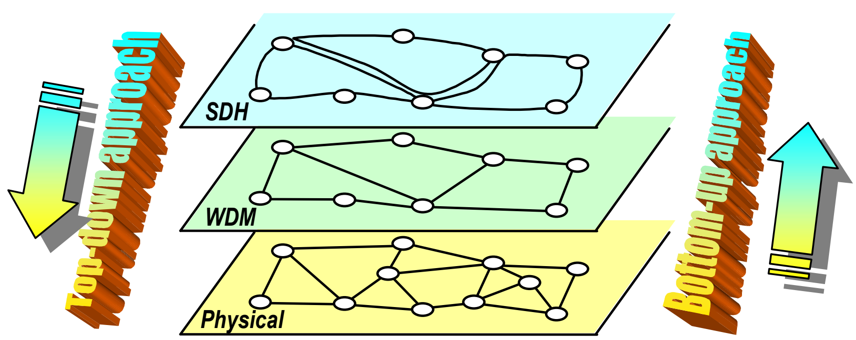


Figure 7. 3 - Multi-layer planning pure sequential approaches

However existing tools require some adaptations before they can be fitted in some sequential multi-layer planning approach. Furthermore, the deployment of a multi-layer survivability strategy requires more interactions between the different single-layer planning modules. Generally it is necessary to arrange some feedback between these modules resulting in some kind of iterative approach.

7.2.3.1 Pure sequential approaches

There are two possible ways of implementing a pure sequential approach, where successively - one at a time - single-layer planning problems are solved (cf. Figure 7. 3 for the OTN case):

- ***Bottom-up planning approach*** (cf. Figure 7. 4): starts with the planning of the lowest layer, i.e. the physical layer, planning successively higher layers. That is, each layer is planned taking into account information (givens, cost information,…) produced by the planning of lower layers, which are already determined at that moment. For instance, the cost model of a layer could take its server layer into account and charge for the use of the server facilities. Indeed, the dominant cost of the network lies in the lower layers.

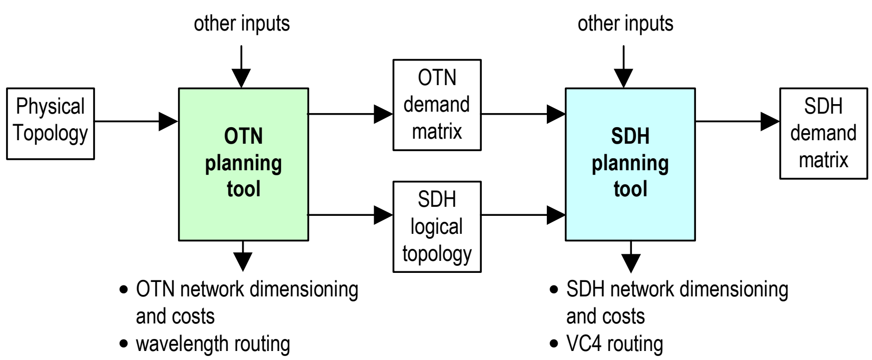


Figure 7. 4 - Bottom-up sequential approach

- ***Top-down planning approach*** (cf. Figure 7. 5): starts with the planning of the highest layer, and then successively plans lower layer, with the information (demand) extracted from the layers already planned. For instance, the links and the capacities of a client layer are additional for the planning of a certain layer, and should be added of the native demands (i.e. injected at that particular layer…)

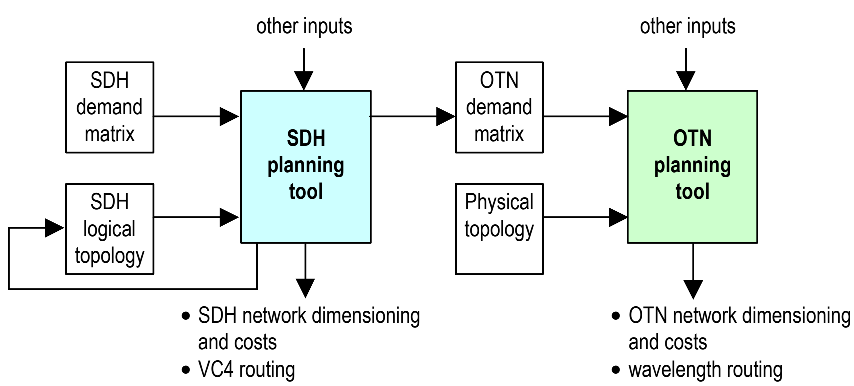


Figure 7. 5 - Top-down sequential approach

7.2.3.2 Problem issues with a pure sequential approach

The most fundamental problem with sequential approaches to multi-layer planning is that

- in lower layers the planner does not dispose of demand matrix (because the demand matrices are typically introduced at the higher layers), while
- in higher layers he has no idea about which costs should be taken into account (because the cost of carrying a VC-4 in the SDH layer strongly depends on the mapping of the SDH resources on the WDM layer).

To put it simple, a sequential approach has to start planning at some layer, but in every layer some important input data are missing (required by current planning algorithms).

A traditional (single-layer) planning algorithm needs a demand matrix as input to be able to allocate a feasible capacity distribution. However, demand matrices are always given as a matrix of desired semi-permanent connections in the highest layer (i.e. SDH in the considered two-layer network). As a consequence, planning cannot easily start at lower layers because there is no explicit idea of what the demand will be in the lower layers (cf. Figure 7. 6).

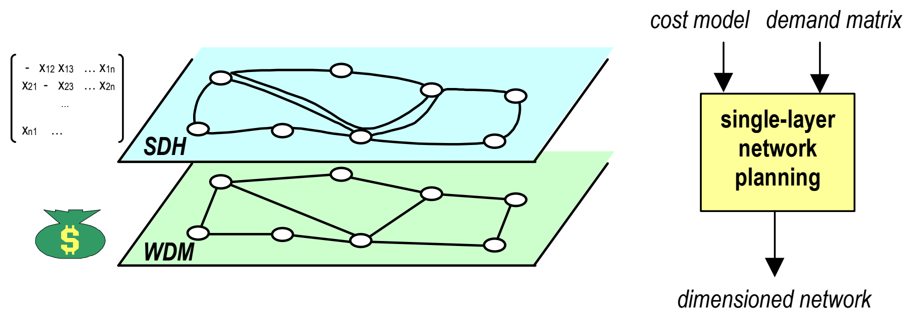


Figure 7. 6 - Problem issues with pure sequential planning approaches

On the other side, with a sequential planning approach starting at the higher layers, we have no idea about the impact on the total network cost of capacity allocation in higher layers. In fact the cost of capacity in a higher layer link connection strongly depends on how the connection is implemented in the server layer (usage of fibres, their availability, eventual transits in the server layer). As a consequence, we cannot foresee if a capacity increase of a higher layer link will have serious or negligent impact on the network cost (cf. Figure 7. 6).

Summarised, the drawbacks of pure sequential planning approach are:

- Bottom-up planning approach - the considered demand is a very rough estimate of the real demand, and therefore, it could be that the planned network is either over-dimensioned, or either unable to carry the actual demand;
- Top-down planning approach - the cost model is incomplete, as it does not consider the influence on lower layer (most dominant) costs, when planning a layer.

Anyway, it is much easier to overcome the top-down approach drawback (at least in an approximated way) on the basis of the experience, so that this approach is suggested.

7.2.3.3 Iterative approach for multi-layer planning

An ***iterative planning approach*** consists in successively solving single-layer planning problems to tackle the multi-layer planning problem. The top-down planning approach was the starting point of research on a better methodology, because the network resulting from this multi-layer planning methodology is at least able to carry the real demands. One solution to the cost model problem could be to improve the top-down planning approach by adding a feed-back loop which propagates cost information from the lowest layer to the higher layers. This means that the different layers are planned several times, and every time a higher layer is planned, lower layer cost information is propagated to the cost model of this higher layer. Of course, every re-dimensioning of a higher layer implies the adaptation of the demand matrix of lower layers, and as a consequence the design of these lower layer is resumed. This alternating process is applied several times, until a satisfactory solution is reached.

7.3 LTP, MTP and STP coherence

The combined action of LTP, MTP and STP allows completely defining the network evolution. Ideally LTP, MTP and STP should act coherently, so that the network long-term evolution defined by the LTP is translated by the MTP in scheduled upgrading of nodes and links, which are used by the STP to support demands. Unfortunately this coherence is far from reality and a more pragmatic approach is often adopted by NOs. This chapter deals with the coherence of the overall planning activities, trying to understand the additional issues raised by the introduction of the optical layer and to define some possible countermeasures.

7.3.1 *Causes of incoherence in planning*

There are several causes of incoherence among LTP, MTP and STP.

As already stated in Chapter 2, ***condition of high uncertainty*** justifies a "pragmatic" interpretation of MTP in which a time scale shorter than the one of LTP is taken into account. That happens because it is no use to implement a detailed planning exercise in case of unreliable forecasts. Generally the shorter the time scale considered the more accurate the forecasts: only the nearest-term single-period MTP processes are worth to be performed. In this way the LTP results can be still viewed as a target for MTP, but the relation between the two is farther than in the traditional MTP approach. Mentioned uncertainty surely regards the demand forecast, but often also the equipment cost and/or technological evolution. In a competitive market, characterised by high bandwidth new services, the uncertainty is likely to become more and more usual and dramatic.

The uncertainty (especially in demand forecasts) affects MTP vs. STP coherence as well. An unforeseen demand to be supported in STP cannot find available resources in the MTP planned network, which deploys resources on the bases of the foreseen demands only.

Another reason for incoherence is ***budget limitation*** in building the network. It directly affects the MTP results and generally imposes a delayed deployment of some resources. As a consequence the LTP routing policies cannot be applied to the MTP network at a certain time, due to lack of resources: a different routing policy leads to different resource utilisation and, finally, different resource requirements between LTP and MTP.

MTP and LTP can use ***different optimisation criteria***. Under the hypothesis of no uncertainty in the input data and no budget constraints, the optimal solution obtained with LTP is obtained with MTP as well when an optimisation criteria "optimise at the same time the network cost in each period" is applied . In fact the application of this criteria in the ideal conditions considered above, leads MTP exactly to the LTP target results, simply scheduling all the LTP investments. However other two optimisation criteria for the MTP have been proposed:

- optimise the discounted sum of the investments from the beginning to the considered period;
- optimise the next MTP period, only considering the structural part (network architecture, network structure) of the LTP results as a weak constraint.

These criteria, generally adopted in case of budget limitation and/or input uncertainty, definitely lead MTP to different results in respect of LTP.

In addition, also the ***resource utilisation rules*** of STP and LTP/MTP are generally different. The main reason for that lies on the fact that the time-to-market feature[2] can be obtained by a telecommunication network only if no lack of resources happens and its STP rules are very simple. The need of simplicity of STP procedures becomes more and more important when the network itself gets more complex (like in case of the introduction of the OTN under the SDH existing network layer). It turns into very simple and "quick" rules for the demand routing, as soon as the demands "appear". These simple rules are generally not coherent with the complex optimisation performed in LTP/MTP.

It is important to note that different resource utilisation policies among STP, MTP and LTP are generally unavoidable, because LTP and MTP deals with static demand evolutions while STP with dynamic demand evolution. In fact the single period demand matrix used by LTP and the multiple single period matrices used by MTP do not contain any information about the order of apparition of the demands: this generally allows dimensioning the network with an order-insensitive algorithm, which leads to a better utilisation of network resources than an order-sensitive one. The additional constraint of demand order, which appears in STP, gives less space for optimisation.

Table 7. 3 summarises the causes of incoherence, specifying which planning phases are affected.

Incoherence cause	***between***
High uncertainty	LTP - MTP and MTP -STP
Budget limitation	LTP - MTP
Different optimisation criteria	LTP - MTP
Different resource allocation rules	LTP - MTP and MTP -STP

Table 7. 3 - Causes of incoherence in planning phases

7.3.2 *Countermeasures against incoherence*

Avoiding incoherence should avoid lack of resources during the STP phase: doing so we obtain a "time-to-market" network, in which all the demands can be actually routed in the network as soon as they appear. Besides the impact on the network optimisation of the different causes of incoherence is not the same:

- when the affected phases are LTP and MTP, the consequence is that the strategic and fundamental choices, like best network architecture and structure, cannot be the best one anymore; however it does not affect the time-to-market feature;
- when the affected phases are MTP and STP, the consequence can be that some demands cannot be routed on the network, due to lack of resources.

