

Summary

The main purpose of this document is to describe the Railway Capacity Management (RCM) model developed for the national freight model in Sweden – Samgods. As a background to RCM, the Samgods model is described in some detail, in particular the logistics model. Uncommon terms and notations are explained in some detail in Chapter 1.2.

In Samgods 1.0 (the first version of the model) capacity constraints with respect to flow on different links were not considered. This is a limitation since modeling capacity is required to enable feasible solutions with respect to railway flows in particular. In order to handle this it has been decided to use a linear programming model (LP), with column generation, for railway capacity modeling as an a posteriori model applied after the standard (basic) logistics model – *Standard LogMod*. The capacity is measured in terms of summarized number of loaded and empty freight trains per year on bidirectional railway links (entered as trains per day in input data). Efficient computation of the train loads on the links are enabled by using the spanning trees determining the railway paths with the least generalized costs. In Samgods it is easily accomplished since only a single path is used per railway origin-destination-relation. Should multiple paths be used, the data volumes would be larger, and also the distribution of flows between paths would require attention. An add-on factor, to estimate the sum of loaded and empty flow per PWC-relation (also denoted firm-to-firm relation or f2f-relation), is determined on the basis of the empty and loaded OD-matrices respectively. To ensure that a feasible solution can be identified we initially add a second column, not using railway if possible, selected from a few alternative transport chain solution alternatives. In practice this will allow us to identify a feasible solution, but just in case we also introduce the possibility to acquire additional, fictitious, capacity inside the LP model.

The idea is to use the dual values (= marginal costs) from these capacity constraints and feed them back into the cost function of the logistics model. With this functionality in place we will iterate a reasonable number of times (between the logistics model and the LP model) and generate additional columns to the LP model where appropriate, managing only transport chains initially using railway, to find a good enough solution using the introduced marginal costs.

When finalized, the set of transport chains with and without railway use will be assembled into a final solution that satisfies the capacity constraints. The linear programming solution is implemented directly, i.e. we use the transport chains (= columns) selected by the optimizer. Optionally a rounding off procedure of some kind could have been used, but it was decided to split the solution into its fractional parts. A more advanced alternative then could be to formulate a smaller mixed integer linear programming problem for the split alternatives concerned, and by this enforce the use of integer solutions.

The purpose of this paper is to present how the column generation LP method works, both the basic idea and its implementation. Some insight into its performance on regular size problems for the national freight model Samgods is also presented. The project is financed by the national Transport Administration in Sweden.

Keywords: logistics, capacity management, railway transports, column generation, linear programming

See Chapter 1.2 for more details.

Content

1	Introduction	3
1.1	A Small Linear Programming Example	12
1.1.1	First linear programming model – LP0	14
1.1.2	Second linear programming model – LP1	15
1.1.3	Third ++ linear programming models – LP2, ... and convergence	16
1.2	Terms and definitions	17
2	The Logistics Model in Samgods and Associated Components	19
2.1	Producer-Consumer Flows in the Base Matrices	19
2.2	Network Model: LOS matrices	19
2.3	Logistics Model	20
2.4	Network Model: Assignment of OD matrices	21
3	Linear Programming Approach Overview	22
3.1	Update with Samgods version 1.1	23
4	Linear Programming Approach Technical	25
4.1	Linear Programming Formulation	28
4.2	Handling of empty train flows in RCM by LP	31
4.3	Railway Capacity Management Integration with the Standard LogMod	34
4.3.1	Developed Software	37
4.4	Update with Samgods version 1.1	37
5	Some Results	38
5.1	Version 1.0 Results Main 2030	38
5.1.1	Version 1.1 Results	41
5.2	General Observations among the Results during the Model Development	41
5.3	OD matrices for empty trains	47
5.4	Assignment of empty trains	49
5.5	Final Comments	50
6	A few LP solver software options	51
7	Summary and Conclusions	56
	Acknowledgements	57
8	References	58
8.1	Linear Programming Software	59

Appendix A: Some LP model details	60
Appendix B: Method for distribution of empty flows in the Extract-module	62
Appendix C: Spanning tree data	65
Appendix D: Some details on the capacity adjustment procedure using LP - LPnAdjCap	67
Appendix E: Logistics model details	68

1 Introduction

Readers of the version 2014 of this report may wish to know the differences compared to last years report without having to read through all the text. The changes are:

1. We have shifted the linear programming solver to Soplex version 2.01 (see section 8.1) for higher efficiency.
2. An option to use only one solution alternative for small shipment sizes is available (the first, standard logistics model solution is always included by default). It is invoked by setting a lower bound on the shipment size, e g 2 tonnes, for allowing multiple solutions. Using this cuts down the number of generated so called convexity constraints (that guarantee that at most 100 % of the alternative solutions per firm-to-firm-relation are used), and since the solution time in principal is proportional to the square of the number of constraints this has a considerable impact on the solution time with small adverse effects.
3. Now there is also an option to lock system train flows to a particular route by determining a number of dummy transfer points along the route. This must be set up manually by the user in a text file as described in Section 4.4.

Studies made of the national freight model in Sweden, Samgods, around the year 2000 came to the conclusion that it had to be updated to better serve as an instrument for policy analysis and cost-benefit analysis of infrastructure investments at the national level. Rather than using statistics on origin destination matrices for different modes as the basis for the model, a basic underlying idea was to acquire data on commodity flows from producers to consumers, and to construct a demand model for goods based on those data. Yet another fundamental approach was to move towards a model handling both the goods transport and the impact of logistic considerations at individual firm levels. A summary of the analysis and the motivation is presented in SIKÅ [2004]. This led to a number of ambitious commodity flow surveys (2001 and 2004/05). The flow surveys hold data on transported quantities, locations of senders and receivers, border passing locations, business sector of the senders, aggregate information on transport chains and ordinary up-scaling factors for the sampled data. Together with foreign trade statistics, business registers and socioeconomic data, the survey provided the base for construction of so called base matrices for the year 2005.

In the old Samgods model all network analysis of the goods demand was handled by the multi-modal, multi-product network analysis tool STAN develop by INRO, in which the system optimum equilibrium, in terms of generalized costs, was the criteria. The above cited analysis concluded that a new model enabling, in a more general manner, the handling of the issues below was needed.

1. Firm to firm decisions related to logistics matters.

2. Logistics issues such as inventory holding, capital costs during transport and handling, transfer costs at terminals.
3. Economies of scale.
4. More realistic model response to cost changes, i.e. more realistic elasticities.

The implemented logistics model is described in de Jong et al [2011]. A key concept introduced is the ADA-principle, which stands for Aggregate-Disaggregate-Aggregate principle, c f Figure 1.1. It means that the model first operates at aggregate levels on the demand side, then secondly at a disaggregate firm to firm level with logistics solutions, and thirdly at the aggregate level again with final assigned flows of vehicles (loaded and empty) and tonnes in the transport network. The disaggregate step involves:

1. Splitting up the zone-zone demand into combinations of firm-to-firm demands, where each pairwise combination of small, medium-sized and large firms comprises a number of firm to firm relations.
2. Simulation of logistics behavior in the firm to firm relations, comprising inventory holding considerations and selection of transport chains providing the minimum total costs. Decisions for each firm to firm relation concern the choice of shipment size, transport chain with transfer locations between modes (using a set of optional, predefined transport chain types) and vehicle numbers and sizes for each leg in the transport chain.

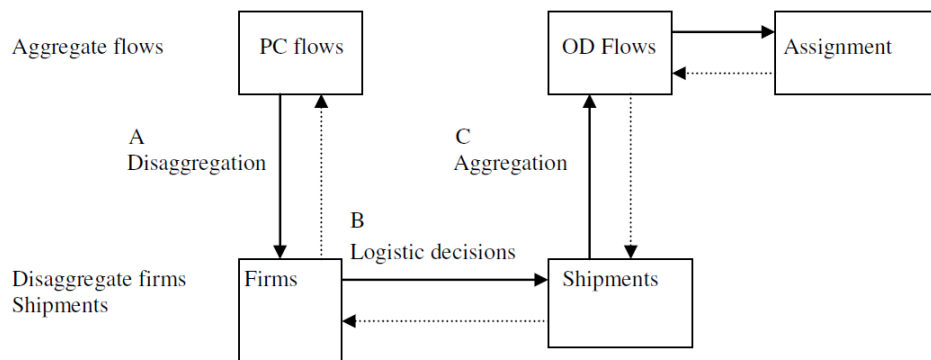


Figure 1.1 A description of the so-called ADA-principle. Source: deJong et al [2011].

The model above does not handle constraints in terms of link flow capacities, which is a very important aspect for domestic railway transports. Therefore a post processing model, RCM, has been developed to manage the capacity relevant matters.

Railway Capacity Management. We first illustrate the current task which is to ascertain that the number of trains (loaded + empties) stays within the capacity limits. The capacities are expressed as the sum of trains per year in both directions on all national railway links (trains per day in input data). After the standard logistics model, denoted Standard LogMod, we may have a situation as shown in the Figure 1.2 network. We see that the rail usage is well above the actual capacity, in particular on Södra Stambanan through Skåne.

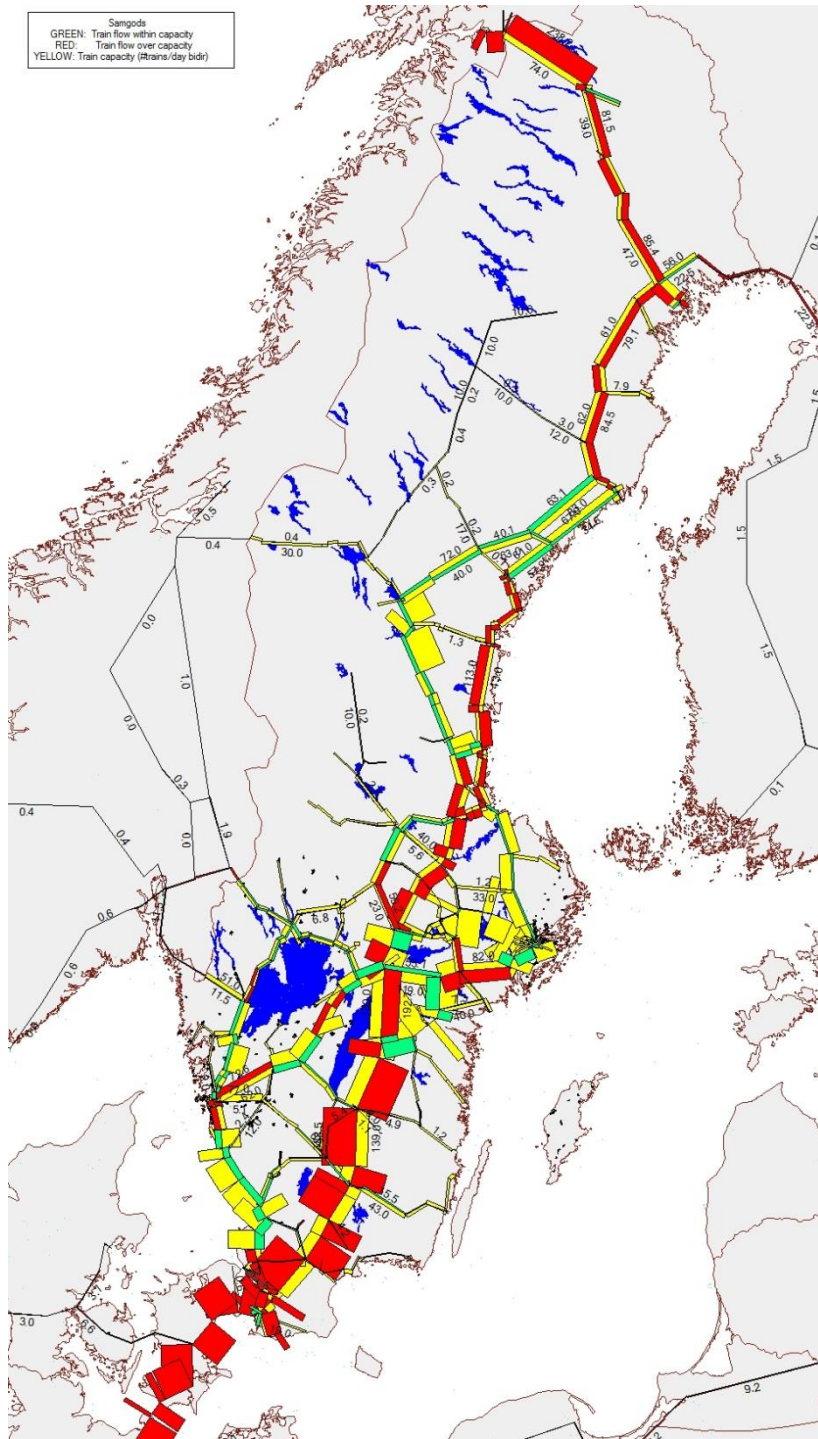


Figure 1.2 Train flows after Standard LogMod in green (within capacity) and red (over capacity). The capacity is shown as yellow bars.

The purpose of this paper is to present how the column generation LP method works, both the basic idea and its implementation. We present how strict capacity constraints for railway, with capacity expressed in terms of an allowed number of trains (loaded + empties) per year (trains per day in input data) on all bi-directional railway links in Sweden, can be handled using linear programming. The linear programming model defines mathematically the problem of selecting suitable (=cost minimizing) combinations among the first best solution and a number of viable alternatives, for all firm-to-firm flows using railway in Standard LogMod. For a basic reference on linear programming, see for example Bazaraa and Jarvis [1977]. The best solution is the one minimizing the additional cost for using the selected alternatives while simultaneously cutting down the rail capacity usage to stay within the capacity limits. The simultaneous consideration of the capacity constraints is handled within a system of linear equations, in which the left hand side represents the capacity usage and the right hand side the capacity limitations. Not all constraints are used up to 100 %, quite few actually, and this is handled by using extra so called slack variables (or variables for unused capacity). The selection of combinations among the firm-to-firm flows are forced to be done in such a manner that the equations are satisfied. In general, the use of an alternative solution (than the Standard LogMod solution) will cost more but use less rail capacity on at least one link. The linear programming algorithm identifies an optimal solution to this problem, i.e. a cost minimum solution satisfying the capacity constraints. A simple analogy is to say that the LP model identifies the firm-to-firm transport solutions that cost as little as possible to move, while simultaneously removing as much capacity usage as possible.

RCM can be stopped with the solution from the first LP model, called LP0. Then a feasible solution from the LP model is constructed by using the transport chain solutions from the LP model. Any non-integer solution is handled by allowing fractions of two alternatives to be used in the actual solution (the fractions sums to one). An example is the mix of two solutions with fractions 0.6 and 0.4 respectively, e.g. 60 % of the Standard LogMod solution and 40 % of the second best solution. Then we only need to finalize by carrying out the extraction of OD-matrices and assign everything to the network.

In order to find better solutions we set up a so called column generation procedure in which the LP model representation of the actual problem is described in more detail by adding more options (in LP1, LP2, etc). In general, more options will enable better solutions to be found. In our case more options means providing more alternatives to the LP model (more columns). This is accomplished by doing three tasks:

1. First we use the “price” or marginal cost on the links used up to the capacity limit. These prices express the marginal cost associated with moving the rail transports away from the link, actually the last one moved away. The LP model begins “moving” transports with low costs compared to the reduced capacity usage, but is successively forced to move transports with higher and higher costs. Any new transport chains must be willing to pay these marginal

costs, and we compute their impact on all train OD-relations in the network. Therefore these costs are summed over the paths for all train OD-relations, and entered as a new set of LOS-matrices in terms of cost per train for using different OD-relations. Note that the consolidation levels and volumes from Standard LogMod are not modified in this process. Any new transport chains only have an effect on the link flow volumes on their paths in case they are used, which of course affects all rail OD-relations using the links concerned.

2. Now a special version of the logistics model is run (BuildChain4RCM and ChainChoi4RCM) that only considers the firm-to-firm relations using rail in the first place, and that considers also the marginal costs in the cost minimization process. Only the third iteration of the Standard LogMod procedure is run, the one that does not change the consolidation levels and consolidation volumes. Changing these would change the conditions for many/all present solutions produced so far and that is deemed unwise. The procedure will produce new options with the current marginal cost structure to be entered into the LP model. Before entering them, they are checked for uniqueness since often already existing firm-to-firm solutions are produced. Solution copies should of course not be entered into the model. **Note!** The introduced marginal costs will only be used during the construction of new transport chains. The marginal costs will NOT be included in the cost calculations in any other place (such as the output reports or as imbedded in objective coefficients for new alternatives), their purpose is to guide the solution process. The effect on the total cost appears as the sum of the extra costs associated with not using the first best solutions.

3. Before the next LP model can be setup, we must construct the loaded and empty train OD-matrices associated with the latest LP-solution to enable us to predict the share of empty trains that are associated with each loaded train. This is done by implementing the latest LP-solution as intermediate ChainChoi-solutions containing both non-rail firm-to-firm transport solutions and the LP-guided firm-to-firm transport solutions. With these in place the Extract-procedure will derive the OD-matrices.

In some more detail RCM is described in the four steps below:

1. Setting up the first best solution from Standard LogMod and suitable second best alternatives that may be used to reduce the railway capacity usage.
2. Selection of transport chain solution combinations, in principle the first solution or any one of the possible alternatives, for all firm-to-firm-relations using railway in Standard LogMod. The selection shall minimize the total additional costs (for not always using the first best solutions) while satisfying the capacity constraints.
3. Computation of marginal costs per train on capacitating links. These will be used as additional costs for the logistics model in the search for good alternative solutions while recognizing the overload through the cost mechanism. Alternatives may be generated that use alternative transport chains, or use different routes within the current transport chain (by shifting to other transfer points). However, in many cases the generated transport solutions are identical to the ones already in the model (in which case they are skipped during construction of the next LP model).
4. The process in steps 1-3 is iterated a number of times until the search for new alternatives does not improve the solution enough to make it worthwhile to continue.

In Figure 1.3 we show a very aggregated flow chart of the system.

