AMO - Advanced Modeling and Optimization, Volume 10, Number 1, 2008

Optimization Models for Routing in Switching Networks of Clos Type with Many Stages

Kaj Holmberg

Email: kahol@mai.liu.se
Department of Mathematics
Linköping Institute of Technology
SE-581 83 Linköping, Sweden

December 17, 2007

Abstract

We present optimization models for the problem of simultaneous routing of connections through a symmetric Clos network, and for the problem of minimal rerouting of previously routed connections when a new connection is to be routed. The models can be used as base for solution methods, such as heuristics for rerouting combined with Lagrangean relaxation. These approaches can together give bounds on the optimal number of rearrangements needed. This is done for Clos networks with three stages, five stages, seven stages, and for an arbitrary number of stages.

Key words: *Telecommunication switches, Clos networks, routing, rearrangement*

AMO - Advanced Modeling and Optimization. An Electronic International Journal.
ISSN: 1841-4311

1 Introduction

In order to be able to route a large number of connections through a switch, it is usual to couple a number of smaller switches together into a network of special structure, so called Clos networks, [Clos, 1953]. The question then is how to route the connections through this special network. We distinguish between two situations, one where a number of connections simultaneously will be routed through the network, and another where there already is a number of connections routed through the network, and we wish to route an additional connection. In the second situation, it is sometimes possible to route the new connection without changing any of the already existing ones. However, sometimes one needs to change the routing of the already routed connections. In this case, we wish to change, i.e. rearrange, as few as possible of the existing connections.

The first of these situations is studied fairly much, while the second is less so. Furthermore, these problems are studied for three stage Clos networks, but much less so for five stage Clos networks and more stages. Heuristics for simultaneous routing in three stage Clos networks can be found in [Hwang, 1983], [Jajszczyk, 1985], [Gordon and Srikanthan, 1990], [Carpinelli and Oruç, 1993], [Franaszek, Georgiou, and Li, 1995], [Lee, Hwang, and Carpinelli, 1996], and [Hwang, 1997]. This means that there is no shortage of proposed heuristic methods for simultaneous routing. Unfortunately, few of the proposed algorithms have proved convergence, and there exists published comments to several of them, claiming that the method doesn't always work. We conclude that the theoretical/mathematical base for these methods is weak, and would like to contribute with a more stable theoretical base. Furthermore, is is hard, in some cases impossible, to extend these methods to more than three stages.

Our main goal with this paper is to construct mathematical models for these problems, models that should enable optimization approaches. The models need to be correct and solvable, and we will to some extent discuss solutions approaches.

We will formulate mathematical models for symmetrical switching networks of Clos type [Clos, 1953] with three stages, five stage, and so on, up to an arbitrary number of stages. This will be done first for simultaneous routing, and then for the more interesting case of minimal rearranging of existing connections in order to route additional connections. We will mention the usage of heuristics and Lagrangean relaxation for the model of minimal rearranging. A constructive, primal heuristic may produce a feasible solution to the problem, and thus an upper bound on the number of rearrangements needed. Lagrangean relaxation and subgradient optimization can then be used to find lower bounds on this number, so that we can get an estimate of how close to the optimum the obtained solution is.

The main contribution of this paper is the mathematical treatment of Clos networks of more than three stages. We will, however, start with a thorough treatment of the three stage case, mainly in order to try to find structures and principles that can be extended to higher numbers of stages.

2 Three stages

2.1 Symmetric Clos networks

The basic building block is a square crossbar switch with n inputs and n outputs. Any input can be connected to any output. Each input can be connected to at most one output, and although it might be technically possible to connect several inputs to the same output, we assume that each output is connected to at most one input. The switch can thus simultane-

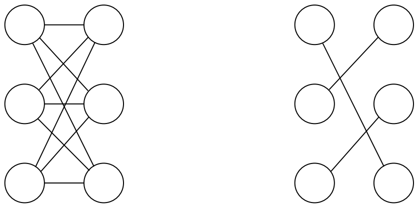

Figure 1: One switch.

ously make n connections between inputs and outputs, and any one-to-one pattern can be realized.

Mathematically, one could say that this kind of switch corresponds to an assignment problem, where n inputs are assigned to n outputs. In figure 1, the case when $n = 3$ is illustrated; in the left figure all possible connections are drawn, in the right a certain feasible set of connections is drawn.

Switches are usually organized in stages, where each stage is a column of switches, and each output of a switch in a stage is connected to an input of a switch in the next stage.