The second point above seems more severe than the first, because it implies the inability of the NO to accept a new customer, with direct damages in loss of profit, image and competitive strength. On the contrary incoherence between LTP and MTP seems to become more and more acceptable from NOs, when moving from the stable condition of monopoly to the dynamic state of competition. Nevertheless, accepting LTP as a never reachable "moving"

[2] It is the ability to answer immediately a request coming from the market. It is intuitive that it can give an advantage to the operator in the competitive environment.

target, we do not reduce its importance in taking strategic and fundamental decision in a high-risk environment.

Two kinds of action can be taken against incoherence:

- Preventive countermeasures;
- Reactive countermeasures.

These two classes of countermeasures are briefly described in this section.

7.3.2.1 Preventive countermeasures

A preventive countermeasure tries to act on the cause of incoherence, with the aim of eliminating it or, at least, to reduce its effects.

Table 7. 4 summarises some of the possible preventive countermeasure that can be adopted.

Incoherence cause	*Preventive countermeasure*
High uncertainty	Improving the forecasting methodologies
	Reducing the considered time-scale
Budget limitation	Improving the overall network creation process
	Demand reconfiguration
Different optimisation criteria	-
Different resource allocation rules	Improving the overall network creation process
	Reducing the considered time-scale
	Demand reconfiguration
	Selecting simpler network structures and architectures

Table 7. 4 - Preventive countermeasures against incoherence of network evolution

A general countermeasure consists in ***improving processes***: a more sophisticated forecasting methodology can give a "more realistic" view of the future network demands and a more robust network creation process can establish tighter links between planning phases. However even the best processes cannot completely avoid incoherence and its effects.

Another good but not decisive action is the ***reduction of the considered time-scale***. It should reduce both the uncertainty and differences between static and dynamic demand evolution.

Besides ***simpler network architectures, structures and routing and protection schemes*** can significantly reduce a different use of network resources between MTP and STP.

Demand reconfiguration can reduce the impact of budget constraints and the differences in the resource utilisation among different planning phases. All the new demands that do not require additional resources are routed on the existing one, eventually on a longer or more expensive path. When in a next time slot new resources are deployed, allowing a better routing policy for some already routed demands, it is possible to exploit more advantages from the new resources reconfiguring (i.e. re-routing) some of the already routed demands.

7.3.2.2 Reactive countermeasures

All the preventive countermeasures can improve coherence, but never avoid incoherence. A wise and pragmatic policy is to accept incoherence, trying to quantify its effects on the planning results (reactive policy). Note that this way of doing does not exclude the preventive countermeasures mentioned above.

Table 7. 5 summarises the reactive countermeasures proposed.

Risk analysis deals with uncertainty problems, with special care for the demand forecast uncertainty and equipment offer upgrading (in terms of functionality and cost), without neglecting technology, costs, regulation, competition and performance aspects. On the basis of the collection of the possible risk factors for the network development, it classifies them in order of severity. Then a model for the risk description and a method for the risks' effect evaluation is used to understand which is the additional investments that allow a NO to keep under a certain percentage the risk of having demands that the network cannot support.

Dynamic demand evolution deals with incoherence problems due to different rules of resource utilisation between MTP/LTP and STP. The main idea is to suggest some simple and practical rules for resource consumption in case of dynamic demand evolution and to reckon the additional network resources needed when the practical rules are applied instead of the optimal ones.

Incoherence cause	***Reactive countermeasure***
High uncertainty	Risk analysis
Budget limitation	
Different optimisation criteria	
Different resource allocation rules	Dynamic demand evolution studies

Table 7. 5 - Reactive countermeasures against incoherence of network evolution

7.3.3 *An additional task for network planners*

On the basis of the brief discussion of this chapter, it is our belief that a new task related to incoherence issues should be added to the traditional network planner work (distribution of network functionality and network dimensioning). To make it simple network planner should be expected to provide not only the amount of needed resources to carry a certain demand matrix, but also the amount of additional resources that should be deployed to avoid the risk of resource underestimation.

The optical layer, enabling a very high utilisation of the transport resources and providing very high bandwidth communication pipes, seems to be a good lever to enhance the basket of high bandwidth offered services and, finally, a very dynamic telecommunication market. That is why incoherence studies will take an important role in the future planner work.

7.4 Overcome the capacity planning?

The "traditional" approach currently used for planning a network is based on the forecast of the demand of circuits at different bit-rates and the dimensioning of the network layers in order to minimise the required network resources (fibre and equipment). This approach is oriented to the minimisation of the investments required to provide a given amount of bandwidth and is very suitable for situations where traffic increases smoothly.

However, a number of changes both in telecommunication market and in transmission network technology are foreseen that could affect the network planning process. The main drivers to the evolution of the telecommunication market are the strong competition between network operators and the impressive growth of data traffic (mainly IP based), compared to the traditional voice traffic. In the near future, the competition between a number of operators will probably make very difficult to have reliable forecasts of the demand. On the other hand, competition will ask for the realisation of transport network that can provide abundant bandwidth with low cost and short provisioning time. Moreover, some analysts foresee that the reduction in the price of the telecommunication services will positively stimulate the market producing a further increase of the demand. This behaviour is called "price elasticity" and the consequence is that in the future bandwidth will be probably considered a "commodity".

For example, optical networking promises to solve the network operators' problems, because it offers

- Abundant bandwidth: some vendors announce to have soon optical equipment that are capable of multiplexing more than 160 wavelengths onto a single fibre.
- Easier-to-grow networks: optical equipment can accept a signal with different bit rate and frame format on each optical channel, therefore providing carriers with more flexibility in growing their networks. So, carriers can adopt a 'pay as you grow' upgrade strategy.

In this new situation, *the uncertainty in the demand forecast could make the traditional planning approach quite risky, because it produces a network that is optimised for a given demand matrix, but it does not guarantee that the network is simple to upgrade in case the demand is underestimated.*

An alternative approach could be to build a network that can provide a significant excess of bandwidth and is simple to upgrade. The operator could, for instance, install WDM systems that, when fully equipped, can provide a large number of optical channels: initially only few wavelengths are used on each system, then new wavelengths are added, when required.

The main differences between the two approaches can be summarised as follow. In the traditional approach

- the focus is on the planning process,
- the network is optimised for a given demand forecast,
- the network growth is driven by the Short Time Planning process;

while in the innovative approach

- the focus is on the filling of the network capacity,
- the network is not optimised; it is built in order to provide bandwidth in excess,
- the network growth is driven by the amount of used resources.

The two approaches described here are the result of two opposite points of view on the problem, and the selection of the more profitable strongly depends on the characteristics of both the market and available equipment. However it is always possible to combine them in order to look for the solution that better fits the needs of the operator at a certain time.

7.5 Core networks

7.6 Access networks

7.7 Mobile communications networks

7.8 Converged fixed and mobile networks

7.9 Basic network design and optimization algorithms

This is a comprehensive review of basic network design and optimization algorithms that has been adapted from the surveys in [3] and [2].

7.9.1 *Manual Planning*

An expert in network design is likely to produce reasonable results in very short time and without sophisticated modeling and (expensive) tools. Manual planning methods can produce almost optimal results for small-sized networks, and almost any constraint can be incorporated in the design.

7.9.2 *Analytical Models*

In some cases, analytical models (AM) can be applied to network design problems. These analytical models, however, are known to be hard to handle and often rely on assumptions like "independence of events" or "system in steady-state" that do not hold in practice. Sometimes, approximation or relaxation can transform a discrete problem into a continuous one that can be tackled by analytical models. Flow deviation (see e.g. [7]) and queuing theory (see e.g. [8]) are examples of analytical models.

In general, network design problems belong to the field of combinatorial optimization and cannot completely be described by analytical models.

7.9.3 *Linear Programming*

Mathematical Programming (MP) is the core of Operations Research. Almost any network design problem can be formulated as a mathematical program. The interested reader may consult [9][10][11] for an overview.

Linear Programming (LP) provides solvers for the following type of model:

$$\text{minimize } o : \mathbb{R}^n \to \mathbb{R} \quad x \to c^\top x \text{ subject to } Ax \geq b$$
$$c \in \mathbb{R}^n,\ b \in \mathbb{R}^m,\ A \in \mathbb{R}^{(m \times n)},\ \llcorner n, m \in \mathbb{N}\lrcorner$$

If there are non-linear constraints, they have to be transformed into linear approximations, e.g. demand constraints or Economies of Scale. Known solvers are the *simplex algorithm* or *Karmarkar's interior point* method. LP is known to be of polynomial time complexity, the simplex algorithm, however, has exponential worst case behaviour but performs well in the average case. An example of a network design problem modelled as a linear program and enforcing integrality conditions by Branch & Bound (see 7.9.12) is given in [12].

7.9.4 *Integer Linear Programming or Combinatorial Optimization*

Whenever there are integer conditions imposed on the vector x of the equation in Section 7.9.3, i.e. $\llcorner x_i \in \mathbb{N}_0 \lrcorner$, the LP becomes an *Integer Linear Program* (ILP). Some authors use the term *Mixed ILP* if integrality conditions are only imposed on some variables, but here both types of problems will be denoted as ILP. ILP problems are known to be strongly NP-complete in general. Unfortunately, most network design problems fall into this category. The decision whether a facility should be deployed or not is an example of such an integer condition in a model. Solvers for ILP problems are e.g. *Branch & Bound*, which is an implicit enumerative strategy that uses relaxation to prune unprofitable solutions, or Branch & Cut that combines Branch & Bound with cutting plane algorithms (see e.g. [13]). A whole branch of research, *Combinatorial Optimization*, is devoted to the theory for solving problems with integrality conditions. Additional material on this topic can be found in [14] and [13].

In the future, techniques that combine heuristics with ILP are expected to produce very good

results in network design but due to their complexity are limited to medium sized problems. An example can be found in [15].

7.9.5 *Dynamic Programming*

Dynamic programming is a computational method which uses recursion to solve the problem in stages. A problem is decomposed into a sequence of nested sub-problems, and the solution of one sub-problem is derived from the solution of the preceding sub-problem. In this sequentially dependent process, each stage involves an optimization over one variable only. Preceding decisions have to be independent.

Dynamic programming applies to some sub-problems of network design, e.g. finding the cheapest realization of a certain capacity requirement given a set of discrete capacities [16]. In some applications, dynamic programming is applied to multi-stage planning problems.

7.9.6 *Meta-Heuristic Optimization*

Due to the intractable complexity of optimal solvers for Combinatorial Optimization problems, the field of *Meta-Heuristic Optimization* (MHO) has developed. Unlike the constructive heuristics of 7.9.20, MHO-algorithms are not restricted to one specific problem in general. The applicability of MHO-algorithms to an arbitrary optimization problem depends on coding the problem in an appropriate representation.

Many of the MHO-algorithms are motivated by analogies in nature, e.g. *Evolutionary Algorithms*, *Simulated Annealing* or *Tabu Search*. In contrast to ILP-solvers like Branch & Bound, MHO algorithms do not guarantee optimal solutions, but are known to be near optimal under certain conditions.

Due to their eminent importance, *Randomized Search Techniques* (RST) as a special case of MHO are regarded separately. Examples of RST are Simulated Annealing, Genetic Algorithms, and Randomized Tabu Search.

General characteristics of RST are:

- RST are likely to produce near optimal solutions
- RST can be tuned to produce very fast solutions (compared to ILP solvers like Brand and Bound) even for large problem sizes
- RST have difficulties with incorporating arbitrary constraints

A successful application of Simulated Annealing and Genetic Algorithms to network design problems is presented in [17].

