orbit #X

ISMP 2003 COPENHAGEN

august • 2003

Special ISMP issue of the newsletter of the Danish Operations Research Society

A word from the editors

As editors we are happy and proud to present you an issue full of hopefully exiting stories about what researchers and companies from Denmark currently are doing within operations research, optimization and mathematical programming.

The stories in this issue are all collected among Danish researcher and companies to give you an insight in some of the areas of research that are currently in focus in Denmark.

Although we are the newsletter for the Danish Operations Research Society we consider ourselves not only a Danish newsletter, but more a Scandinavian. True to this we publish papers in Danish, Norwegian, Swedish and English in regular issues.

Beside that the entire contents of this newsletter are in English it generally reflects the contents and style of a regular issue.

We aim to keeps our readers up-to-date with latest research from the universities and interesting stories from the companies.

Jesper Larsen   Rene M. Jørgensen

Editors

ORbit
newsletter for
the Danish Operations Research Society

Editors:
Rene Munk Jørgensen (rmj)
(in-Chief) Jesper Larsen (jla)

IMM, building. 305 - DTU
DK-2800 Kgs. Lyngby
Phone : +45 4525 3385
Fax: +45 4588 2673
E-mail: orbit@dorsnet.dk

Next deadline:
September 15. 2003

Next issue:
Beginning of November 2003

Tryk:
XXXXXXXXXXXXX
Circulation: 1500
ISSN 1601-8893

Support:
Britt Morelli Hansen

Photographs: Pictures on page 13 and 14 are taken by René Strandbygaard and are curtesy of HUR Trafik.

Special issue sponsors:

EURO

COWIfonden

Information about DORS August 2003

Membership
Membership rates until December 31th 2004. Applies only in Scandinavia.

Personal member (including. Ph.D.-student):	250 kr./yr
Student:	50 kr./yr
Company member:	3000 kr./yr
Department member:	1500 kr./yr

Payable to Giro 9123865 (reg.nr. 1199)

General office:
DORS
IMM, Building 321
Technical University of Denmark
DK-2800 Kgs. Lyngby
Denmark
e-mail: dors@dorsnet.dk

Internet
www.dorsnet.dk

Welcome
to ORbit Extra

As President of the Danish Operations Research Society (DORS) I have the pleasure of introducing you to this special issue published on the occasion of the International Symposium on Mathematical Programming taking place in Copenhagen.

ORbit is the leading OR newsletter in Scandinavia. It is published four times a year and has a circulation of 300 copies. Kind donations from the EURO General Support fund and from the COWI Foundation have made it possible for DORS to publish ORbit Extra with a circulation of 1500 copies.

The articles in ORbit Extra reflect the contents of a normal issue of ORbit. We aim to present theoretical work within OR as well as practical experiences from using OR methods. ORbit thereby helps to fulfill the main objective of DORS, which is to spread the knowledge and use of OR methods.

Besides publishing ORbit, DORS hosts 6-7 OR events a year - again with an even focus on theoretical research and practical experiences. Every second year DORS awards a price for the best Danish Masters Thesis within OR.

DORS is founded in 1962 and we were happy to celebrate its 40th anniversary last year. But even with 40 years of experience, it is still not an easy task to spread the messages of OR. However, increasing quality of companies data and easy data access world wide combined with increasing global competition could cause a major OR breakthrough in the coming years. We aim to make a small contribution to this!

A parallel objective of DORS is to establish contacts to OR researchers, organizations and societies from other countries. We hope ORbit Extra will inspire to this, and we will be very pleased to discuss possible cooperation with you and your organization.

I hope you will enjoy reading the articles in ORbit Extra. If you have any comments or ideas regarding ORbit, we will also be pleased to hear from you. DORS will be present at its own stand at the Symposium. We hope you will find time to stop by. You can also visit DORS at our (Danish) web-site www.dorsnet.dk.

Søren K. Nielsen

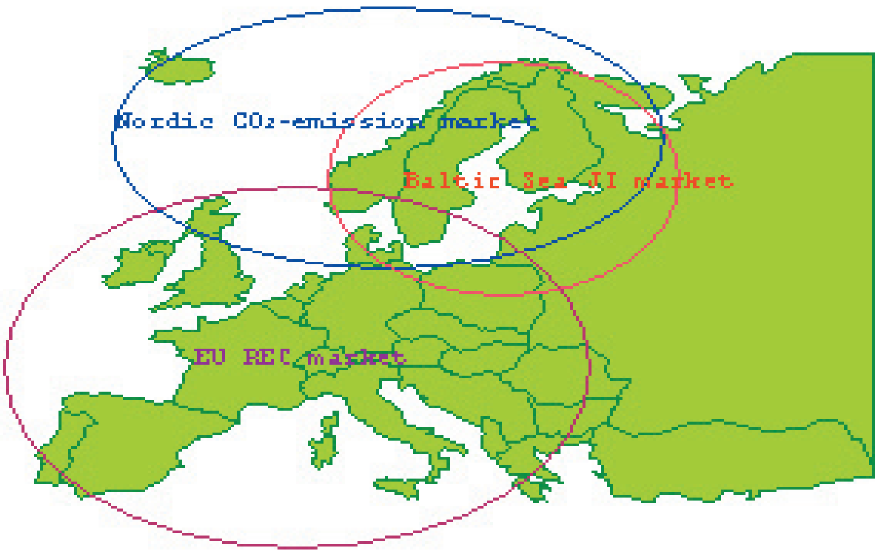
President of DORS

Contents

By Hans F. Ravn

Balmorel - energy system modelling with open source code and data

Energy research has a long tradition of applying modelling and optimization. In this article Danish energy expert Hans Ravn will describe the Balmoral energy model covering the countries in the Baltic Sea.



The Balmorel model is an energy model covering the countries in the Baltic Sea Region with emphasis on the electricity and combined heat and power sectors. The model may be used for analyses that cover some or all countries bordering the Baltic Sea, and include aspects of energy, environment, and economy.

The purpose of the Balmorel project is to support modelling and analyses of the energy sector in the region. The project maintains and develops the Balmorel model. Thus, the project contributes:

1. A model - the Balmorel Model. This model consists of the model structure and the corresponding data for the Baltic Sea Region.

2. Experience exchange in relation to modelling.

3. Cooperation on application of the Balmorel model for analyses of aspects of energy, environment, and economy in the region.

Model characteristics include representation of electricity and heat; environmental aspects; international aspects, including electricity transmission lines; representation of several handles for energy and environmental policy, e.g. taxes and emission limitations. Two energy modelling traditions are represented (sometimes referred to as the top-down and the bottom-up approaches). Thus technical and engineering (bottom up) aspects and aspect from the economic tradition (top down) are integrated. Examples of applications include:

• Analysis of market power in the electricity sector is presently undertaken. As a first step, the classical Cournot and Bertrand models will be implemented in the Balmorel model.

• Application in relation to evaluation of Danish environmental and energy policy in the nineties.

• Economic analyses in the electricity sector in Lithuania.

• An analysis of two environmental instruments in relation to the electricity sector: tradable green certificates (TGC) and tradable emission permits (TEP) in the Baltic Sea Region. See the figure which illustrates that these markets need not cover the same geographical areas.

Open source model and data

The characteristic point in relation to the Balmorel project, which probably makes it unique in the energy sector, is the *open source* spirit. Hence, project generated information, including all details, can be freely downloaded from the home page of the project, *www.Balmorel.com*. This in particular includes the model structure source code (the model is formulated in the GAMS modelling language) and the corresponding data for the Baltic Sea Region.