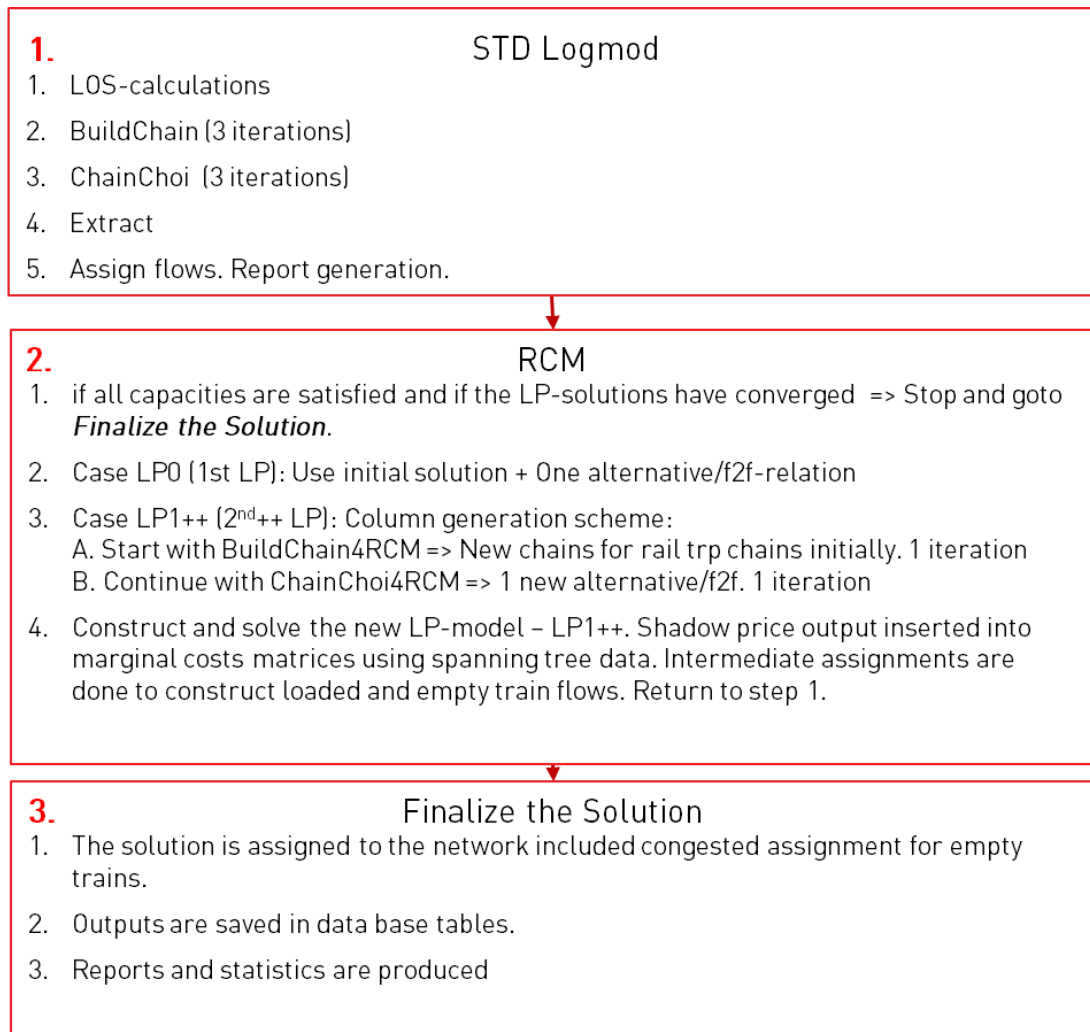


Figure 1.3 Flow chart of the logistics model and RCM.

After having applied the above procedure in one step (using the so called LP0-formulation) we may obtain a solution like the one in Figure 1.4, the so called RCM-solution (Railway Capacity Management). As shown in the figure almost all the capacity constraints are satisfied, and the remaining overflows (in red) are not so far from the capacities (yellow).

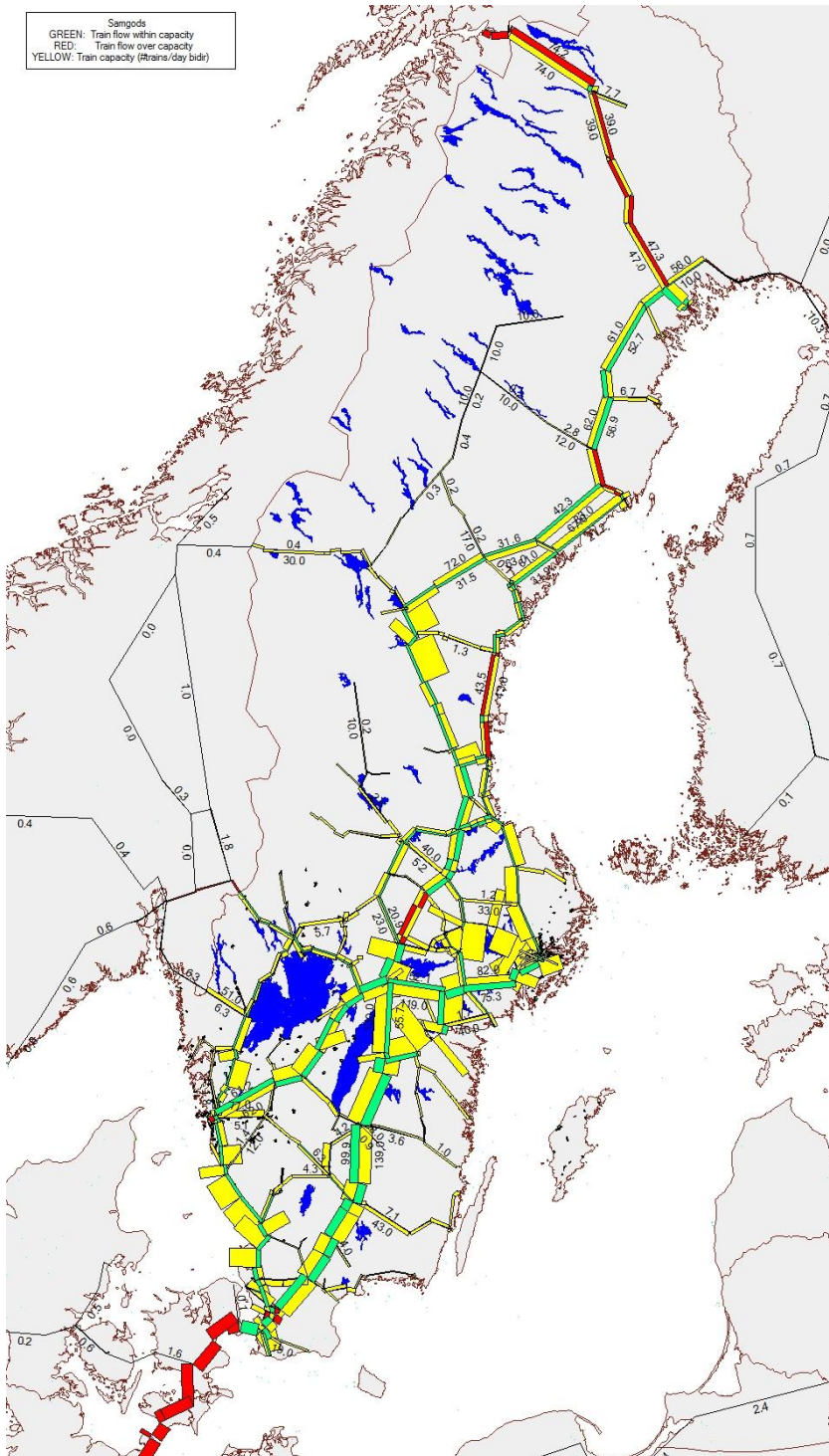


Figure 1.4 Train flows after RCM logmod in green (within capacity) and red (over capacity). The capacity is shown as yellow bars.

Following this initial introductory part, Section 1.1 provides a small example of how the linear programming method is used in rail capacity management. The example only holds a single capacity constraint to enable a model that easily can be understood, and that can be subject to the readers own checking. It can be skipped for now, and possibly be returned to in conjunction with reading the Chapters 3 or 4.

Section 2 presents the logistics model in Samgods and the interactions with other components of the model system. Then in Sections 3 and 4 the linear programming approach will be presented, first in a less detailed description and secondly in more technical terms. A few LP-solver alternatives are presented in Section 5 and Section 6 holds some results. Finally, in Section 7 a summary of the method and its applicability is provided. A number of suggested improvements are presented in Section 7.1.

1.1 A Small Linear Programming Example

As a practical introduction to the presented railway capacity management method we start with a small example having 5 firm-to-firm transports using railway. The capacity on the single constrained link is 500 trains per year. On average the number of trains include 60 % loaded trains and 40 % empty trains, where the empty train are estimated from the number of loaded trains in each relation and the number of empty trains in the same relations (derived from the extract module after Standard LogMod).

Table 1.5 First and second best alternatives from Standard LogMod.

Nr	First best solution		Second best solution	
	Cost	Nbr trains on constrained link	Cost	Nbr trains on constrained link
1	100	100	150	0
2	200	180	300	125
3	300	250	510	0
4	400	300	680	0
5	150	75	500	0
SUM	1150	905	2140	125

From Table 1.5 we have that the Standard LogMod cost is 1150 but the capacity used is 905 which is way over the capacity at 500. The second best solution¹ together cost as much as 2140 but only uses a capacity at 125. What we look for is a mix of these two solutions that minimizes the total cost (or the extra cost) while satisfying the capacity constraint and

¹ A technical aspect is that the second best solution should not require the use of more railway capacity anywhere. If not satisfied, the third or higher best will be used under the same condition.

ascertaining that for each firm-to-firm transport all the transport is carried out, i.e. that the first and second best solution usage must sum to 100 %. It is assumed that any transport solution can be split into fractional parts (otherwise it is not a linear model), i.e. we do not require an integer solution requiring either that the first best solution is used to 100 % or that the second best solution is used to 100 %. Such a model would be an integer programming model, and it would be immensely more complicated to solve.

The mentioned small, mixed integer programming model mentioned in the Summary would be a heuristic method for dealing with only the fractional parts of the solutions. An integer programming model in principle requires that the constructed sub-problems in a branch-and-bound-tree can be solved quickly (within seconds preferably), and the sub-problems here may very well in practice be the full LP-problem with a few extra conditions requiring that a number of variables either are set to zero or one.

1.1.1 First linear programming model – LP0

Introduction of variables for the different alternatives in a small example with a single capacity constraint (see Chapter 4 for a model with multiple capacity constraints):

x_{ij} = share of firm-to-firm transport nr i that is transported with alternative j ($i = 1, 2, \dots, 5$, $j=0, 1$). The first best solution which is the Standard LogMod solution is labeled $j = 0$. The alternatives will be $j=1, 2, \dots, 9$ as they are constructed in the solution process. In this example we have only $j=0$ and $j=1$.

Equation (1.1) is the objective function, equation (1.2) the capacity constraint and equations (1.3) – (1.7) the so called convexity constraints requiring 100 % of the transports to be carried out one way or another. On the last row in Table 1.6 we have the required non-negativity constraints, all variables must be greater than or equal to zero.

Table 1.6 Linear programming formulation of this problem.

	First best solution	Second best solution		RHS	
min z =	100 * x10 + 200 * x20 + 300 * x30 + 400 * x40 + 150 * x50 +	150 * x11 + 300 * x21 + 510 * x31 + 680 * x41 + 500 * x51			(1.1)
subject to	100 * x10 + 180 * x20 + 250 * x30 + 300 * x40 + 75 * x50 +	0 * x11 + 125 * x21 + 0 * x31 + 0 * x41 + 0 * x51	≤	500	(1.2)
	1 * x10 +	1 * x11	=	1	(1.3)
	1 * x20 +	1 * x21	=	1	(1.4)
	1 * x30 +	1 * x31	=	1	(1.5)
	1 * x40 +	1 * x41	=	1	(1.6)
	1 * x50 +	1 * x51	=	1	(1.7)
	x10, x20, ..., x51 ≥ 0				

The optimal solution to this problem is presented in the framed parts in Table 1.7. Transports number 2 and 5 ($x_{20}=1$, $x_{50}=1$) are using the Standard LogMod solution, transports number 1 and 3 ($x_{11}=1$, $x_{31}=1$) are using the second best alternative, and finally transport number 4 use 81.67 % of the Standard LogMod solution and 18.33 % of the second best alternative ($x_{40}=0.8167$ and $x_{41}=0.1833$). In total it results in a cost increase to 1461.3 and a capacity usage equal to exactly 500.

Table 1.7 Optimal solution to the problem LP0 in Table 1.6.

LPO											Optimal LP-solution			
Variables	First best solution					Second best solution					RHS	Sum of costs + RHS	Marginal cost	
	x10	x20	x30	x40	x50	x11	x21	x31	x41	x51				
	0	1	0	0.817	1	1	0	1	0.183	0				
Costs	100	200	300	400	150	150	300	510	680	500		1461.3		
CAP001	100	180	250	300	75	0	125	0	0	0	≤	500	500.0	-0.933
Cvx01	1					1					=	1	1.0	150.0
Cvx02		1					1				=	1	1.0	368.0
Cvx03			1					1			=	1	1.0	510.0
Cvx04				1					1		=	1	1.0	680.0
Cvx05					1					1	=	1	1.0	220.0
Non-neg x10, x20, ..., x51 ≥ 0														

The marginal cost (or shadow price) for the capacity constraint at -0.9333 tells us that the objective value on the margin may be reduced by 0.933 per unit of increased capacity. This means that should the capacity have been 501 instead of 500, the optimal cost would have been $1461.3 - 0.933 = 1460.367$. We do not presently use the marginal costs for the convexity constraints (although they may be used in the future for selections among alternative solutions to include in the model).

1.1.2 Second linear programming model – LP1

Now the idea is the use the marginal costs for finding additional solutions (or columns, each variable is associated with a column in the two tables above) that will be added to the model so that better solution combinations can be identified. This is done by using the marginal costs for the capacities as cost input data to a **column generation modules** of the Standard LogMod programs that operates only on the f2f-relation transports using railway after Standard LogMod (the modules are denoted *BuildChain4RCM* and *ChainChoi4RCM*). The marginal costs are entered as additional costs on the capacitated links, in this case as a cost of 0.933 per train using the single capacity link. These marginal costs are transformed into additional marginal cost LOS-matrices for rail using path info for all rail transports (with the use of so called spanning tree data constructed during the LOS-calculations). It can be viewed upon as applying a select link analysis on the different links with non-zero marginal cost, and after that adding the marginal costs to the concerned train OD-relations derived from the set of select link cases.

With the new marginal cost LOS-matrices in place the **column generation module** will identify a new set of alternatives while considering the marginal costs on the constraining, binding, capacity constraints. When completed, these alternatives will be added to the linear programming model and it may look like in Table 1.8 (same format as Table 1.7)

Table 1.8 *Optimal solution to the problem LP1 (same as LP0 + 5 new alternatives/columns). Note! The marginal costs are not included in the objective functions.*

LP1															Optimal LP-solution					
Variables	First best solution					Second best solution					Third solution alt after Column generation					RHS	Sum of costs + RHS	Marginal cost		
	x10	x20	x30	x40	x50	x11	x21	x31	x41	x51	x12	x22	x32	x42	x52					
	0	1	0	0.67	1	0	0	0	0	0	1	0	1	0.33	0					
Costs	100	200	300	400	150	150	300	510	680	500	125	250	405	540	325		1326.7			
CAP001	100	180	250	300	75	0	125	0	0	0	10	131	25	30	7.5	≤	500	500.0	-0.519	
Cvx01	1					1					1					=	1	1.0	130.2	
Cvx02		1					1					1				=	1	1.0	293.3	
Cvx03			1					1					1			=	1	1.0	418.0	
Cvx04				1					1					1		=	1	1.0	555.6	
Cvx05					1					1					1	=	1	1.0	188.9	
Non-neg	x10, x20, ..., x51 ≥ 0																			

The optimal solution to this problem is presented in the framed parts in Table 1.8. Transports number 2 and 5 ($x_{20}=1$, $x_{50}=1$) are using the Standard LogMod solution, transports number 1 and 3 ($x_{12}=1$, $x_{32}=1$) are using the third best alternative, and finally transport number 4 use 66.67 % of the Standard LogMod solution and 33.33 % of the second best alternative ($x_{40}=0.6667$ and $x_{41}=0.3333$). In total it results in a cost increase to 1326.7 and a capacity usage equal to exactly 500. The marginal cost (or shadow price) for the capacity constraint has dropped to at -0.518 which tells us that the objective value on the margin may be reduced by 0.518 per unit of increased capacity. The reduction of the optimal cost from 1461.3 to 1326.7 is a result of finding better alternatives while recognizing price control invoked by recognizing the marginal cost LOS matrices in the **column generation module**. As you see the second best solutions have all been abandoned in this example.

1.1.3 Third ++ linear programming models – LP2, ... and convergence

The solution process will in principle continue as the LP1 case. The only difference between LP1 and subsequent LP:s (LP2, LP3 etc) are that they hold more columns for better descriptions of the RCM-problem, and possibly also additional convexity constraints.

Convergence: It stops when one of the following conditions is met:

1. The consecutive optimal objective values are sufficiently close to motivate a stop.
2. The last iteration does not generate any new alternatives.

From the latest LP-solution we do the following:

1. Construct intermediate logmod solutions using the LP model solution, the x-variable values. This enables the construction of revised empty train OD-matrices. These are needed for estimating the capacity for empty train flows.
2. Retrieve the marginal costs from the optimal LP-solution.
3. Revise the marginal cost LOS- matrices.
4. Run the column generation module of the logistics model, that initially will provide many new alternatives, but with higher iteration numbers the new alternatives will successively be fewer and fewer. Actually, already for LP1 there will be cases where the solutions for some firm-to-firm transports will be repeated. Either they may not be affected by the marginal costs, or there are simply not any better options available. Stop in case of Convergence (case 2).
5. The new alternatives are inserted into the LP model and solve it. Stop in case of Convergence (case 1). Otherwise a new iteration is performed with these five steps.

1.2 Terms and definitions

Linear programming = a linear model with an objective function that is to be minimized or maximized, while ascertaining that a number of conditions (all formulated as linear equations or inequalities) are satisfied. In RCM the objective function is to minimize the additional costs associated with the satisfaction of all capacity constraints and with carrying out all the demanded transports. These constraints are formulated as the maximum number of loaded and empty trains per year (trains per day in input data) on bidirectional links in Sweden, and by requiring that the sum of alternatives used per f2f-relation amounts to 100 %.

Convex function = a function is convex if any point on a straight line between any two points on the function's surface is on or above the function's surface.

Convexity constraint = the sum of a set of variables must be equal to one, and all the variables should assume values between 0 and 1.

Extract = a module in Standard LogMod that constructs OD-matrices for loaded vehicles and tonnes directly from the first solution alternatives from the ChainChoi-module output. From the loaded vehicle matrices also empty vehicle flows are constructed in the return directions based on a piece-wise, non-increasing, linear function with distance. In principle the empty vehicle add-on factor decreases with distance reflecting a higher propensity to find return flow. Also an imbalance in the loaded flow may generate empty flows, but this is dependent on input data, and it can be completely over-ruled. See also Appendix B.

First alternative solution = in general this refers to the second best ChainChoi-solution in Standard LogMod being the first one of the second to fifth solution alternatives that does not use more rail capacity than the initial solution anywhere and use less capacity somewhere

Initial solution = in general this refers to the first ChainChoi-solution in Standard LogMod

LP = acronym for Linear programming, see linear programming

LP-RHS = the Right Hand Side in an LP model, see RHS and LP respectively

f2f-relation = firm-to-firm relation, see PWC

Objective function = a linear function with coefficients describing the contribution to the objective (the objective being cost minimization in RCM) per unit of the variables used (the variables being what proportion between 0 and 1 to use of the alternative transport solutions per f2f-relation)

PWC = represents demand in terms of production, warehouse and consumption. Freight demand is modelled on the basis of performed commodity flow surveys and foreign trade statistics. P and W represent senders of goods and C represents receivers. The total demand in a relation from origin node O to destination node D is further split into so-called firm-to-firm relation (f2f-relations) representing demand between pairs

of small-medium-large size companies at both ends (in total 9 possible combinations) or as very large flows (term used?).

RCM = acronym for Railway Capacity Management procedure. RCM is a post processing procedure after Standard LogMod that identifies transport solution alternatives that in a cost minimizing manner satisfy all the railway capacity constraints in terms of bidirectional link flow upper bounds.

RHS = common notation for Right Hand Side or Right Hand Side values (both relating to the constant in the right hand side of an equation, when formulated with the variables on the left hand side)

Second best solution = same as first alternative solution

Slack variable = it can represent the difference between a used resource and the available resource, or the difference between achieved quality and a required quality. In RCM the slack variables per link represents the difference between available capacity and the actual link flow. The unit in the model is the number of loaded and empty trains per year on bidirectional links, which in input data and some outputs are shown as number of trains per day.