7.9.7 *Approximation Algorithms*

The theory of *Approximation Algorithms* (AAs) is a comparatively young field in theoretical computer science. It is concerned with finding good algorithms, i.e. of polynomial time complexity, that yield (sub-optimal) solutions to NP-complete problems, which are bounded by being at most *δ(n)* times worse than the optimum solution (see e.g. [18] for an introduction). *δ(n)* may be a constant or depend on the problem size *n*.

The theory of AAs reveals an interesting structural classification of network design problems of size $n := |V_e|$:

- no reasonable polynomial time approximation algorithm exists, i.e. *δ(n)* grows faster than polynomial with *n*
- a *δ(n)-AA* exists and $\delta(n) = O(n^x)$ with $x \in \mathbb{R}^+$
- a δ(n)-AA exists and $\delta(n) = O(\log n)$
- a δ(n)-AA exists and $\delta(n) = c,\ c \in \mathbb{R}^+, c > 1$ is independent of *n*
- a polynomial time approximation scheme exists, i.e. for every $\delta > 1$ there exists a polynomial time approximation algorithm.

A promising application of approximation algorithms may be the generation of lower bounds.

An example of an O(log(|Ve|)) - approximation algorithm solving a minimum-cost network design problem is presented in [19], some further approximation algorithms for sub-problems of network design can be found in [20].

7.9.8 *Artificial Intelligence*

Network designers have to be experts in their field. The knowledge of an expert is valuable in many situations of the network design process. Knowledge processing and engineering belong to the field of *Artificial Intelligence* (AI) (for an overview see e.g. [21] and [22].

As already mentioned, manual planning by "rules of thumb" is one way of solving network design problems. Rule or case based inference systems (see e.g. [23]) using expert knowledge can serve as a decision support system, offering assistance to the network designer. Decision support systems in general provide assistance by provision of relevant information, reasoning with information to provide arguments for possible decisions, and identifying qualifications, ramifications or risk associated with possible decisions. A rule based expert system for network design problems can be found in [24], an example of a more general inference system based on probability theory is described in [25].

Neuronal Networks that are capable of forecasting water and power consumption should be able to forecast network traffic as well, but there is still some additional research necessary (see e.g. [26]).

(Fuzzy) Logic, Evidence Theory and Probability Theory are the formalisms available in AI. Fuzzy concepts could for example be used in the objective of network design problems representing "favourable" or "non-favourable" properties. Fuzzy clustering is another technique that was already successfully applied to network design problems [27][28].

7.9.9 *Scalable Architectures*

In a highly structured and regular network design problem, regular graphs, such as hypercube, shuffle exchange networks and DeBruijn Graphs (for an overview see e.g. [29] and [30]) may be an alternative to classical network design algorithms. Regular graphs are advantageous in routing and scalability and, depending on their type, have guaranteed upper bounds on the number of hops, capacity etc. Network designs that depend on regular or structured graphs are called scalable architectures (SCA).

Above all, WDM-technology with its almost unlimited bandwidth opportunity is a candidate for modelling light-wave networks as regular graphs. Examples can be found e.g. in [31].

7.9.10 *Hybrid Combinations*

Naturally, between the single modelling and solving-techniques no crisp line can be drawn. Most successful solvers for network design problems are combinations of different techniques, i.e. hybrid combinations. In many cases heuristics are used to speed up convergence. In the case of problem decomposition different solvers can be applied to each sub-problem by choosing the most promising solver.

7.9.11 *Enumeration*

By enumeration is meant the crude approach of evaluating all possible permitted solutions. In general there is a huge number of permitted solutions for a certain network planning problem, each of which needs a complete dimensioning process to have the needed network description to be evaluated. It is clear that enumeration is only possible for problems with a very limited number of permitted solutions and a limited set of simple evaluation criteria.

7.9.12 *Branch and Bound*

Branch and bound is a technique to speed up the enumeration. Crucial for this to use some methods to get lower bounds of the evaluation criteria for the optimal solution. By use of this it is not needed to evaluate all possible permitted solutions, some of them can be excluded with help of the established bounds. If a subset of network elements $e \in E_i \subset E$ set the values of z_e to 0 or 1, we might find that independent of the values of the other 0-1 variables the cost of the solution will be higher than an already found solution. In this case we can short-cut the enumeration without evaluating the cost for each individual combination of values for the variables z_e with $e \in E_i \backslash E$. The lower bound can e.g. be found by relaxation techniques. Although the branch and bound solution method considerably can reduce the number of solutions to be cost evaluated, the remaining number will often still be so large that it heavily reduces the size of the problems that can be handled.

7.9.13 *Relaxation techniques*

Relaxation is a technique to simplify a constrained optimization problem. The idea is simply to remove some constraints or to substitute them by other looser constrains. The problem

$$P = \underset{x \in S}{Min}\, F(x)$$

with $S \subset \Re^n$,

could thus be replaced by an another more easily solved

$$RP = \underset{x \in S'}{Min}\, F(x)$$

where $S \in S'$.

Normally the solution of RP is not a feasible solution to P so we only get a lower bound P ≥ RP. To get a feasible solution we then need some additional method. This might e.g. be a heuristic algorithm starting from the solution of RP. The lower bound will then give information on the quality of the feasible solution obtained, i.e. a limit on the distance to the true optimum. Normally we can not obtain a feasible solution with the same cost as the best lower bound from the relaxation, instead there will be a certain *relaxation gap*.

To reduce the relaxation gap it is often essential not only to remove constraints but also add new constraints. These constraints should be redundant in the original problem P while strengthening the relaxed problem RP. It is also important that they do not complicate the solution of RP. The two types of relaxation most frequently used are LP-relaxation and Lagrangean relaxation.

LP-relaxation is applied in problems with integrality constraints and substitutes such ones with linear constraints. The constraint $z_e \in \{0,1\}$ may thus be substituted by $0 \le z_e \le 1$. A *mixed integer linear programming* problem is except for the integrality constraints pure linear, and after relaxation can be handled by an ordinarily LP method such as *simplex*.

In LP-relaxation we have some special conditions for the approach of adding constraints to strengthen the relaxation. In this case the convex hull of the feasible set *S* of the problem P forms a polyhedron, that also can be defined by a set of linear constraints. To find these *facet-defining inequalities* requires an analysis specific to the problem P. The gain is the possibility to get an LP-problem with a solution that is also feasible and thus optimal to the original problem P. *The cutting plane algorithm* is a practical way to iteratively introduce strengthening constraints [32][33].

In Lagrangean relaxation we do not only modify the set of constraints but also introduce new variables, *Lagrangean multipliers*. The problem

$$P = \underset{\substack{x \in S \\ Ax \le b}}{Min}\, F(x)$$

is charged to

$$LRP(\lambda) = \underset{x \in S}{Min}\, F(x) + \lambda^T (Ax - b)$$

where $\lambda \ge 0$.

As the additional term in the cost function in LRP(λ) is negative for all x that are feasible solutions to P, we will have P ≥ LRP(λ). In order to have a lower bound as tight as possible we get a dual optimization problem

$$\underset{\lambda > 0}{Max}\, LRP(\lambda)$$

To solve the dual, one can use ascent methods or sub-gradient optimization techniques. There has been recent research on improved methods for the dual optimization, such as *bundle method* [34], where the collected information from all obtained sub-gradients is utilised.

7.9.14 *Marginal improvement techniques*

Marginal improvement or descent method is a very natural method. It is sometimes also denoted by neighbourhood search, but following the terminology of [35] we consider it only as the simplest example of the *neighbourhood search*. The latter is more general and includes the three following techniques, simulated annealing, tabu search and successive smooth approximation that all tries to overcome the limitations of the marginal improvement.

Suppose that we for each solution **x** to P have a set of solutions V_x that relatively easily can be obtained from **x**. The marginal improvement starts with a feasible solution **x** and then generates improved solutions by iterating

Search a **y** $\in$ V_x with F(**y**) < F(**x**)

If such a **y** is found set **x** = **y,** otherwise stop

In this way the marginal improvement will always find a solution that is locally optimal with respect to the type of neighbourhood V. In order to find a good solution it is essential that the neighbourhoods tested are large enough in order to avoid a large number of local minima of highly varying costs. At the same time this makes the search more time consuming. The definition of neighbourhoods thus is crucial and it is also specific to the type of problem P.

A special version is the *greedy* marginal improvement technique. Here the iterative step is instead defined by

Search a **y** $\in$ Vx that minimizes F(**y**)

If F(**y**) < F(**x**) set **x** = **y**, otherwise stop

7.9.15 *Simulated annealing*

In order to avoid the effect of the marginal improvement that it could stop at a local minimum that is far from the global one, the simulated annealing also can allow changes that increase the cost.

As its name implies, the Simulated Annealing [36], exploits an analogy between the way in which a metal cools and freezes into the minimum energy crystalline structure and the search for a minimum in a more general system. The method is a technique that has attracted significant attention as suitable for optimization problems of large scale, especially ones where a desired global maximum/minimum is hidden among many, poorer, local maxima/minima. Surprisingly, the implementation of the algorithm is relatively simple.

This algorithm is based upon of that of Metropolis et al, [37]. The algorithm employs a random search, which not only accepts changes that decrease the objective function, but also some changes that increase it. The following elements must be provided:

- A representation of possible solutions or system configurations. The configurations can be represented as a set of configuration parameters $\{x_1, x_2,..., x_n\}$
- A generator of random changes in solutions. The solution generator should be able to introduce small random changes and allow reaching all possible solutions. Usually, the changes are done by modifying any of the configuration parameters.
- A means of evaluating the problem functions, that is, an objective function E whose minimisation is the goal of the procedure. The evaluation of the objective function is essentially a "black box" operation as far as the optimization algorithm is concerned.
- An annealing schedule - an initial temperature T and rules for lowering it as the search progresses. This requires mastering the fundamental physics of the process or on the other hand to follow a trial-error method. First some random rearrangements are generated and used to determine the range of values of ΔE that will be encountered from move to move. Then, it is chosen a starting value for the parameter T, which is considerably larger than the largest ΔE normally encountered. Afterwards, it is proceeded

downward in multiplicative steps each mounting to a 10 percent decrease in T. Each new value of T constant is hold for, say 100N reconfigurations, or for 10N successful reconfigurations, whichever comes first. When efforts to reduce E further become sufficiently discouraging enough, it is stopped.

Figure 7. 7 shows the simulated annealing algorithm's basic structure.

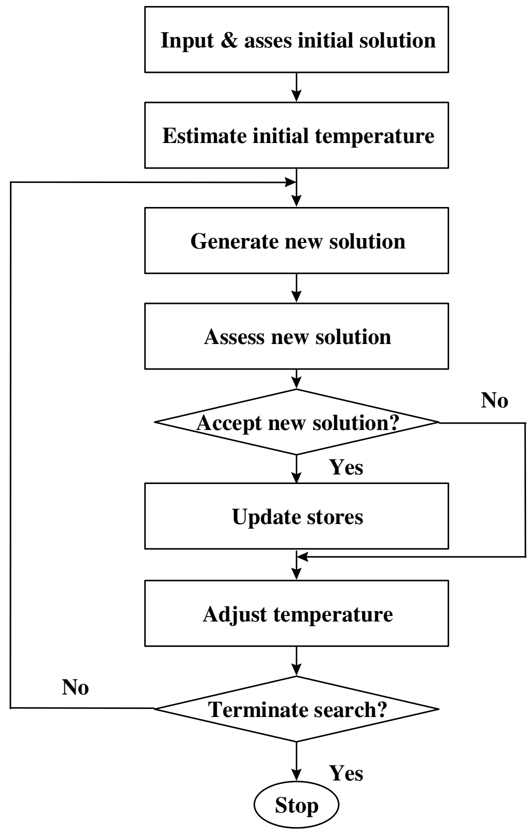


Figure 7. 7 - Simulated annealing algorithm

The effectiveness of simulated annealing is significantly determined by the definition of domain-dependent neighbourhood function. In addition, the cooling schedule sometimes plays an important part. If the temperature is reduced too rapidly, it may not increase the probability of finding better solutions. On the other hand, the slower the temperature is reduced, the longer the time the algorithm will take to determine.