A connection or “call” is a request to connect a certain (left-most) input to a certain (right-most) output. Such a connection needs to be assigned a path through the network in order for the request to be satisfied. The path determines what switches and what inputs/outputs the connection will use. One switch in each stage needs to be used.

Often there are already a number of connections assigned through the network, and sometimes a new request cannot be satisfied. We assume that a new request involves an unused left-most input and an unused right-most output. Otherwise it is rejected for obvious reasons.

Nevertheless, there are situations where no path for a requested connection can be found. In such a case, the network is called “blocking”. If this never happens, the network is called “nonblocking”. In a “strict-sense nonblocking” network, no existing connection need to be rerouted, but in a “rearrangeably nonblocking” network, some existing connections may need to be rerouted in order to allow the new one to be set up.

In [Clos, 1953] so called Clos networks were first studied. There are three stages, the first one with r $n \times m$ switches, the second one with m $r \times r$ switches and the third one with r $m \times n$ switches. Thus there are nr different inputs and the same number of outputs. If $m \geq 2n - 1$, one can show that a Clos network is strict-sense nonblocking, [Clos, 1953]. According to the Slepian-Duguid theorem, [Beneš, 1965], a Clos network is rearrangeably nonblocking if $m \geq n$.

One can extend Clos networks by replacing the center stage by three stage Clos networks, thereby obtaining a five stage network. This can be repeated in order to obtain higher numbers of stages.

Rearrangeably nonblocking Clos networks with $m = n = 2$ are called Beneš networks, [Beneš, 1965]. If there are n inputs, there will be $\log_2 n - 1$ stages with $n/2$ switches. The total number of switches will then be $n \log_2 n - n/2$.

In this paper we focus on symmetrical Clos networks based on symmetrical and identical $n \times n$ switches. Compared to general Clos networks, we have $r = m = n$, so such a network is rearrangeably nonblocking, but not strict-sense nonblocking.

The switches are arranged in a symmetrical three stage Clos network as follows. In the first stage, there are n switches, yielding n^2 inputs, and in the third stage there are n switches, yielding n^2 outputs. Between these stages, there is an intermediate stage, also with n switches.

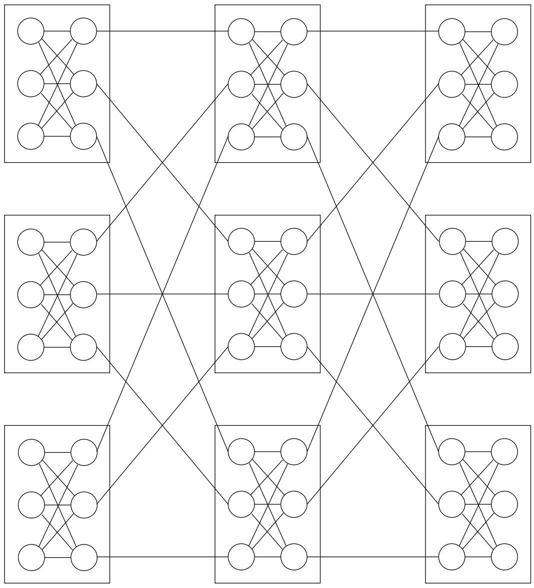

Figure 2: 3-stage switch.

The interconnections between the stages are fixed as follows. The first output of each switch in the first stage is coupled to the first switch in the second stage. The second output of each switch in the first stage is coupled to the second switch in the second stage, and so on. The outputs from the first switch in the first stage is coupled to the first input in each switch in the second stage. The outputs from the second switch in the first stage is coupled to the second input in each switch in the second stage, and so on.

The second and third stages are coupled in exactly the same way as the first and second. This is pictured for $n = 3$ in figure 2. There are never two connections between one pair of switches. This arrangement can accommodate n^2 connections as discussed above.

2.2 Mathematical model

We now wish to make a mathematical model of the possible ways of connecting the inputs to the outputs. We consider a situation where m specific connections are requested. For connection l there is a specific input, the origin, o_l, and a specific output, the destination, d_l.

One way of looking at the problem is as a multicommodity network flow problem. The different connections are modeled as different commodities of flow, where we wish to send one unit of flow of commodity l from the origin, o_l, to the destination, d_l. The capacity on each arc is equal to one. The network has the structure pictured in figure 2.