Spanning tree = all minimum cost paths from an origin node, the root, in a network (with its current fixed attributes) to all reachable destination nodes form a spanning tree. This enables the construction of compact, implicit definitions of all the paths in a data structure in which each node N in the network holds the link number L for the preceding link to all paths passing node N. A path from any reachable destination is constructed through a simple backtracking process starting from the destination node, successively following the preceding links to their start nodes until the preceding link number is identical to zero. Then the origin has been reached. See also Appendix C.

SRAIL = set of demand/f2f-relations using railway transports defined by a railway transport superindex, product and key to the f2f-solution

Standard LogMod = the standard logistics model developed for Samgods

2 The Logistics Model in Samgods and Associated Components

For a complete presentation of the model and its implementation the reader is referred to de Jong et al [2011] and de Bok et al [2011]. In a paper by Edwards [2011] a short summary of the model and some of its shortcomings are presented. Since then the model has been revised along some of the suggestions. The demand input to the logistics model is handled independently from the logistics model, see Edwards et al [2008] and Edwards [2008]. The presentation of the input data in general for the network and the logistics model is presented in Vierth et al [2009].

2.1 Producer-Consumer Flows in the Base Matrices

There are 34 base matrices representing 33 product groups and a 34th air transport product group constructed as a weighted combination of the first 33. The number of zones in Sweden is 290 and there are 174 foreign zones. As a consequence of the disaggregation approach the demand in each relation is split into demand between combinations of small, medium-sized and large firms, all in all up to a maximum of nine combinations. On top of this there is yet another possibility in terms of a so called singular flow. These are in principle very large existing demands in Sweden that will not be constructed by the general base matrix model. Each firm to firm combination is also associated with a number of existing, identical, transport relations. All of this has been constructed using socio-economic data, commodity flow survey, foreign trade statistics and business register data for the Swedish side. In the foreign zones the medium-size is the assumed one.

2.2 Network Model: LOS matrices

The network model represents the transport networks for road, rail, sea and air transports. All in all there are 5 lorry types, 8 train types, as many as 17 sea vessels, 4 ferry types and 1 air cargo type. The goods are shifted between different vehicles at approximately 655 terminals around the world, most of them in Sweden though. At the road and rail ferries the goods remain on the vehicles, or at least on the trailer or wagon part of the vehicle. The supply of transport services are computed as level of service-matrices, LOS matrices, in terms of time, distance and special costs (road tolls, km-taxes etc) for the paths with the lowest generalized cost determined as:

$$\sum_{\text{links in path}} [Time \times (vehicle\ cost / hour) + Distance \times (vehicle\ cost / km) + Special\ costs]_{\text{link}} \quad (2.1)$$

It is worth to mention that congestion or capacity constraints are disregarded in the LOS-calculations. These aspects are also disregarded in the assignment step in Section 2.4. A fourth type of LOS matrix is determined, namely the transport distance inside the Swedish

border, including sea transport close to the coastal line. These are used to compute national transport measures, such as tonne and vehicle kms.

The spanning tree information required for RCM is saved to text files as a by-product of the LOS-calculations when all the information is available. See appendix C for further details on the spanning tree information.

2.3 Logistics Model

The logistics model is designed to reflect firm to firm (f2f) decisions. For further details on the logistics model we refer to Appendix E and de Jong et al [2011].

The cost function to be minimized for each f2f-flow consists of the following components:

1. Order setup costs and inventory holding costs (both carrying costs and interest on product capital).
2. Transport costs for the OD-legs in the transport chain from origin to destination, through one or more transfer points where transport mode/vehicle change-overs are made unless it is a direct transport. The transport costs are computed for entire vehicles for unconsolidated transports, and for fractions of vehicle loads for consolidated transports. Consolidation is applicable for all vehicles for which the current shipment size is below a consolidation level, endogenously determined (inside an exogenously determined interval [LBD,UBD]) per OD-relation and vehicle sub-mode². Otherwise an integer number of vehicles are used only for the shipment size in question. System trains are never consolidated.
3. Handling costs per tonne and vehicle cost per hour for loading/unloading at transfer points. For container transport there are also stuffing and stripping costs at the origin and destination respectively.
4. Capital costs based on product values during transportation and waiting times.

The decisions to make for each transport are:

- a. Which transport chain to use?
- b. Which vehicle size to use in each OD-leg?
- c. What shipment size should be used (equivalent to annual demand/frequency)?

The current method for solving the above issues is to perform the following two steps in an iterative process (3 iterations) for each of the 34 product groups. The same cost function is used in both steps, but with a set of predefined input parameters (vehicles and shipment size).

² A sub-mode is a number of vehicles grouped together into a “letter group”. For non-container transports, C for example often includes the two largest lorry types 104 and 105, and H typically includes the wagon load trains 207 – 209. The groups A and F respectively, represent the same vehicles when used for container transports.

BuildChain step (BC): Determine a number of optimal transport chains, consisting of combinations of road, rail, ferry, sea and air transports, for all origins and destinations. The combinations considered are predefined. For each product group there is one typical shipment size used per origin zone (CPU time would increase a lot for individual shipment sizes). The vehicles considered are a designated vehicle for each sub-mode defined, where the sub-modes may for example, be either a small or a large lorry. A shortest path procedure is used to determine the transfer points for each transport chain.

ChainChoi step (CC): Here the set of optimal transport chains are evaluated at a detailed level. The optimal combination of individual vehicles and shipment frequencies are determined given the set of available transport chains. A crucial input here, since the costs incurred are vehicle based, is the consolidation approach that results in the vehicle utilization levels. See de Jong et al [2011] for further details. In principle all rail, sea and air transports are considered to be consolidated. The only exception now is system trains. After three iterations with the BuildChain and ChainChoi modules as described above the results are summarized in the so called Extract module into vehicle- and tonne-OD-flows respectively per vehicle/vessel type. In the latter module also empty flows are constructed. See Appendix B for a more detailed description of the Extract module.

2.4 Network Model: Assignment of OD matrices

With the disaggregate solution from Section 2.3 aggregated into OD-flows (loaded vehicles, tonnes and empty vehicles) we are now back at the aggregate level. Now the OD-matrices are assigned to the network using the same routes as used for their construction, c f eq (2.1). All in all there are up to 105 OD-matrices to assign to the network, i e up to 35 OD-matrices for total vehicle flows (or loaded trains in case the empties are first subtracted from the totals), 35 OD-matrices for tonne flows and 35 OD-matrices for empty flows. Should this assignment procedure be flow dependent, i e involving non-constant volume-delay functions, we would need to handle multi-class assignments with up to 17 classes (if all sea vessels should be assigned together). However, as stated earlier, they can be assigned individually since volume based delays on links are not included.

3 Linear Programming Approach Overview

A linear programming (LP) approach has been developed to allocate the railway freight flows with respect to available capacity. We use the notation RCM, Railway Capacity Management, for the entire solution procedure involving all the steps of incorporating the LP model into the Samgods model system. The capacity is measured in terms of number of freight trains per day (loaded and empty) for all bidirectional link flows in Sweden. The objective is to change some of the existing transport chain solutions in a cost minimizing manner. Only f2f-flows (firm-to-firm) from the PWC-matrices using rail transports need to be considered. From the Standard LogMod, the basic logistics model, solution up to five of the best (= lowest cost) transport chains are available. From this we have two solutions per initial transport chain (including rail transports), which are the initial solution and the first alternative solution (see also Chapter 1.2). For each of the solutions their rail capacity consumption is derived per link (a number of loaded trains, an empty flow add-on factor). This derivation requires full information on the paths used to be readily available. The empty flow add-on factors are estimated from the OD-matrices with loaded and empty trains, where the latter are determined by an empty train model inside the logistics model, see Appendix B and de Jong et al [2011]. The treatment of the add-on factors for empties is very important for the result since they constitute approximately 40 % of the capacity usage³. In the current model the add-on factors are determined as the ratio of *the number of empty trains per train type and relation in both directions* to *the number of loaded trains per train type and relation in both directions*. Should there not be any empty train in the relation, e.g. in the case of a new column, then a default add-on factor from ASEK⁴ is used.

The concept of an LP model can be viewed upon as a formulation in which these two options, alternative 0 and alternative 1 per f2f-relation, are entered side by side in the model (a huge table) together with their respective costs. The variables in the model are how much alternatives 0 and 1 respectively are used per f2f-relation. Their total use must amount to 100 % (described by a condition on their sum). For each one of the capacity constraints, the capacity requirement is obtained as the sum of the products of the variable values and the capacity consumption. These sums should not exceed the available capacity.

Two extreme solutions are either:

1. to use the initial solution everywhere which gives the capacity usage from Standard LogMod, or
2. to use alternative 1 everywhere which would provide an entirely different capacity usage.

Most likely the second alternative would mean that all capacity constraints would be satisfied, although extreme cases can occur if there for any capacitated link exist alternatives that cannot be combined into a feasible capacity allocation (e.g. on the iron ore railway).

Obviously we do not have all the means to establish the optimal solution using the LP model with only 2 alternatives per f2f relation. A standard approach to deal with this is to use a so

³ In december 2014 the Extract module is programmed to allow for user input of empty train allocation data to allow for better adherence to official ASEK values on empty train shares

⁴ ASEK = Arbetsgruppen för SamhällsEkonomiska Kalkylvärden

called column generation method, in which some method is used to generate new viable alternatives to create an LP model that better describes the possibilities available (in principle an LP model holding all the potential transport solutions possible would describe the problem at hand in detail, but it would be practically impossible to solve). The purpose of the column generation method is to generate a few options per f2f-relation that efficiently and accurately enough formulates the RCM. In this process the current LP-solution output, in the form of its marginal cost values for insufficient capacity is used for guiding the method used. Here a version of Standard LogMod is used to find efficient new alternatives, i.e. new columns, based on the marginal costs. This iterative column generation leading to successively larger and larger LP-problems will eventually identify the optimal solution if the underlying problem has a “nice” underlying objective function, looking for example like a bowl. In technical terms the function should be convex.

The introduced marginal costs will NOT be included in the cost calculations in places other than the column generation step, their purpose is to guide in the search for viable firm-to-firm transport solutions. The effect on the total cost of the logistics model will appear as the sum of the extra costs associated with not using the first best solutions.

When the solution is completed the model tells us which alternatives to use in order to satisfy the capacity constraints in a cost minimizing manner. An overview of the full system comprising Standard LogMod and RCM is shown in Figure 3.1.

3.1 Update with Samgods version 1.1

With this version a different LP-software has been put into operation. It reduces the LP-solution times considerably, in particular for the first solution, LP0. This alone was not sufficient, and therefore the many small railway flows are handled in a simplified manner by only providing one alternative solution for them (the initial one). Small flows are defined as those below a user-defined value.

A second new feature is the possibility to lock system train flows to particular routes by defining a set of compulsory rail terminal passages. This has been put in place to facilitate dedicated routes for known, large scale system train flows, such as the steel shuttle Luleå-Borlänge. These flows must not be reallocated by the RCM procedure. Their capacity consumption is accounted for in the LP model through reduced available capacity for other trains.

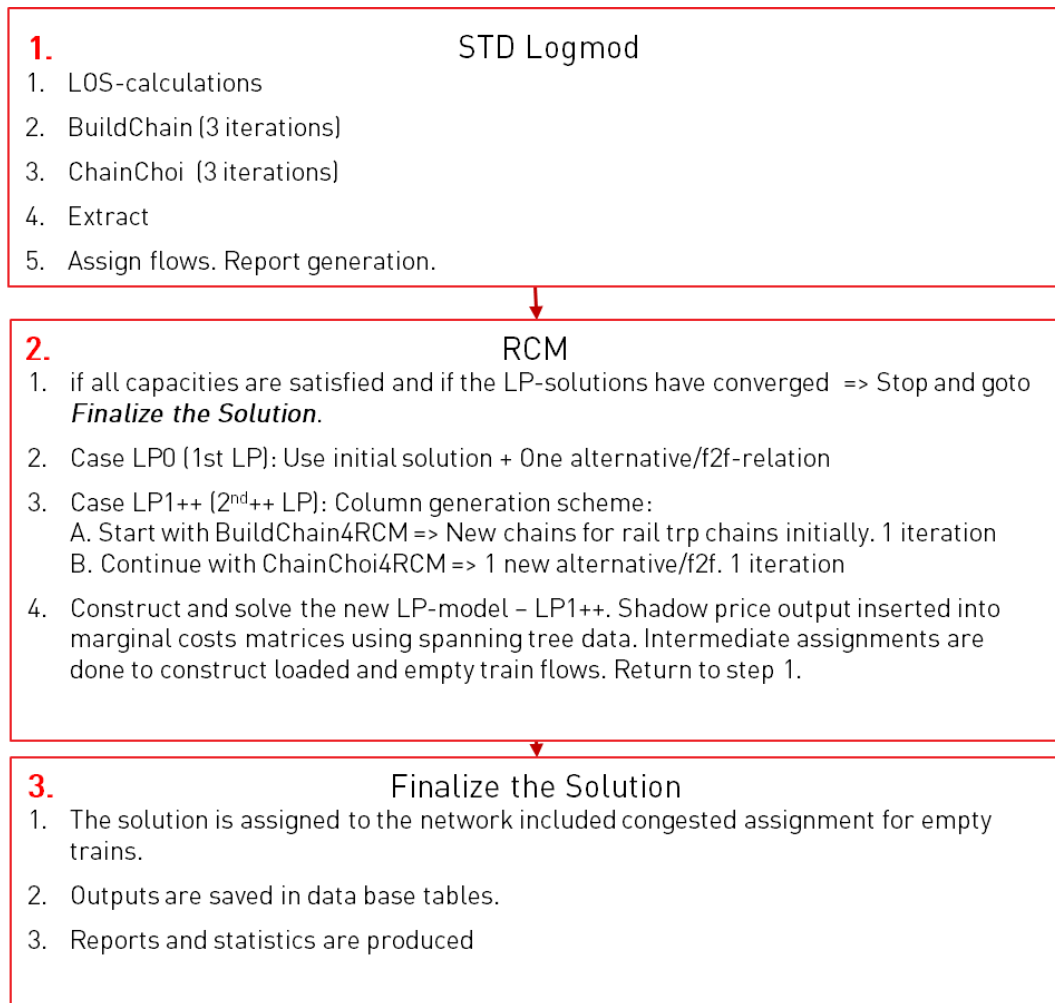


Figure 3.1 Flow chart of the logistics model and RCM.

4 Linear Programming Approach Technical

A linear programming (LP) approach has been developed to allocate the railway freight flows with respect to available capacity. We use the notation RCM for the entire solution procedure involving all the steps of incorporating the LP model into the Samgods model system. The capacity is expressed in terms of number of freight trains per day (loaded and empty) for all bidirectional link flows in Sweden. The objective is to change some of the existing transport chain solutions in a cost minimizing manner. From the basic logistics model solution up to five of the best (= lowest cost) transport chains are available (Note! At most one solution per chain type can be used). From this we have two solutions per initial transport chain (including rail transports), alternative 0 == the initial solution, and alternative 1, the first alternative solution. For each of the solutions their rail capacity consumption is derived (a number of loaded trains, an empty flow add-on factor plus a vector with the concerned capacitated link numbers). The estimated number of trains are the number of loaded trains multiplied by (1 + empty flow add-on factor), the add on-factor should be in the interval 0 to 1. This derivation requires full information on the paths used to be readily available. The empty flow add-on factors are estimated from the OD-matrices with loaded and empty trains, where the latter are determined by an empty train model inside the logistics model. The treatment of the add-on factors for empties is very important for the result since they constitute approximately 40 % of the capacity usage. Further details on this matter are discussed in Chapter 3.3. In the current model the add-on factors are determined as the ratio of *the number of empty trains per train type and relation in both directions* to *the number of loaded trains per train type and relation in both directions*. Should there not be any empty train in the relation, e.g. in the case of a new column, then a default add-on factor from ASEK⁵ is used.

The set of demand/f2f-relations using railway transports are defined in a set SRAIL⁶. It holds a coupling between the variables in the LP model and the combinations of *Key* and *product group number* in the Standard LogMod results.

The concept of an LP model can be viewed upon as a formulation in which these two options, alternative 0 and alternative 1 per f2f-relation, are entered side by side in the model (a huge table) together with their respective costs. The variables in the model are how much alternatives 0 and 1 respectively are used per f2f-relation. Their total use must amount to 100 % (entered into so called convexity constraints). For each one of the capacity constraints, the capacity requirement is obtained as the sum of the products of the variable values and the capacity consumption. These sums should not exceed the available capacity.

Two extreme solutions are either:

1. to use the initial solution everywhere which gives the capacity usage from Standard LogMod, or
2. to use alternative 1 everywhere which would provide an entirely different capacity usage.

⁵ ASEK = Arbetsgruppen för SamhällsEkonomiska Kalkylvärden

⁶ In Samgods version 1.0 the text file holding this information is named JLISTA.DAT, cf Bergström [2015]

Most likely the second alternative would mean that all capacity constraints would be satisfied, although extreme cases can occur if there for any capacitated link exist alternatives that cannot be combined into a feasible capacity allocation (e.g. on the iron ore railway).

Obviously we do not have all the means to establish the optimal solution using the LP model with only 2 alternatives per f2f-relation. A standard approach to deal with this is to use a so called column generation method, in which some method is used to generate new viable alternatives to create an LP model that better describes the possibilities available. In this process the current solution output, in the shape of its dual variable values (c.f. below) is used for guiding the method used, here a version of Standard LogMod, towards efficient new alternatives, i.e. new columns. This iterative column generation leading to successively larger and larger LP-problems eventually will identify the optimal solution if the underlying problem is convex (i.e. has a “nice” underlying objective function, looking for example like a bowl). The introduced marginal costs will NOT be included in the cost calculations in places other than the column generation step, their purpose is to guide in the search for viable firm-to-firm transport solutions. The effect on the total cost of the logistics model will appear as the sum of the extra costs associated with not using the first best solutions.

Important results from the LP model are the so called shadow prices (dual variables values). Their interpretation is the marginal value of having an extra unit of train flow capacity available. For all links with sufficient capacity, the shadow prices are zero. Thus, in order to improve the LP model with new alternatives that may help to reduce the total cost, the shadow prices are introduced as marginal costs for all concerned rail-OD-relations. Any solution not being able to carry the marginal costs will NOT be a part of the optimal LP solution, and therefore this price control, or “congestion tax”, is an efficient method for sorting out non-viable solutions and keeping the good options.

With these in place a special version of Standard LogMod are run again, using only the third iteration in Standard LogMod (where consolidation factors and volumes respectively are unaltered), for the concerned demand cases in SRAIL, and promising solution alternatives can be added to the linear programming model. The special version will handle only the rail solutions. After each new run of the logistics model in this so called column generation scheme, the linear programming model is solved again. When the LP-solution has converged (improvement $< \epsilon$ or no new columns are generated), the column generation scheme is interrupted, and the identified solution is inserted as the final solution in the logistics model.

For the sake of efficiency the LP model will be formulated in such a manner that the initial solution is considered as the starting point, and all coefficients on the left hand side will constitute differences compared to the initial solution. It will cut down the number of variables in the initial LP model by a factor 2, and an upper bound only per f2f relation will be required instead of convexity constraints. When more columns are generated, then a convexity constraint must be inserted for all concerned cases, but with an inequality instead of an equality (since the initial solution, the default column, represents the difference = 1 – the sum of the column variables per f2f relation).