Although the simulated annealing cannot guarantee that we do not reach a bad local minimum it generally gives good results. During the optimization process it is more likely that the current solution will be in an area of low costs than in one of high. Simulated annealing is simple to implement and the neighbourhoods tested can be made simpler than for the marginal improvement, so it has become popular for combinatorial optimization. Drawbacks are the requirement for CPU time and that it does not provide any simple way to look for additional improvements to a good already known solution.

7.9.16 *Tabu search*

Tabu search (cf. [38] and [39]) is a recent neighbourhood search method for overcoming the limitations of marginal improvement. The approach is much greedier than that for simulated annealing and follows at the start the marginal improvement technique. It is also used for optimization applications in which near optimal solutions are acceptable. Like simulated annealing, the aim of the tabu search is also to escape local optima.

The main idea is to explore the search space of all the possible solutions by a sequence of movements. Thus, the selection of one solution to another (named move) will be the best of the possible ones. A tabu list disallows certain moves from taking place. For example, it could be the list of the previous k states, which have been visited and disallowed from being revisited by the current move. It can also be restrictions on the direction of moves. After each move is made, the tabu list will be updated. Different tabu searches may use different strategies to manipulate the tabu list.

The choice of which moves are forbidden is based on the short-term and long-term history of the sequence of moves. For instance, a very simple implementation would be to declare a move as tabu if its inverse has been produced recently (short-term history) or many other times before (long-term history). However, a move can be considered as tabu if it is judged favourably.

The tabu method starts with an arbitrary solution $s \in S$ that has a set of moves $M=\{m_1,....,m_k\}$. The result of the moves to the solution drives to k different solutions $M(s)=\{m_1(s),...,m_k(s)\}$. From those solutions, the feasible ones compose the neighbourhood $N(s) \subseteq M(s)$ of the solution s. Inside the neighbourhood, some solutions are labelled as tabu, $T(s) \subseteq N(s)$ and among them some solutions are selected as valid solutions despite of being tabu. These tabu and valid solutions are declared as candidates, $A(s) \subseteq T(s)$. Afterwards, it is chosen the solution that minimises the cost function among the non-tabu solutions and the candidates ones $((N(s)-T(s)) \cup A(s))$.

It is wished to avoid the convergence to local minimum. The algorithm should avoid the moves to solutions where the cost function increases instead of decreasing. That is achieved by choosing the criterion to label a solution as tabu or candidate as a function of the problem to be optimised.

The effectiveness of tabu search depends solely on the way that the tabu list is defined and manipulated. Since there is no limit to how that is done, tabu search is a very general strategy.

Although tabu search is somewhat more complicated to implement than simulated annealing, the principles are general and are not unique for any specific problem. It appears a promising technique when considering the merits and drawbacks of simulated annealing.

7.9.17 *Successive smooth approximation*

This is a newly suggested method [40] with a similar general approach as simulated annealing. However there are no random variables affecting what changes of the solution are accepted. Instead the target function is successively manipulated in a deterministic way, and the changes accepted are those favourable with regard to the current target function. The idea is that the target function in the beginning should be close to linear (a small second derivative) reflecting global properties, while during the process finer and finer details are included, providing a convergence to the true target function.

For a separable target function $F(x)=\sum_e f_e(x_e)$ with $x_e \in \Re$, the manipulated target function can be defined by $F_\delta(x)=\sum_e f_{e\delta}(x_e)$, where $f_{e\delta}$ is given by a convolution $f_{e\delta} = f_e * f_\delta$ and $h_\delta(t)=\frac{1}{\delta}h(\frac{t}{\delta})$ with h e.g. selected as

$$h(t)=\begin{cases}1-2t^2 & \text{for } |t|\le \tfrac{1}{2}\\ 2(|t|-1)^2 & \text{for } \tfrac{1}{2}\le |t|\le 1\\ 0 & \text{for } 1\le |t|\end{cases}$$

The procedure starts with a large δ that gradually is decreased to 0. As during a certain phase of the procedure the solution updates are made according to a target function only slowly changing, we can expect that they will be coordinated so that also a number of small updates can cause a considerable change to the current solution. This is in sharp contrast to the random nature of simulated annealing, and we can expect that successive smooth approximation would need a much lower number of iterations. On the other hand the evaluation of the cost function gets more complex, instead of evaluating e.g. a piece wise linear function f_e one would need to make a number of calculations of a 4th degree polynomial to evaluate $f_{e\delta}$.

7.9.18 *Simulated assignment*

Simulated Allocation (SAL) is a generic stochastic combinatorial optimization approach with certain similarities to Simulated Annealing. The SAL algorithm generates a trajectory of a discrete time stochastic process, called the allocation process. The aim of the allocation process is to provide network capacities resources to realise transmission capacity demands. The main advantage of the method, that different allocation rules (e.g. taking into account path diversity) can be specified and applied for the allocation process.

Each state $x\in X$ of the state space X is a flow pattern. State x is maximal if for each demand $d\in D$ all its demand capacity units (DCU) are allocated. The time epochs (steps) i=0, 1, 2, ... of the process correspond to consecutive arrivals and disconnections of DCUs (or modules of DCUs) from all the demands.

The pseudo-code for the SAL algorithm is given below (where "*allocate*" and "*disconnect*" identify the allocation and disconnection procedures, they both change the current state "x").

For a constant allocation probability function ($q(x)=q_0$) the expected number of steps required to reach a maximal allocation state from a state with exactly l allocated units is equal to $(L-l)/(2q_0-1)$, provided $q_0>1/2$ (where L is the total number of DCUs to be allocated).

```
begin
step:=0; min_cost:=∞; x:=0;
repeat
     step:=step+1;
     if   random<q(x)  then  allocate(x)  else  disconnect
(x);
     if |x|=L and current_cost < min_cost then
          begin minc_cost:=current_cost; x_opt:=x; end
until step=step_limit or min_cost=cost_lower_bound
end
```

An effective procedure for generating the allocation trajectory is obtained with the use of the allocation vector A[j] (j=1,2,...L). For state x^i in time epoch i, the constant length L of A is given L=N(i)+M(i), where N(i) the current number of DCUs still waiting for allocation, and

M(i) it the total number of DCUs already allocated in state x^i . In each state x^i the first N(i) entries of vector A describe the DCUs to be allocated (by specifying for each unit the demand d to which it belongs), and the last M(i) entries describe the currently allocated DCUs (by specifying for each unit its demand d and path $p \in P(d)$).

Suppose the trajectory is in epoch i. If the next event is a DCU arrival (this happens with probability $q(x^i)$, one of the records A[1], A[2],...,A[N(i)], say record j corresponding to a unit of demand d, is chosen at random from a uniform distribution. Then the allocation rule is applied to choose a path $p \in P(d)$, and the unit is allocated by appropriate adjusting the current flow function and by interchanging the record A[j] and A[N(i)], writing additionally to A[N(i)] the identification of the chosen path p. If the next event is a disconnection, one entry from the sequence A[N(i)+1], A[N(i)+2],, ... A[L] is selected at random from a uniform distribution, a DCU corresponding to it is disconnected and the chosen entry is interchanged with A[N(i)+1].

7.9.19 *Genetic programming*

The genetic algorithms (GA) (cf. [41],[42], and [43]) attempt to simulate the phenomenon of natural evolution first observed by Darwin and elaborated by Dawkins. In natural evolution each species searches for beneficial adaptations in an ever-changing environment. As species evolve these new attributes are encoded in the chromosomes of individual members. This information does change by random mutation, but the real driving force behind evolutionary development is the combination and exchange of chromosomal material during breeding.

GAs differ from traditional optimization algorithms in four important aspects:

- They work using an encoding of the control variables rather then the variables themselves.
- They search from one population of solutions to another, rather than from individual to individual.
- They use only objective function information, not derivatives.
- They use probabilistic, not deterministic, transition rules.

Of course, GAs share the last two attributes with Simulated Annealing and have found applications in many of the same areas. The basic structure of GA is shown in Figure 7. 8. One minor change from the standard optimization flow diagram is the use of the word “population” rather than “solution”. A major difference is that the usual operation of generation of a new solution has been replaced by three separate activities -population selection, recombination and mutation.

Candidate solutions must be encoded by binary bit strings (named as chromosomes). This is suitable for integer and decision variables while continuous control variables must be approximated to integer variables. Then an initial population of chromosomes is selected randomly or using specific information about the problem to solve. Within the algorithm, population selection in each generation is based on the principle of survival of the fittest through a formula that represents the nature of the problem to solve. The best-fitted chromosomes are selected to produce descendence that will constitute the next generation and that will inherit the best features of both progenitors. After many generations, it is expected that the final population will more adequate than the original one.

It is in the recombination phase that the algorithm attempts to create new and improved solutions. The key procedure is the crossover in which GA seeks to construct better solutions by combining the features of the good existing ones. The crossover consists of cutting off the parent strings at a random point and exchanged. Strings produced in this way are subject to occasional mutation, in which random elements of the string may change values. The purpose of the mutation stage is to provide insurance against the irrevocable loss of genetic information and hence to maintain diversity within the population. For instance, if every solution in the population has a 0 as the value of a particular bit, then no amount of crossover will produce a solution with a 1 instead.

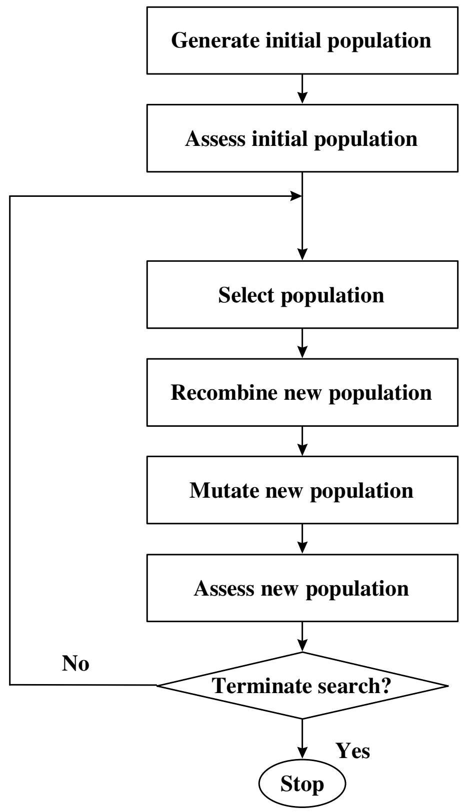


Figure 7. 8 - Genetic algorithm procedure

The evaluation of the new strings is essentially a "black" box operation as far as the GA is concerned. In that way, the new population can be accepted or rejected. This process is repeated until satisfactory schedules have been generated or resources exhausted.

In general GA require a non-trivial number of iterations to find good quality solutions. This means that, in general, a non-trivial amount of time is needed before they can be expected to produce good results. Sometimes, it is difficult to converge to a global solution, as there is a danger of all the strings having the same building blocks. On the other hand, they allow searching in a wide space. Domain knowledge plays an important role in the representation as well as the choice of the many parameters and operations.

Genetic programming could be implemented as a post-optimiser to any network structure optimization method that can produce a variety of solutions. If the problem is formulated as a multi-commodity flow problem with possible additional flow constraints each only related to the flow on one edge, we can use the following approach:

> for the two solutions to be mated we look for a cut (a set of network elements separating the network in two) where the two solutions have identical flows while the solutions are different for both sub-networks on either sides of the cut. In this case we

can create two new feasible solutions by combining the flows on one side of the cut from one solution with the flows on the other side of the cut from the other solution.

7.9.20 *Heuristic algorithms*

They are an alternative to operations research techniques. They are based in the extraction of practical rules from the way an expert solves a specific problem (e.g. first consider the bigger demands and then the smaller ones). These rules may not have a mathematical proof as they are based in good results obtained in the practice. These rules define a "path" for solving the problem. Although initially may be considered as a simplistic approach for solving this kind of problems, they are surprisingly spread in the optimization world; as an example, they have been proposed in the dimensioning of intelligent network, [43]. The heuristic algorithms may be combined with other optimization techniques for reducing the complexity of the problem.

