Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ARIE M.C.A. KOSTER

Re-Optimization of Signaling Transfer Points

ZIB-Report 00-18 (May 2000)

Re-Optimization of Signaling Transfer Points

Arie M.C.A. Koster*

Abstract

In this paper we describe the results of a computational study towards the (re)optimization of signaling transfer points (STPs) in telecommunication networks. The best performance of an STP is achieved whenever the traffic load is evenly distributed among the internal components. Due to the continuously changing traffic pattern, the load of the components has to be re-optimized on a regular basis. Besides the balancing objective also the number of rearrangements have to be taken into account. In this paper we present two alternative formulations to deal with both requirements. Computational results show that for both formulations (near) optimal solutions can be obtained within reasonable time limits.

1 Introduction

Telecommunication companies maintain a so-called signaling network apart from their communication transport network. This signaling network is used for management tasks like basic call setup and tear down, wireless services (wireless roaming, mobile subscriber authentication), and enhanced call features (call forwarding, number display). Moreover, GSM cellular phone network providers utilize the signaling network to offer Short Message Services (SMS). In recent years, the use of this service has been increased very rapidly. The signaling network is a digital network that uses the Common Channel Signaling System No. 7 (SS7) protocol. The SS7 protocol is a global standard for telecommunications defined by the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T). The standard defines the procedures and protocol by which network elements in a communication network exchange information.

A signaling network consists of signaling points (SP), and signaling links in between them. Three different main types of SPs are distinguished: Service Switching Points (SSPs), Signaling Transfer Points (STPs), and Service Control Points (SCPs). Moreover, also Mobile Switching Centers (MSCs), Service Management System Centers (SMSCs), Home Location Registers (HLRs), and other application specific equipment are connected to the signaling network. An SSP is a switch which sends signaling messages to other SSPs for the setup and release of phone calls. In addition, maintenance requests and other service requests can

*Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Takustraße 7, D-14195 Berlin-Dahlem, Germany. E-mail: koster@zib.de. This research has been done in cooperation with E-Plus Mobilfunk GmbH.

Keywords: Telecommunications, Modeling, Integer Programming

be sent to other SPs. An SCP contains a centralized database with services like routing information for toll-free calls and alternate billing services.

Both SSPs and SCPs are connected with each other through STPs. An STP is a high speed, reliable, special purpose packet switch for signaling messages in an SS7 signaling network. It has the same function as the switches in the transport network: it serves as an intermediate between SCPs, SSPs and other network elements. Direct links between SSPs and SCPs can be avoided by the use of STPs. SCPs, SSPs, and STPs are connected with 64 kbit/s data links. Depending on the adjacent SPs, those links are classified with the characters A, B, C, D, E and F, for respectively, Access links, Bridge links, Cross links, Diagonal links, Extended links, and Fully Associated links (we omit a further explanation, see [3]).

The signaling network is critical with respect to the setup of calls in the transport network. Therefore, high performance guarantees are required. In case of an isolated failure of one of the components, the signaling tasks should still continue network-wide. For this reason, many security measures have to be fulfilled in the signaling network. STPs are usually deployed in mated pairs (at different locations), each SSP/SCP is connected to multiple STPs, and the STPs themselves are connected by multiple links as well. Moreover, inside an STP certain diversification rules are applied to guarantee that in case a part of the STP fails, a certain percentage of the signaling traffic within the STP still continues.

Another security measurement is that the load of a link should never exceed a certain percentage. The load of a link is expressed in Erlang, where 1 Erlang is equivalent to 100% use of the 64 kbit/s link. If in normal operation loads above 50% are avoided, then in case of emergency the links can receive a duplication of the load without problems (in fact, the ITU recommends a load of 20 - 40%). For the same reason the total load of internal STP components should be balanced to avoid a loss of too much traffic in case of component failures (the total load of a component is defined by cumulative load of links connected to that component).

Due to the fast growth of telecommunication traffic in recent years, also the load of signaling links grows steadily over time. Especially in GSM cellular networks the grow rate is spectacular. For signaling the figures show an even more tremendous increase due to the introduction of Short Message Service (SMS), which uses the signaling network instead of the transport network. Therefore, continuously new SPs are connected to STPs and traffic load is redistributed among the new links. The traffic load, however, does not increase equally for all links, which causes that over time the total load of the internal STP components becomes unbalanced. To re-balance those loads a reconfiguration of the assignment of the links to the components have to be carried out. A complicating factor for reconfiguration is that the reassignment of a link interrupts the operating service in the associated part of the network during the time of reconfiguration. This implies that also the actual unbalanced situation should be taken into account. The old configuration and the optimized configuration should be as similar as possible to avoid loss of quality of service for a long period.

This paper is devoted to (re)optimization of signaling transfer points. We discuss several models that take different requirements into account and present a case study for these models with real-life data of E-Plus. The paper continuous with a description of the relevant STP properties and the optimization problem in Section 2. Next, in Section 3 we present several models for the formulation of the optimization and the re-optimization of STPs. Computational results are presented in Section 4, whereas conclusions are stated in Section 5.