This means that the capacity constraints minus the allocated train flow will constitute the constraint value on the right hand side. Then the differences between any alternative solution

and the initial column (number 0) will be represented among the alternatives on the left hand side. The cost will be the cost difference between the costs in column N and the costs in column 0, and the capacity will be the added capacity from column N minus the capacities cut out when not using column 0. Any use of an alternative means that its net additional cost will be added to the total costs, and the same thing holds for the constraints – the capacity usage of alternative N will be added and that from the alternative 0 will be balanced out (neutralized /nulled). Details and further examples on the LP model formulation are presented in Appendix A.

The method is laid out in the item list below.

1. Run the logistics model in STD mode, including module Extract to get total rail OD-flows and empty rail OD-flows.
2. We construct a list of transport chains using rail in Sweden (SRAIL). The common identifier in the list is denoted superindex and it is associated with a product group number and a key.
3. The capacity model is applied to the transport chains in SRAIL. The first time, iteration 0, the first best solution and the second best solution are extracted as two alternatives per list item. They constitute two columns in the model, but as explained above the model are reformulated to hold differences between columns 1,2,...,N and column 0. In consecutive iterations, at most one new alternative is added per list item as part of the column generation scheme.
4. The LP model (linear programming model) finds the minimum cost combinations of variables (==columns) that satisfies the link flow constraints. From the LP-solution we use the marginal values of additional capacity, the dual values (shadow prices), to calculate a new set of LOS-matrices that we denote MC-matrices (MC = marginal cost).
Help variables are used to enable purchase/investment in new flow capacity at very high costs to ensure the existence of a feasible solution.
5. From the solution we compute intermediate transport chains solutions that primarily are used to update the empty train OD-matrices (using the Extract module in Standard LogMod)
6. Should the capacities be satisfied and the improvement in the LP model objective be small enough we can stop and feed back the LP-solution into the logistics model, and then STOP and goto step 7. The solution is fed back in terms of suggested transport chains to use, not in terms of marginal costs to use for reoptimizing these transport chains. Should the latter approach be used, it is not possible to say

whether or not the capacities will be satisfied.

Otherwise we do the following:

- a. run the logistics model in COLUMN_GENERATION mode to generate possible new alternative solution with the latest set of MC-matrices and
 - b. go back to step 3.
7. After feeding the LP-solution into the logistics model, revised report files and OD-matrices can be constructed by running the logistics module in mode OUTPUT_REVISION. Most of the LP solution variables are either 0 or 1 (0 meaning that the initial solution is kept, 1 meaning that the alternative is used). However, in general there are a few variables with a solution between 0 and 1 which means that the total volume will be split in the final solution.

The logistics model in mode COLUMN_GENERATION are special versions only handling the transport chains in SRAIL. The ChainChoi-version is ran with the input argument from the third iteration in mode STD, /RDVOLUMES=YES, during which utilization factors etc are NOT updated. Variations in these would compromise the solution process considerably. Separate RCM-output files are used to store column generation output from **BuildChain4RCM** and **ChainChoi4RCM**.

Unique column information input from the RCM-modules are stored for LP model setup, the first best solution is denoted column 0, and the alternatives are numbered 1, 2, Except information in the original output format, each data row holds information about superindex, column number and marginal costs. The LP-solution needed to update the solution is saved in a designated file. It holds only those variables where an alternative transport chain is used.

4.1 Linear Programming Formulation

The mathematical formulation of the problem is shown below:

$$\begin{aligned}
 \min \quad & z = c^T \cdot x + d \cdot y \\
 \text{subject to} \quad & A \cdot x - y \leq b \\
 & C \cdot x \leq l \\
 & x \leq u \\
 & x, y \geq 0
 \end{aligned} \tag{4.1}$$

The variables in the model are:

x = vector with usage of the second best (second best without rail if available) solution. Unit: A share between 0 and 1.

y = vector with volume of purchased link capacity to enable a feasible solution as result.
Unit: Number of trains per year.

The input data in the model are:

c = vector with additional costs for using the second best solution. Unit: SEK for a transport chain shift.

d = vector with costs for acquiring additional rail capacity. Unit: SEK for one additional train per year

A = matrix with changes in number of trains required should second best solutions be used. Unit: trains (loaded and empty) for a full transport chain shift.

C = matrix for convexity constraints with a row for each f2f-relation with at least two columns and with ones in the concerned columns.

1 = a vector of ones with one element for each f2f-relation with at least two columns

b = right hand side vector with available freight train capacity per year per bidirectional capacitated link.

u = upper bound on the use of the variables x . u is a vector with ones.

We illustrate the LP model construction by a small example in Tables 4.1 and 4.2 with input for five f2f-solutions and with five capacity constraints CAP001 – CAP005 representing five domestic bidirectional railway link flows with limited capacity that needs to be managed. Domestic railway links with plenty of remaining capacity available after Standard LogMod actually can be left outside the LP model since they, according to experience, are unlikely to attract any new railway transports.

Table 4.1 *Transport chains using trains, initial solution and possibly one alternative solution.*

	Initial solution w trp chain.					Alternative solution w trp chain.				
						alt train	alt train	truck+sea	alt train	same as x50
<i>x-values</i>										
<i>Variables</i>	x_{10}	x_{20}	x_{30}	x_{40}	x_{50}	x_{11}	x_{21}	x_{31}	x_{41}	
cost/objective	100	200	300	400	150	150	300	510	680	
CAP001	100			80	10				10	
CAP002	100	70			10		20			
CAP003		70	60			10	20			
CAP004			100			10	20			
CAP005			100	80	10				10	

In Table 4.2 we have initial x-values $x_{10} - x_{50} = 1$ that give cost and capacity usage according to column **Initial solution**. The default solution thus results in a capacity deficit according to column RHS (Right Hand Side). As presented we formulate the LP-problem by setting up the alternatives as the variables, where the effect of each alternative is the difference between the columns x_{11} and x_{10} , x_{21} and x_{20} , etc in Table 4.2. Setting for example $x_{11} = 1$ will increase the cost by 50 ($= 150 - 100$) and capacity CAP001 will impact the left hand side with -100 ($= 0 - 100$), i.e. we would move from an infeasible left hand side at 0 ($> \text{RHS} = -40$) to a feasible left hand side at -100 ($< \text{RHS} = -40$).

Table 4.2 Effect of initial transport chain solution. The RHS-value simply represents the remaining, available capacity. Any negative values indicate the overloads on the links.

	Initial solution w trp chain.					Alternative solution w trp chain.					Capa- city	Initial solution	RHS = Cap - Init soln
	x_{10}	x_{20}	x_{30}	x_{40}	x_{50}	alt train x_{11}	alt train x_{21}	truck+sea x_{31}	alt train x_{41}	same as x_{50}			
x-values	1	1	1	1	1	0	0	0	0			1150	
Variables	x_{10}	x_{20}	x_{30}	x_{40}	x_{50}	x_{11}	x_{21}	x_{31}	x_{41}				
cost/objective	100	200	300	400	150	50	100	210	280				
CAP001	100			80	10	-100			-70		150	190	-40
CAP002	100	70			10	-100	-50				150	180	-30
CAP003		70	60			10	-50	-60			150	130	20
CAP004			100			10	20	-100			150	100	50
CAP005			100	80	10			-100	-70		150	190	-40

The solution to the problem of finding the least cost alternative solution with the available, optional alternatives is shown in Table 4.3 in the rightmost two columns. We have removed the initial solution corresponding to columns $x_{10} - x_{50}$ from the model. Their capacity consumption ($=$ initial solution) is subtracted from the capacity which results in the RHS-value. This means that the RHS-value is transformed into **Residual capacity = Input capacity – Initial flow**. A non-negative residual capacity of course means that the initial solution satisfies the capacity constraint, a negative residual capacity means that the link is subject to overload.

The capacity usage at optimum is presented in the column denoted *LP-RHS-USAGE* in Table 4.3 and the minimum cost is shown below *LP-cost for alt's* (the reader can verify that a multiplication of the solution variable values $x_{11} - x_{41}$ with the coefficient in the left hand side table result in the presented values, e.g. as shown for the minimum cost value in equation (4.1)). We also have that

Column *LP-RHS-USAGE* values \leq RHS-values for all capacity constraint rows.

Otherwise the solution would not be feasible, that is at least one capacity would be overloaded.

We see that the marginal costs (MC) appear where the LP model capacity usage equals the RHS-values. Here the solution variables are fractional which means that a split of the solution would be required when used in the logistics model. Optionally a rounding off/truncation⁷ could have been used.

Table 4.3 Optimal LP-solution with minimum cost for using alternative transport chains.

	Alternative solution w trp chain.					Capacity	LP-cost for alt's =		104	MC
	alt train	alt train	truck+sea	alt train	same as x50		Initial solution	RHS = Cap - Init soln		
<i>x-values</i>	0.4	0	0.4	0			1150			
<i>Variables</i>	<i>x11</i>	<i>x21</i>	<i>x31</i>	<i>x41</i>						
cost/objective	50	100	210	280						
CAP001	-100	0	0	-70		150	190	-40	-40	-0.5
CAP002	-100	-50	0	0		150	180	-30	-40	0
CAP003	10	-50	-60	0		150	130	20	-20	0
CAP004	10	20	-100	0		150	100	50	-36	0
CAP005	0	0	-100	-70		150	190	-40	-40	-2.1

The LP model cost for the alternatives are:

$$0.4 \times 50 + 0.4 \times 210 = 104 \quad (4.2)$$

The shadow prices (dual values) are -0.5 (COST/USAGE=50/-100) and -2.1(210/-100) for CAP001 and CAP005 respectively. The interpretation is that on the margin, each unit of additional capacity for constraint number 1 leads to a lower LP model cost by 0.5 units. A general property of the solution is also that multiplication of the shadow prices with the RHS in the model results in the optimal minimum cost value:

$$-0.5 \times -40 - 2.1 \times 40 = 104 \quad (4.3)$$

4.2 Handling of empty train flows in RCM by LP

The initially used fixed add-on factors for empties worked fine inside the LP model, but when inserting the LP-solution into a final solution and running of the Extract module it

⁷ It is expected that a truncation to 1 in most cases would solve the issue. Should two or more alternatives display fractional values, a simple measure would be to select any of them. A possible method would also be to set up a small integer programming problem with all fractional variables and their companion alternatives, and solve the reduced programming problem to optimality, but it is probably not worthwhile.

displayed a solution that not quite completely satisfied the capacity constraints. This is hardly surprising for the following reasons:

1. The add-on factors are of course approximations.
2. There are not any margins left in the formulated LP model. 100 % of the capacities on the capacitating links are used. With a high probability this will result in some overloads

With the detailed description of the extract procedure, it can be concluded that the approximations were caused by the asymmetric flows, see Appendix B. Since it now (December 2014) has been decided to use add-on factors without consideration of asymmetric empty flows for rail, the description of the empty flows in the LP model is perfect as long as the congested assignment model is not used. When using the empty congestion results, then the path distribution of the empty flows would be needed to maintain a perfect description in the LP model.

The updating of the add-on factors are carried out in the following manner:

1. The latest LP-solution was inserted into a temporary final solution detailed enough to allow for the Extract module to be run, resulting in updated OD-matrices for loaded and empty trains.
2. When preparing the columns for the forthcoming LP-solution, add-on factors for all columns were updated.

due to the changes made to Extract and its input parameters the 2 steps above are actually only needed for new columns inserted into the LP model, but it could also be skipped since the constant values from the files emptyfrac.dat could be used directly (see Appendix B).

Here we present in some more detail options considered/used in the current implementation. Option 3 is presently not used directly for the LP model but as a post-processing, a congested assignment model for empties, to squeeze in the empty flows in the gap between the capacities and the loaded flows.

Option 1. This option encompass the use of constant empty add-on factors, i.e. a constant ASEK-based share of the loaded trains per OD-relation are added as empty trains. This allows for an exact representation of the empty flows in the LP model for the capacity utilization. Since December 2014 this is the option used in practice. It should also be noted that the exact representation is valid not only with a constant add-on value for all distances, but also for the case of add-on values following any distance function as long as the symmetric flow modeling only is used.

Option 2. The main approach in the LP model is to assess a relation between empty trains and loaded trains per OD-relation and train type. In principle it may look like Figure 4.4. Supposing we have 200 loaded trains from T01 to T02 and 120 trains in the opposite direction, we may compute an empty train factor at 0.6 ($=120/200$) for this relation. Since the capacity constraints are expressed in number of trains for bidirectional flow we add the

loaded and empty flows in both directions to come up with the empty train factor. Should the ratio be over 1.0 the ratio is set to 1.0.

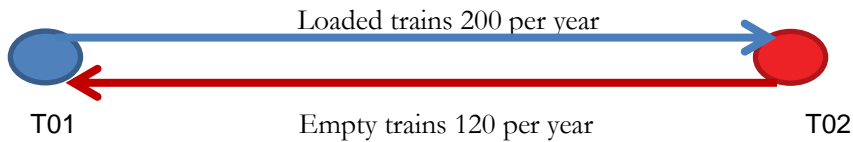


Figure 4.4 Loaded and empty trains in an OD-relation.

This method is the one used in the implementation, but as stated it gives the same result as option 1 with the current settings, but also for any distance function as long as the asymmetric flow option is not included.

Option 3. A multi-class network assignment model with congestion components is introduced to enable other paths than the LOS-paths to be used. A required feature is the need to incorporate bidirectional flows into the flow dependent cost function. The solution would in general turn out to use multiple paths as illustrated in as illustrated in Figure 4.5 where the empties are split into three paths. The paths cannot be saved without complicating additional programming, but it is sufficient to obtain the link flows in a post processing application.

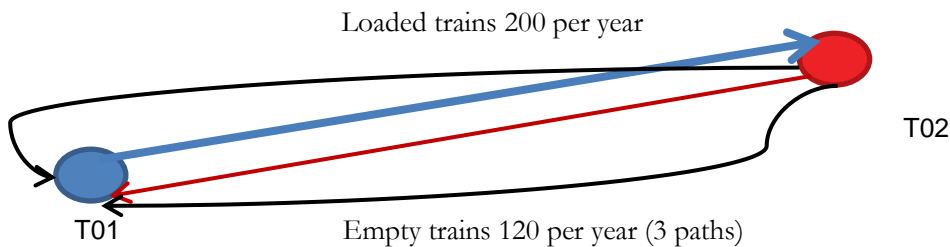


Figure 4.5 Loaded and empty trains in an OD relation. Multiple paths for empties according to a multi-class assignment with congestion modeling.

Allocation of empty trains according to a congestion assignment model allowing for multiple routes can be accomplished by setting up a model that:

1. assigns the empty trains based on their generalized time, their regular costs, on the links
2. incorporates the capacity constraints with a penalty function that assumes increasingly higher values when the capacities available for empties are overloaded.

The resulting assignment model is formulated as:

$$\min \sum_{traintypes} \sum_{links} (regular\ costs) \square (empty\ train\ flows / empty_length_factor) + \sum_{capacities} Penalty(overload) \tag{4.4}$$

with respect to

1. fix assignment of loaded trains in the network
2. assign of the calculated empty train OD matrices from the logistics model
3. network flow constraints

where

Penalty(.) = an optional function for penalizing loading above capacity.

Empty_length_factor = a factor (> 1.0) to simulate that the empty trains may be longer and thus requiring less capacity than loaded trains having **a corresponding factor equal to 1.0**

The penalty-function used is the one in equation (4.5)⁸.

$$Penalty(overload) = 10000 \times \left(\max \left[0, \left(\sum_{traintypes} \sum_{directions} (empty\ and\ loaded\ train\ flows) \right) - Capacity \right] \right)^2 \tag{4.5}$$

The main benefit of this approach is that it may open up for more railway transports in the system and lower marginal costs. In the current implementation this is used in finalizing capacity adjustment steps in RCM, the so call LPnAdjCap-procedure.

4.3 Railway Capacity Management Integration with the Standard LogMod

In this chapter it is described in words and with a flow chart how Railway Capacity Management is integrated with Standard LogMod. In Figure 4.6 a flow chart of the

⁸ But it ought to be normalized in this manner to focus somewhat more on relative errors:

$$Penalty(overload) = 10000 \times \left(\max \left[0, \left(\sum_{traintypes} \sum_{directions} (empty\ and\ loaded\ train\ flows) \right) - Capacity \right] \right)^2 / \sqrt{Capacity}$$

integration of RCM with the logistics model is presented. Comments to the various boxes are presented below.

Box 1. In combination with the LOS computations for a scenario also the spanning trees are saved, cf Section 2.2, for handling the mapping between OD-flows and link loads.

Box 2. The analysis starts with running Standard LogMod (the basic logistics model).

Box 3. The first LP model in iteration 0 is constructed using the 1st and 2nd best solutions (or higher order best solutions to find a transport chain not using rail). It is formulated with implicit entering of the 0-columns, and therefore no convexity constraints are required. However, upper bounds at 1 must be inserted to limit the variable usage to 100%. The input ascii-file is a free-format mps-file (a standard format).

Box 4. The LP model is solved with the LPsolve program, see Berkelaar, Eikland and Notebaert [2004]. When solving the model, the *presolve*⁹ option is used which considerably reduces the problem at hand before the basic optimization algorithms are invoked. The output level is piped to an output ascii-file, and is selected so as to obtain the needed solution variables and the dual variable values for the capacity constraints.

Box 5. The dual values are collected from the output file and the LOS files with marginal costs are recomputed with the latest solution. As described in Section 4 we insert the current LP solution into the logistics model and run the Extract module to obtain updated empty train add-on factors. The summed flows are checked against the capacities.

Box 6. The quality of the solution is checked here. Should the difference between consecutive iterations be small enough (for example with a total cost difference less than 1 % or so), the solution is considered satisfactory.

Box 7. Special versions of the logistics modules BuildChain and ChainChoi that only operates on the transport relations defined in JLISTA.dat (the ones initially using railway) are used for computing a new first best solution given the updated marginal costs. The modules are run with the same conditions as in iteration 3 of Standard LogMod in which a large number of endogenously generated input data, such as consolidation factors and volumes, are left unchanged. Should these be altered in this process, all input data to the LP models would need a revision in all iterations, and we would most likely run into convergence problems and perhaps worse.

Box 8. In this last step, the new columns generated in Box 7, if any, are added to the LP model. Each candidate must be checked against existing columns to avoid adding an already existing column to the model (it would not add any new information to the model). When the number of alternatives is 2 or more a convexity constraint must be introduced to limit their combined usage to at most 100 %.

⁹ Presolve is an option in the LP-solver that use efficient rules to reduce the problem at hand by removing uninteresting options.