The underlying idea is that data and modelling should be common to all parties dealing with common problems. Access to data and models is a source of strength which should be available to all interested parties. In complicated processes involving several possibly conflicting parties a model may serve as a means of communications. Thus, it provides a set of data; it provides interpretations of the working mechanisms in the system; and it may serve the purpose of identifying and formalising relevant questions for discussion.

Anybody that is interested can take the model from the home page and use it as desired, including modifying it. Hopefully, such activities will feed back into the Balmorel project, such that model improvements, data updating and experiences gained by the users may be made available to others.

The Balmorel model is in its present state indeed suitable for a number of relevant analyses, therefore the challenges are the following:

• To maintain and improve on the model in the open source perspective. It is quite easy to make the present version available to all interested parties, using the internet. The difficult part is to ensure that the experience, the updates and the improvements that are made when somebody is using the model, can be channelled back to the project and in turn made available to and benefit other parties. How can this process be organised such that it functions smoothly and efficiently?

• The basic version of the model aims at covering the whole geographical area of the Baltic Sea Region, handles the long term perspective, e.g. up to 2030, and represents the basic elements of the energy sector. Most surely, any user will like to include some specific functionality. This will be possible in many cases, and quite easy due to the application of a modelling system. However, if all new functionalities are included as options in the basic version of the model, then it becomes more difficult to understand and apply. How can it be organised that many special options are kept alive without clouding the basic model?

• Since the basic version of the model aims at the large perspective with respect to geography and time, some detail has been sacrificed for computational reasons. However, most users would like to include some more details because they feel that the results will be better this way. Sometimes they are right. But computational times easily grow uncomfortably long, see next.

Computational challenges

The present basic version of the model applies deterministic linear programming. This gives acceptable computational times with the standard level of detail. But we experience increased pressures to extend the model in ways that may imply unacceptable computational times. Three extensions are much desired:

• A finer resolution of the time within the year (presently the year is divided into 144 time segments).

• Introduction of mixed integers to represent the unit commitment aspects (i.e., when to stop and start production units)

• Stochastic optimisation to represent hydro power and wind

Any comments or suggestions? We will be pleased to receive your ideas at the home page, www.Balmorel.com.

By Hans Ravn.

Hans Ravn has through noumerous years worked with matematical programming as a problem solving tool.

Hans Ravn is Msc., Ph.D. and doctor of science from the Technical University of Denmark. He has worked in both the academia at the Technical University of Denmark and Risø Research Facility and industry, where he was part of the management at Elkraft. Generally he is considerede on of the leading energy experts in Denmark, and has been involved in noumerous energy policy-related projects during the last 20 years.

By Tim Helge Hultberg, Critical Software SA/University of Aveiro

A presentation of FLOPC++

FLOPC++ is an open source algebraic modelling language, implemented as a C++ class library, which can be downloaded at www.mat.ua.pt/thh/flopc. Using FLOPC++, linear optimization models can be specified in a declarative style, similar to algebraic modelling languages such as GAMS and AMPL, within a C++ program.

Introduction

Real world optimization problems typically arise as specific data instances of generic algebraic models, while the software used to solve these problems (the solver) expects the problems to be passed to it in an explicit matrix format (typically with a sparse column compressed storage of the coefficient matrix). The benefits of using algebraic modelling languages to automate the generation of problem instances are well recognized and a number of algebraic modelling languages have emerged. However, traditional algebraic modelling languages have shortcomings with respect to:

• **Embedding of optimization models in applications**. The data entering an optimization model typically comes from sources such as databases, spreadsheets and GUI's and its solution might be needed as input for all kinds of further processing. An optimization model is often just a minor part of a larger software application, but this embedding is hard to achieve with traditional algebraic modelling languages.

• **Implementation of model tailored solution algorithms**. The algebraic description of a model convey structural information that is hidden in the matrix representation required by the solver. Sometimes this information can be used to develop efficient specialized solution algorithms while still depending on a general linear optimization solver. Examples of such algorithms include decomposition, column generation and model specific cutting plane algorithms. The development of such algorithms is greatly facilitated by working in an environment where the algebraic description of the model is available. But efficiency considerations make traditional algebraic modelling languages a bad choice.

These shortcomings can be remedied to a certain extent by incorporating procedural features and links to external software (such as databases and spreadsheets) into the languages. But bringing modelling features to a programming language (rather than the opposite) is a much more natural and versatile approach.

A small step in this direction has been taken with the ILOG Concert Technology, which provides support for mapping of variable/constraint indices to instance column/row indices, but offers only an executable (as opposed to declarative) model representation. FLOPC++ has been developed with the purpose of making declarative model representation (similar to traditional algebraic modelling languages) possible within C++ programs.

Optimization modelling and C++

Optimization modelling and C++ is a perfect match. Due to the power of objectoriented programming and the standard template library (STL), an efficient algebraic modelling language, including sparse multi-dimensional subsets and index arithmetic, can be implemented as a C++ class library using less than 3000 lines of code. Linear optimization models can then be conveniently specified within a C++

program using modelling objects (sets, data, variables and constraints) declared as instances of classes provided by FLOPC++, making it easy to integrate such models in software applications.

An example of a FLOPC++ represen-tation of a sparse transportation model is shown in figure 1.

```
class Transport : public MP_model {
   MP_data COST;
   MP_variable x;
   MP_constraint supply, demand;
public:
   Transport(MP_set& S, // Sources
   MP_set& D, // Destinations
   MP_subset<2>& Link, // Transport. links (sparse
                          subset of S*D)
   MP_data& SUPPLY, MP_data& DEMAND, MP_data& DISTANCE) :
   MP_model(new OsiOslSolverInterface),
   COST(Link), x(Link), supply(S), demand(D) {

   // Assignment of derived data
   COST(Link) = 90 * DISTANCE(Link) / 1000.0;

   // Definition of constraints
   supply(S) = sum( Link(S,D), x(Link) ) <= SUPPLY(S);
   demand(D) = sum( Link(S,D), x(Link) ) >= DEMAND(D);

   // Objective function
   minimize( sum(Link, COST(Link)*x(Link)) );
  }
};
```

Figure 1.: Example of the representation of a sparse transportation problem

The use of a class (derived from MP model) to specify the model is not required, but can be advantageous when several models are used. (A number of complete example models are available at the FLOPC++ Home page: www.mat.ua.pt/thh/flopc.)

The declarative representation is possible because expressions in FLOPC++ evaluate to abstract syntax trees, which can subsequently be analyzed and used for generating problem instances. Reference counting is used to avoid memory leaks due to the dynamic allocation of the data-structures representing the abstract syntax trees.

FLOPC++ uses the COIN (www.coin-or.org) Open Solver Interface (OSI) (a uniform API for calling MP solvers), and may be linked to any solver with an OSI interface (currently CLP, CPLEX, dylp, GLPK, OSL, SOPLEX, VOL, and XPRESS-MP). A FLOPC++ model has a pointer to an OSI object through which the methods of its class can be invoked (using the overloaded dereferencing operator ->). This can be useful, for example, to output an MPS file of a problem or to modify it without regenerating the problem.

Conclusion

The syntax and expressiveness of FLOPC++ for linear optimization modelling is similar to traditional modelling languages, but its implementation as a C++ class library implies several additional benefits:

• The relatively small size of the source code makes changes and extensions easy.

• Fast problem generation. Sparse generation efficiently implemented and problem generation done by compiled code.