Their main features are derived from their practical basis:

- To provide a solution in a reasonable execution time (polynomial order according to the problem size).
- Not to guarantee the optimum solution for every situation.
- To achieve less optimum solutions that despite being away of the optimum one has certain characteristics that can be labelled as acceptable. This will depend on the quality of the heuristic approaches.

Due to their practical basis, the quality of the heuristic will be reinforced if it is possible to estimate a percentage of the error.

Constructive Heuristics (HE) are used most intensely in network design algorithms. In general, constructive heuristics have the following characteristics [2]:

- To achieve less optimum solutions that despite being away of the optimum one has certain characteristics that can be labelled as acceptable. This will depend on the quality of the heuristic approaches.
- they are easy to understand
- they can be adapted to changing requirements with less effort
- they produce feasible and "sufficiently good" solutions
- they have acceptable time complexity
- they allow interaction with the designer
- they usually give no lower bounds on the optimum solution

Center of Mass [45], Unified-Algorithm [46], Cut Saturation [47] and MENTOR [[48], [49] are examples of well-known and intensely used heuristics in network design.

ANNEX 1 – Identification of Different Planning Processes

It is proposed to add this contribution ***to an annex of the manual****.*

This annex is to illustrate an implementation of identifying different optical transmission network (OTN) planning processes [2].

A planning process is a sequence of consecutive steps and iterations to elaborate network plans, or, in more details, to elaborate alternatives for different planning problems, to evaluate them and make decisions to choose among them. Several planning processes have been identified that deal with combinations of the different planning sub-problems, which have been described before, such as dimensioning a network cluster, dimensioning a network structure, optimisation of a network structure and optimisation of the physical level. These problems are formulated with a similar approach of the sub-problem descriptions.

As it has been described in Section 7.1.1, Chapter 7, the complexity of the planning processes makes necessary to split the planning problem into several solvable size tasks (sub-problems).

The different sub-problems are normally fully coupled; that is, it exists a meshed interrelation between them. Different alternatives may be thought for dealing with the large problem size and complexity.

The first one is the translation of the meshed relation of the sub-problems into a linear chain of sub-problems that are solved separately. So, several interrelations between sub-problems are ignored, only sub-optimal solutions are obtained. The less important the discarded interrelations, the better the sub-optimal solutions. In consequence, a heuristic component is introduced in the decision of the discarded interrelations. For reducing the effect of the discarded interrelations, feed-backs, iterations and tolerance analysis may be introduced between the different sub-problems. In spite of this drawback, the decomposition is wildly applied in practice. This alternative may be called as the **sequential decomposition approach** when no iteration is performed, and simply **decomposition approach** when some iterations are present.

A different strategy is the application of gradually refined models. In this case, the size and the complexity of the planning problem are reduced by the simplification of the applied models. Practically, some of the sub-problems are modelled in a simplified way and only the more important sub-problems are more precisely modelled. Once more, a heuristic component is introduced in the decision of the sub-problems to be simplified and the degree of the simplification of each one. Different solutions are obtained according to the previous used criteria; the less important the simplified problems, the better the obtained solutions. This alternative may be called as the **gradual-simplification approach**. An example for this approach may be seen in the location of clusters (ring and meshes) problems. It is very hard to jointly optimise the location of domains and the routing and grouping inside each cluster, but it is possible to simplify the problem by using the gradual-simplification approach. In this case, a possible alternative is to prepare a function for costing a domain in terms of the number of nodes and demands but without performing in details the routing and grouping; with this function it is possible to focus in the domain location optimisation problem without losing computation time in the grouping and routing problems.

A third approach is to use both of the previous approaches in a combined way. In this case, some interrelations between the different sub-problems are discarded and some sub-problems are modelled in a simplified way. This alternative may be seen, as well, as solving a set of the sub-problems with the gradual-simplification approach and discarding the relations of these sub-problems with the other ones. An example for this alternative may be seen in the mesh topology optimisation problem. To evaluate a candidate topology the whole dimensioning problem should be solved. Simplified models for the routing, grouping, stand-by network planning are used. In this situation the topology identification, which is the core of the method, can be speed up. After the topology is identified, the routing, grouping and stand-by network problems may be solved in a more accurate way. In this approach, the interrelations between the topology and the other three sub-problems (routing, grouping and stand-by network) are discarded.

In the following sections some planning processes are described coherently with the proposed atomic breakdown planning approach.

A1.1 Dimensioning a network cluster

Network architecture prototypes are applied to realise different network clusters. Different network architectures are based on different routing and protection mechanisms, node functionality and node architectures. Each specific architecture has its own specific dimensioning problems. In a general description the main points are not the details of the different specific features but the identification of the general characteristics of the dimensioning of architectures.

To dimension specific network architectures routing, grouping and protection planning sub-problems should be solved. However, depending on the planning model the routing, grouping and protection planning sub-problems can be solved separately or jointly.

General Problem

- to elaborate the full dimensioning of a specific network cluster
- the problem is combination of the following elementary sub-problems
  - routing
  - grouping
  - stand-by planning
  - transmission system allocation
  - equipment allocation and evaluation

Input

- network cluster
  - nodes
  - transmission demands
  - topology in the physical level
  - architecture prototype
- engineering rules to dimension the architecture prototype

Output

- dimensioned node configurations
- dimensioned transmission systems and multiplex bundles

Target function

- minimal cost

Restrictions

- routing, grouping and protection/restoration restrictions

The practical implementation of the planning process that allows dimensioning a network cluster depends on several factors like:

- the cluster structure,
- the type of demands: normal demands or demands that require an extra protection,
- the involved network layer(s),
- the adopted grooming strategy,
- the implemented recovery mechanism provided by the cluster.

As a consequence different initial hypotheses imply different cluster dimensioning planning processes. In this subchapter the most important classes of cluster dimensioning problems for a SDH network are taken into account.

A1.1.1 Protected meshed network dimensioning

Figure A. 1 applies the breakdown approach to the planning process of dimensioning a meshed network with the following characteristics:

- local or intermediate grooming strategy,
- LO protection scheme,
- given topology of the HO layer network,
- LO native demands without extra protection.

The local and intermediate grooming strategies allow the planner grouping the LO demands into HO containers after the identification of the working and the protection routes. When the routing of the demands is performed in a sequential way (demand by demand), the protection routes can be found in a sequential way as well. In this case, typically, the routing and stand-by sub-problems are solved together in an interleaved way. In addiction to the protection routing table, the solution of the stand-by planning sub-problem can identify some critical conditions in the cluster, such as no link/node disjoint protection path for one or more demands. In order to solve the routing and stand-by sub-problems in an optimal (i.e. the cheapest) way it is necessary to know the cost of the LO demand transport over the HO resources, i.e. it is necessary to have a projection of the HO cost in the LO layer. These cost figures are given by a first solution of the costing function, which, generally, is applied in this stage in a very simplified way: often the cheapest route is simply the shortest one or the one with a minimum number o crossed nodes.

After the solution of the routing, stand-by and grooming sub-problems the transmission system and equipment allocation sub-problems can be solved. As in this process we, practically, de-couple the LO and HO network layers the transmission system allocation only regards the HO layer, while two instances of the equipment allocation sub-problem are required: one for the LO and the other for the HO pieces of equipment. Finally the costing sub-problem solution (not simplified this time) allows evaluating the cost of the planned network.

This general dimensioning process can be modified in several ways: the main ***alternatives*** are described and discussed in the following.

The *routing* and *stand-by sub-problems* are sometimes *solved together* (i.e. a merging relationship is applied, cf. Figure A. 2). In fact

- the additional complexity due to this merging action can be kept under control through some well-known (and still simple) algorithms,
- a common solution leads to a more optimal solution.

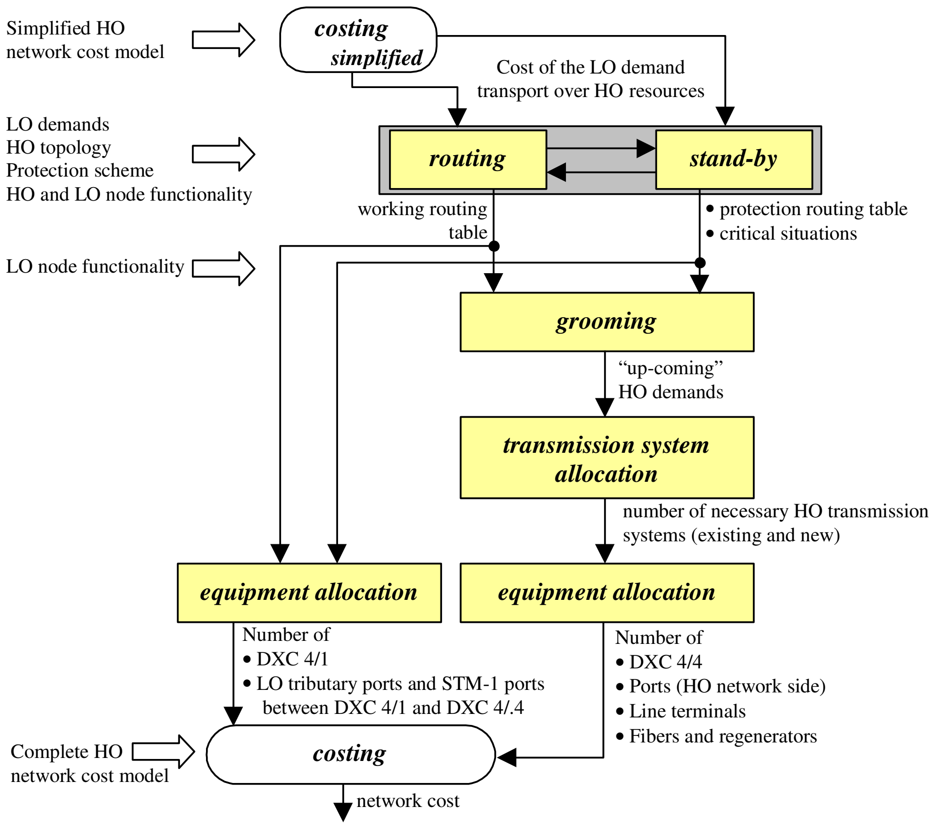


Figure A. 1 - Dimensioning a protected meshed network

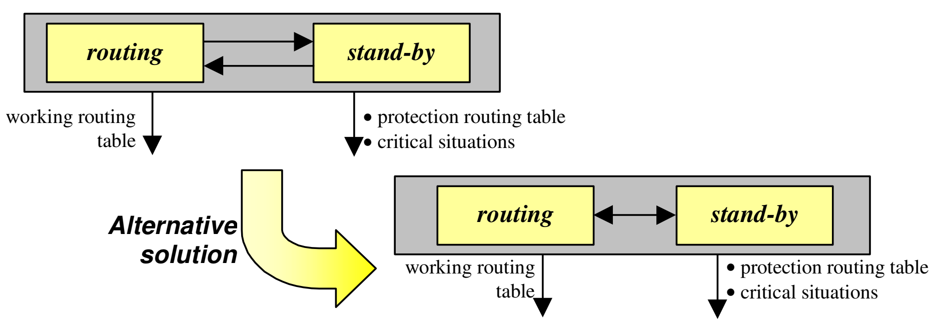


Figure A. 2 - Merging the routing and the stand-by network planning sub-problems

A simple example of this solution is the usage of a modified version of the Dijkstra algorithm to find a minimum weight (note that "weight" generally means "cost" in planning processes) pair of disjoint paths for each demand. First of all let's suppose to keep the two sub-processes separated. For a certain demand the Dijkstra algorithm can solve the routing sub-problem; the same algorithm can be used to find the link/node disjoint protection route (after the modification of the weight of some links in the network). In this way two routes are identified: the working one has the minimum weight, while the protection one has the minimum weight among all the possible link/node disjointed paths. But searching these two routes together often allows finding two disjointed routes with a less sum of weights: neither of the two identified routes is the minimum weight one, but the cumulative weight of the two paths is the minimum (i.e. cheapest) one.