2 Problem Description

An STP is decomposed in clusters, which on their turn consist of routing units called Common Channel Distributors (CCDs) and interface cards called Common Channel Link Controllers (CCLKs). To establish connections every link has to be connected with a CCD as well as a CCLK. Although CCLKs and CCDs in different clusters are connected through bus-connections, a link should be connected to a CCD and a CCLK within the same cluster to avoid unnecessary internal traffic. Figure 1 shows a schematic diagram of an STP with 4 clusters, each containing two or three CCDs and 20 CCLKs. Every CCLK has a certain

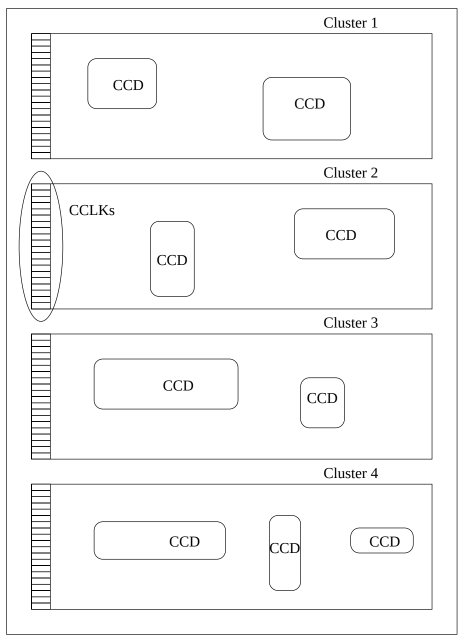

Figure 1: Schematic view of an STP

capacity, which denotes the number of links that can be connected to that CCLK. The CCLKs are continuously indexed. Every CCLK can be connected with every CCD of the same cluster through internal connections.

For every link that has to be connected to an STP a load in milli-Erlang) is given. The total set of links is partitioned in so-called *linksets* depending on their source/destination (another SP in the signaling network). To guarantee survivability of the STP in case of cluster failures, at most half of a linkset can be connected to a single cluster. Moreover, the links of a linkset assigned to the same cluster (a subset of the CCDs) have to be equalized among the odd and even indexed CCLKs.

The objective of the STP optimization problem is to balance the load of the CCDs. In other words, the load of the links has to be distributed as evenly as possible among the CCDs. The partition of links among the odd and even numbered CCLKs does not influence the objective, and therefore will not be considered in the sequel of this paper. Only the capacity of the CCLKs plays a role in the assignment of links to CCDs of a cluster, i.e., the number of links assigned to a cluster is limited.

An additional condition that should be taken into account deals with the actual situation. In fact, the optimization of STPs is a re-optimization of the assignment. However, the reassignment of links to other CCDs has to be carried out manually. During the time of reassignment the capacity of the associated part of the signaling network is reduced which can cause a loss of network service. As a consequence, the number of reassignments should be kept small. In the next section we present mathematical formulations for the optimization of STPs and alternatives to formulate this additional condition.

3 Integer Linear Programming Formulations

In this section we present four different integer linear programming formulations for the optimization and re-optimization of signaling transfer points. The first formulation is introduced by Kühn and Mayer [2] and recalled in Section 3.1. Next, we present a slightly different formulation for the same problem in Section 3.2. The advantage of this formulation is that it can be extended for the case of re-optimization of the current solution after an update of the traffic load. In Sections 3.3 and 3.4, we present two different ways to extend the model of Section 3.2 to incorporate the re-optimization aspect.

For all models, we use the following notation:

L	Set of links that have to be assigned to CCDs (in total $m = \|L\|$ links).
C	Set of CCDs that are available to assign links to (in total $n = \|C\|$ CCDs).
P	Index set of linksets.
Q	Index set of clusters.
$L_p \subset M$	Linkset $p \in P$ containing a subset of the links. Each link is in exactly one linkset, i.e., $\cup_{p \in P} L_p = L$ and $L_{p_1} \cap L_{p_2} = \emptyset$ for all $p_1 \neq p_2$.
$C_q \subset C$	Cluster $q \in Q$ containing a subset of the CCDs. Each CCD is in exactly one cluster, i.e., $\cup_{q \in Q} C_q = C$ and $C_{q_1} \cap C_{q_2} = \emptyset$ for all $q_1 \neq q_2$.
e_i	Load of link $i \in L$.
c_q	Capacity of cluster $q \in Q$. The capacity of a cluster is given by the number of CCLKs in the cluster times the number of link slots of a CCLK.

3.1 Original Formulation

In Kühn and Mayer [2] a first model is presented for the optimal balancing of the traffic load of an STP. They define the variables x_{ij}:

$$x_{ij} = \begin{cases} 1 & \text{if link } i \in L \text{ is connected with CCD } j \in C \\ 0 & \text{otherwise} \end{cases}$$

Then their integer linear programming formulation for the STP optimization model reads

$$
\begin{aligned}
\min \quad & \sum_{i \in L} e_j (x_{i1} - x_{in}) && && (1)\\
\text{s.t.} \quad & \sum_{j \in C} x_{ij} = 1 && \forall i \in L && (2)\\
& \sum_{i \in L} e_i (x_{ij} - x_{i,j+1}) \geq 0 && \forall j = 1, \dots, n-1 && (3)\\
& \sum_{i \in L_p} \sum_{j \in C_q} x_{ij} \leq \left\lceil \frac{|L_p|}{2} \right\rceil && \forall p \in P, \forall q \in Q && (4)\\
& \sum_{i \in L} \sum_{j \in C_q} x_{ij} \leq c_q && \forall q \in Q && (5)\\
& x_{ij} \in \{0, 1\} && && (6)
\end{aligned}
$$

The objective (1) computes the difference between the traffic load of the first and the last CCD. Since constraints (3) enforce that the traffic load of CCD j is at least as large as the traffic load of CCD $j+1$, the objective models the minimization of the difference between maximum and minimum traffic load of the CCDs. The condition that every link $i \in L$ has to be assigned to a single CCD is modeled by constraints (2). Inequalities (4) model the diversification of the links in a linkset L_p, $p \in P$, across the available CCDs. The number of links in a linkset L_p that can be assigned to the same cluster C_q, $q \in Q$, is restricted by half the cardinality of the linkset (rounded up for linksets of odd size). Hence, in case of the failure of a cluster, at most 50% of the links are lost. Finally, the capacity constraints (5) model that the number of links assigned to the same cluster is restricted by the number of slots available at the CCLKs in the cluster.