• Seamless integration with applications.

• Convenient and efficient implementation of model tailored solution algorithms. Model modification (without regeneration) available.

• Modelling objects are first class C++ types (i.e. they can be passed around as parameters to functions). This allows higher level modelling extensions (such as a »minimize max« objective function, etc.) to be implemented.

By Tim Helge Hultberg. Critical Software SA/University of Averio. E-mail: thh@mat.ua.pt.

Tim Helge Hultberg has a MSc from the Department of Computer Science at the University of Copenhagen, and was awarded a Ph.D. in OR from the department of Informatics and Mathematical Modelling at the Technical University of Denmark in 2001.

Tim has since worked for the software company Critical Software and he also holds a position a postgraduate fellow at the University of Aveiro.

By Niklas Kohl, Carmen Consulting Denmark

Solving the World's Largest Crew Scheduling Problem

In the airline and railway industry we find many examples of the benefits of operations research and optimization. Cases from this research area has also been an important field for noumerous Danish researchers. In this article Niklas Kohl of Carmen Consulting in Denmark describes the process of solving the largest crew scheduling problem in the world.

Airline crew scheduling has for decades been a major industrial application area of mathematical programming and operations research. The goal of crew scheduling is to assign the tasks to be carried out (the flights) to named individuals such that rules and regulations are respected and costs minimized.

The problem naturally decomposes into the *crew pairing problem,* in which trips, i.e. sequences of flights, starting and ending at a crew base are constructed and the *crew rostering problem*, where the anonymous trips are assigned to named individuals. In this article we limit the scope to the crew pairing problem.

Virtually all major airlines and most midsize airlines use optimization software as part of their crew pairing process. More recently railways have started to use optimization. There are a lot of similarities between the crew pairing problems faced at airlines and railways, but also important differences.

The purpose of this article is to give some insight into the particularities of the railway crew pairing problem and to briefly outline how we at Carmen have chosen to solve some of the difficulties. In particular we will discuss the crew pairing problem at Europe's largest railway Deutsche Bahn, the German state railways.

About Carmen

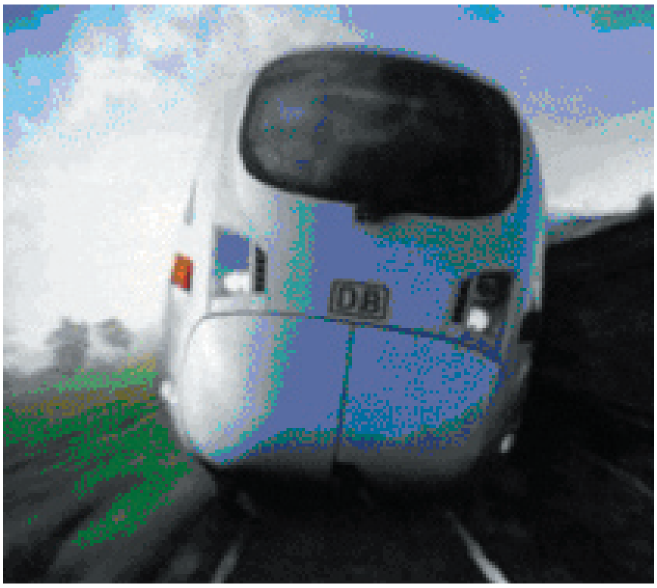
Carmen Systems is the leading developer of crew scheduling software for the transportation industry. The crew pairing system discussed in this article is in use at all major European airlines as well as Singapore Airlines, Northwest Airlines and other major international airlines. In addition to Deutsche Bahn, railway customers include the Swedish State Railways and the freight operator Green Cargo.

About Deutsche Bahn

Deutsche Bahn (DB) is the state owned railway company in Germany and one of the worlds largest transportation companies. DB Regio, consists of many partly independent companies, which operate regional and commuter traffic. DB Reise & Touristik operates long line passenger traffic. Total crew (train drivers and conductors) number around 30,000. The DB timetable contains several thousand trains per day and more than 100 crew bases are in use. A train is operated by one train driver and from zero to seven conductors depending on the type of train, the expected number of passengers as well a number of other factors.

Mathematical Model of the Crew Pairing Problem

In its simplest form the crew pairing problem can be described as follows; The timetable specifies a number of *atomic* tasks which must be staffed by a crew. In case of a railway an atomic task consist of staffing the train between two stations where crew change can take place. Following the airline terminology these tasks are denoted *legs*. A *trip* is a sequence of legs, which define a possible working pattern for a crew. It must start and end at a crew base, must respect legal and contractual rules regarding work and rest time as well as operational rules ensuring that the trip can be ope-



rated in practice. The cost of a trip is the crew cost associated with operating the trip as well as other direct costs such as hotel costs and costs for passive transports (for example taxis). Mathematically, the crew pairing problem formulates as a set partitioning problem

$$
\begin{array}{ll}
\text{Minimize} & \sum_t C_t X_t \\
\text{Subject to} & \sum_t A_{lt} X_t = 1, \forall l \in L \\
 & X_t \in \{0,1\}, \forall t \in T
\end{array}
$$

Where the sets T and L are the sets of possible trips and legs, respectively. C_t is the cost of trip t and the coefficient A_{lt} is 1 if trip t contain leg l and 0 otherwise. The binary decision variable X_t is 1 if trip t is part of the optimal solution and 0 otherwise. The constraints set expresses, that each leg should be contained in exactly on trip. Numerous extensions and modifications to this model have been proposed, see e.g. Andersson et al. (1998). Most importantly, the partitioning constraints (=) are usually replaced with covering constraints (≥), and base constraints, which specify the distribution of work time between crew bases, are added.

Though simple, this model suffers from two drawbacks. The set partitioning problem is NP-hard and even worse – the set T is extremely large for virtually all problems of reasonable size.

Differences between the Airline and the Railway Crew Pairing Problem

The vast majority of academic and industrial research on the crew pairing problem has focused on the problem faced by airlines. In the airline problem the set L is rather small – hundreds or a few thousands of legs. On the contrary the long line traffic of DB consists of more than 1000 trains per day and many trains correspond to more than 10 legs since it is possible to change crew at most major stations. Several conductor positions must be staffed on a train, and since preparation times and qualification requirements depend on the position, each position must be considered separately. However, a crew member can change position between legs within a trip, so all positions must be considered as one problem. Hence, the set L for a weekly pairing problem is more than 100,000.

Further, the number of 1-coefficients per variable is higher than in the airline case. An airline trip will typically contain 2-4 legs per day and rarely have a duration of more than 4 days. A railway trip is never more than two days long, but since the average duration of a leg is only 30 minutes for the long line traffic and 20 minutes for the regional traffic, pairings with 20 or more legs are not uncommon.

It is known that the average case difficulty of a set partitioning problem increase with the average number of 1-coefficients per variable, so the railway set partitioning problems are not only more difficult than their airline counterparts due to their size, but also due to their inherently complex structure.

It is not only the mathematical structure of the problem, which is more complex for railways than for airlines. The calculation of legality and cost of a trip is substantially more complex at Deutsche Bahn. There are several reasons for this. Most importantly train crew carries out a number of duties in addition to driving trains. In particular train drivers have a lot of preparation tasks, which must be derived from the locomotive and wagon rotations.

Rest calculations are also complicated by the possibility of taking a rest while the train is running - obviously this is only possible for conductors! Such rest will be paid, but will extend the possible working time. Normal rest at a station is generally unpaid if the rest is required by the working time rules, but will be paid if it is not required. Since rest occurs naturally in connection with changing trains, a trip with more work can actually produce less pay than a trip with less work because the rest will be unpaid in the first case.