When an *end-to-end grooming strategy* is applied, the order of the grooming, routing and stand-by sub-problem can be modified. In this case, typically, the protection is performed in the HO layer: the grooming sub-problem is solved first and provides a HO demand matrix starting from the LO one; hence the routing and stand-by sub-problems act on the HO demands (cf. Figure A. 3). Note that in this case there is no need of the costing function at the beginning of the process: demand grooming is performed in a deterministic way (by destination and, maybe, by service) and the routing and stand-by sub-problems work at the HO layer using directly the HO network cost model.

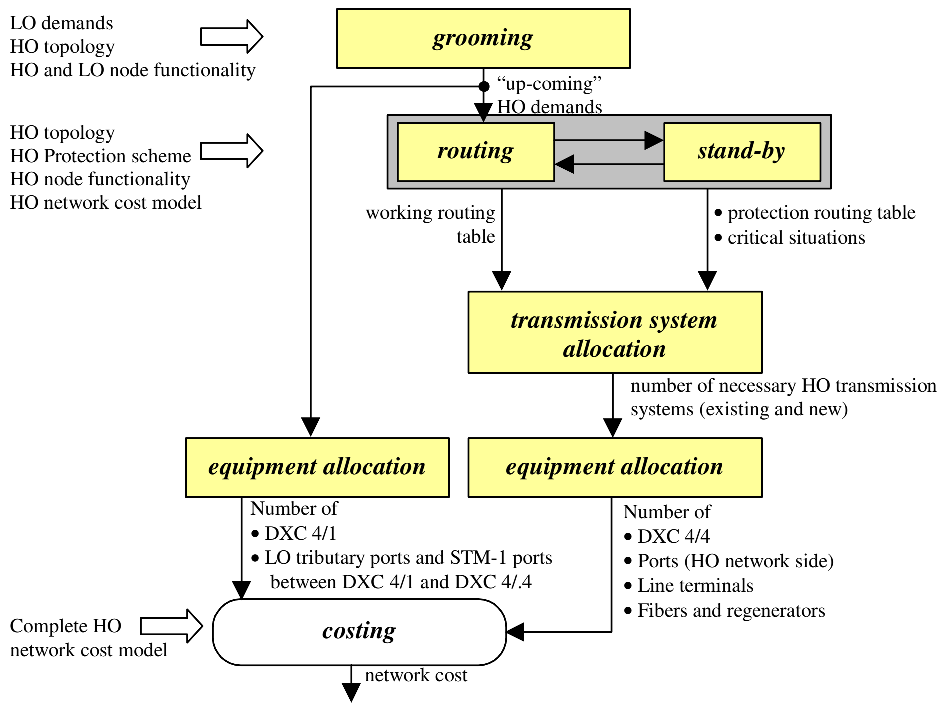


Figure A. 3 - Dimensioning a protected meshed network with end-to-end grooming

A complication of the planning process happens when *both HO and LO native demands* are present. Different alternative dimensioning processes can be used according to different possible policies for the HO and LO demand protection. Figure A. 4 shows one of these alternatives that corresponds to case in which two different protection mechanisms are present, one at the LO layer for the LO native demands and one at the HO layer for the HO native demands. In this case practically the native VC-4 are selectively protected in the HO layer network.

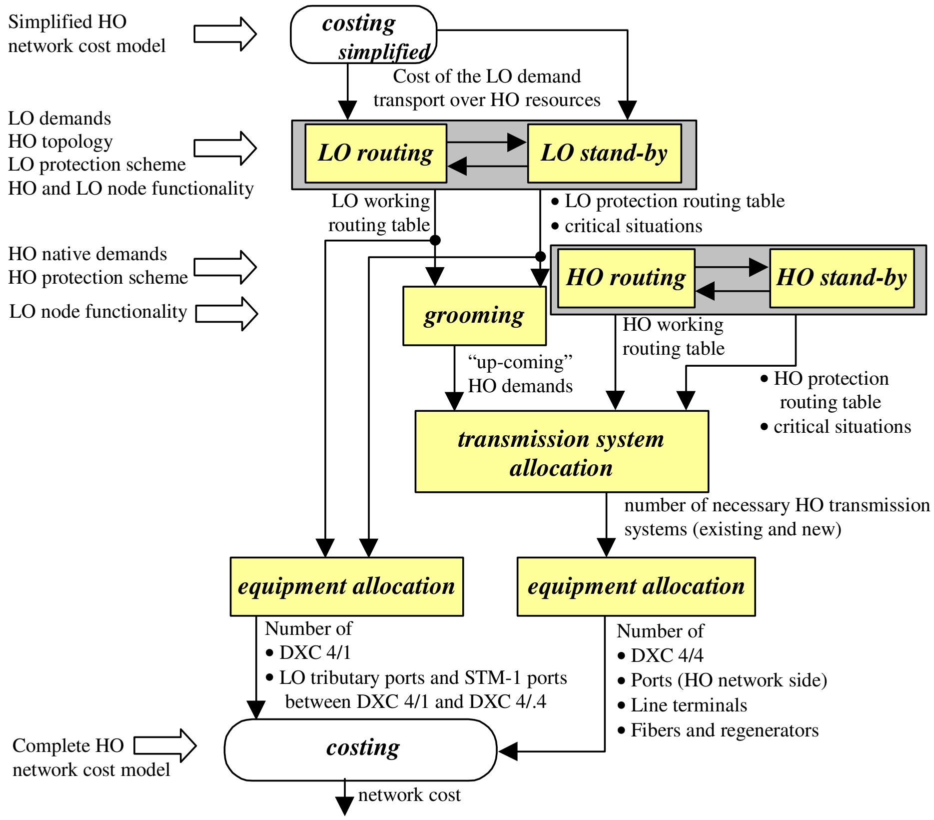


Figure A. 4 - Dimensioning a protected meshed network with HO and LO native demands and HO and LO protection mechanisms

A different alternative with a single protection mechanism in the HO layer network is illustrated in Figure A. 5. In this case all the VC-4s are protected in the HO layer network both when they are native demands and when they are the result of the grooming of VC-12s/VC-3s and, practically, the LO demands inherit the VC-4 protection.

A1.1.2 Restored meshed network dimensioning

Two characteristics of the restoration mechanisms applied to an SDH network affects the dimensioning process:

- restoration is slower than protection,
- restoration is often applied in a sequential way to one disrupted path connection at a time.

These characteristics make the restoration well suited to the HO layer network, which has less disrupted path connections than the LO layer in case of network failure.

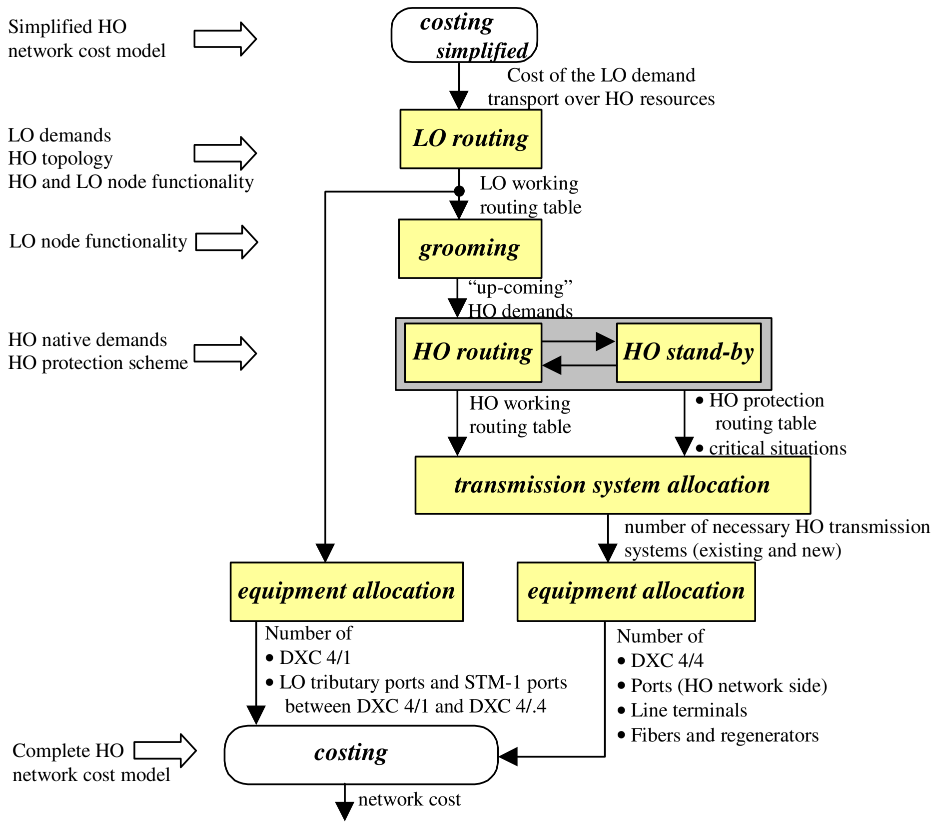


Figure A. 5 - Dimensioning a protected meshed network with HO and LO native demands and HO only protection mechanisms

The typical dimensioning process for a restored meshed network is illustrated in Figure A. 6. At a first glance this process is simpler then the one applied to a protected meshed network. In fact in case of restoration the routing and the stand-by sub-problems are clearly divided: first all the demands (VC-4s) are routed, then their restoration paths are identified. Actually, in spite of this supposed simplification, the overall dimensioning process is more complex in case of restoration than in case of protection. The reason for that is the higher complexity of the HO stand-by sub-process, which, in case of restoration, generally has

- to simulate all the expected failures,
- to identify all the disrupted HO path connections,
- to re-route the disrupted HO path connections on the HO layer network, keeping track of the cumulative resource requirements on each element of the HO layer network.

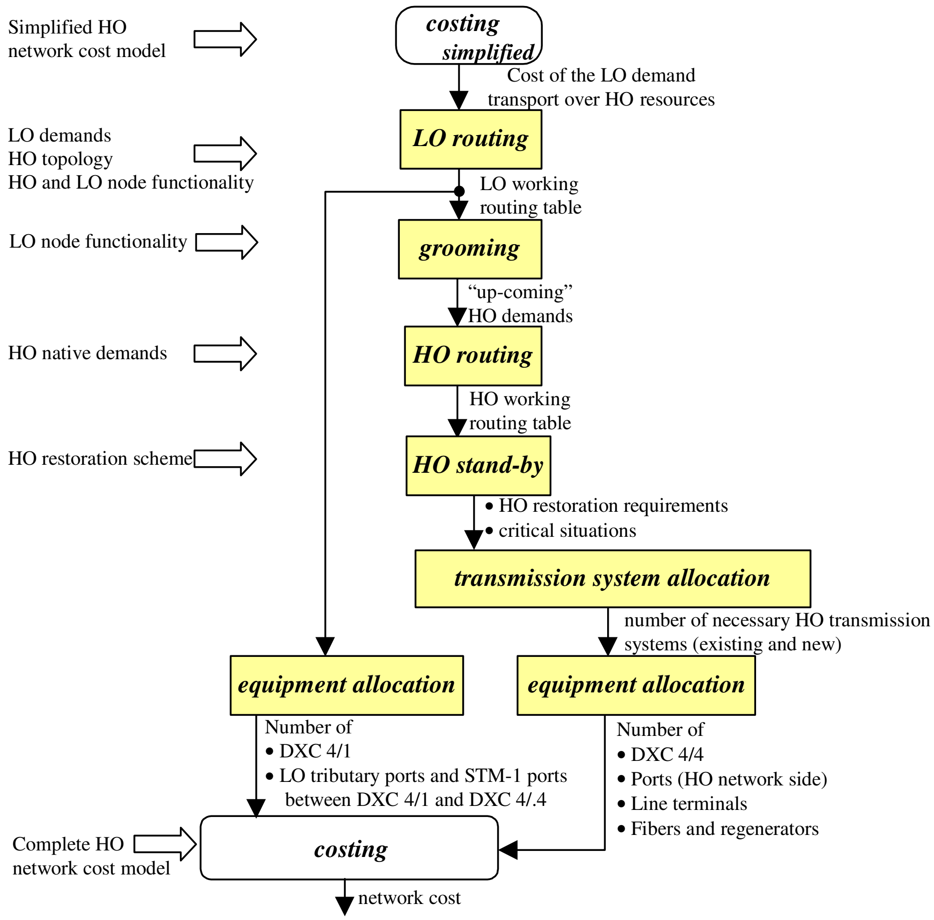


Figure A. 6 - An approach to dimensioning a restored meshed network

Another ingenious approach that considers a recovery mechanism for the demands independently of the restoration mechanism, a local grooming strategy and a multiplex section restoration mechanism is reflected in Figure A. 7. First of all, it is necessary to route the demands in the physical layer in order to assure disjoints paths for the individual recovery of the demands. In that sense, the demand routing in the physical layer and the recovery mechanism for a transmission demand is done simultaneously. Based on a local grooming strategy, it is possible to infer a restricted LO and HO topology in which LO routing, grooming and HO routing is performed. In this approach the stand-by is done in the multiplex section layer assuring that all the VC-4 that are using a specific multiplex section are restored allowing a better reutilization of the transmission capacities. It is important to note that this restoration mechanism assumes a non-simultaneously single-failure model.