Note that the right-hand side of (4) slightly differs from the one in [2]. For integral solutions, it is equivalent to $(|L_p| + 1)/2$, but in the linear relaxation our right-hand side is tighter for $|L_p|$ even. Moreover, not that the constraints (5) were not modeled in [2].

3.2 Reformulated Model

The constraints (3) cause that the load of the consecutive CCDs is non-increasing in a solution that satisfies them. Assuming that CCDs in the same cluster have successive indices, the model in the previous section has the disadvantage, that the CCDs in the first cluster have a higher traffic load than the CCDs in the last cluster.

Moreover, the current situation does not satisfy (3), which implies that many connections have to be changed to obtain a load pattern that satisfies (3). As mentioned earlier, the rearrangement time of an STP is positive correlated with the number of rearrangements that should be carried out. Hence, the number of rearrangements should be kept as small as possible. The non-increasing ordering of CCD-loads enforced by (3), however, increases the number of rearrangements unnecessary. For above reasons, we modify in this section the model (1)-(6) in such a way that it does not enforce a non-increasing order of the CCD-loads anymore, without changing the objective. To do so, we introduce two additional variables

y and z, representing the maximum and minimum traffic load of the CCDs, respectively. Then the STP optimization problem reads

$$
\begin{aligned}
\min \quad & y - z & & (7)\\
\text{s.t.} \quad & \sum_{j \in C} x_{ij} = 1 & \forall i \in L \quad & (8)\\
& \sum_{i \in L} e_i x_{ij} \leq y & \forall j \in C \quad & (9)\\
& \sum_{i \in L} e_i x_{ij} \geq z & \forall j \in C \quad & (10)\\
& \sum_{i \in L_p} \sum_{j \in C_q} x_{ij} \leq \left\lceil \frac{|L_p|}{2} \right\rceil & \forall p \in P, \forall q \in Q \quad & (11)\\
& \sum_{i \in L} \sum_{j \in C_q} x_{ij} \leq c_q & \forall q \in Q \quad & (12)\\
& x_{ij} \in \{0, 1\} & & (13)
\end{aligned}
$$

The assignment constraints (8), as well as the diversification constraints (11), and the capacity constraints (12) remain the same in comparison with the previous model. The objective, however, now only deals with the difference between the maximum traffic load y and the minimum traffic load z. Constraints (9) enforce that y is at least as large as the traffic load of CCD j for all $j \in C$. The positive coefficient of y in the objective implies that y equals the maximum traffic load. In the same way the minimum traffic load z is enforced by the constraints (10) and the negative objective coefficient.

Compared to the previous model, this model does not require that the CCDs are ordered with respect to non-increasing traffic load, which gives the freedom to the optimization algorithm to find solutions with not only small difference in traffic load among the CCDs, but also a small number of changes to the current solution. A disadvantage of this model is that given a solution, we can find $m! - 1$ solutions with the same value by interchanging the CCDs (note that a small number of rearrangements is not yet enforced by the model, it only allows for those solutions). This so-called degeneration aspect makes it harder for general purpose optimization software (based on branch-and-bound) to solve the problem.

3.3 Re-Optimization with a Restricted Number of Changes

Although the model (7)- (13) allows for solutions with a small number of rearrangements of the current assignment, this *second objective* is not yet included in the formulation. Since two objectives cannot be taken into account by an integer linear program at the same time, we have to find an alternative way to formulate the second objective. An often used procedure is the following. First, the model without the second objective is solved. Next, we add the constraint that the first objective equals the optimal value, we change the objective to the second one, and we solve the new problem. This procedure will find among the solutions that are minimal on the first objective, the solution that is minimal on the second objective. This means that highest priority is given to the first objective and a lower priority to the second. In our case, however, it is not clear which objective should have a higher priority. It may be interesting to allow a minor increase of the difference between maximum and minimum traffic load if that results in a substantial decrease of the number of changes that have to be carried out. Therefore, we reject the above described procedure. In this and the next subsection, we propose two other ways to deal with the two objectives.

Instead of adding an additional objective, in this subsection we introduce an additional constraint, which right-hand side is a parameter that can be tuned. The additional constraint restricts the number of changes that may be applied to the current solution. For that reason, we define $j^*(i)$ to be the currently assigned CCD of link $i \in L$. Hence, for a new solution it holds that

$$x_{i,j^*(i)} = \begin{cases} 1 & \text{if the assignment of link } i \in L \text{ is not changed} \\ 0 & \text{otherwise} \end{cases}$$

Stated differently, $\sum_{j \in C, j \neq j^*(i)} x_{ij}$ indicates whether or not the assignment of link $i \in L$ is changed. Hence, the number of changes can be restricted with

$$\sum_{i \in L} \sum_{j \in C, j \neq j^*(i)} x_{ij} \leq B \tag{14}$$

where $B \geq 0$ is the maximum number of allowed changes to the current solution. The model (7)-(14) then minimizes the difference between maximum and minimum CCD-load with the additional constraint, that the number of changes to the current assignment is restricted by B.

3.4 Re-Optimization with a Maximum Difference Constraint

Another way to model the fact that the number of changes should be kept small, is by minimizing this number, subject to a constraint that the difference between the CCD-loads is restricted to a certain value $D \geq 0$. So, the objective now reads

$$\min \quad \sum_{i \in L} \sum_{j \in C, j \neq j^*(i)} x_{ij} \tag{15}$$

whereas the additional constraint reads

$$y - z \leq D \tag{16}$$

Then, the model (15), (8)-(13), (16) re-optimizes the STP to an assignment with a minimum of changes to the current solution in which the maximum difference in traffic load among the CCDs is at most D.