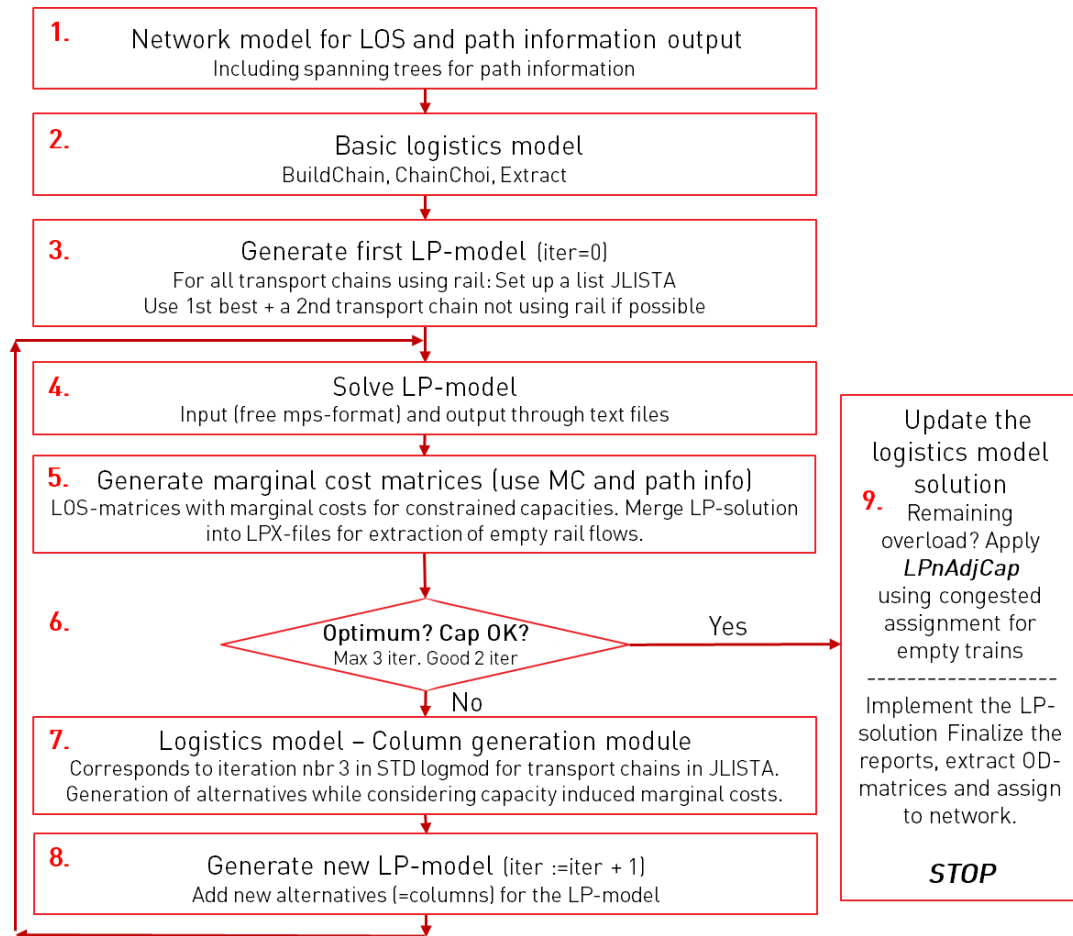


Figure 4.6 Flow chart of RCM integrated with the logistics model.

Box 9. With an existing, accepted solution it is time to combine it with the solution from the basic logistics model. Most of the variables turns out to be either 0 or 1 but fractional values exist. In the latter case it has been decided to split the solutions into the existing fractional values.

Should any of the links be overloaded over an acceptance level of more than 1 train/dy or 1% of the capacity level, then the current LP model is run again with the only change in input data that the capacity levels are reduced with $r\%$ of the overload (currently used is $r = 50$)¹⁰.

This is carried out in the function labeled **LpAdjCap** (to be interpreted as: *LP model after iteration n with Adjusted Capacities*). The process may in theory be run several times, but due to the current (fall 2014) long CPU times the procedure is unlikely to be activated. Actually also links with reduced capacities may have them reset to their original values to some degree, should the process lead to a large enough remaining capacity.

¹⁰ Further details on this matter are presented in Appendix D.

4.3.1 Developed Software

The standalone software in Delphi and Java for generation of logmod columns (BuildChain4RCM and ChainChoi4RCM), for MPS-file construction (MPS.jar) and the logmod cost evaluation and LP solution feedback to the ChainChoi-solution (LP2CC) has been developed by Jon Bergström, BeBits Systemutveckling. These programs are invoked by running different command files (bat-files) which are started from the Samgods GUI in Cube.

The spanning tree data information derived from the LOS calculations are implemented by Gabriella Sala at MIC in a special version of the Voyager model for Samgods/Trafikverket. Also the congested assignment model is implemented by Gabriella Sala inside the Cube/Voyager system, requiring a non-standard handling of the sum of bi-directional flows instead of handling the flows individually.

4.4 Update with Samgods version 1.1

With this version a different LP software, SOplex, version 2.01, has been put into operation. See section 8.1 under SOplex for further information. The SOplex solver offers much shorter solution times, in particular for the initial problem solved. This change alone was not sufficient to obtain acceptable CPU times (maximum 15 hours or so). Therefore also an option to handle the many small flows in a simplified manner was introduced, namely to include only one alternative solution selected among the STD-solution. This is managed by defining small flow limit (in the range 0.5 – 5 tonnes), and only for first choice solution above this limit multiple columns may be generated.

A second new feature is the possibility to lock system train flows to particular routes by defining a set of compulsory rail terminal passages. This has been put in place to facilitate dedicated routes for known, large scale system train flows, such as the steel shuttle Luleå-Borlänge. These flows must not be reallocated by the RCM procedure. Their capacity consumption is accounted for in the LP model through reduced available capacity for other trains. Should the locked route coincide with the first best solution the right hand sided will be the same. The only difference would be that there is not any variable on the left hand side that allows for an alternative solution.

Table 4.7 Example of control file for locked chains.

Commodity	FromNode	ToNode	SubCell	Freq	Mode	NNodes	Node1	Node2	Node3	Node4
15	952300	958400	0	-1	U	2	952311	958012		
15	958400	958400	0	-1	U	2	958411	958412		
17	958000	908100	0	-1	I	3	958012	926013	908111	

The input data for this is primarily a text file as exemplified with Table 4.7. The first four columns in the input define the locked demand relations in the PWC matrices¹¹ (used instead of keys). *Freq* equals -1 means that the frequency is optimized, positive values determines the shipment frequency. Modes for system train transport are either *I*, *U* or *T*, and in future version there will also be *i* (lower case *i*, separate from *I*) representing longer system trains (maximum 750 meters). Then follows the number of rail terminal passages (not incurring any costs), and the terminal numbers that are expected to well enough determine the desired, locked route.

The three examples in Table 4.7 are:

1. A flow between Gällivare and Kiruna.
2. An in-zone flow in Kiruna between two terminals (would otherwise by default be included among lorry transports, although not assigned to the network).
3. The steel shuttle from Luleå to Borlänge which is forced to pass 926013 (Östavall)

5 Some Results

In Chapters 5.1-5.2 we present some typical results using the final calibrated version 1.0 of Samgods. To provide more details on the models performance also results and observations made during the development phase are presented in Chapters 5.3-5.5.

5.1 Version 1.0 Results Main 2030

The basic 2030 scenario Main2030R37, the IMO-medium case defined by Trafa, has been selected for presentation of results. In Table 5.1 we show the transport work per mode after Standard LogMod and two iterations with the RCM-procedure.

Table 5.1 Transport work per mode at various stages. (LP1 results were obtained after as much as 107 CPU-hours on the machine used, a PC with an Xeon processor).

	STD	RCM:LP0	RCM:LP1
	Tonne kms [*10 ⁹]		
Road	65.50	72.92	75.37
Rail	50.52	32.79	47.41
Sea	62.66	71.56	48.22
Air	0.00	0.00	0.00
Total	178.68	177.27	171.00

¹¹ Note that this does not apply in case of multiple entries of the same type.

The transport options included in the LP0-problem are not sufficient to allow for a feasible solution to be identified, and therefore the safety exit option to acquire extra capacity has been utilized to allow for 625 trains per year over capacity on two links (numbered 83 and 293). The estimated cost, the minimum objective value in the LP model, to accomplish this solution amounts to 7060 MSEK/year (of which approximately 620 MSEK are due to the infeasibilities).

The LP1-problem has been solved with a different LP-solver that is being tested for release in version 1.1 but also with the one in version 1.0. The estimated cost in this case has been cut down to 2410 MSEK/year, i.e. a cost reduction with almost 66 %, and the solution is feasible. We present the marginal costs from the two LP models in Figure 5.2 (and two LP1 sets of marginal costs). You may notice that the same marginal costs are obtained but in two instances they appear on different links (the blue and yellow bars appear on links 1 and 5 and on 211 and 212 respectively).

The large difference in costs are also reflected in the transport work flow we see in Table 5.1, as much as 47.4 out of 50.5 billion tonne kms are still on rail after LP1.

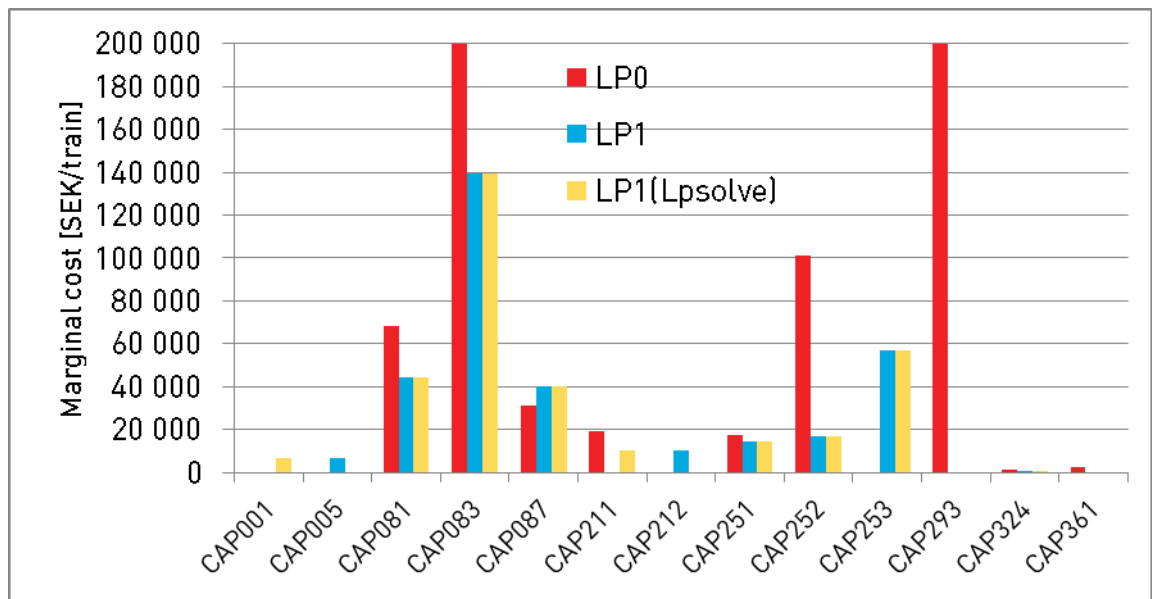


Figure 5.2 Marginal cost [SEK/train] in the LP models. The marginal cost for the unsatisfied capacities are 499 800 SEK/train (cut off in the diagram).

The locations of these capacitated bidirectional links are presented in Figures 5.3 – 5.5 as centered node markers on the links. From the maps it can be seen that the “jumping” marginal costs occur on neighbor links.

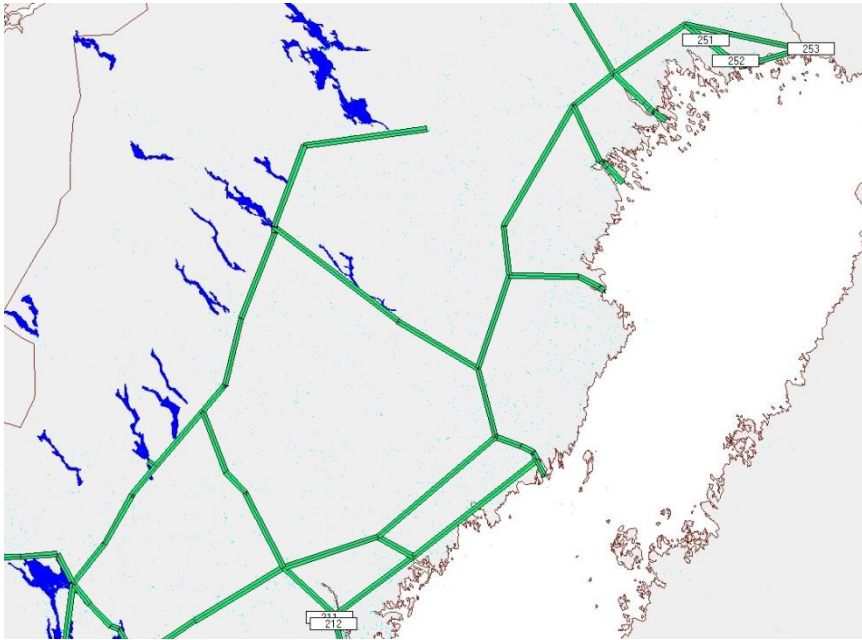


Figure 5.3 Capacitated links with number marks centered on the links: Høga kusten – Haparanda.

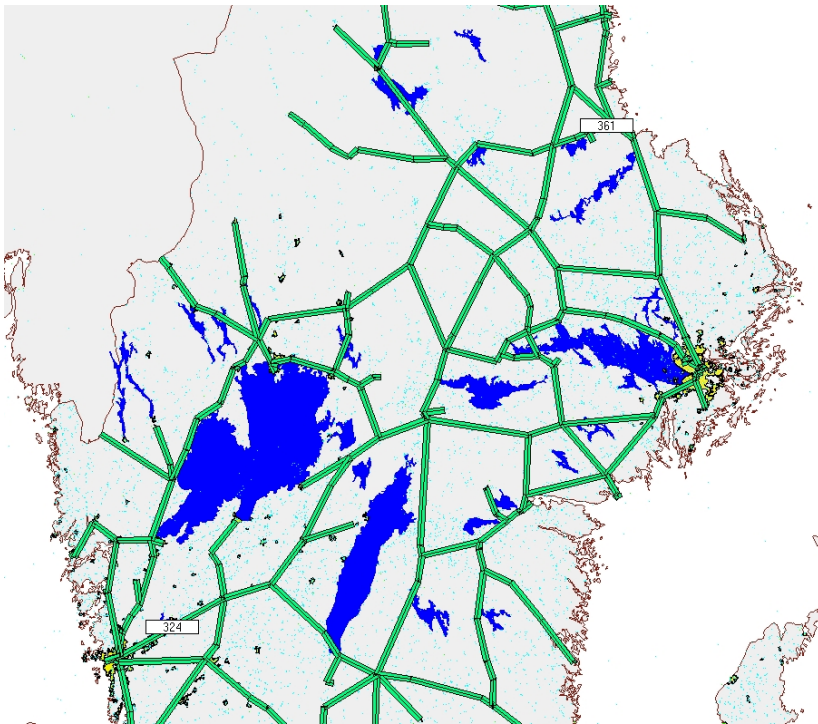


Figure 5.4 Capacitated links with number marks centered on the links – Gothenburg – Gävle.

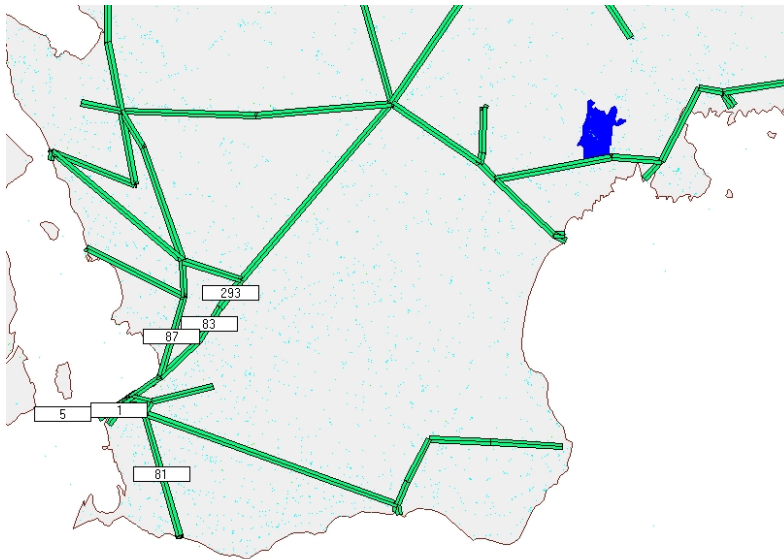


Figure 5.5 Capacitated links with number marks centered on the links – Skåne.

5.1.1 Version 1.1 Results

The results from using the new software SOPLex is practically identical to the ones obtained from LPSolve. For the base year 2012 problem the solution time for the LP0 problem often is in the range 5-15 minutes and for the LP -problem in the range 30-120 minutes. The solution time for 2030 will in general be longer since the constraints are tougher. A solution time of approximately 7 hours has occurred for the LP1 problem with 2040-demand matrices.

In general the 2012-solutions are feasible after LP0, i.e. all capacity constraints are satisfied (no need to acquire any extra dummy capacity). During LP1 the solution is improved which leads to a lower total cost.

So far the experience with 2040 solutions is limited but it seems to be the case that unsatisfied capacity constraints prevail after LP0, but are solved after invoking LP1.

Since the LP model exactly replicates the empty train flows through the return paths using the reverse forward paths compared to the loaded trains, there is not any need for a post assignment process for empties.

5.2 General Observations among the Results during the Model Development

A number of cases have been used in developing the method with data for the base year 2006 and for a forecast year 2030. Initially empty flows were handled with fix add-on factors for empties, before ending up with the solution described in Chapter 4.2. The results here

represents a 2030 case with a revised Extract module limiting the empty flow in all relations to maximally the loaded flow, representing an add-on factor at 100%. A full feedback of the LP solution into the basic logistics model format has been performed after each iteration to account for dynamics in the empty flow results. This means that we after each LP-solution merge the basic ChainChoi-results with the current LP-solution (without recomputing all the costs and statistics). For each basic ChainChoi-output file we obtain an LPX-version where the optimal LP solution columns are inserted. These results are processed with the Extract-module for rail vehicles (201, 202, 204, 205, . . . , 209) only (the others are not needed). We obtain total and empty vehicle OD-matrices in an LPX-version of the standard OD-matrices. In subsequent iterations all empty add-on factors are modified to implement the most current loading of empties. Previously computed add-on factors are not changed when the updated ones are lower or non-existing, the reason being that we want to avoid under-estimation of the capacity needed.

We first show the total cost table in Table 5.6. In terms of improvement per iteration it displays a different performance compared to the previous cases with fixed add-on factors, namely that the total costs increase by each iteration (this is however not always the case). With fixed add-on factors the representation of the empty flows are exact in the LP model and thus consecutive iterations results in non-increasing costs.

From 1517 MSEK per year it increases to 1591 and 1615 MSEK after iterations 1 and 2. This is caused by the successively modified add-on factors for empty trains, determined from the intermediate inclusions of the current LP-solutions and the consequential empty train loadings. The higher optimal costs are of course also reflected by, on the average, higher marginal costs, a natural effect since fewer loaded trains can be fitted in when the empty trains successively require more capacity (possibly as an effect of a misrepresentation in the empty train generation). A further observation is that after the first inserted two columns per f2f relation, very few new columns were generated, see Table 5.11.

Table 5.6 Total costs for the consecutive solutions in the iterations.

RCM Result cases	Total cost [MSEK/yr]
LP0	1517
LP1	1591
LP2	1615

We find that the change from LP iteration number 0 to number 1 is very small, thus we have for all practical purposes determined the optimal solution in the process. Also the fact that very few new columns were generated in iteration 1 tells us that we are close to the optimal solution already.

Checking also the marginal costs in Figure 5.7 tells us the same thing. The changes between the results after two iterations are very small. Within the current case the differences are

marginal, but they are more substantial for some of the capacities when comparing with the case with fixed add-on factors for empties.