A general mathematical model for this multicommodity network flow problem is to find a feasible solution to the constraints below.

$$\sum_{j:(j,i)\in A} x^l_{ji} - \sum_{j:(i,j)\in A} x^l_{ij} = \begin{cases} 1 & \text{if } i = o_l \\ -1 & \text{if } i = d_l \\ 0 & \text{otherwise} \end{cases} \qquad \forall i\ \forall l$$

$$\sum_{l=1}^{m} x^l_{ij} \le 1 \qquad \forall i,j$$

$$x^l_{ij} \in \{0,1\} \qquad \forall i,j,l$$

The network has $6n^2$ nodes, $3n^3 + 2n^2$ arcs and n^2 commodities. ($3n^3$ of the arcs lie inside the switches.) One might believe that the arcs between the stages can be eliminated, since there is only one possibility between a pair of switches. However, there is a very important upper bound of one on the total flow in these arcs, so this elimination cannot be done. The problem thus has $3n^5 + 2n^4$ variables. For $n = 20$ this is 1600 nodes, 24000 arcs, 400 commodities and 9 920 000 variables.

The LP-relaxation of this feasible set is obtained by replacing $x^l_{ij} \in \{0,1\}$ by $x^l_{ij} \ge 0$. The constraint matrix of a multicommodity network flow problem is in general not totally unimodular, so the extreme points of the LP-relaxation are not necessarily integer. Thus we need to keep the integrality requirements. The integer multicommodity network flow problem is *NP*-complete. However, in practice, these problems often are fairly easy to solve.

Here the network is represented by the arc list A. Unfortunately the special structure of the network is somewhat hidden in this formulation.

While this is a possible way of solving the problem, it is probably not very efficient, due to the special structure of the network. Therefore, we will look for a more compact model.

Consider a specific connection, l, with input o_l and output d_l. We assume that $0 \le o_l \le n^2 - 1$ and $0 \le d_l \le n^2 - 1$. In each stage, we must decide which switch, which input and which output the connection shall use. Let us therefore define the following parameters. Connection l will in stage t use switch k^t_l, input i^t_l, and output j^t_l. Letting $N = \{0, \ldots, n-1\}$, we have $k^t_l \in N$, $i^t_l \in N$, and $j^t_l \in N$.

Knowing o_l and d_l, we can calculate what switches in the first and third stage that will be used. If $0 \le o_l \le n-1$, the first switch in the first stage will be used. If $n \le o_l \le 2n-1$, the second switch in the first stage will be used, and so on. We simply get $k^1_l = \lfloor o_l/n \rfloor$. Similarly, we get $k^3_l = \lfloor d_l/n \rfloor$. Furthermore we have $i^1_l = o_l - k^1_l n$, and $j^3_l = d_l - k^3_l n$.

According to the fixed couplings between the first and the second stage, the choice of output on a switch at stage one exactly determines the choice of switch at stage two. We simply have $k^2_l = j^1_l$. Similarly, the choice of input on a switch at stage three exactly determines the choice of switch at stage two. We have $k^2_l = i^3_l$. For the same reasons, we also have $i^2_l = k^1_l$ and $j^2_l = k^3_l$. Thus the choice to be made is which switch in stage two to use, i.e. k^2_l. Everything else follows uniquely. Let us sum this up.

Connection l starts at o_l and ends at d_l. It will use the following switches and inputs/outputs. We denote k^2_l by v in order to emphasize that it is the only variable.

Stage	Switch	Input	Output
1	$k^1_l = \lfloor o_l/n \rfloor$	$i^1_l = o_l - k^1_l n$	$j^1_l = v$
2	$k^2_l = v$	$i^2_l = k^1_l$	$j^2_l = k^3_l$
3	$k^3_l = \lfloor d_l/n \rfloor$	$i^3_l = v$	$j^3_l = j_l - k^3_l n$

Thus k^2_l is the only choice we have to make for connection l. All the other parameters are determined by this choice. Another way of illustrating this is shown in figure 3. Solid circles indicate parameters that are directly calculated from o_l and d_l and solid lines indicate the dependencies. Dashed circles indicate parameters not fixed and dashed lines indicate dependencies between parameters not fixed.

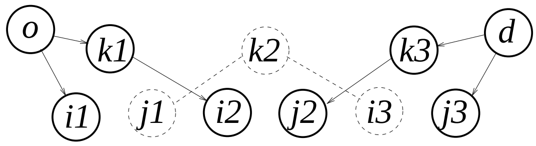

Figure 3: Dependencies between parameters.