4 Computational Results

In this section, we describe the results of a case study with real-life data of one of the STPs owned by E-Plus, a German GSM cellular phone network provider. This STP consists of 5 clusters, a total of 18 CCDs, and 112 linksets with 336 links. The number of links per linkset varies from 1 to 16. In Table 1, the other characteristics of the considered STP are displayed. Moreover, Figure 2 shows a histogram of the traffic loads. Finally, the current solution with a difference of 2180 can be viewed in Figure 3.

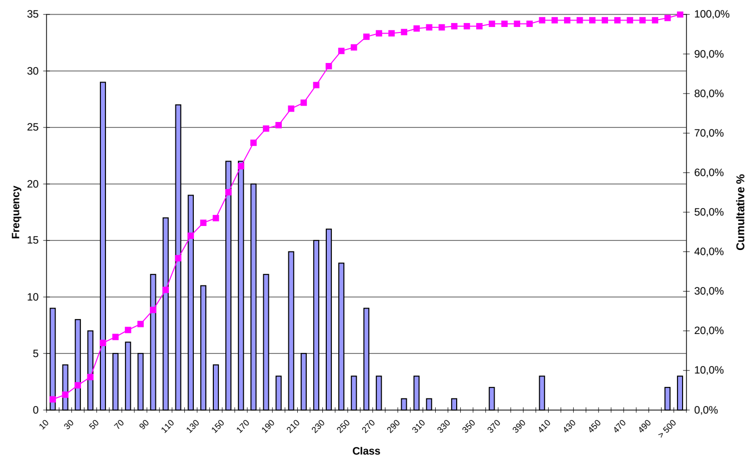

Figure 2: Histogram traffic load of the connected links.

# links n	336	# CCDs m	18
# linksets $\|P\|$	112	# clusters $\|Q\|$	5
CCDs cluster 1	0 1	capacity c_1	40
CCDs cluster 2	3 4 5 6	capacity c_2	80
CCDs cluster 3	9 10 11 12	capacity c_3	80
CCDs cluster 4	15 16 17 18	capacity c_4	80
CCDs cluster 5	21 22 23 24	capacity c_5	80

Table 1: Data characteristics case study STP.

In the subsequent subsections we present the results of respectively applying the models of the Sections 3.2, 3.3, and 3.4. The computations were carried out on 1 of the 4 CPUs of a UltraSPARC E3000 with 2.5 Gb internal memory. The integer linear programming problems were solved with the callable library of CPLEX, version 6.53 [1]. We used all standard parameter settings. Among others, this implies that CPLEX automatically determines whether it is effective to separate clique and cover inequalities. We have tried to improve the performance of the solver by changing some other parameters without success.

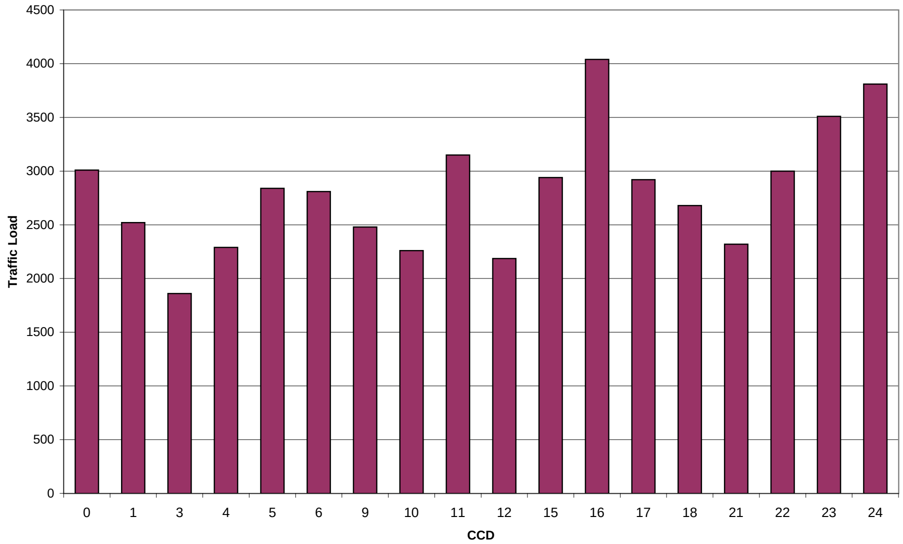

Figure 3: Current solution for the STP with value 2180.

4.1 Reformulated Model

First of all, we tried to solve the model (7)-(13) to optimality. Figure 4 shows the decrease of the upper bound during the first 3000 branch-and-bound nodes. After a solution of value 30 is found no further improvement has been obtained during the next 472,000 branch-and-bound nodes. The lower bound given by the linear programming programming relaxation equals 0 and is not improved during those 475,000 nodes of the branch-and-bound procedure. Due to memory limits (and the unpromising situation of 470,000 remaining nodes), we stopped the program at this point of time. The solution of value 30 was found after 1200 seconds of CPU time, whereas it took 58,000 seconds to investigate the 475,000 nodes.

In the solution of value 30, the number of changes to the current solution is quite large: 311 of the 336 links have to be reassigned. Note that, this does not mean that for every solution of value 30, we need to do 311 reassignments. Only this particular solution found by the algorithm, that does not take any aspect of the current solution into account, needs a large number of rearrangements. The solution is displayed in Figure 5. Note that the vertical axis starts at 2780 instead of 0 as in Figure 3.