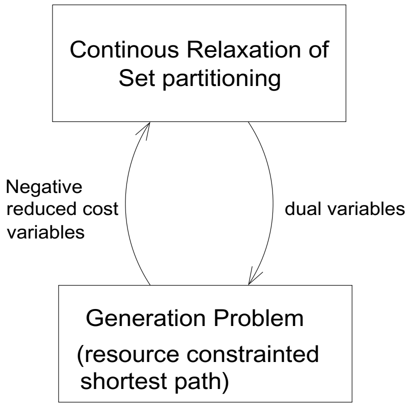
A further complication is the amount and quality of data. The time table is huge. Wagon and locomotive rotations are needed to calculate preparation and termination activities as well as to calculate connection times. And these data must be consistent with the time table. In some cases infrastructure data is needed as well. For a locomotive driver it can for example be important which route the train takes between station A and B, since each route will require a specific qualification profile.

How to Generate the Variables

As previously stated, the number of possible trips will be enormous for a large crew pairing problem. Early crew pairing algorithms tried to reduce the number of trips considered by heuristically generating some thousands of trips, solving the set partitioning problem, updating the solution and repeating generation. One implementation of this idea is to randomly select a small number of trips and generate all possible trips, which "cover" a subset of legs covered by the legs "covered" by the selected trips. This can be repeated as long as time permits. The number of trips selected in each iteration can be chosen to ensure that the set partition problems remain solvable. There are many other ways to limit the number of generated trips.

More recent work on the crew pairing problem has focused on the column generation (or Dantzig-Wolfe) approach. In this approach the generation problem is to find the variable (trip) which will minimizes the reduced cost with respect to the solution to the continuous relaxation of the set partitioning problem defined on the currently known variables. In theory, this can be repeated until no variable with negative reduced cost exist and at this point the continuous relaxation of the set partitioning problem, defined on all trips, has been solved without explicitly generating all possible trips. However, two problems remain. The solution obtained is generally not integer and – in practice much worse – it is not trivial to find the variable which minimize the reduced cost. In fact the latter problem turns out to be a resource constrained shortest path problem.

In many cases dynamic programming can solve the resource constrained shortest path problem. Two different methods are in use. Either multidimensional state variables are used, where each dimension represent a resource such as work time, or a k-shortest path



algorithm is used. In the later case the *k* shortest paths are generated and those paths which corresponds to resource infeasible trips are discarded, see Hjorring and Hansen for a further discussion of this idea. Recent work by Carmen focus on integrating these two approaches.

Technologies Used

Due to the size of the integer programs to be solved, it is not possible to obtain acceptable solution times with commercial integer programming solvers such as ILOG CPLEX or XPRESS-MP. Instead we use the proprietary PAQS optimizer developed in cooperation with Chalmers University of Technology. PAQS has been designed to yield near-optimal solutions to huge set covering type problems and has consistently outperformed the standard commercial optimizers on this class of problems. An early reference on PAQS is Wedelin (1995)

Another important technology used is the Rule And Value Evaluator (RAVE). RAVE is a programming language used to define legality, costs and other objectives. Compared to a normal programming language such as C++ or Java, RAVE programming is simple because it has been developed to reason about sequences of activities and because the programmer only need to state the

definitions and rules, but do not need to consider how the rules will be tested during the optimization.

Results Obtained

Good solutions to huge optimization problems require lots of CPU time! For the long line conductor problem, discussed in this article, a total of more than 200 CPU hours is typically necessary in the production environment to obtain a good weekly solution. Compared to current, manually constructed solutions the optimized solutions are significantly cheaper to operate. In addition to this some more “soft” quality aspects have been improved and the inevitable rule violations associated with manual planning have been eliminated.

References

Erik Andersson, Efthymios Housos, Niklas Kohl, and Dag Wedelin. *Crew Pairing Optimization*. In G. Yu editor, *Operations Research in the Airline Industry*, Kluwer Academic Publishers, 1998.

Curt A. Hjorring and Jesper Hansen. Column Generation with a Rule Modelling Language for Airline Crew Pairing. In Proceedings of the 34th Annual Conference of the Operational Research Society of New Zealand, December 10-11, 1999, Hamilton, New Zealand. This paper is available from the URL below.

Dag Wedelin. *An algorithm for large scale 0-1 integer optimization*. Annals of Operations Research 57. 1995.

A lot of papers on crew scheduling, authored or co-authored by Carmen employees, can be found at

http://www.carmen.se/research_development/research_reports.htm

By Niklas Kohl, Ph.D., Senior Consultant, Carmen Consulting Denmark, e-mail: niklas.kohl@carmenconsulting.com

Niklas Kohl is Msc and Ph.D. from the department of Informatics and Mathematical Modelling at the Technical University of Denmark.

Upon receiving his Ph.D. he worked as postdoctoral fellow at the department of operations research at the University of Copenhagen before he joined the industry. First working for COWI since going to Carmen Systems in Gothenborg. In 2000 he returned to Denmark to lead the Copenhagen-office of Carmen Consulting.

By Michael Berliner Pedersen, Centre for Traffic and Transport, Technical University of Denmark

Minimizing Passenger Transfer Times in Public Transport Timetables

Her noget text

INTRODUCTION

A lot of effort has previously been put into urban planning and hereby into planning of urban public transport. Many issues are covered depending on the level of planning. In the end a number of factors determine the quality of public transport service. Examples of factors that represent good quality are regularity, direct connections, convenient terminal layout, high and fixed frequencies and short travel times. Especially the latter is an important factor in the choice of public transport vs. private transport.

Planning process in public transport

It is often the case that most planning issues in public transport are performed at a local or regional planning authority. Here strategic issues spanning from demographic, economical and developmental studies to infrastructure and strategic network design are considered. Also planning authorities often cover tactical planning issues like service network design and timetabling.

If planning authorities own their own fleet operational issues like crew scheduling and fleet management are performed. However the actual operation of the public transport network can be carried out by a number of carriers. E.g. in Copenhagen; trains, busses and metro are run by different private enterprises. There are even several bus companies to share the operation of bus routes. Timetabling of all bus routes is centralized at the Greater Copenha-

gen Authority. The routes are divided into smaller groups. Based on the timetables carriers are then invited to bid on the route groups.

The process of timetabling is performed interactively with customers and carriers. The planning authorities study their system and collect requests from customers and carriers. Timetable planners evaluate the requests and a revised timetable is proposed. The time horizon for this process is up to six months.

Much of this process is performed manually and relies on planners' experience. This is especially true for the synchronization of routes at transfer points. It is a quite difficult task considering the size of the system with respect to both the number of routes and the number of possible transfers.

Transfers and waiting times

It is obvious that it is economically infeasible to have direct connections between all source and destination points in a public transport system. Transfers between routes are therefore an unavoidable aspect of travelling with public transport.

Transfers are unpopular because they require a physical transfer from one vehicle to another and they can generate important waiting times. The inconvenience of physical transfer is portrayed by factors like bad weather, the carrying of heavy luggage over large distances within transfer terminals, and physical barriers like roads and stairs. Only rarely is it possible to step directly from one vehicle to another and therefore some time waiting for the connecting vehicle is expected. This prolongs the total travel time of the trip. Extra waiting time because of irregularity of service in urban areas is an additional inconvenience.