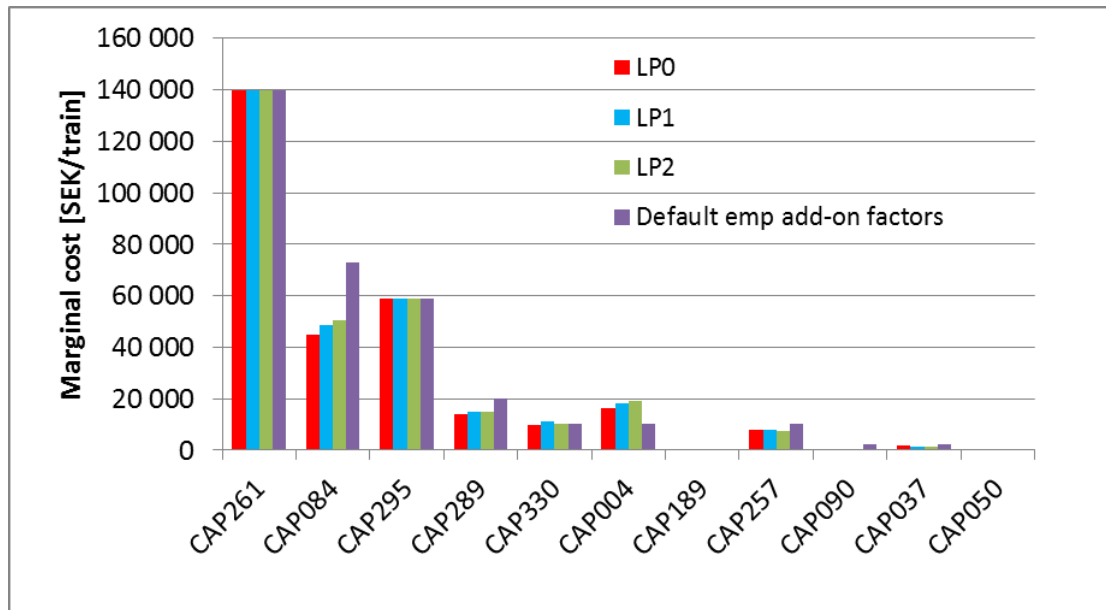


Figure 5.7 Marginal costs for the capacities with marginal costs arranged for the same capacities as in a previous case with fixed add-on factors according to the empty train allocation after running the basic logistics model.

The location of these capacity bottlenecks in Sweden are shown on the rail network in Figure 5.8.

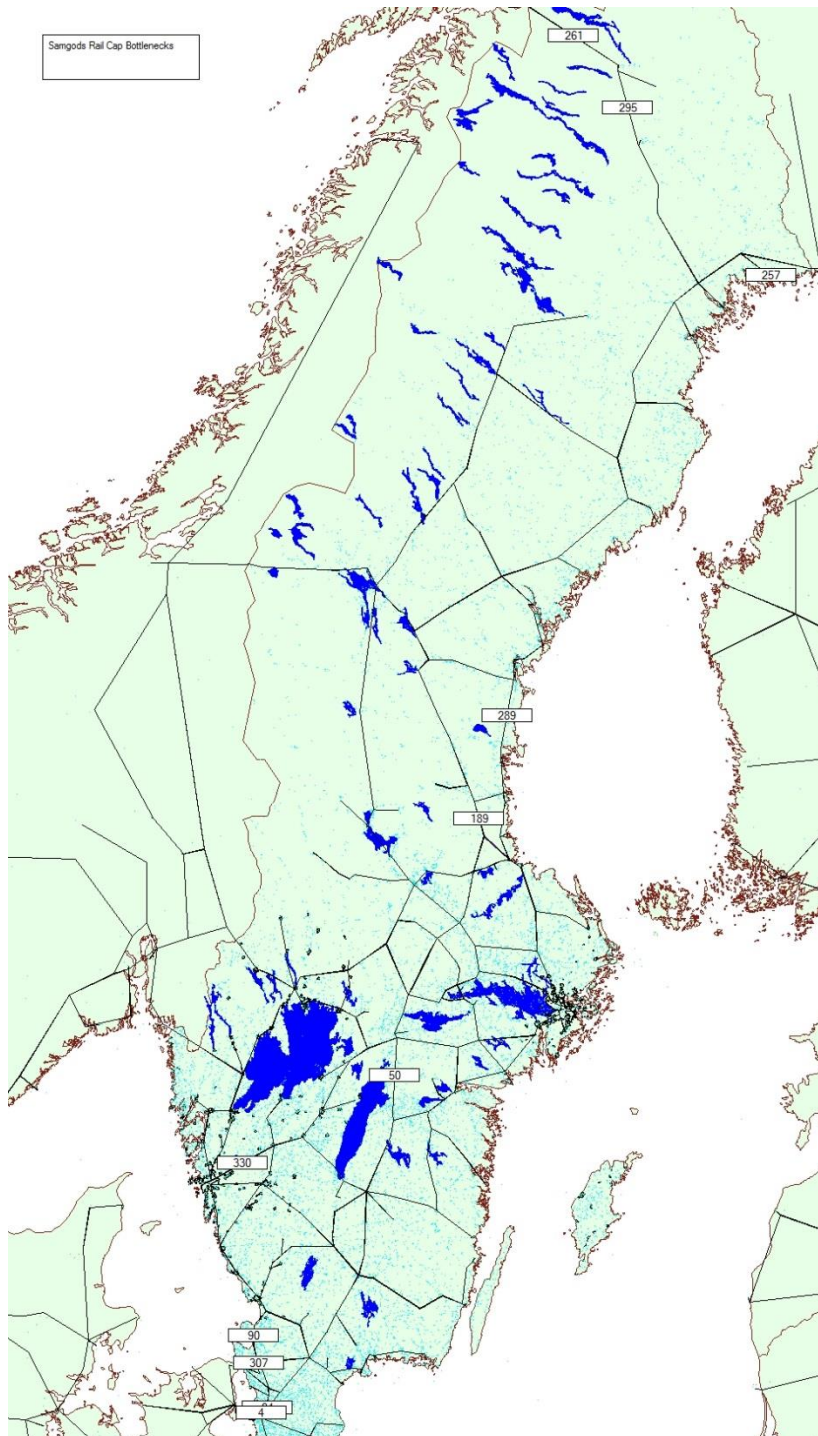


Figure 5.8 Location of constraining capacities, with marginal costs > 0 , in the 2030 scenario (link nbr 84 is hidden behind link nbr 4).

The result with initial loading ratios and final loading ratios are presented in Figure 5.9 for all bi-directional links with initial utilization rate over 50 %. As shown in Figure 5.9 there are after iteration 2 only one constraint with a load exceeding 100 %, namely CAP257 at 100.9 %. However, the absolute value is low since the capacity is only 13 freight trains per day.

The distribution of the solution variables in Table 5.11, shows that the solution is almost integer. Only 8 variables assume fractional values out of almost 132 000 non-zero variables. The CPU-time required on a PC with an Intel Core i7-3770 CPU at 3.40 GHz is less than 10 minutes. The LPSolve code is compiled and linked with the C-compiler in Intel C++ Composer XE 2013.

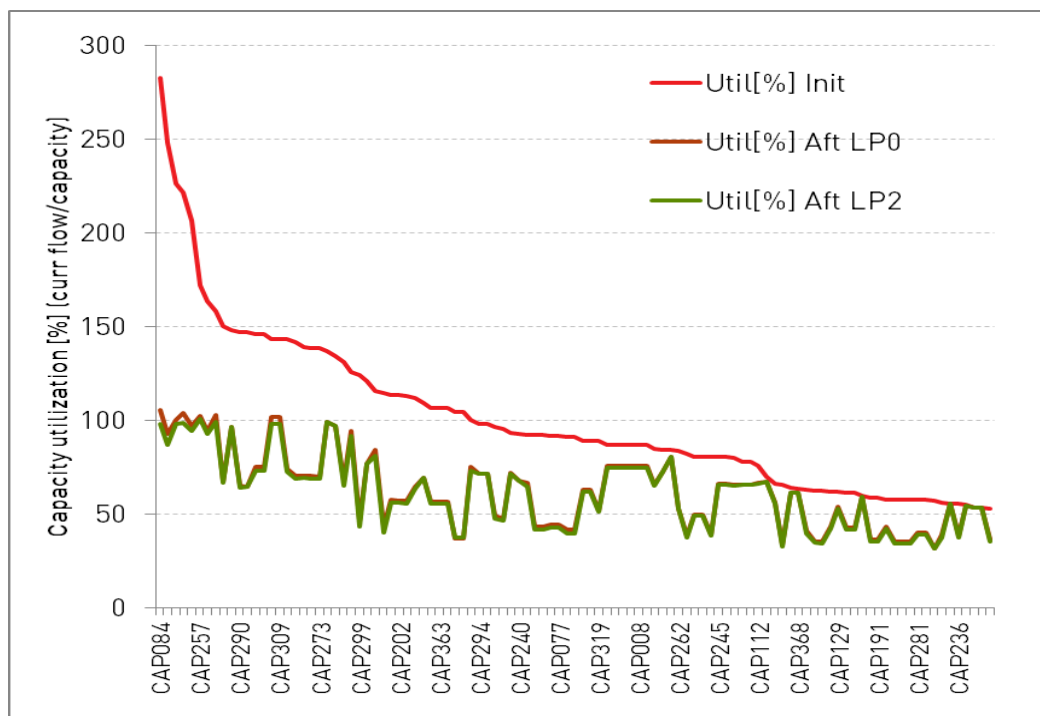


Figure 5.9 Railway link utilization before and after application of RCM with implemented, intermediate LP-solutions and consequential empty train assignments according to the extract-module.

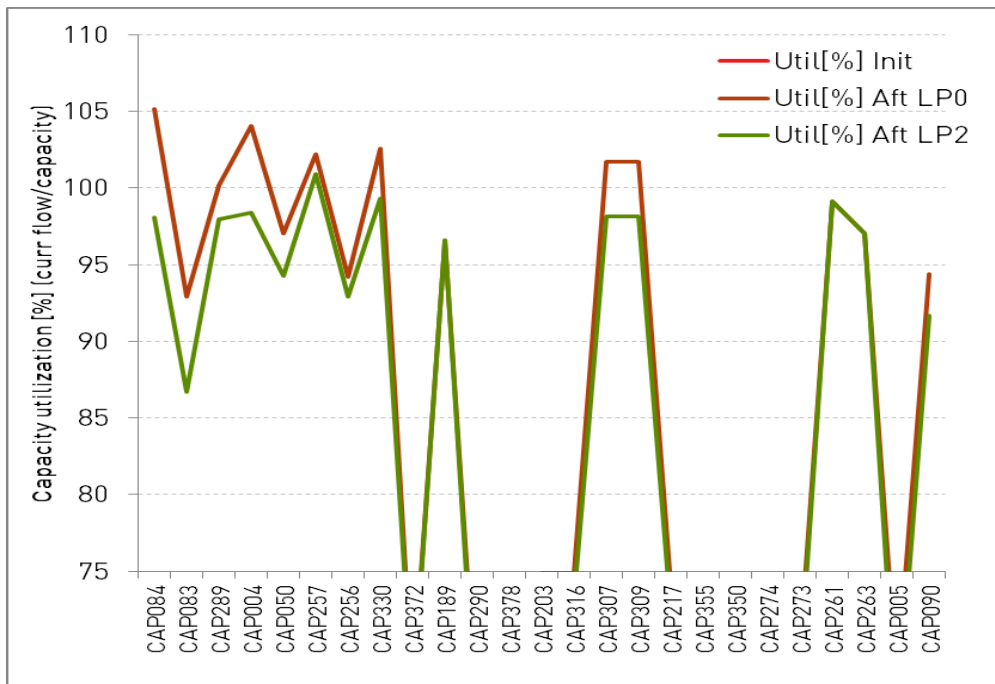


Figure 5.10 Display of the utilizations (in the same order as in Figure 5.9) zoomed in on the link flows closest to the capacity.

With the data in Table 5.11 and the model construction in mind we conclude that the good performance in terms of CPU time, to a very large extent relies on the model structure that firstly cuts down the number of variables in the model by almost a factor 2, and secondly, and even more important, almost eliminates the need for convexity constraints (only 425 are required for this case). The solution times for LP problems in general increase at least proportionally to the number of constraints, and without the elimination of columns 0 we would need 1 478 965 convexity constraints!

The result in terms of domestic transport work distribution among modes is an issue of considerable interest. As displayed in Table 5.12, the impact on domestic rail transports in the model is quite substantial, 34 % is moved to other transport modes. The basic model needs to be better calibrated in general, and probably the important load factors need revisions, but the volume 19 375 Mtonkm in case LP for this 2030 case is lower than the actual transport volumes during 2006 at approximately 22 300 Mtonkm. During the further calibration and scenario analyses carried out in December 2014, the costs for loading/unloading were modified (lowered) which lead to a substantial change in the tonne kms on rail after RCM. Probably this lead to railway solutions being used on shorter distances which in turn opened up for higher utilization of the rail network.

Table 5.11. Distribution of solution variables. The total number of $f2f$ -relations are 1 478 965, which also is the number of columns 0 and 1 per $f2f$ -relation in the LP model. The number of generated columns 2 and 3 are as low as 425. Since columns 0 are handled implicitly, the total number of variables in the LP model is 1 478 965 + 425.

x-values	LP-col 1	LP-col 2	LP-col 3	Total
0.181	1			1
0.298	1			1
0.397	1			1
0.46	1			1
0.803	1			1
0.838	1			1
0.955	1			1
0.964	1			1
1	131 596	72	80	131 748
Total	131 604	72	80	131 756

Table 5.12 Domestic tonnekms distribution with the basic logistics model and the result after applying the post processing with RCM.

	Domestic tonnekms [Mtonkm]		
	BASE	LP	Difference
Road	63 156	66 654	3 498
Rail	29 289	19 375	-9 915
Sea	91 637	97 033	5 396
Ferry	506	541	36
Air	0	0	0
Total	184 587	183 603	-985

5.3 OD matrices for empty trains

Since the capacity is measured in number of trains the actual load factors of the trains are of paramount importance, both as loaded trains and when accounting for the empty trains. On the matter of this *first aspect* we summarize the results in Figures 5.13 and 5.14 regarding the loaded trains. We find, for example, that also trains with high load factors are cancelled by RCM to a considerable extent.

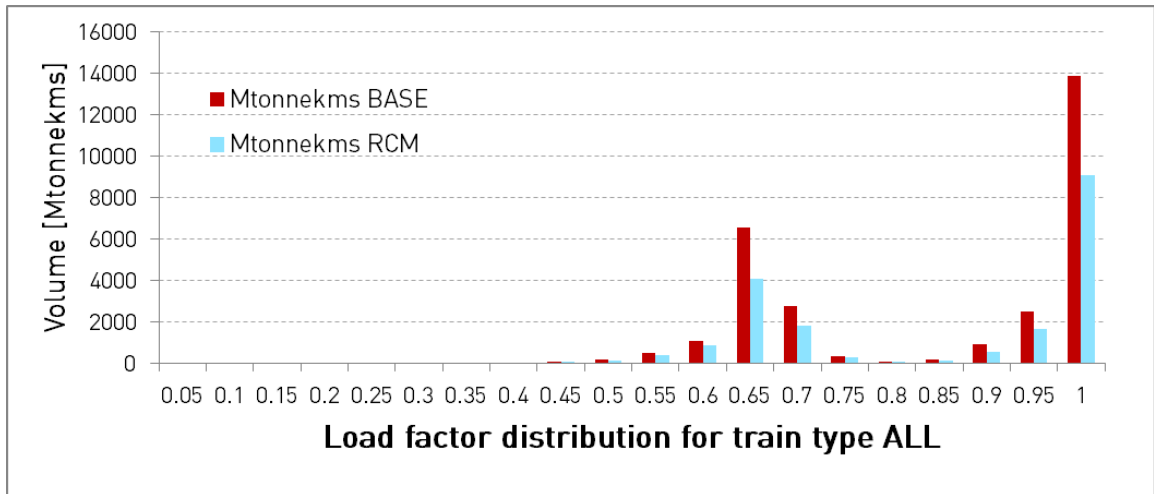


Figure 5.13. Railway transports moved to other modes by RCM. All trains types are considered, but only loaded trains.

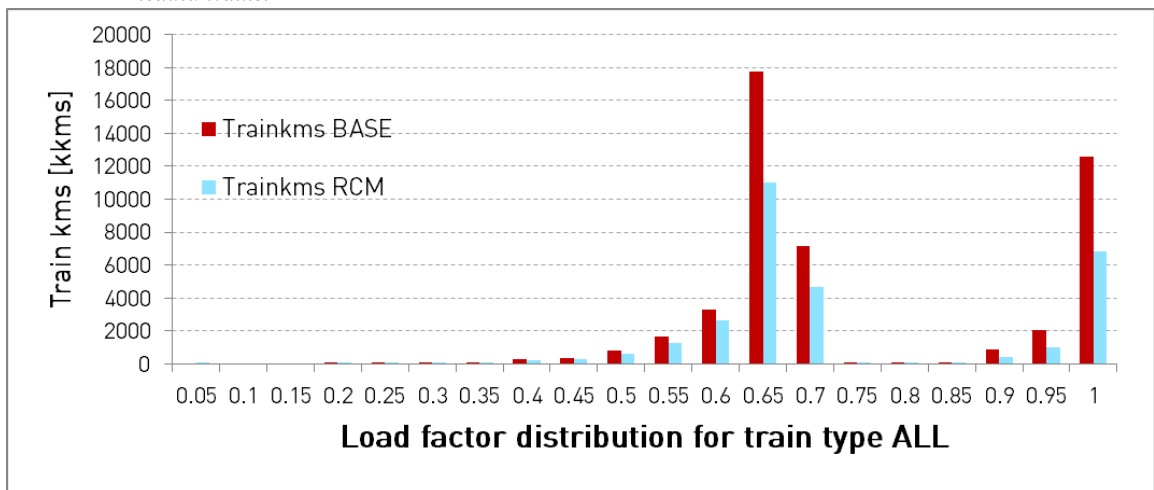


Figure 5.14. Reduction of train movements by RCM. All trains types are considered, but only loaded trains.

Since we need the full picture with empty train flows also we show some statistics in Tables 5.15 and 5.16. All train kms are provided in Table 5.15 to allow for a computation of the total train utilization levels, and we find that as much as 40 % of the freight trains are running empty according to the empty train assignment model.

The total increase of the average load per train for all trains in Table 20 (column Diff ...) from 620 to 661 tonnes, despite a number of reduced average loads, are explained by the fact that most of the system train transports (trains 204-206) are not subject to reassignments to other modes. Although some of the utilization levels for loaded trains are very good, almost all average utilizations are below 50 % when considering the empty flows. Based on these

observations and their impact on the results, we find it necessary to compare with observed data on railway transports and pay more attention to the empty train assignments in general.

Table 5.15 The allocation of loaded and empty trains after RCM.

RCM, domestic trainkms			
Train	Loaded train kms	Empty train kms	Empty trains [%]
201	9 924 187	5 092 837	33.9
202	1 077 064	678 966	38.7
204	4 220 550	3 925 585	48.2
205	1 369 281	1 368 654	50.0
206	333 922	333 922	50.0
207	10 561 000	6 395 980	37.7
208	1 083 472	1 004 035	48.1
209	657 132	626 158	48.8
ALL	29 226 608	19 426 137	39.9

Table 5.16 Results on utilization of loaded trains for the basic logistics model and after application of RCM. The utilization factors after RCM are shown in the second rightmost column.

Train	BASE: E[load/train]	RCM: E[load/train]	Diff: E[load/train]	BASE: Uload[%]	RCM: Uload[%]	RCM: Utotal[%]	Capacity per train [tonnes]
201	378	375	-3	62	61	41	610
202	313	303	-10	64	62	38	488
			0				0
204	943	943	0	98	98	51	960
205	3 458	3 469	11	95	95	48	3 652
206	5 274	5 274	0	88	88	44	6 000
207	366	338	-28	70	64	40	526
208	675	677	2	94	95	49	716
209	790	755	-35	87	83	43	907
ALL	620	661	41				

Statistics from the Transport Administration concerning 2006 display an average load at 440 tonnes, whereas the post-RCM value here is $661 / (1 + 19.4 / 29.2) \approx 397$ tonnes (the pre-RCM value would approximately be 370 tonnes).