4.2 Restricted Number of Changes

The next case we studied was the model with the number of changes restricted by B (i.e., inequality (14)). The results for $B = 0, \dots, 20$ are presented in Table 2 as well as Figures 6, and 7. Table 2 and Figure 6 show for the various values of B, lower and upper bounds

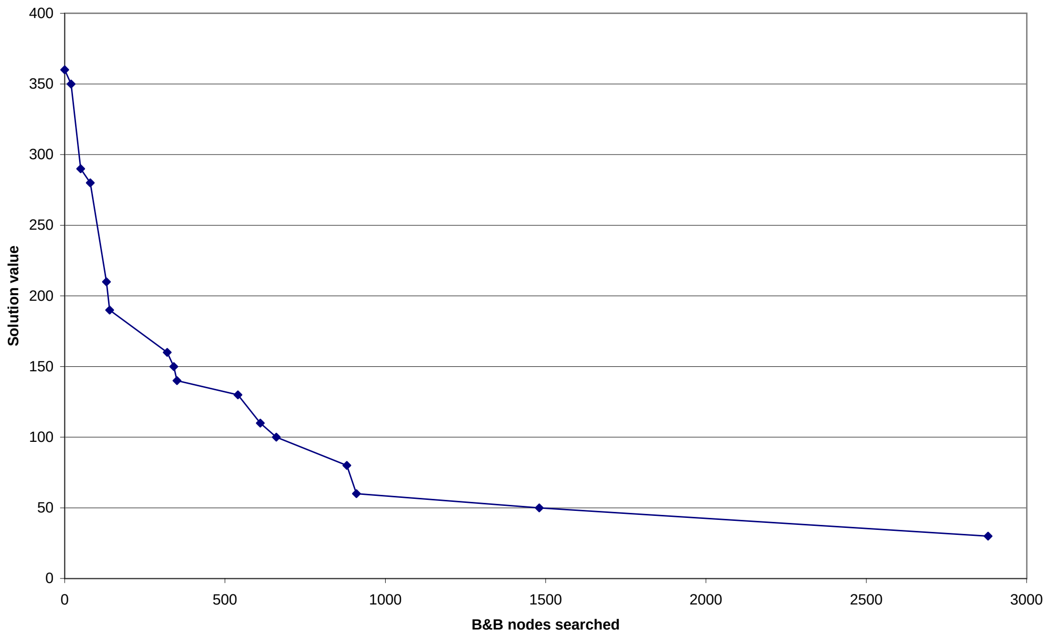

Figure 4: Solution values found by the CPLEX branch-and-bound procedure for the reformulated model.

for $y - z$ found by the branch-and-bound procedure, the value of the linear programming relaxation, and the value of the solution found by a rounding heuristic applied in the root node. Moreover, in Table 2 as well as Figure 7 the number of B&B nodes as well as the CPU time is given. The computation time was restricted to one hour per B. In case the optimal solution was found within the hour, the lower and upper bound are equal. In case the branch-and-bound algorithm could not guarantee an optimal solution within that hour, the best solution available gives the upper bound, whereas the global lower bound of the branch-and-bound procedure serves as lower bound.

In case $B = 0$, no changes are allowed and the solution should equal the current situation. However, the integer linear program turned out to be infeasible, which implies that the current situation does not satisfy all conditions enforced by the model. Investigation of this infeasibility lead to the conclusion that for 3 linksets the constraint that at most half of the links can be assigned to a cluster was violated. As a consequence, at least 3 rearrangements are necessary to obtain a feasible solution, which implies that for $B = 1, 2$ the problem is infeasible as well. Moreover, for $B = 3$ only the infeasibility is resolved in the most efficient way, resulting in a solution of 1740. For $B \leq 6$, as well as $B = 8$, the integer linear program can be solved within one hour to optimality. For $B = 7$ and $B \geq 9$, however, solving the problem will take more than a hour. The number of branch-and-bound nodes increases very rapidly with increasing B in case the problem can be solved. For larger B the number

Table 2: Computational results for the model with a restricted number of changes to current situation

B	$y - z$ lower bound	$y - z$ upper bound	LP value	Solution root node	# B&B nodes	CPU time (sec)
0	2180	2180	Current solution not valid!			
1	-	-	Infeasible			
2	-	-	Infeasible			
3	1740	1740	1740.0	1740	1	1.55
4	1550	1550	1406.3	1550	317	12.76
5	1290	1290	1221.3	1320	551	20.17
6	1160	1160	1067.0	1230	21228	523.15
7	1059	1094	934.2	1160	135731	3608.33*
8	900	900	811.7	930	25622	552.01
9	744	794	698.1	840	140339	3615.37*
10	637	710	592.4	760	185835	3636.33*
11	530	634	486.7	690	142189	3629.37*
12	408	470	383.7	480	108631	3618.39*
13	344	420	296.0	540	159520	3638.32*
14	298	330	220.8	390	121058	3610.35*
15	253	330	159.1	430	89798	3613.34*
16	164	320	101.3	380	88142	3612.33*
17	111	270	43.5	360	83291	3610.43*
18	0	246	0.0	370	44863	3604.37*
19	0	200	0.0	260	37105	3603.60*
20	0	140	0.0	230	44120	3604.39*

* Optimization procedure exceeded maximum CPU time (1 hour). Best lower and upper bound are given.