There exist solutions to improve the perception of transfers. These include efficient terminal layout, acquisition of additional vehicles to increase frequency and thereby shortening expected transfer times, ITS systems to improve regularity and passenger information and thereby minimizing disruption and passenger discontent. However these all require investments in physical infrastructure or equipment and therefore require important allocation of resources.

The question is whether it is possible to develop a method to reduce transfer waiting times that only requires limited investment. We will present an optimization model and a solution procedure that reduces waiting times for transfers by modifying the timetables and thereby improving synchronization between routes.

Existing work and literature

Only little work can be found on the problem of optimizing timetables by synchronizing routes in order to minimize transfer waiting times. Existing literature on optimization within public transportation is generally focused on fleet management, crew scheduling, frequency optimization or the application of ITS. The first two generally focus on optimizing the use of resources to perform the timetable of a given service network and not the scheduling of the timetable itself. Within ITS there are a number of papers that touch upon the issue of optimizing passengers waiting times (e.g. Aisu *et al* (2000) and Dessouky *et al* (1999)). Likewise papers like Constantin *et al* (1995) and Mekkaoui *et al* (2000) focus on optimizing frequencies.

The only paper we have found that resembles our approach to optimizing timetables is Ceder *et al* (2001). In there synchronisation is only achieved if two runs arrive at a stop at the same time and hence it applies a strict definition of synchronisation. The objective is to maximize the number of synchronisations. In our approach we relax the definition by allowing "semi-synchronisation". A run arriving at a stop prior to another run can be said to be synchronised to the later run because passengers can perform a transfer. Obviously the transfer is not possible from the later run to the earlier one and thus there is no synchronisation. Our objective is to minimize the transfer waiting time between the first and the later run. The transfer times are weighted with the number of passengers at the transfer and their value of times.

In section 2 our model is presented. The solution method will follow in section 3. Section 4 contains the data on which the solution method was tried and the results are found in section 5. Finally section 6 concludes the paper and presents areas of future work.

A TACTICAL MODEL FOR MINIMIZING TRANSFER TIMES IN TIMETABLES

This section will present a tactical optimization model based on a given service network. The model formulates the problem of finding an optimal timetable for all bus routes so that passenger transfer waiting times are minimized.

Model limitations

Given the time horizon of the process of collecting customer requests, evaluating them and presenting a manually revised service network and timetable this can be considered a tactical planning issue. Often, however, only small or no changes are made to the existing service network and most changes are made to the timetables. Therefore the model is formulated on a fixed service network. Frequencies on the individual routes are considered as a strategic decision, and are considered fixed as well. Having fixed frequencies is equivalent to having a fixed number of departures on any given route. Increasing frequency means extra departures and thus requires extra resources.

The limitation on the no additional resources requirement also implies that in vehicle times (IVT) remain constant and cannot be reduced. Thus the model can not change the time it takes to dirve between two stops. Reducing IVT would require investing in technology or infrastructure like bus lanes, ITS, new vehicles etc. The IVT are limited to being deterministic as well. Obviously having deterministic times in public transport systems is unrealistic. However it is a common assumption in planning practice. Uncertainties can be handled by adding buffer times into the timetables. The size of these buffer times is often subjective though, and the idea of having an optimal timetable somewhat vanishes with the inclusion of buffer times. In section 6 we will present an idea on how stochasticity could be included in the model.

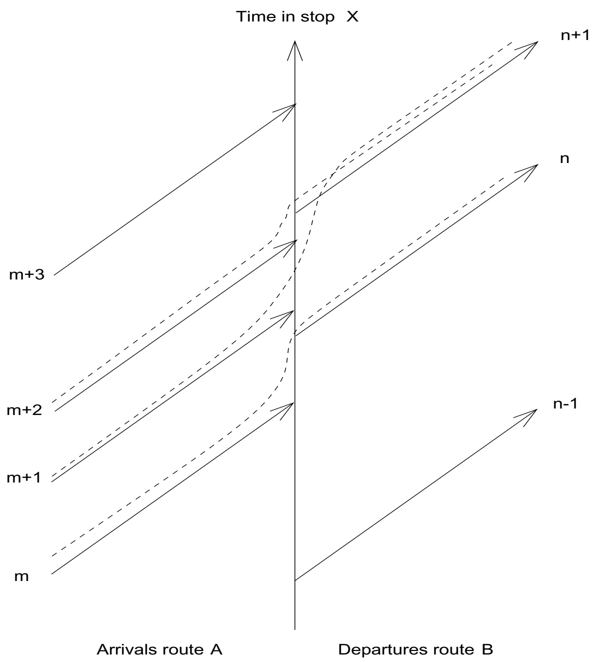


Figure 1.: Definition of a connection between an arrival and a departure

The objective of the model is to minimize the weighted sum of transfer waiting times for the entire system. It is obvious that passenger transfer patterns can reasonably be applied as weights. The problem is that the passenger transfer patterns are not independent of the timetable. Thus we will here just describe the weights as some general function.

Only passenger transfers between vehicles are considered. For instance the passenger flows on board of vehicles will not be included into the model. This requires stopping times at terminals to be fixed. If stopping times were variable the model could find an optimal solution in which a bus is put to stop in a terminal awaiting transfers from a later connection, even though passengers on board would have added extra waiting time to their total travel time. We have chosen to omit passenger flows on board of vehicles in order to reduce the complexity of the model, and thus keep stopping times constant.

The model takes into account the possibility of inter-terminal passenger transfers. A bus terminal and a train station might not be considered the same stop, but passenger transfers are often significant if walking distances are sufficiently limited. The transfer times between stops are also considered deterministic.

Model definitions

It is important to describe how a transfer is defined in the model. For any arrival at a stop it is defined that there is only one possible connection to another departure within each departing route. This is shown in figure 1.

The figure shows a stop *X* with a time axis. The left side of the axis shows arrivals in time *m*, *m+1, m+2* and *m+3* of route *A* at stop *X*. The right side shows three departures in time *n-1*, *n* and *n+1* of route *B* from stop *X*. The dashed lines show the connections between arrivals and departures. One notices that arrival *m* is connected to departure *n* and arrivals *m+1* and *m+2* are connected to departure *n+1*. The connections imply that there are solely passenger transfers between route *A* and *B* on the three connections. Hence there are no passenger transfers between arrival m and departure *n+1* although it is physically possible to change from *m* to *n+1*. Departure *m+3* has no connection to any departures and thus does not comply with the definition. Adding a pseudo departure at some later time to which it can be connected can solve this. Obviously passenger transfers to the pseudo departure are not allowed. The following sets are defined:

S: Set of stops in the network.

L: Set of lines in the network

R(l): Set of routes for each line

S(r): Set of ordered stops for each route

N(r, l): Set of runs for each route in each line

Ń(r, l): Set of time constrained runs

The following parameters are used:

β $_{n(r,l),m(r,l),s,t}$**(·)**: Weight in objective function of waiting time from arrival of run n(r,l) at stop s to departure of run m(r,l) from stop t. The weight is here generalised as some function of the system.

t$^{(trans)}_{n(r,l),m(r,l),s,t}$**:** Transfer time between run *n(r,l)* at stop *s* and run *m(r,l)* at stop *t*. Typically the same for all runs on a route.

f$_{n(r,l)}$: Headway between departure of run n(r,l) and run n(r,l)+1

t$^{(ivt)}_{n(r,l),s,t}$**:** In vehicle time for run n(r,l) between stop s and t.

t$^{(stop)}_{n(r,l),s}$**:** Stopping time at stop s for run n(r,l).

t$^{(first)}_{n(r,l)}$**:** Earliest time run n(r,l) may start at its first stop.

t$^{(last)}_{n(r,l)}$**:** Latest time run n(r,l) may start at its first stop.