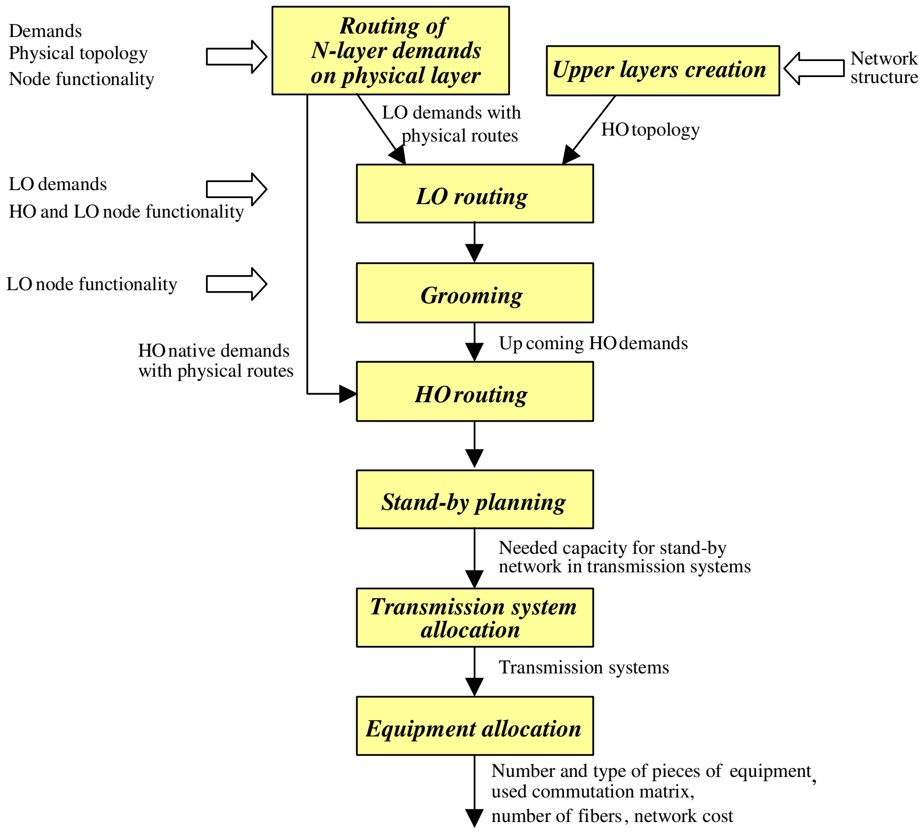


Figure A. 7 - Approach to dimension a restored mesh subnetwork

A1.1.3 LO SH ring dimensioning

A LO Self Healing ring is a SDH ring in which ADMs have

- LO add/drop capabilities,
- LO (SNCP) protection functionality.

The HO view of the LO ring is practically a sequence of link connections between adjacent node pairs (cf. Figure A. 8).

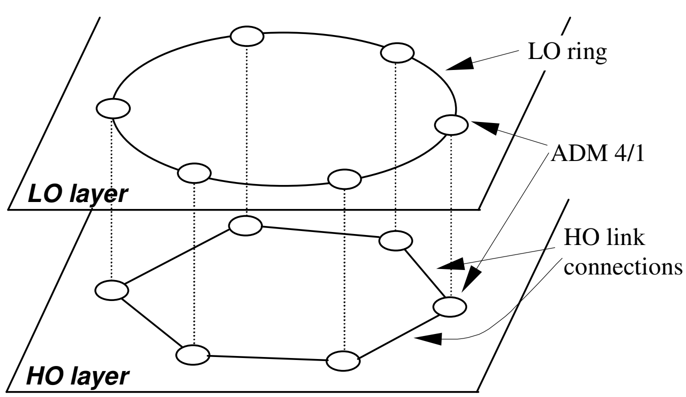


Figure A. 8 - Layered view of a LO ring

The atomic breakdown approach for the dimensioning process for such a SDH self-healing ring is shown in Figure A. 9. The process itself is rather simple; besides some of the involved sub-problems can be either solved in a really trivial way or skipped from the process. In fact the routing, grooming and stand-by problems provides in this case unessential information to the following sub-problems (that is why they are greyed in the figure).

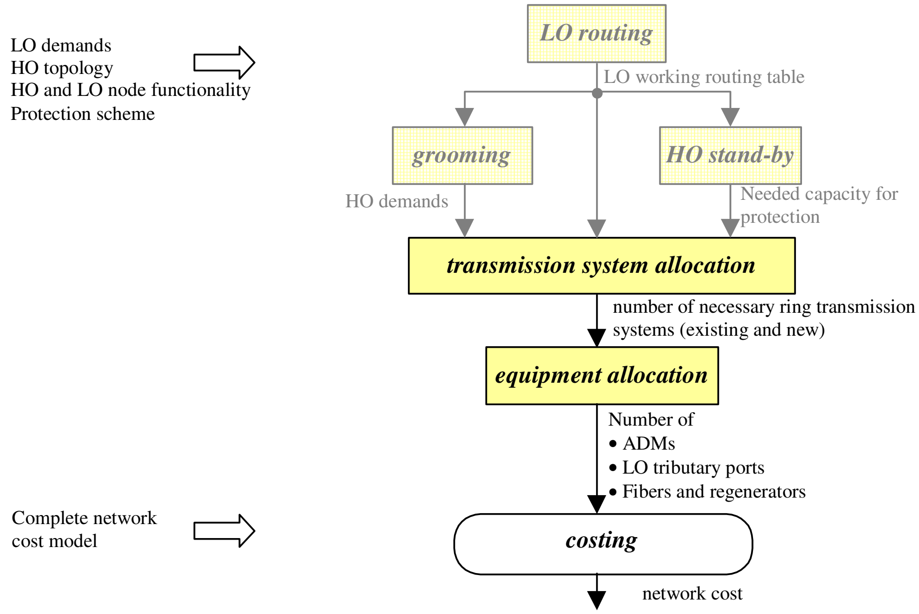


Figure A. 9 - Dimensioning process for a LO Self-Healing ring

A1.1.4 HO SH ring dimensioning

A HO Self Healing ring is a SDH ring in which ADMs have

- HO add/drop capabilities,
- HO (SNCP or MSP) protection functionality.

The LO view of the HO ring is practically a (fully) meshed link connection network (cf. Figure A. 10). The grooming of the LO demands into HO containers is performed by DCX 4/1s outside the ring, while routing and protection of the HO containers is performed by ADM 16s in the ring.

The atomic breakdown approach for the dimensioning process of such a SDH self-healing ring is shown in Figure A. 11. The process itself is again rather simple. As the grooming strategy is the end-to-end one the grooming sub-problem is generally solved first; then the resulting HO demands can be routed on the HO ring. The routing problem is again trivial (and in principle avoidable) in case of SNCP Ring, but it is not trivial at all in case of MSP Ring. In the latter case in fact a routing algorithm should be implemented (e.g. a sequential greedy algorithm); besides when the routing is performed in a sequential way (demand by demand) a “side” problem is the individuation of the optimal routing sequence (i.e. which demand should be routed first). In any case the stand-by sub-problem is trivial and can be skipped.

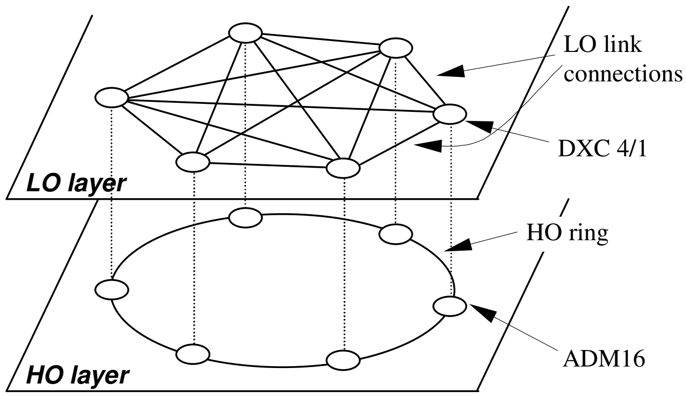


Figure A. 10 - Layered view of a HO ring

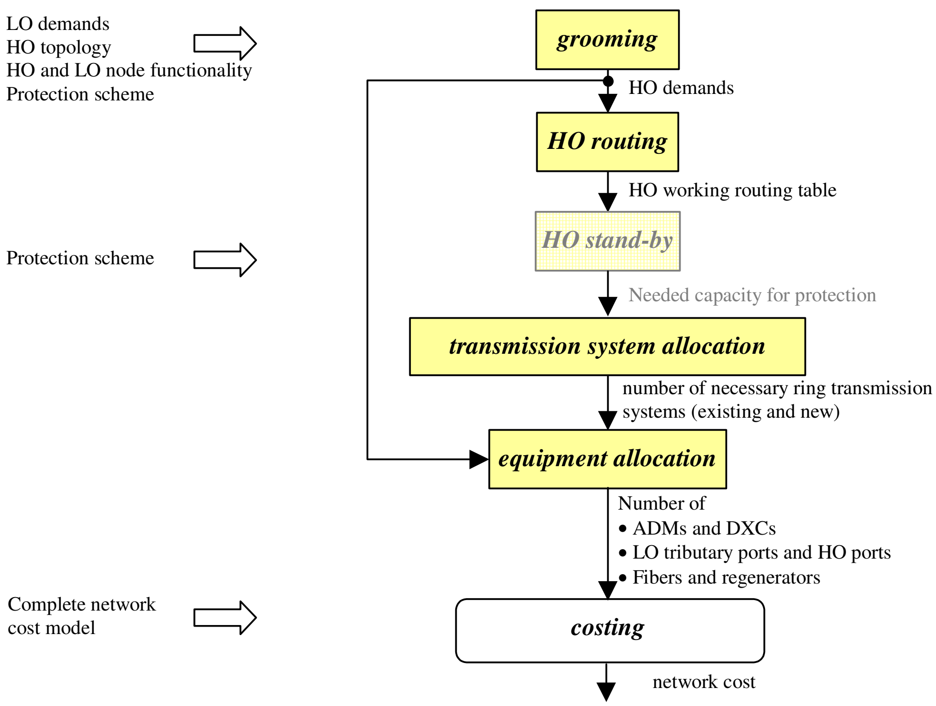


Figure A. 11 - Dimensioning process for a HO Self-Healing ring

A1.2 Dimensioning the network structure

The network structure can be a flat or clustered. A flat structure is realised with a given architecture, a mesh or a ring. In this case the dimensioning of the flat structure can be performed like the dimensioning of a given architecture realising a single cluster, as it described above. Clustered hierarchical or non-hierarchical networks may include different architectures realising the different clusters. There are nodes with specific functionality to interconnect the clusters. The dimensioning of a network structure includes the complex problem of routing the transmission demands on the network structure and the dimensioning of the architectures realising the different network clusters. It is important to remark that in the general planning practice the assignment of network nodes to the network structure is not included in the problem.