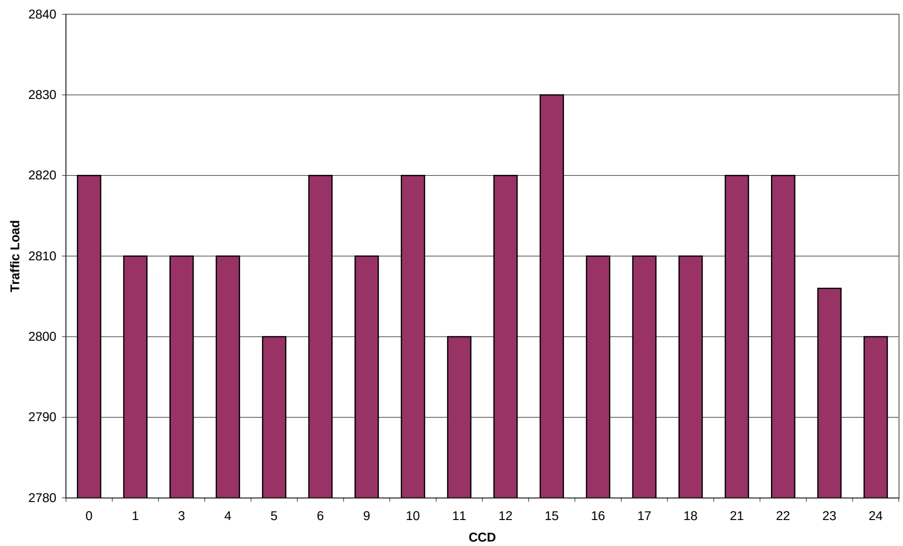

Figure 5: Solution with value 30 found by the CPLEX branch-and-bound procedure for the reformulated model.

of searched nodes decreases again which indicates that it becomes more difficult to solve the linear relaxation in every node. Finally, note that for $B \geq 18$ the linear programming relaxation becomes 0 and cannot be improved within one hour by branch-and-bound. Since, we had the same result for the model without constraint (14), this indicates that for $B \geq 18$ this constraint does not influence the linear programming relaxation anymore.

4.3 Maximum Difference Constraint

In the third part of our computational study we consider the model (15), (8)-(13), (16), in which the number of changes is minimized subject to the constraint that the difference between maximum and minimum load is restricted by a value D. We varied the value D from 2500 down to 0. In Table 3 the results for the tested values of D are given. Like in the previous section we restricted the computation time to one hour. In case the optimal solution was found the lower and upper bound equal the optimal number of changes; in case optimality could not be proved, the lower bound of the branch-and-bound as well as the best available solution can be found in the table. Moreover, the 4th and 5th column contain, respectively, the value of linear programming relaxation and the value of the solution found by the root node rounding heuristic. The last two columns contain again the number of branch-and-bound nodes searched and the CPU time. The lower and upper bounds are also displayed in Figure 8, whereas Figure 9 shows the number of branch-and-bound nodes and the CPU times.

Although, the difference in the current solution equals 2180, the infeasibility of it enforces

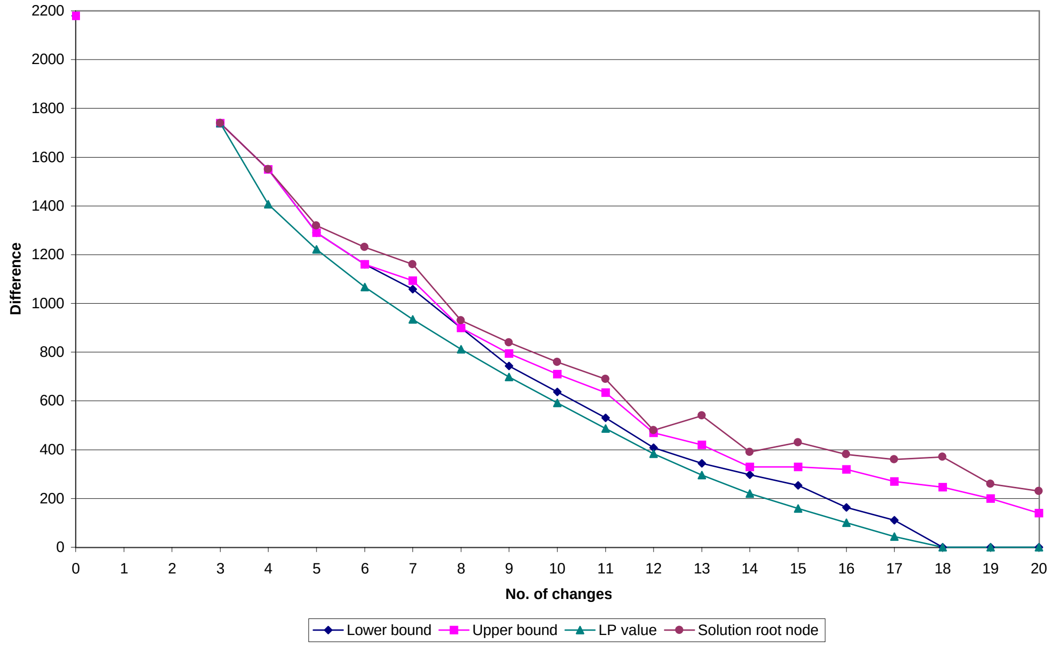

Figure 6: Solutions and lower bounds for the model with a restricted number of changes.

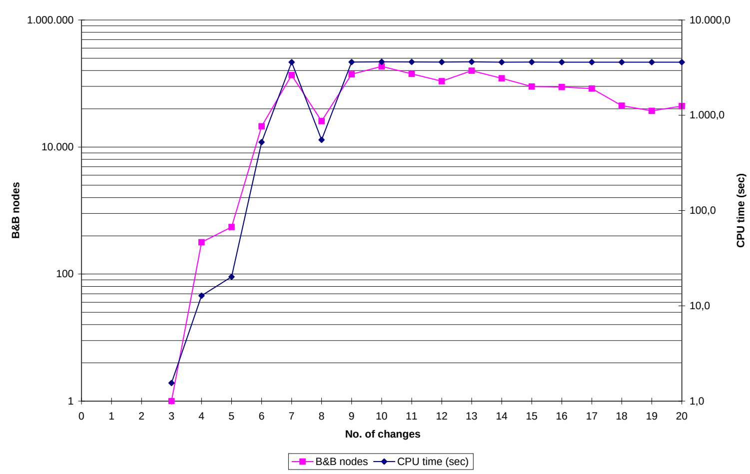

Figure 7: Number of branch-and-bound nodes as well as CPU times for the model with a restricted number of changes.