The following decision variables are used:

T$^{(d)}_{n(r,l),s}$**:** Departure time for run n(r,l) at stop s

T$^{(a)}_{n(r,l),s}$**:** Departure time for run n(r,l) at stop s

ρ$_{n(r,l),m(r,l),s,t}$**:** Is equal to 1 if there is a connection between run n(r,l) at stop s and run m(r,l) at stop t. Else it is 0.

T$^{(wait)}_{n(r,l),m(r,l),s,t}$**:** Waiting time when transferring from run n(r,l) at stop s to run m(r,l) at stop t.

The Model

Given the definitions from section 2.2 we will now present an optimization model (shown in figure 2), that solves the problem of finding the optimal timetable for a service network. In the formulation the dependencies of sets are omitted and replaced by the following notation (·). The same notation is used for the weights β in the objective function to illustrate that it is a function of the decision variables of the model.

Equation (1) is the objective function of the model and states that the optimal timetable is found by minimizing the sum of the weighted transfer waiting times. Equation (2) limits the number of connections from an arriving run on a route to other runs on the same other route to 1 as defined in section 2.2. Since the objective function in equation (1) will minimize the total weighted waiting time, all the ρ will be chosen so

$$\min \sum_{n(\cdot),m(\cdot),s,t} (\beta_{n(\cdot),m(\cdot),s,t} \cdot T^{(wait)}_{n(\cdot),m(\cdot),s,t}) \tag{1}$$

subject to

$$\sum_{n(\cdot)} \rho_{n(\cdot),m(\cdot),s,t} = 1 \qquad \forall m(\cdot) \in N, \forall s,t \in S \tag{2}$$

$$M(1-\rho_{n(\cdot),m(\cdot),s,t}) + T^{(wait)}_{n(\cdot),m(\cdot),s,t} \geq T^{(d)}_{n(\cdot),s} - T^{(a)}_{n(\cdot),t} - t^{(trans)}_{n(\cdot),m(\cdot),s,t} \qquad \forall m(\cdot), n(\cdot) \in N, \forall s,t \in S \tag{3}$$

$$T^{(a)}_{n(\cdot),t} = T^{(d)}_{n(\cdot),s} + t^{(ivt)}_{n(\cdot),s,t} \qquad \forall n(\cdot), t = s+1, s \in \{1,2,\ldots,|S(\cdot)|-1\} \tag{4}$$

$$T^{(d)}_{n(\cdot),s} = T^{(a)}_{n(\cdot),s} + t^{(stop)}_{n(\cdot),s,t} \qquad \forall n(\cdot), s \in \{2,\ldots,|S(\cdot)|-1\} \tag{5}$$

$$T^{(d)}_{n(\cdot),s} = f_{n(\cdot)} = T^{(d)}_{n(\cdot)+1,s} \qquad n(\cdot) \in \{1,2,\ldots,|N(\cdot)|-1\}, s=1 \tag{6}$$

$$T^{(d)}_{n(\cdot),s} \leq t^{(last)}_{n(\cdot)} \qquad n(\cdot) = \acute{N}(\cdot), s=1 \tag{7}$$

$$T^{(a)}_{n(\cdot),s} \geq t^{(first)}_{n(\cdot)} \qquad n(\cdot) = \acute{N}(\cdot), s=1 \tag{8}$$

$$T^{(wait)}_{n(\cdot),m(\cdot),s,t} \geq 0, \rho_{n(\cdot),m(\cdot),s,t} \in [0,1] \tag{9}$$

Figure 2.: The optimization model for the optimal timetable problem

that there are connections to the first possible other runs on the other routes. On the right hand side of equation (3) the transfer waiting times are calculated as the difference between the arriving run and the departing run minus the transfer time between the arrival point and departure point. If there is a connection between the two runs, the first half of the left-hand side will be zero, and minimization will make sure the waiting time is computed as wanted. However, if there is no connection, the first half will become *M* (*M* being a sufficiently large number) and the waiting time will be computed to zero under minimization and hence will not change the objective function value. Equations (4) and (5) are flow constraints connecting the departure and arrival times at the stops for each run. Equation (4) states that the arrival time at the next stop is equal to the departure time of the previous stop plus the in vehicle time between the stops. Equation (5) states that the departure time at a stop is equal to the arrival time plus the stopping time at the stop. Equation (6) is a headway constraint that spaces out the runs over the timetable period. Equation (7) and (8) constrain certain runs to start respectively before or after some given time at its initial stop. By setting the earliest departure time equal the latest departure time one may specify any runs departure time. Finally equation (9) is non-negativity constraints on the waiting times and binary constraints on the connection variables.

SOLUTION METHOD BASED ON TS

One may notice that equation (1) is nonlinear. In equation (1) the weights β are a function presumably of the decision variables of the model. Adding to the complexity are the binary constraints on the connection variables. Also considering the size of realistic problems (public transportation network in cities) no exact solution method was considered. Consider a simple example of 1 transfer point between two lines, each line having 2 routes (one each way) and each route having 3 runs. Assume that runs can only connect to runs on either routes on the other line. This will yield 4 routes x 5 potential connections for each run = 20 binary connection variables.

Instead the metaheuristic Tabu Search (TS) was used to develop a solution method to the model. The general TS algorithm will not be discussed here, but a good insight can be found in Pirlot (1996).

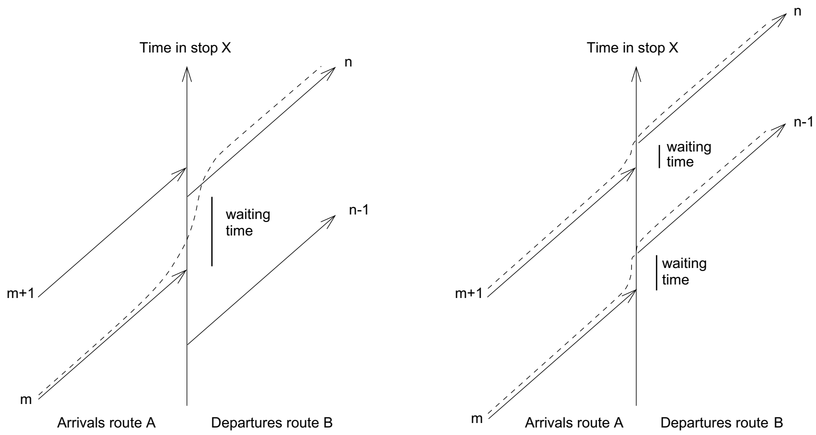


Figure 3.: Reason for aggregation of passenger transfer patterns.

Definition of and choosing neighbour timetable solutions

The definition of a neighbour solution (almost identical solution) to a timetable comes instinctively. Changing all the runs on one route by a minute either forward or backward will yield a slightly different timetable. Hence a neighbour timetable solution is defined here as changing all the runs on one route by +/-*n* minutes. The neighbourhood is the set of timetable solutions that can be created by changing any one of the routes in an initial timetable by +/-*n* minute.

In a timetable of reasonable size e.g. 100 routes there will be 200 neighbour solutions in the neighbourhood. Evaluating the entire set of solutions in each iteration would prove time consuming. Therefore only a subset of solutions will be evaluated. A reasonable size of the subset cannot be estimated as such and must be assessed experimentally. The subset neighbour solutions from the neighbourhood are determined by randomly picking a route and the minute change *n* within some limits.