5.4 Assignment of empty trains

A **second aspect** on the empty train flows concerns the assignment of them to the network. The current practice is to consider them assigned according to the LOS paths when computing the add-on factors as discussed in Section 4. Only in the post-process after the

last LP-solution in the column generation scheme the congested assignment of empties is used to carry out a final check of whether or not the solution is feasible. In case the adjustment capacity phase is used, *LPnAdjCap*, the congested assignment procedure is used in all those iterations.

This process is not required when handling empties according to the current Samgods model design, in which the empty train volumes are strictly proportional to the loaded flows for trains. See also comment in Section 5.1.1,

5.5 Final Comments

In summary we would like to claim that the method provides very good results, in particular with the dynamic updates of the empty loading factors through insertion of intermediate LP solutions and updated consequential empty loading factors. A number of issues regarding the empty train assignments probably need further attention as discussed in Chapter 4.2, in particular if the asymmetric flow case would be incorporated. In general it would most probably be worthwhile to enable the use of multiple paths for the train allocations, not only through changes of transport chains and changes of transfer points inside the transport chains. This is particularly important for direct system railway transports that only may use a single path. Since the rail network is a rather sparse one, an option to have one alternative path to the first one would most likely be highly valuable. The important steel shuttle between Luleå and Borlänge constitutes a typical example.

6 A few LP solver software options

The current version 1.1 is operating with the SOplex software, see Wunderling [1996]. In the work with the forecast scenario the problem sizes have grown considerably due to calibration efforts for the base year 2012 involving more railway transports than in Samgods model instance used during the development of RCM. In the main forecast scenario 2040 this has become a matter of increasing importance with seriously increased CPU times. The stress tests made on the system during the development phase were too undemanding.

The experienced too long CPU times for 2030-scenarios led to a revision of RCM for version 1.1 by changing the model formulation strategy in the following manner:

Handle the many small railway transport volumes, demanding little capacity, in a more standardized manner. It is accomplished by only adding the first alternative solution from Standard LogMod in the LP model for transports with shipment sizes below a certain cut off limit. During the calibration 2012 this limit was set to 2 tonnes.

This has worked out quite well, although the solution times for iteration 1 (the 2nd after iteration 0) is around 2 hours. However, more alternatives exist that may be used to keep down the problem size and yet achieve good solutions. One option would be to reevaluate entered options in a more elaborate manner, and sustain from keeping options in the model that are unlikely to be used. Such cases would be handled in a manner similar to the low shipment size cases.

The SOplex linear programming software, cf Wunderling [1996], has been tested after comparison with the previously used LP-solve. It is allegedly one of the fastest non-commercial LP-solvers available, see Figure 6.1 for an illustration. Conditions for a commercial license can be agreed upon by contacting Thorsten Koch at Koch@zib.de. Soplex is developed and maintained by a number of well-known universities and also in co-operation with many well-known companies such as for example Siemens and SAP, see <http://soplex.zib.de/>.

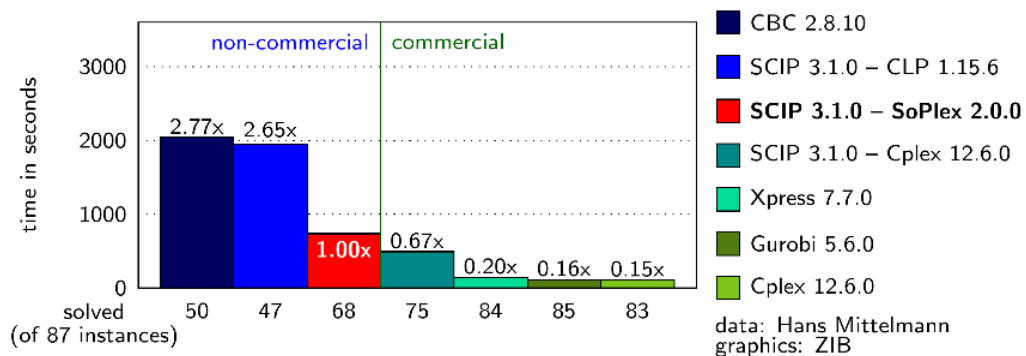


Figure 6.1 Comparison of CPU-time performance. Source: <http://scip.zib.de/>

Citing the homepage from the soplex developers we are informed that:

“SoPlex is a Linear Programming (LP) solver based on the revised simplex algorithm. It features preprocessing

techniques, exploits sparsity, and offers primal and dual solving routines. It can be used as a standalone solver reading MPS or LP format files as well as embedded into other programs via a C++ class library. SoPlex has been implemented as a part of Roland Wunderling's Ph.D. thesis Paralleler und Objektorientierter Simplex-Algorithmus (in German) and is available in source code. SoPlex is free for academic research and can be licensed for commercial use.”

An important fact to consider for the two mentioned non-commercial softwares above is that they utilize only a single CPU-unit. The reason is that the algorithms used are very difficult to parallelize, despite the title of the mentioned Ph D-thesis. However, it should not pose any problem, except perhaps licensing conditions, to run multiple Samgods scenarios in parallel.

For Trafikverket’s evaluation of the options available we are using one instance of an LP1-problem form *MainSc2030R0/R4*. Available aggregate information for the three discussed softwares are presented in Table 6.2. The PC used for the CPU-time test was a Dell Precision T3500 (model year 2012), Processor: Intel(R) Xeon®, W3670 @ 3.20 GHz 3.20 GHz. RAM: 12 GB. OS: Windows 7, 64-bit.

As an alternative PC we used a Dell Optiplex 7010 (model year 2012), Processor: Intel(R) Core(TM)-i7-2600 CPU @ 3.40 GHz. RAM 16.0 GB. OS: Windows 7, 64-bit. Further tests made in January 2014 with SoPlex version 2.0.1 (recommended by Thorsten Koch) showed even better performance, see Table 6.2.

Table 6.2 LP software properties. Note! The CPU-time is 42 % longer on the Xeon machine.

Software	CPU-time for solving LP1-instance MainSc2030R0/R4	License cost	Maintenance cost	Update frequency	Other information
LP-solve	69 hrs 27 min Alt: 48 hrs 33 min	Free	Free	V 5.1 2006-10 V5.5.2 2013-04 Non-regular updates	License terms: GNU LGPL
SoPlex ver 1.7	6 hrs 10 min Alt: 4 hrs 22 min	Aprr 500 euros	None?	Once a year the last 5 yrs	Licence terms: ZIB Academic License
SoPlex ver 2.0.1	Alt: 3 hrs 17 min	As above	As above	As above	As above
CPLEX	38 minutes, see details below Result from IBM 2014-12-01. Note. The test is run by IBM on their machine using 12 cores.	Minimum 8778 + 18087 = 26 865 euros. Possible TrV discounts.	Appr 15 %/yr	Minor 12 month interval, Major 18 month interval	Requirement: 1 development licence and 1 or more deployment licences. See further details from e-mail excheange Sweco-IBM.

The tests of the CPLEX software has so far been done by IBM, see e-mail below. A trial license from IBM. It has been somewhat difficult to obtain straight forward license terms.

IBM-information in e-mail 2014-11-16 (heading CPLEX pricing):

TrV will need at least 1 development license and 1 or more deployment licenses.

Development licence => D0CV0LL IBM ILOG CPLEX OPTIMIZATION STUDIO DEVELOPER EDITION AUTHORIZED USER LICENSE + SW SUBSCRIPTION & SUPPORT 12 MONTHS

Public price is 8778 euros, such price does not take into account special conditions TV might have. SEK Price of the final configuration will have to be confirmed by a local sales rep.

Deployment license : 3 possible solutions

1) BM ILOG CPLEX Optimizer Single User Edition Client Device License + SW Subscription & Support 12 Month

- License attached to one single device, 1 application only, 6 cores max, limited to 1 concurrent access (1 user at a time)

- Public price 18087 euros. Such price does not take into account special conditions TrV might have. SEK Price of the final configuration will have to be confirmed by a local sales rep.

2) IBM CPLEX Optimizer Deployment Entry Edition Processor Value Unit (PVU) License + SW Subscription & Support 12 Months

- License limited to one application, limited to 140 PVUs (power of the CPU depending on the type of machine/processor you work with). FYI New professional laptops now are at least 140 PVU.
- Public price 161 euros. per PVU i.e. 22540 euros for a 140 PVUs machine. Such price does not take into account special conditions TrV might have. SEK Price of the final configuration will have to be confirmed by a local sales rep.
- Such license is not very appropriate for multiple users.

3) IBM ILOG CPLEX Optimizer Deployment Edition Processor Value Unit License + SW Subscription & Support 12 Months

- Optimization server allowing unlimited concurrent users. Most adopted license.
- Public price 267 euros per PVU i.e. 26700 euros. for a 100 PVUs machine, 53400 euros for a 200 PVUs machine, etc.
- Such price does not take into account special conditions TrV might have. SEK Price of the final configuration will have to be confirmed by a local sales rep.

Result of CPLEX tests

For the current problem instance CPLEX has reported errors in input data and stopped. The reported error is a complaint about “unwanted input”. The known issues are cases where upper bounds are provided for variables NOT being in the constraint matrix. Below an example is provided from a small cleaning program constructed by Sweco:

Skipped: UP BND1 V26686251 1.
 Skipped: UP BND1 V26686252 1.

Result from IBM 2014-12-01:

In general, an important remark is that the barrier algorithm of CPLEX works much better and by far than the other algorithms (dual or primal), also the barrier benefits (much more than the other algorithms) from multiple cores to solve a model. Notice also that SOplex doesn't include a barrier algorithm.

Machines used: Xeon X5650 2.66 GHz, Red Hat 6, 24 GB, 32 cores

Result of the reference test: Problem solved to optimality within 2228.63 sec using 12 threads.

This satisfies the goal of solving the problem in less than an hour (around 40 minutes). Performance is degraded when using more than 12 threads, so a machine with 12 cores using barrier should achieve their goal, the option of using 6 cores is also a valid choice using the barrier algorithm.

6.1 Linear Programming version in Samgods since 2016

The current linear programming software is CLP, which is a result of the COIN-OR projects (www.coin-or.org). Samgods use version 1.17.6 built using a 64-bit compiler. When comparing the performance of this software compared to SOplex 2.01, CLP results were identical and the solution times were much shorter.

7 Summary and Conclusions

A general conclusion is that RCM by LP seems to work well. Already with the first LP-solution we have good approximations of the marginal costs for the most constraining capacities. In the tested cases the second LP solution differs only marginally from the third, so solving at most two LP models ought to be good enough, especially when considering that the basic logistic model involves only three iterations.

After only a few additional iterations the solution has converged. The LP solution contains only a few fractional variables, so the split of the solutions related to fractional solution variables does not cause any problems. The costs after the few splits obtained when reevaluating the final solutions are almost the same as the cost from the LP model.

The formulation of the LP model with implicit use of the first best solution, and letting the explicit variables represent differences compared to the first best solution has proven very efficient. It has led to a model (in our tested cases) with almost half of the variables (appr 1.4 M), and only a small fraction of constraints, a few hundred convexity constraints instead of one for all first best solutions (i.e. 1.4 M), we would otherwise have had with a straightforward setup.

With the current handling of empty train flows as strictly proportional the loaded train flow, the modelling of the train loads in the LP model becomes very exact, and this is also displayed by a full satisfaction of all capacity constraints in the tested cases so far. Nevertheless, there is a potential for utilizing the rail network better in the model by assigning the empty flows prior to solving the LP models, and then using add-on factors for empties that in general would be using a combination of routes. This would require further development of RCM, and we would then come back to the issues raised below.

As shown in the analysis in Chapters 4.1 – 4.2 there are some issues that need further attention in the future in order to properly reflect the train utilization model:

1. How to make better use of the congested assignment model for empties to improve the utilization of the railway network?
2. How to set up the basic model for loaded and empty¹² trains to reflect observed actual train load averages?

In summary, we claim that the result is rather successful with an almost complete satisfaction of all capacities within a limited additional running time, in particular with respect to all other aspects that may be applied to the logistics model in terms of cost assessments, network routing, vehicle utilization model, empty train assignment procedure etc.

However, in light of the experiences during the calibration of the model and running the model for the 2030 scenario we see some potential in revising the handling of empties, and

¹² As mentioned earlier the extract module was revised in December 2014 to allow for user input of empty train usage. The basic idea is to set it up to reflect average empty train flows provided in ASEK. All data presented in this report deals with the version with fixed, hardcoded empty trains allocation parameters in the extract module.

also in a more demanding change of the model to handle multiple paths between the OD-pairs. The distribution of the problem involves many small transport volumes resulting in the main part of the variables, whereas the dramatically fewer larger transport volumes constitute the main part of the allocation problem. Since all the variables representing small transport volumes are as computationally demanding as the ones representing larger transport volumes, we could save on computational effort by a special handling of the small transport volumes in some manner without losing very much in terms of addressing the allocation problem.

Obviously we also see a large potential in changing the LP-solver software to something else, c f Chapter 6.

Acknowledgements

The author has very much appreciated the possibility to develop RCM for Trafikverket. Also the detailed comments on the manuscript from Petter Hill, Trafikverket, and Gabriella Sala, Citilabs were very useful and welcome.

8 References

- Bazaraa M S and Jarvis J J (1977): *Linear Programming and Network Flows*, John Wiley, New York.
- Bergström J (2015): *Documentation of mps.jar*, Compose IT.
- de Bok Michiel, Baak Jaap and de Jong Gerard (2011): *Program documentation for the logistics model for Sweden*, SIGNIFICANCE, January.
- de Jong Gerard, Ben-Akiva Moshe and Baak Jaap (2011): *Method Report - Logistics Model in the Swedish National Freight Model System (Version 2)*, Significance, Deliverable 6b for the Samgods group D6b, Project 10060, January.
- Edwards Henrik, Swahn Henrik and Bates John (2008): *Swedish Base Matrices Report. Estimates for 2004, estimation methodology, data, and procedures*, PM 2008-03-13, Vägverket Konsult, Solna.
- Edwards Henrik (2008): *Update of Samgods Base Matrices 2008*, PM 2008-09-25, Vägverket Konsult, Solna.
- Edwards Henrik (2010): *Example of results from the Logistics Model in Samgods (in swedish)*, Vectura 2010-09-17, PM.
- Edwards Henrik (2011): *The Logistics Model in the Swedish National Freight Transport Model – A Discussion of Unresolved Issues Conference, Kuhmo Nectar Conference and Summer School on Transportation Economics*, Stockholm, Sweden, June 2011.
- Edwards Henrik (2013): *Railway Capacity Management in the Swedish National Freight Transport Model, Samgods, using Linear Programming*, 2nd Symposium of the European Association for Research in Transportation (hEART), Stockholm, Sweden, September 2013, Stockholm, Sweden, June 2011.
- Ortuzar Juan De Dios and Willumsen Luis G. (1996): *Modelling Transport*, John Wiley, New York.
- Sheffi Yosef (1985): *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.
- SIKA (2004): *The Swedish National Freight Model: A critical review and an outline of the way ahead*, Samplan 2004:1, SIKA (now partly transformed to Trafikanalys, Sweden (www.trafa.se)), Stockholm.
- Vierth I, Lord N and McDaniel J (2009): *Representation of the Swedish Transport and Logistics System (Logistics Model Version 2.00)*, VTI notat 17A.

Vägverket [2008]: The Road Transport Sector - Sectoral Report 2008, Publication 2009:34, Borlänge.

8.1 Linear Programming Software

LPSolve (used in version 1.0 of Samgods)

Michel Berkelaar, Kjell Eikland and Peter Notebaert (2004): LPSolve citation data:

Description : Open source (Mixed-Integer) Linear Programming system

Language : Multi-platform, pure ANSI C / POSIX source code, Lex/Yacc based parsing

Official name : lp_solve (alternatively lpsolve)

Release data : Version 5.0.0.0 dated 1 May 2004

Co-developers : Michel Berkelaar, Kjell Eikland, Peter Notebaert

License terms : GNU LGPL (Lesser General Public Licence)

Citation policy : General references as per LGPL

SOPlex 2.01 (used in version 1.1 of Samgods)

Roland Wunderling (1996), Paralleler und Objektorientierter Simplex-Algorithmus, ZIB technical report TR 96-09, Berlin 1996.

Bibtex-reference:

@PhdThesisWunderling1996,

author = Roland Wunderling

title = Paralleler und objektorientierter Simplex-Algorithmus

School = Technische Universit, Berlin

Year = 1996

Note = \url{http://www.zib.de/Publications/abstracts/TR-96-09/}

Homepage: <http://soplex.zib.de/>

CLP-1.17.6 (used in version 1.2 and later of Samgods)

The COIN-OR LP code is designed to be a high quality Simplex code provided under the terms of the Common Public License. CLP is written in C++, and is primarily intended to be used as a callable library (though a rudimentary stand-alone executable exists). The first release was version .90. The current release is version .99.9.

Authors: John Forrest, IBM Research, jjforre@us.ibm.com, David de la Nuez, Columbia University & IBM Research, dmd57@columbia.edu Robin Lougee-Heimer, IBM Research, robinlh@us.ibm.com.

CLP and this documentation are provided under the terms of the Common Public License ("CPL"). Any use, reproduction or distribution of the programs constitutes the recipient's acceptance of the license. The CPL is approved by the Open Source Initiative. IBM Corporation, the author of the CPL, has a CPL FAQ available which is based on IBM's understanding of the CPL.

Homepage: www.coin-or.org

Appendix A: Some LP model details

A first implementation of the linear programming model has been made according to the following principles:

1. It is assumed that the first best solution per PWC-f2f-relation is used in full unless a fraction of the second best solution (or second best without rail if available) is used.
2. The cost for using the second best solution in the LP model $\text{cost}_{2\text{nd}} - \text{cost}_{1\text{st}}$
3. The constraints with a utilization (link flow load / capacity flow > 0.50) of at least 50 % are included in the LP model. The motivation is to keep down the problem size, but it probably does not matter much.
4. The right hand side (RHS) of the constraints are the available capacities per year minus the required capacities for the initial basic solution. Motivation:

Required capacities on the left hand side \leq Available capacities

=>

$$[\text{Cap } 1^{\text{st}} \text{ best sln}] \times \mathbf{x}^{1\text{st}} + [\text{Cap for } 2^{\text{nd}} \text{ best sln}] \times \mathbf{x}^{2\text{nd}} \leq \mathbf{CAP}$$

To reduce the problem size we subtract the first best solution from both sides. This represents that the net effect of using the alternative solution should be on the left hand side

=>

$$[\text{Cap for } 2^{\text{nd}} \text{ best sln} - \text{Cap } 1^{\text{st}} \text{ best sln}] \times \mathbf{x}^{2\text{nd}} \leq \mathbf{CAP} - [\text{Cap } 1^{\text{st}} \text{ best sln}] \times \mathbf{1} \quad (\text{A.1})$$

In equation (A.1) we see that the net effect of introducing any of the alternative columns will add its capacity requirements and at the same time cancel out those from the first best solution.

Note! The idea of using the add-on factor in all relations both for the approximation of their impact regarding changes of the solution in the left hand side, and for the derivation of the right hind side utilizing the congested assignment model results may need to be revised as discussed in Section 7.1.1.