Table 3: Computational results for the model with a maximum difference constraint

D	# changes		LP	Solution	# B&B	CPU time
	lower bound	upper bound	value	root node	nodes	(sec)
2500	3	3	3.00	3	1	0.5
2000	3	3	3.00	3	1	0.9
1900	3	3	3.00	3	1	1.7
1800	3	3	3.00	3	1	1.5
1700	4	4	3.07	4	1	1.6
1600	4	4	3.25	4	1	1.6
1500	5	5	3.55	5	14	2.5
1400	5	5	4.03	5	1	1.1
1300	5	5	4.56	6	91	4.5
1200	6	6	5.13	7	7550	103.8
1100	7	7	5.75	7	17	2.5
1000	8	8	6.51	8	22	3.8
900	8	8	7.26	9	600	17.9
800	9	9	8.10	10	51490	800.2
700	9.6	11	8.98	-	197913	3614.3*
600	10.2	12	9.93	14	243675	3673.3*
500	12	12	10.87	15	1867	53.2
400	12.2	14	11.83	16	191100	3644.4*
300	13.9	15	12.95	-	174000	3600.0*
200	15.3	18	14.31	19	142694	3617.3*
100	17.0	21	16.02	-	114142	3611.4*
50	18.2	24	16.89	-	102893	3612.3*
20	18.5	-	17.41	-	72683	3607.3*
10	18.7	-	17.58	-	70227	3607.3*
0	18.9	-	17.75	-	67285	3607.3*

* Optimization procedure exceeded maximum CPU time (1 hour). Best lower and upper bound are given.

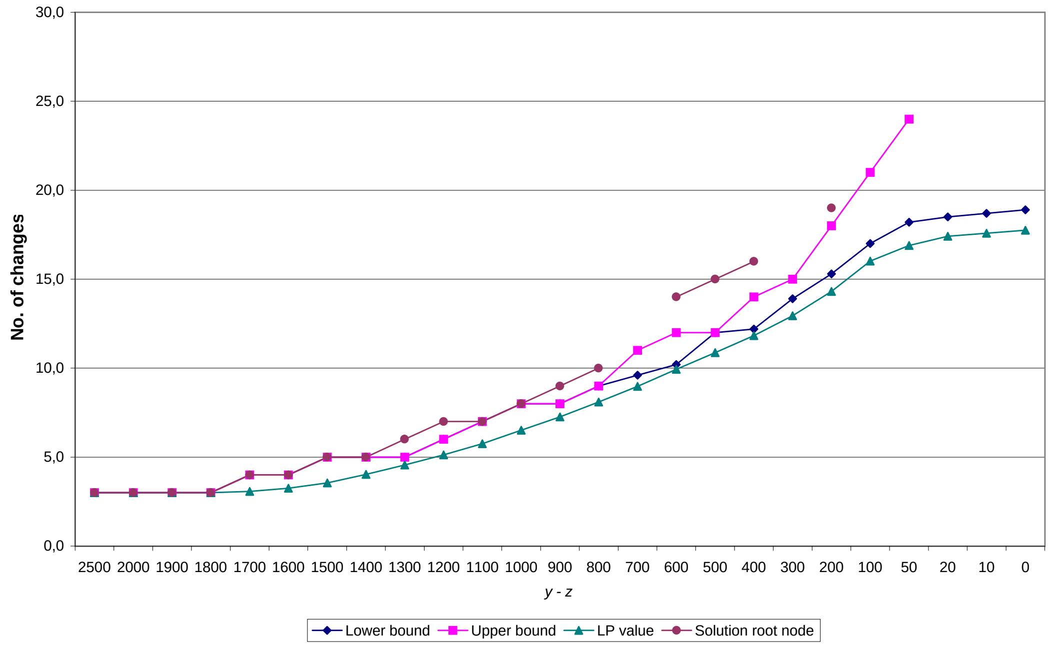

Figure 8: Solutions and lower bounds for the model with a maximum difference constraint.

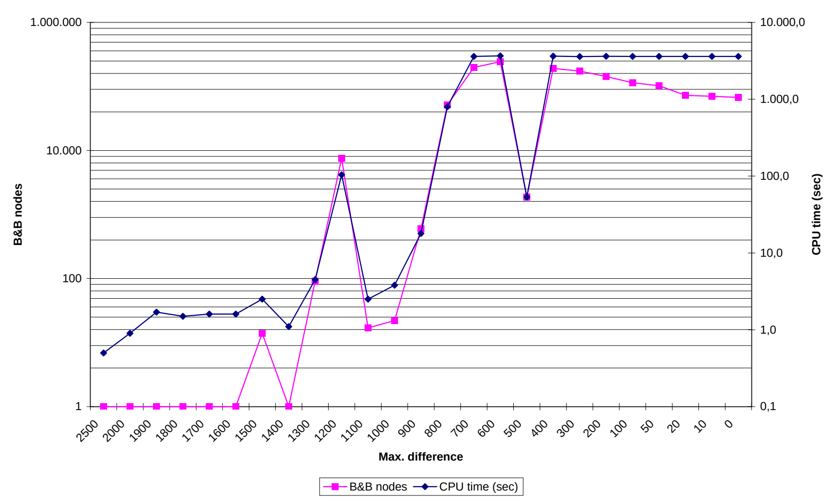

Figure 9: Number of branch-and-bound nodes and CPU times for the model with a maximum difference constraint.