Tabu list, aspiration criterion and stop criterion

It is not possible to keep entire timetable solutions in the tabu list because that would require large amounts of memory to keep and processing time to read. Even keeping just the starting times of the first runs for each solution would require important computation. The content of the tabu list is restricted to contain only the number of the changed route and the number of minutes it has been changed. Limiting the content of the tabu list to this will result in the exclusion of several feasible solutions via the tabu list even though they have not been examined. Therefore an aspiration criterion will be used to permit tabu solutions. The criterion used here is to allow a tabu solution if its objective value is better than the current best found value.

The algorithm is put to a stop if the improvement of the best solution is less than α % over *X* iterations. The *X* iterations will be referred to as GAP.

Evaluating timetable solutions

To evaluate the timetable solutions the objective function from the model presented in section 2 was used. The determination of the weights is discussed in section 4. In each stop with possible transfers (i.e. 2 or more lines use the stop) all runs for each route are connected to another run from another route using the definition from section 2. The connection is done by applying a simple search procedure. The waiting time is calculated similarly to equation (2) in the model knowing the actual connections.

For the initial solution value all connection and waiting times are calculated. This is not necessary when evaluating new solutions. Here only the runs affected by a change on a route, i.e. runs on the changed route and other runs with possible connections to the route's runs, are evaluated and the difference in weighted waiting time is subtracted the old solution value.

APPLYING TABU SEARCH

In order to test the TS-algorithm, a realistic case was constructed on which TS could be applied. The data for the case originate from the Copenhagen-Ringsted model (CRM). The data for the CRM contain a slightly aggregated version of all the routes and schedules of Eastern Denmark. More information on the CRM can be found in Nielsen *et al* (2001).

The CRM-data was reduced to contain only the routes within the greater Copenhagen metropolitan area and further reduced to the time interval 16:30 to midnight. Using the CRM had the benefit of having access to the EMME/2 route choice model, in a version where it has been extended by macro programming to include stochastic to find a stochastic user equilibrium Nielsen *et al* (2002).

The route choice model provided the passenger transfer patterns. These data were aggregated from the number of transfers between single runs to the number of transfers between routes.

During the computation of the evaluation of a timetable solution the passenger transfer patterns are then distributed on the runs according to some function of the number of connections and the waiting times. This is an approximation which is acceptable as long as passenger transfer patterns are close to an even distribution of transfers among the runs of the routes. However it is necessary to make the aggregation. This is explained in figure 3.

On the left part of figure 2 only run *m* is connected to run *n*. All passengers transferring from route *A* to *B* use this connection. Optimizing on this example would only minimize the waiting time for the single connection. On the right side a better solution is shown, which can only be achieved if passenger transfer patterns are not fixed on routes. The aggregated transfer patterns are dispersed on the possible connections according to some function of the number of connections and the transfer waiting times. The aggregated passenger transfer patterns were used as weights in the objective function used to evaluate the timetable solutions.

The passenger transfer patterns are divided in three distinct passenger groups 1) business passengers, 2) commuters and 3) recreational passengers. Passengers in each of the three groups have different perceptions of waiting times. Using CRM time values for each of the three passenger groups have been estimated. The values are 270 DKK/hour, 38 DKK/hour and 28 DKK/hour respectively. The values of time can be perceived as the amount a passenger is willing to pay to reduce waiting time in transfers by one hour. Multiplying the time values with the passenger transfer patterns and summing the three values is eventually used as weights. The size of the CRM-data set is:

• 270 lines, 662 routes , 45 fixed (train)

• 7182 runs, 1344 stops and 43,346 non-aggregated transfers.

RESULTS

We performed 4 runs with our Tabu Search algorithm, TS-1, TS-2, TS-3 and TS-4 on our CRM-data. The parameter settings for each run and the obtained results and computation times are illustrated in table 1.

In figures 4 and 5 is shown the value of the best found solutions as a function of the number of iterations and computation times. TS-4 converges well both with respect to iterations and computation time. TS-2 and TS-3 follow TS-4 until they

	TS-1	TS-2	TS-3	TS-4
Initial value (DKK)	714.000			
Value of best found solution (DKK)	685.000	659.000	666.000	634.000
Improvement (%)	4,1	7,7	6,7	11,1
Time consumption (hours)	3,0	33,0	23,1	130,9
Iterations	255	660	365	2120
Parameters				
Size of neighbourhood subset	*10*	*20*	*20*	*20*
Length of tabu list	*50*	*50*	*50*	*100*
Max. numerical change on a route	*10*	*10*	*20*	*20*
GAP	*100*	*200*	*100*	*200*
Stop criterion on GAP (%)	*1*	*1*	*0,1*	*0,1*

Table 1.: Resulta with Tabu Search

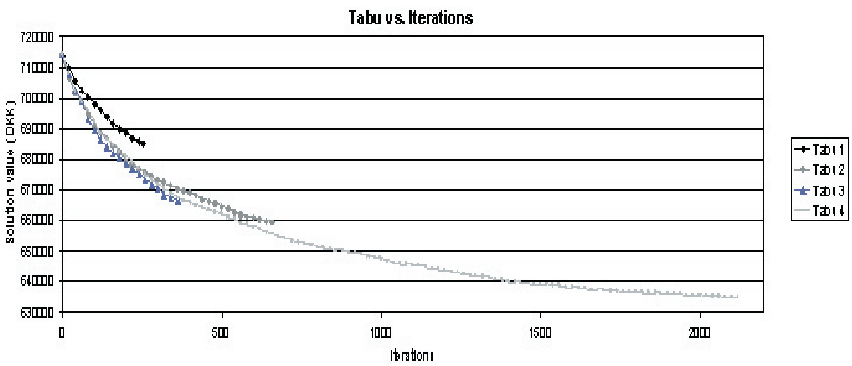


Figure 4. Tabu Search results as function of iteration

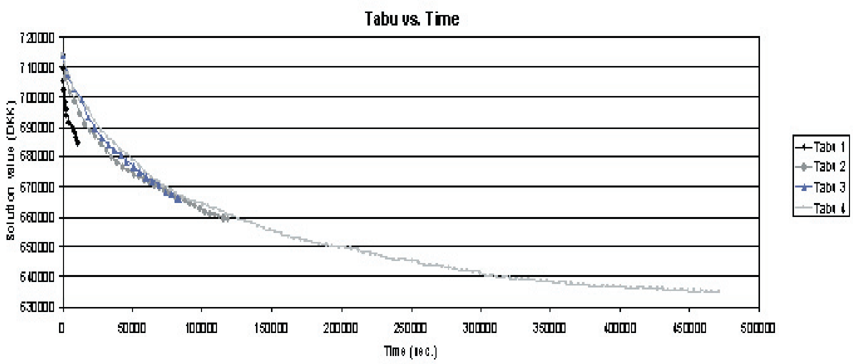


Figure 5. Tabu Search results as function of time

are stopped. Convergence could therefore be expected. Only TS-1 diverges slightly. This can be explained by only a subset of 10 neighbours was examined. Each iteration is therefore performed faster, which yields a faster decrease with respect to time. However the fewer number of examined neighbour solutions means poorer best solution found for each iteration and therefore slower decrease with respect to iterations.

All functions are rather smooth which points to the proper functioning of the algorithm. Nevertheless running times of the algorithm are important. TS-4 spent 131 hours before stopping. The long running times are not excessive if compared to a planning horizon of 6 months. However improving algorithmic efficiency should be considered.