5. For 2nd best solution used the net effect of capacity requirements are added to the left-hand-side, i e any rail capacities required for the 2nd best are added and the rail capacities for the 1st **best solution are subtracted** to balance the subtraction of them on the right hand side.

This can perhaps be better understood by an example (The reason for doing this is to cut down the problem size and computing times.). Assume the capacity is 1000 trains per year and the required capacity for one first best relation is 100 trains. Then the RHS is $1000 - 100 = 900$. Now assume that the first best solution is replaced fully by a sea transport (no rail). Equation (A.1) with this looks like:

$$[\text{Cap for } 2^{\text{nd}} \text{ best sln} - \text{Cap } 1^{\text{st}} \text{ best sln}] \times \mathbf{x}^{2\text{nd}} \leq \mathbf{CAP} - [\text{Cap } 1^{\text{st}} \text{ best sln}] \times \mathbf{1}$$

↔

$$[0 - 100 \dots] \times \mathbf{x}^{2nd} \leq 1000 - [100 \dots] \times \mathbf{1}$$

By setting $\mathbf{x}^{2nd}(1) = 1$ we get

$$0 - 100 \dots = 1000 - 100 \dots$$

We see that the -100 cancels out on both sides and there is a remaining capacity at 1000 for the remaining 2nd best solutions. Another way to see it, is that the RHS with using any new columns is the available capacity minus all capacity used by the initial solution. Then any effect on the left hand side must introduce the new capacity load and remove the previous one (the default one from the initial solution). For a case with two transport chains with two alternatives each using 10 and 20 trains each versus the initial solution using 28 and 41 with a capacity at 50.

The condition will look like this:

$$x_{A1} \times (10 - 28) + x_{A2} \times (20 - 41) \leq 50 - 28 - 41 = -19$$

where

$x_{Ai} = 1$ if alternative A_i is used, 0 otherwise

Using none of the alternatives ($x_{Ai} = 0$) gives: $0 \leq -19$ **INFEASIBLE**

Using alternative A1 ($x_{A1} = 1$) gives: $-18 \leq -19$ **INFEASIBLE**

Using alternative A2 ($x_{A2} = 1$) gives: $-21 \leq -19$ **FEASIBLE**

Using both alternatives ($x_{Ai} = 1$) give: $-18 - 21 \leq -19$ **FEASIBLE** (A.2)

So, just in the same manner as we evaluate the cost component as the difference of the alternatives as compared to the first best solutions, all capacity inputs are in terms of the differences between the capacities of the alternatives (the entering capacities) and the capacities of the initial solution (the exiting capacities).

- Options to purchase additional capacity have been introduced in order to guarantee that a feasible solution can be obtained. In order to avoid these options to be used it will probably be necessary to set the cost high enough and perhaps also to ensure that alternative transport chains are introduced into the model, even if they are only 3rd or higher best.

Upper bounds are introduced for all 2nd best solutions. At most 100 % of the 2nd best solution can be used.

Appendix B: Method for distribution of empty flows in the Extract-module

Add-on factors for empty vehicles follows the distribution according to a piecewise linear function such as the one in Figure C.1. In december 2015 this function was changed into a user-defined function in which the levels and breakpoints (possibly more than 2) must be entered exogeneously through an input file – emptyfrac.dat.

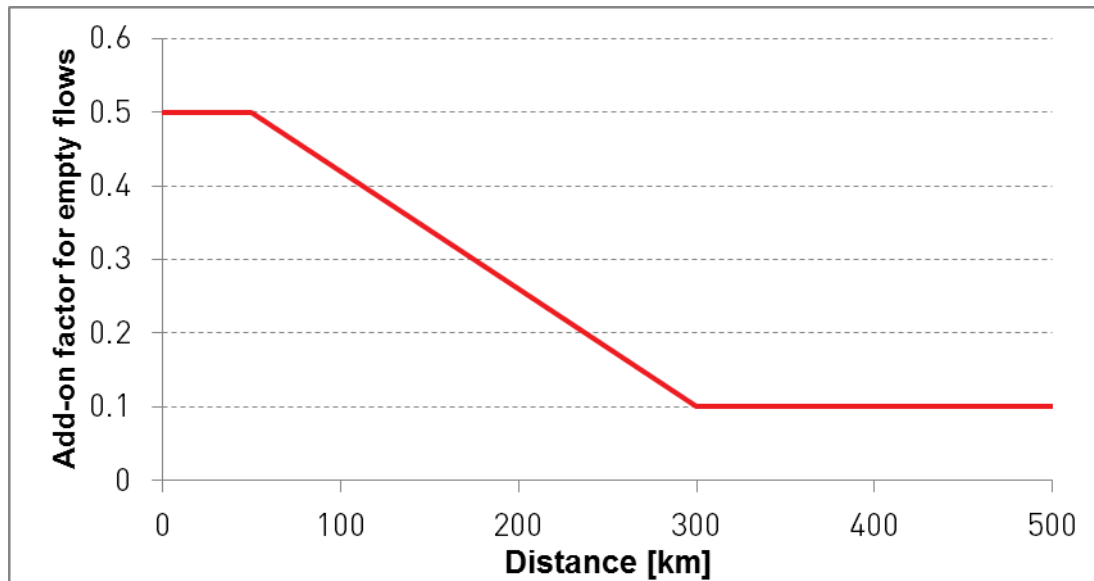


Figure B.1 Basic function for empty vehicle add-on factors in Extract.

Given the value of a distance parameter ASYM, there is on top of the empty flows according to Figure B.1 an allocation due to the asymmetry (or unbalance) of the loaded flows in an OD-relation. The asymmetry is measured as the absolute difference in the loaded flows between OD and DO respectively (DO == return OD-direction). In OD-relations with distances larger than the ASYM-value, the asymmetric loaded flow is assumed to generate the same number of empty flow in its opposite direction. The earlier, hard-coded ASYM-value was set to 50 kms, and it was used for all vehicles.

B.1 Input data for control of empty flow distribution in ctl-files for Extract

Input data for the extract module per per vehicle (in ctl-files):

ASYM=50

EMPTYFR=emptyfrac.dat

The ASYM-parameter tells the model that empty flows related to transports longer than the provided value (here 50 kms) are subject to the addition of so called asymmetric empty flow volumes. Shorter transports are only subject to symmetric empty flow handling.

The EMPTYFR-parameter references the input file containing the empty vehicle fractions.

Example of emptyfrac.dat input (the settings here gives the function in Table B.1):

```
VhclTyp Dist EmptyFrac
101      50  0.5
101     300  0.1
102      50  0.5
102     300  0.1
...
```

In December 2012 Trafikverket decided to use the average empty train add-on factors from ASEK for all train transports. This means that the function in Figure B.1 was changed to a horizontal line with a constant empty fraction value for all distances, and that the ASYM-parameter was set to 999999. Thus the empty train add on factors for trains have been changed into.

```
VhclTyp Dist EmptyFrac
...
201      0    0.200
202      0    0.790
204      0    0.560
205      0    0.560
206      0    1.000
207      0    0.450
208      0    0.450
209      0    0.450
...
```

B.2 Description of asymmetric flows

The ASYM parameter has been added to be able to run EXTRACT in a way that is consistent with the previous version. It defines the threshold distance above which asymmetric flows will generate empty vehicles. As every distance is above zero, setting ASYM equal to zero will always generate asymmetric, empty vehicle flows. Similarly, setting ASYM to a very high value (99999) will never generate empty asymmetric vehicle flows as all distances will be smaller than the threshold value. The threshold value of 50 km has been taken from the norwegian model a long time ago.

To illustrate the calculation consider an OD-relation:

```
Distance = 70 km
OD-flow = 120 vhcls
DO-flow = 100 vhcls
Empty vehicle fraction at 70 km = 0.1
```

If ASYM=50 the distance (70) is above the threshold value so **asymmetric flows will generate** empty vehicles:

$$\begin{aligned} \text{OD-flow empties} &= 0.1 \cdot 100 \\ \text{DO-flow empties} &= 0.1 \cdot 120 + 20 \end{aligned}$$

Maybe it would be an improvement to calculate the empty flows as follows:

$$\begin{aligned} \text{OD-flow empties} &= 0.1 \cdot 100 \\ \text{DO-flow empties} &= 0.1 \cdot 100 + 20 \end{aligned}$$

If ASYM=99999 the distance (70) is below the threshold value so **asymmetric flows will not generate** empty vehicles:

$$\begin{aligned} \text{OD-flow empties} &= 0.1 \cdot 100 \\ \text{DO-flow empties} &= 0.1 \cdot 120 \end{aligned}$$

It is an easy change to do but then EXTRACT performance will not be consistent with the previous version anymore. The author's opinion is that the change above should be implemented. This is motivated by the possible error that the current logic may result in, namely a larger empty return flow than the number of loaded initial flow. Assume we would have

$$\begin{aligned} \text{Distance} &= 70 \text{ km} \\ \text{OD-flow} &= 100 \text{ vhcls} \\ \text{DO-flow} &= 0 \text{ vhcls} \\ \text{Empty vehicle fraction at 70 km} &= 0.1 \end{aligned}$$

$$\text{DO-flow empties} = 0.1 \cdot 100 + 100 = 110$$

Appendix C: Spanning tree data

The fact that a single path is used for each combination of vehicle and OD-pair, makes it easy to handle the necessary mapping from transport chains to capacitated links for the suggested handling of the capacity constraints in Sections 3 - 5. The only thing required is that the so-called spanning trees, being available while computing the regular LOS matrices, are saved into text files. Thus, after having derived the shortest paths from any vehicle – origin – combination (having at least one feasible path) the predecessor links to all nodes are saved, with a format according to Table C.1. From this information it is a straight forward, and fast, procedure to backtrack the path from any destination to its origin.

In the following tables the structure implemented in the current implementation that requires three set of files:

- the spanning tree data itself;
- the list of nodes and the list of links with labels used in setting up the spanning tree file.

Table C.1 Table structure for spanning tree data with predecessor links. 0 indicates absence of predecessor link. (*PathTreeRail.txt*)

Vehicle	Origin	Pred link for Node 1	Pred link for Node 2	...	Pred link for Node N
101	711400	0	L2		LN
...					
401	977400				

By saving the spanning trees only for origins with usable, paths, the number of rows in the above tables can be reduced considerably. The number of active origins for the four different modes are currently as shown in Table C.2.

Table C.2 Number of origins with at least one usable path per mode.

Mode:	Road	Rail	Sea	Air	Total
#vehicles	5	8	21	1	35
#O-zones	1068	368	194	148	
#Active nodes					13037

The node list contains the serial numbers and hierarchical numbers as shown in Table C.3.

Table C.3 Node list [*Nodes_List.txt*].

Internal_Node_Nbr	VY-NodeNvr	EMME2-NodeNbr
1	1	711400
2	2	711401
3	3	711402

4
---	-----	-----

The link list contains the serial numbers and hierarchical from and to node numbers such as the example in Table C.4.

Table C.4 Link list [*Links_List.txt*].

Internal_Link_Nbr	FrVY	ToVy	FrEMME2	ToEMME2
1	1	2659	711400	300025
2	2	2838	711401	300255
3	3	3100	711402	300595
4	4	2659	711403	300025
...

This information on the format above results in files sizes at 1769 MB. You may appreciate the compression possibilities if you start scrutinizing the contents. There are very many zeros for example, and many rows are very similar! With compressed information the file sizes stops at less than 1 % in file size, 16 Mb. There is a very easy conversion of the spanning tree data in Table C.1.

Appendix D: Some details on the capacity adjustment procedure using LP - LPnAdjCap

The implementation is done in the following manner.

Definitions:

$OriginalCap$ = original capacity input value

$PrevCap$ = capacity input in the latest LP model run (initiated to $OriginalCap$)

$Overload$ = $\max(AsgFlow - OriginalCap, 0)$

$SurplusCap$ = $\max(OriginalCap - AsgFlow, 0)$

$min_overload$ = min of all positive $Overload$ – values

Rules used for adjusting the capacity levels entered into RHS:

1. For all links with $Overload > 0$: $CurrentCap = \max(PrevCap - 0.5*overload, 0.5*PrevCap)$
2. For all links with $Overload = 0$ (i.e. with $SurplusCap > 0$):
 - a. If $SurplusCap < 5$: $CurrentCap = \max(OriginalCap - 0.25*min_overload, 0.5*OriginalCap)$
 - b. If $SurplusCap \geq 5$: $CurrentCap = \max(OriginalCap - 5, PrevCap)$

Appendix E: Logistics model details

The logistics model is designed to reflect firm to firm (f2f) decisions. The cost function to be minimized for each f2f-flow consists of the following components:

1. Order setup costs and inventory holding costs (both carrying costs and interest on product capital).
2. Transport costs for the OD legs in the transport chain from origin to destination, through one or more transfer points where transport mode/vehicle changeovers are made unless it is a direct transport. The transport costs are computed for entire vehicles for unconsolidated transports, and for fractions of vehicle loads for consolidated transports. Consolidation is applicable for all vehicles for which the current shipment size is below a consolidation level, endogenously determined (inside an exogenously determined interval [LBD,UBD]) per OD relation and vehicle submode¹³. Otherwise an integer number of vehicles are used only for the shipment size in question, cf the mathematical formulation in eq (E.2). System trains are never consolidated.
3. Handling costs per tonne and vehicle cost per hour for loading/unloading at transfer points. For container transport there are also stuffing and stripping costs at the origin and destination respectively.
4. Capital costs based on product values during transportation and waiting times.

This leads to the following cost expression (index details are left out):

$$C_{TOT} = AnnDem/q \times Setup + (b + i \times V) \times q / 2 + TrnspC(q, \varphi) + HandlingC(q, \varphi) + WIPC(q) \quad (E.1)$$

where

C_{TOT} = total logistics costs [SEK/year]

$AnnDem$ = annual demand [tonnes/year]

q = shipment size [tonnes]

$Setup$ = order set up cost [SEK]

b = inventory carrying cost [SEK/(tonne,year)]

i = interest rate on capital

V = product value [SEK/tonne]

φ = consolidation factor (vehicle capacity utilization ratio)

$TrnspC(q, \varphi)$ = transport cost for the transport chain that depends on q and φ [SEK/year]

¹³ A sub-mode is a number of vehicles grouped together into a “letter group”. For non-container transports, C for example often includes the two largest lorry types 104 and 105, and H typically includes the wagon load trains 207 – 209. The groups A and F respectively, represent the same vehicles when used for container transports.

$HandlingC(q, \varphi) =$ goods handling and vehicle loading/unloading costs for the transport chain are a function of q and φ [SEK/year]

$WIPC(q) =$ capital costs for goods values during transport depends on q [SEK/year]

The decisions to make for each transport are:

- d. Which transport chain to use?
- e. Which vehicle size to use in each OD-leg?
- f. What shipment size should be used (equivalent to annual demand/frequency)?

The current method for solving the above issues is to perform the following two steps in an iterative process (3 iterations) for each of the 34 product groups. The same cost function is used in both steps, but with a set of predefined input parameters (vehicles and shipment size).

BuildChain-step (BC): Determine a number of optimal transport chains, consisting of combinations of road, rail, ferry, sea and air transports, for all origins and destinations. The combinations considered are predefined. For each product group there is one typical shipment size used per origin zone (CPU-time would increase a lot for individual shipment sizes). The vehicles considered are a designated vehicle for each sub-mode defined, where the sub-modes may for example, be either a small or a large lorry. A shortest path procedure is used to determine the transfer points for each transport chain.

ChainChoi-step (CC): Here the set of optimal transport chains are evaluated at a detailed level. The optimal combination of individual vehicles and shipment frequencies are determined given the set of available transport chains. A crucial input here, since the costs incurred are vehicle-based, is the consolidation approach that results in the vehicle utilization levels. A short version of the consolidation method is provided here, see de Jong et al [2011] for further details.

In principle all rail, sea and air transports are considered to be consolidated. The only exception now is system trains. The basic idea is that for a shipment of size q the costs paid for the consolidated OD-leg transports are

$$vehicle\ cost \times q / (\varphi \times Cap) \tag{E.2}$$

where

$vehicle\ cost =$ sum of time and distance cost for the vehicle [SEK]

$Cap =$ vehicle capacity [tonnes]

In summary, the iterative process works as below with φ set to a constant value initially (e.g. 0.75), then according to potential and actual flow per OD-relation in the iterations 2 and 3 (as determined in the preceding iterations). After the iterations we obtain, for every OD-relation, estimated, consolidated flows Z .

Iteration 1: $\varphi = 0.75$ BC CC $\Rightarrow (Z, \varphi)$

Iteration 2: potential BC CC $\Rightarrow (Z, \varphi)$

Iteration 3: OD-flow BC CC $\Rightarrow (Z, \varphi)$

The potential consolidation is determined by summarizing the flow on all transfer to transfer relations (actually often (zone to transfer)/(transfer to zone)-relations also when the zone is outside Sweden). All potential transport chains are considered, i.e. both used and unused OD legs. This means that an OD leg can be included multiple times for a single producer to consumer relation. In the current version these potentials are, for any OD leg, merely used in relative terms in an ordinal ranking. All possible combinations of transfer to transfer relations per sub-mode are ranked according to total potential and allocated ϕ -values uniformly in a user defined range [LBD, UBD], e.g. [0.05, 0.95]. Before the ranking additional weights in terms of total observed flows are weighted into domestic port transfers. In the third iteration, the consolidation is based on the actual OD-flows from iteration 2. Otherwise, the ranking process is repeated with the changed base.

Selection of the optimal shipment quantities in the ChainChoi procedure are determined as described below (in principle a quote from de Jong et al [2011]): The starting point for annual delivery frequency is $AnnDem/q^W$ (rounding off to integer values). Then we generate twenty possible frequencies in the interval $AnnDem/q^W \times [0.2, 1.0]$, at uniform intervals¹⁴. For each of those 20 possible frequencies, we calculate the total logistics costs for each of the available vehicle/vessel¹⁵ type sequences for the available transport chains. From all these discrete alternatives, we select the one with the lowest total logistic costs C_{TOT} , eq (1), and use the corresponding frequency $AnnDem/q^*$ and shipment size q^* in the further calculations. Should the optimum frequency $AnnDem/q^*$ equal the lower bound of the range at $0.2 Q/q^W$, then we perform another search using twenty points in the interval $AnnDem/q^W \times [0.04, 0.2]$. Actually, since the costs for individual OD-legs are independent, the evaluations can be restricted to the choice of the optimal vehicle per OD-leg, thus avoiding the combinatorial explosion when evaluating all sequences. Furthermore, for Ro-Ro-vessels and ferries, now the same vehicle is used at both ends.

The number of vehicles required determines the transport costs as in eq (E.2). A more complete description of the number of vehicles, N , required is:

$$N = \begin{cases} \frac{q^*}{\phi \times Cap} & \text{if } q^* < \phi \times Cap \\ \lceil q^*/Cap \rceil & \text{if } q^* \geq \phi \times Cap \end{cases} \quad (E.3)$$

¹⁴ Optimal order quantity (the so called Wilson formulae): $q^W = \sqrt{\frac{2 \times Setup \times AnnDem}{(h+i \times V)}}$

¹⁵ As a matter of fact the set of vehicles is restricted by a ruling out vehicles with too large capacities as compared to the consolidated flow. Without this restriction the economies of scale would lead to selection of the vehicle/vessels with the largest capacity.

After the three iterations with the BuildChain- and ChainChoi-modules as described above the results are summarized in the so called Extract module into vehicle- and tonne-OD-flows respectively per vehicle/vessel type. In the latter module also empty flows are constructed. See Appendix B for a more detailed description of the Extract module.