that 3 reassignments are necessary also in the case of $D = 2500$. For $D \geq 1800$ (in fact $D \geq 1740$, cf. Table 2) only those rearrangements suffices to obtain a feasible solution. For smaller D the number of rearrangements needed increases steadily. For $D \geq 900$, in all but one cases the branch-and-bound tree remains decent small. For smaller D the number of nodes increases rapidly with an exception for $D = 500$. In that case the difference between linear relaxation value and the optimal value is remarkable small and the optimal solution was found quite fast. Finally, for small enough D it becomes more and more difficult to found a solution in the root node with the rounding heuristic ($D \leq 300$) as well as in the searched part of the tree ($D \leq 20$). Since, from the load traffic data characteristics we already can conclude that the difference is $D \geq 10$ for an integral solution, it is not a surprise that for $D = 0$ no solution could be found within one hour of computation.

To conclude this section, Figure 10 shows the solution with $D = 500$ in which 12 changes to the current situation (cf. Figure 3) are needed. Moreover, also the solution with $B = 8$ and value 900 is displayed (cf. Table 2).

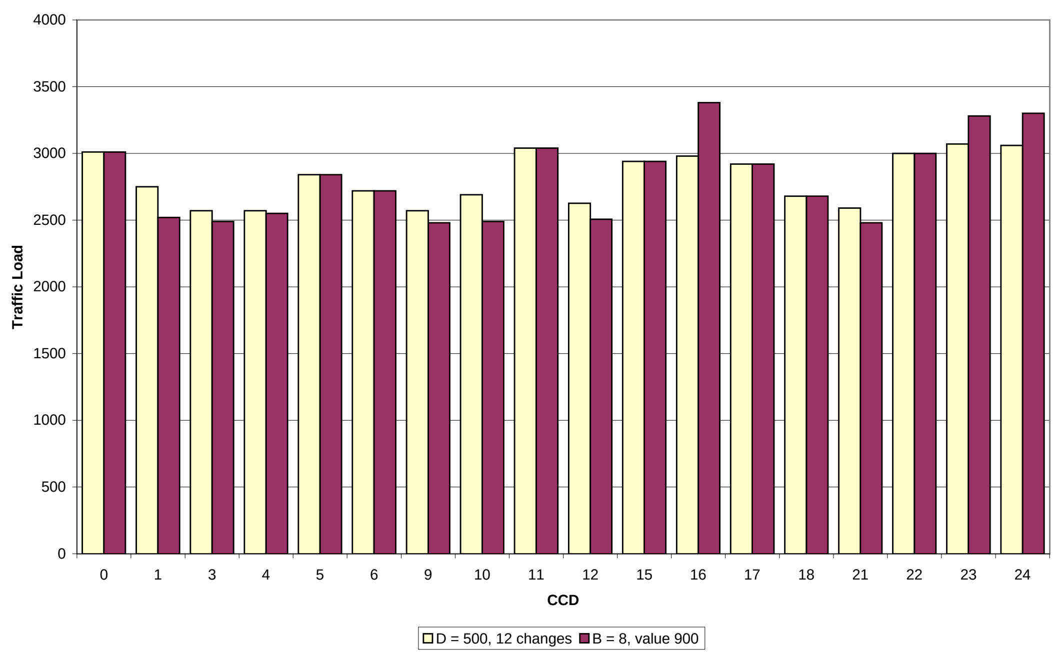

Figure 10: Solutions with value $D = 500$ (12 changes) and $B = 8$ (value 900).

5 Conclusions

In this paper we described a case study to the STP (re)optimization problem. First of all, we presented a formulation of the STP optimization problem as an integer linear program.

Next, we formulated two alternatives for the re-optimization of an STP, given an actual assignment of links to CCDs. On the one hand, we added a constraint in which a parameter restricts the number of changes to the current situation. On the other hand, a new objective is introduced that minimizes the number of changes and a restriction on the difference between the CCD loads is added.

For both cases, we did a computational case study with the data of one of the STPs of E-Plus. The computational results show that for small values for the maximum number of changes and for large values for the maximum difference, respectively, an optimal solution can be found by state-of-the-art integer programming software. However, whenever we allow a larger number of changes or we restrict the difference by a smaller value, the problem becomes more difficult. As a consequence, after certain breakpoint values CPLEX cannot guarantee the optimality of the solution anymore within a reasonable time limit. Most times the solutions found by the branch-and-bound procedure seem to be of high quality. Unfortunately, for small values of the maximum difference parameter, no solution can be found anymore by the integer programming software within an hour of computation.

In conclusion, the computation of solutions for a number of parameter values for both models can be an effective decision support system. From the practical point of view the absence of optimality guarantees should not fear, since they have a limited durability by the continuously changing traffic loads patterns. Within an hour of computation the load difference can be reduced by more than 97% (from 2180 to 50) with 24 changes to the current assignment. Alternatively, with 20 reassignments the difference can be reduced to 140 (or 93.6%).

Acknowledgment

Thanks to Roland Wessäly (Konrad-Zuse-Zentrum für Informationstechnik Berlin - ZIB) for bringing my attention to this problem, as well as to Randolf Wallbaum (E-Plus) for his clear explanation of the problem and the provision of the data for the case study.

References

[1] CPLEX division of ILOG. CPLEX callable library, version 6.53, 1999.

[2] P.J. Kühn and S. Mayer. STP-Optimierungsproblem. Technical report, IND, Universität Stuttgart, October 1999. Forschungsbericht im Rahmen des Kooperationsprojekts von E-Plus und der Universität Stuttgart, IND. In German.

[3] MicorLegend. SS7 Tutorial. web-site: http://www.microlegend.com/whatss7.shtml, April 2000.