The initial value on the timetable given in the CRM-data was 714.000 DKK. All four runs yielded improvements between 4% and 11%. Taking the best run, TS-4, the solution value was improved to 635.000 DKK; a 79.000 DKK improvement. This number is a measure for the improvement in waiting times for one evening in the greater Copenhagen metropolitan area. It adds up to 29 million DKK for a whole year. This number may not be seen as a potential cost reduction for the operator but as the benefit given passengers.

CONCLUSSION

The results presented in this paper indicate that there is a potential 29 million DKK to be saved in time value for the evening hours of the Greater Copenhagen metropolitan area. By this we have proved that there is an interesting potential in reducing waiting times within public transport. On top of the 29 million DKK obtained for evening hours should be added the potential savings for morning and daytime hours.

The CRM-data provide a realistic case, but implementing the model within a local or regional planning authority would not only reveal the time value saving potential but also provide feedback on the applicability of the optimized timetables and thereby provide useful input on how to modify the model concept to solve real problems. An implementation of the model would probably result in a DSS for revising existing timetables. However the model has the advantage that it can give a solution to any system and hence can be used to give an initial timetable guess for a modified system. This e.g. could have been an interesting feature to use when Copenhagen public transport authorities (HUR) introduced new bus lines in connection with the opening of the Copenhagen Metro.

One modification that could prove to be useful is to relax the fixed headway between the runs as expressed in equation (6). Instead the headway could be allowed to vary within some bounds. This would allow more flexibility into the system and probably give rise to more connections between runs and shorter waiting times. However even longer computation times are expected if a new solution method embedding the flexibility is implemented.

An interesting area in which the model could be further improved is the ability to handle the stochastic nature of operating timetables. Having deterministic in vehicle times is an approximation to reality where congestion and other delays affect the system. Introducing stochastic in vehicle times would require simulating the times for a number of scenarios and somehow averaging the optimal solutions to each scenario to obtain a general optimal solution. An alternative approach which could prove to be more efficient would be to find the optimal solution to the deterministic problem and then evaluate the deterministic result by simulation. This simulation would then feed back information to the deterministic problem on how to improve the timetable (e.g. changing connections that are often missed). Iterating between the solver and the simulator would eventually provide a good timetable solution.

Adding an iterative procedure or even integrating the model with a route choice model would provide another way of evaluating the optimized timetables. This would give the possibility to estimate new passenger transfer patterns as a result of an optimized timetable. The new estimated transfer patterns could be fed back to the model and a new re-optimized timetable could be found. Considering the computation times of both the model and a route choice model some kind of trade off is needed with the results obtained in the iteration process.

REFERENCES

Aisu H., Torri K., Tanaka T., Oba Y., Seki Y. (2000) A Dynamical Operation Scheduling for Dual-mode Transport System, *2000 IEEE Intelligent Transportation Systems, Conference Proceedings,* Dearborn (IM), USA, October 1-3, 2000

Ceder A., Golany B., Tal O. (2001) Creating bus timetables with maximal synchronization, *Transportation Research Part A* 35(2001) 913-928

Constantin I., Florian M. (1995) Optimizing Frequencies in a Transit Network: a Nonlinear Bi-Level Programming Approach, *International Transportation Operation Research* Vol.2, No. 2, pp.149-164, 1995

Dessouky M., Hall R., Nowroozi A., Mourikas K. (1999) Bus Dispatching at Timed Transfer Transit Stations Using Bus Tracking Technology, *Transportation Research Part C* 7(1999) 187-208

Mekkaoui O., Palma A., Lindsey R. (2000) Optimal Bus Timetables and Trip Timing Preferences, Université de Cergy-Pomtoise & Université de Paris X - Nanterre

Nielsen, Otto Anker; Hansen, Christian Overgaard & Daly, Andrew (2001). A Large-scale model system for the Copenhagen-Ringsted railway project. Paper in Travel behaviour Research: *The Leading Edge*. Book edited by David Hensher. Pergamon press. Chapter 35, pp 603-626.

Nielsen, Otto Anker (2002). A Multi-class timetable-based transit assignment model with error components in the utility functions. *First International Workshop on the Schedule-Based Approach in Dynamic Transit Modelling.* 27/5. Ischia, Italy.

Pirlot M. (1996): General local search methods. *European Journal of Operational Research 92.*

By Michael Berliner Pedersen, Centre for Traffic and Transportation, Technical University of Denmark.

Lore consequat. Duisim vel in hent enim dit ullum zzriureet am, corem zzrilisl illuptate tatum velisi tat iure molore commy nonullamet, quisl dolortis ad tat. Ut alisi blaor ipsum nullam, sed et nosto eui bla feuisit velis ex elesed tem duis nonum nim dolorperos nibh euguer senit vero odolum do dolor il ing ex endre faccummod et eum dit nostrud dolenibh exerit autat ex eu faccum nulla

Operations Research in Denmark - still going strong after 40 years

Figure 1.: »Rolighed« where the Dansih Orperations Research Society was established back in 1962.

As new research areas appear they will if the ideas are strong start to build their own organisations. This happende for Operations Research (OR) in the years following the second world war. Here OR emerged as a formiddable tool for optimize the usage of scarse resources and coordinate an multinational effort. At the end of the war the lessons learned were transferet to many different planning problems in the civilian society; OR got a solid theoretical foundation and became an academic discipline with England as poineer. Here a group of visionary people already in 1948 founded the *Operational Research Club.* This was followed by the foundation of a number of national societies in both Europe and the United States, and in 1959 IFORS the International Federation of Operations Research Societies was founded. In parallel to this development a number of dedicated OR-journals emerged as *Operational Research Quarterly* (1950), *Operations Research* (1952) and *Unternehmensforschung* (1956).

In 1957 the *First International Conference on Operational Research* was held in Oxford. The meeting was attended by 250 researchers from 21 countries. Among these was the Danish pioneer especially within telecommunications professor Arne Jensen and his younger fellow Eric Johnsen (later professor of OR at the Copenhagen Business School).

After the conference they went home and during a Danish conference i 1962 at the »Roligheden« the Danish Society of Operations Research (DORS) was founded. The first president of the society was Arne Jensen. Within the first year the society had more than 60 members. In many other ways 1962 became important for the udbredelse of Operations Research. Erik Johnsen published the first Danish book on OR and at the Technical University of Denmark the chair in applied mathematics was split into three chairs, so i 1963 Arne Jensen became professor of statistics and OR. Furthermore Peter Pruzan returned from the United States as Ph.D. in OR from Case-Western Reserve Univesity and gave the community fresh impulses just as it had happened a few years earlier when Erik Johnsen returned from Princeton. Peter Pruzan was part of a small group that founded the first OR consultancy company in Denmark i 1967.

OR people were quick to se the potential in the usage of computers. In Denmark today OR is an active field of research at several universities. The largest groups can be found at the Aarhus School of Business, University of Copenhagen and the Technical University of Denmark, but groups also exist at the Copenhagen Business School and The University of Southern Denmark.

As OR has been developing as an reserch field in Denmark also the industry is active. As huge quantities of computer power are getting affordable for almost any company, and as company data are available digitally small companies have emerged to use OR and optimization in a number of different industries. In Denmark the transportation sector has for noumerous year played a vital role and there are several companies in Denmark that are now developing and implementing systems for route planning, supply chain management, and manpower planning. [jla]