General Problem

- to elaborate the dimensioning of network architectures included in a specified network structure to realise a given network

Input

- network structure
  - scheme
  - architectures
  - network nodes assigned to the scheme, interconnection nodes (hubs)
- network topology assigned to the links in the structural scheme
- transmission demands

Output

- transmission demands routed on the given network structure
- full dimensioning of the network architectures realising the different network clusters,

Target function

- minimum cost

Restrictions

- routing, grouping restrictions
- technological limitations

A1.3 Optimisation of network structure

The problem of finding an optimal network structure is an extension of the previously described network structure-dimensioning problem. The aim of the planning is to elaborate an optimal network structure based on the different network structures and architectures. Besides the structural scheme and the architectures realising different network clusters the assignment of network nodes to the given structure should be defined, as well. Thus, besides the selection of a proper scheme, and the specification of architectures realising the network, the clustering of the network nodes and the mapping of the structural scheme to the network topology should be elaborated too. The alternatives of this complex problem can be evaluated by detailed dimensioning of the candidate network structures. A network structure is defined for a longer run, thus not only the optimal realisation of the given transmission demands, but the optimal extension of the structures in case of transmission demand growth should be taken into account.

General Problem

- to define a network structure (including structural scheme, architectures, node clustering) optimal for a longer run taking into account forecast demand growth and planned network extensions.

Input

- network nodes
- transmission demands
- network structure including the physical level

- candidate network clusters

Output

- network structure with clusters and architectures realising the different clusters
- mapping of the network architecture to the topology

Target function

- minimal cost
- additional aspects: flexibility, extendibility, upgradeability

Restrictions

- topological limitations
- technological limitations

In this case, the operators are using a common approach depicted in Figure A. 12.

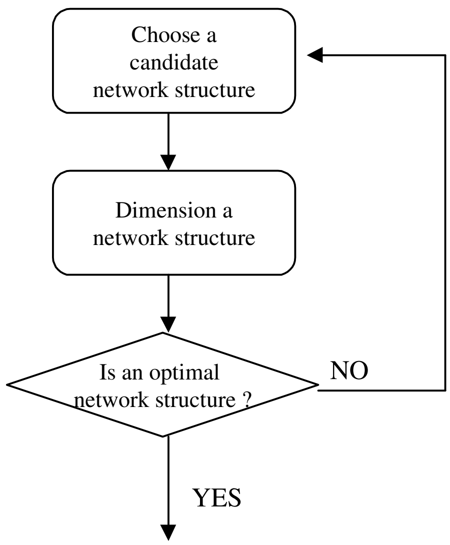


Figure A. 12 - General process of optimisation of a network structure

A1.4 Optimisation of the topology of the physical level

In all previously described planning problems the topology of the physical level is considered as specified. The optimisation of the topology of the physical level is targeting the definition of the infrastructure for a longer run taking into account structural considerations. Different network topologies can be evaluated by general simple quantitative measures like e.g. connectivity, or in more details based on the mapping and dimensioning of desired network structures.

General Problem

- to elaborate the topological configuration of the network infrastructure for a longer run taking into account considerations on the desired network structure.

Input

- structural schemes
- architectures

- candidate network nodes
- permitted links
- transmission demands

Output

- topological configuration of the physical level

Target function

- minimal network cost (based on the dimensioning of candidate structures)

Restrictions

- geographical and technological restrictions on links

Usually an iterative approach is used for the optimisation of the topology of the physical level as it is shown in Figure A. 13.

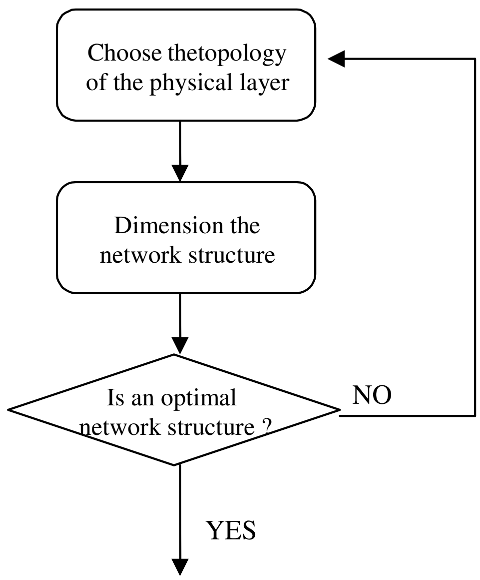


Figure A. 13 - General process for the optimisation of the topology of the physical level

References

[1] Eurescom, *Optical Network Planning*, Project P709, Deliverable 3, Volume 1 of 9, March 2000.

[2] Eurescom, *Optical Network Planning*, Project P709, Deliverable 3, Volume 2 of 9, March 2000.

[3] Eurescom, *Optical Network Planning*, Project P709, Deliverable 3, Volume 3 of 9, March 2000.

[4] *Software Testbed Results*, Deliverable 5, ACTS Project 0205 PANEL - Protection Across NEtwork Layers, 1998.

[5] Michael Gryseels and Piet Demeester, "A multi-layer planning approach for hybrid SDH-based ATM networks, " *Proceedings of the 6th International Conference on Telecommunication Systems: Modeling and Analysis*, Nashville, March 1998.

[6] V. G. Fischer, *Evolutionary Design of Corporate Networks under Uncertainty*, Ph.D. thesis, Institut fur Informatik der Technischen Universitat Munchen, 2000.

[7] Ravindra Ahuja, Thomas Magnanti, and James Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall Inc., 1993.

[8] Leonhard Kleinrock. *Queuing Systems* - Volume I and Volume II. John Wiley & Sons, Inc., 1975.

[9] Hamdy A. Taha. *Operations Research: An Introduction*. MacMillan Publishing Company, 1987.

[10] Wolfgang Domschke. *Einfuhrung in Operations Research*. Springer-Verlag Berlin Heidelberg New York, 1995. 3. Auflage.

[11] Joseph Kallrath and John M. Wilson. Business Optimization using Mathematical Programming. MacMillan Press Ltd., 1997.

[12] Bezalel Gavish, "Topological Design of Computer Communication Networks – The Overall Design Problem, " *European Journal of Operational Research*, 58:149–172, 1992.

[13] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., 1988.

[14] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization - Algorithms and Complexity*. Dover Publications Inc., 1982.

[15] Andreas Bley, Martin Grotschel, and RolandWessaly. *Modellierung zur "Topologie-, Kapazitats und Routingpanung fur das B-WiN"*. Technical report, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, September 1997.

[16] Mario Gerla and Leonard Kleinrock, “On the topological Design of Distributed Computer Networks, “ *IEEE Transactions on Communications*, COM-25(1):48–60, January 1977.

[17] Mhammad Shahbaz. *Zufallsgesteuerte Verfahren zur Topologieoptimierung von Telekommunikationsnetzen*. PhD thesis, Fakultat fur Elektrotechnik der Rheinisch-Westfalischen Technischen Hochschule Aachen, 1997.

[18] Dorit S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1995.

[19] Yishay Mansour and David Peleg. *An Approximation Algorithm for Minimum-Cost Network Design*. Technical report, Computer Science Department, Tel-Aviv University, Israel, 1994.

[20] Sven Olver Krumke. *On the Approximabilityof Location and Network Design Problems*. PhD thesis, Bayerische Justus-Maximilian-Universitat Wurzburg, 1996.

[21] Thomas Dean, James Allen, and Yiannis Aloimonos. *Artificial Intelligence – Theory and Practice*. The Benjamin/Cummings publishing Company Inc., 1995.

[22] George J. Klir and Mark J. Wierman. Uncertainty-based Information – Elements of Generalized Information Theory. Physica-Verlag, 1998.

[23] Anthony Hunter and Simon Parsons, editors. *Applications of Uncertainty Formalisms*. Number 1455 in Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg New York, 1998.

[24] Hany I. Fahmy and Christos Douglігeris, “END: An Expert Network Designer, “ *IEEE Network*, pages 18–27, November/December 1995.

[25] Volker G. Fischer and Manfred Schramm, “Probabilistic Reasoning with Maximum Entropy - The System Pit, “ In Francois Bry, Burkhard Freitag, and Dietmar Seipel, editors, *Twelfth Workshop on Logic Programming*, Ludwig Maximilians Univeristat Munchen, September 1997.

[26] Petr Horacek, Radek Sindelmar, Zdenek Vleck, Martin Vojtisek, and Milon Vojnar. Midrange *Prediction of Data Throughput in a Communication Network*. Technical report, Czech Technical University in Prague, Faculty of Electrical Engineering, 1999.

[27] Thomas Kindshofer. *Optimierung der Netztopologie in zellul¨aren Mobilfunknetzen unter Berucksichtigung eines bestehenden Netzes*. Master’s thesis, Technische Universitat Munchen, 1997. Institut fur Informatik.

[28] Daniel Lang. *Plazierung von Backboneknoten mit Fuzzy-Clustering*. Master’s thesis, Technische Universitat Munchen, 1999. Institut fur Informatik.

[29] Harald Richter. Verbindungsnetzwerke f¨ur parallele und verteilte Systeme. S pektrum Akademischer Verlag, 1997.

[30] Harald Richter. *Verbindungsnetzwerke - Strukturen und Leistungsanalysen*. Technische Universitat M¨unchen, Institut fur Informatik, Lehrstuhl fur Rechnertechnik und Rechnerorganisation, 1998.

[31] D. Guo, A.S. Acampora, and Z. Zhang, "Hyper-Cluster: A Scalable and Reconfigurable Wide-Area Lightwave Network Architecture, " *IEEE Global Telecommunications Conference – Globecom96*, pages 863–867, 1996.

[32] Geir Dahl, *Contribution to the design of survivable directed networks*, Ph.D thesis, University of Oslo, 1991.

[33] M. Grötschel, C. L. Monma, M. Stoer, *Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints*, Op. Res. Vol. 40. No 2. March-April 1992.

[34] U. Brannlund, *On relaxation methods for nonsmooth convex optimization*, Doctoral thesis, Royal Institute ot Technology, Stockholm, 1993.

[35] F. Glover, M. Laguna, *Tabu Search* (A chapter in Modern heuristic techniques for combinatorial problems), North Holland, 1992.

[36] Otten R.H.J.M., Van Ginneken L.P.P.P., *The Annealing Algorithm*, 1989, (Boston:Kluwer).

[37] N. Rosenblouth, A. Rosenbluth, M., Teller, *Metropolis*, E. 1953, Journal of Chemical Physics, vol.21, pp 1087-1092.

[38] Glover F., *Tabu search Part I*. ORSA Journal of Computing 1, 1989, pp 190-206.

[39] Glover F., *Tabu search Part II*. ORSA Journal of Computing 2, 1990, pp 4-32.

[40] P. Lindberg, A heuristic method for optimiation with a separable non-linear target function, Telia Research Sf 032/92, 4 Sept. 1992 (in Swedish).

[41] Goldberg D.E, Genetic algorithms in search, optimization and machine learning, Addison Wesley Pub Co Inc, 1989.

[42] J. Holland, *Genetic Algorithms Scientific American*, July 1992.

[43] L. Davis (ed.), Handbook of genetic algorithms, Van Nostrand Reinhold (1991).

[44] EURESCOM, Methods and specification for Tools to Dimension Intelligent Networks. General specification and examples, EURESCOM P-308 Deliverable 4, March 1997.

[45] Aaron Kershenbaum. *Telecommunications Network Design Algorithms*. McGraw-Hill, 1993.

[46] G.R. Ash, R.H. Cardwell, and R.P. Murray, "Design and Optimization of Networks with Dynamic Routing, " *The Bell System Technical Journal*, October 1981.

[47] Robert Boorstyn and Howard Frank, "Large-Scale Network Topological Optimization, " *IEEE Transactions on Communications*, 25(1):29–47, January 1977.

[48] A. Kershenbaum, P. Kermani, and G. Grover. *MENTOR: An Algorithm for Mesh Network Topological Optimization and Routing.* Technical Report RC 14764 - 7/14/89, IBM Research Division, T.J. Watson Research Center, 1989.

[49] Robert S. Cahn. *Wide Area Network Design - Concepts and Tools for Optimization.* Morgan Kaufmann Publishers Inc., 1998. The Morgan Kaufmann Series in Networking.