In the mathematical model, we introduce the following variables.

$x_{il} = 1$ if connection l uses switch i (in the second stage).

(The relation to the previously used notation is that $x_{il} = 1$ for $i = k_l^2$, while $x_{il} = 0$ for all $i \neq k_l^2$.)

For each switch, each input can be used by at most one connection, and each output can be used by at most one connection. We will assume that the overall inputs, o_l, and outputs, d_l, obey this, i.e. that all o_l are different and all d_l are different. This takes care of i_l^1 and j_l^3. However, we have to ensure that this is true also for j_l^1, i_l^2, j_l^2 and i_l^3. Since $j_l^1 = i_l^3 = k_l^2$, we must ensure that the connections that use the same switch in the first stage or in the third stage do not use the same switch in the second stage. Because of this, we introduce the following sets.

$$L_k^1 = \{l : k_l^1 = k\} \qquad \text{and} \qquad L_k^3 = \{l : k_l^3 = k\}$$

This means that L_k^1 is the set of connections that use switch k in the first stage, and L_k^3 is the set of connections that use switch k in the third stage. Note that since we can calculate k_l^1 and k_l^3 from o_l and d_l, these sets are given by the indata.

One may note that $L_k^1 \cap L_{k'}^1 = \emptyset$ for all $k \neq k'$ and $\bigcup_k L_k^1 = \{1, \dots, m\}$, since each connection uses exactly one switch in the first stage. Similarly $L_k^3 \cap L_{k'}^3 = \emptyset$ for all $k \neq k'$ and $\bigcup_k L_k^3 = \{1, \dots, m\}$, since each connection uses exactly one switch in the third stage.

Let us also, for future use, introduce the sets

$$L_k^2 = \{l : k_l^2 = k\},$$

i.e. L_k^2 is the set of connections that use switch k in the second stage. These sets are, as opposed to L_k^1 and L_k^3, not given by indata, but a way of representing a solution. Again $L_k^2 \cap L_{k'}^2 = \emptyset$ for all $k \neq k'$ and $\bigcup_k L_k^2 = \{1, \dots, m\}$.

Now the constraint

$$\sum_{l \in L_k^1} x_{il} \leq 1$$

means that there is at most one connection that uses switch k in the first stage and switch i in the second stage. This ensures both that output i on switch k in the first stage is used at most once, and that input k on switch i in the second stage is used at most once. In other words it takes care of j_l^1 and i_l^2.

Similarly, constraint

$$\sum_{l \in L_k^3} x_{il} \leq 1$$

means that there is at most one connection that uses switch k in the third stage and switch i in the second stage. This ensures both that input i on switch k in the third stage is used at most once, and that output k on switch i in the second stage is used at most once. So it takes care of j_l^2 and i_l^3.

We know that there is a feasible solution if $m \leq n^2$, while if $m > n^2$ there is none, so we assume that $m \leq n^2$, and consider the question of how to route all the connections simultaneously. We must ensure that each connection is created, i.e. that each connection use exactly one switch in the second stage.

$$\sum_{i\in N} x_{il} = 1 \qquad l = 1,\dots m$$

Thus a feasible solution to the following model tells us how to route the connections.

$$\begin{aligned}
\sum_{i\in N} x_{il} &= 1 \qquad l = 1,\dots m \qquad (1.1)\\
\sum_{l\in L_k^1} x_{il} &\leq 1 \qquad i\in N, k\in N \quad (1.2)\\
\sum_{l\in L_k^3} x_{il} &\leq 1 \qquad i\in N, k\in N \quad (1.3)\\
x_{il} &\in \{0,1\} \quad \forall i,l
\end{aligned} \qquad [\mathrm{P1}]$$

This model has mn variables and $2n^2+m$ constraints (not counting the binary requirements). For $n = 20$ and $m = n^2$, this is 8000 variables and 1200 constraints. In practice it is not very difficult to solve for a general MIP-code, such as the free `lp_solve` or the more efficient `CPLEX`. One may note that P1 is much smaller (less variables and constraints) than the multicommodity flow model.

An interesting question is if the integrality requirements really are necessary. Considering the LP-relaxation where $x_{il} \in \{0,1\}$ is replaced by $x_{il} \geq 0$, the question is if all the extreme points of the feasible set are integral. If this was true, P1 could be solved as an LP-problem.

We have so far not been able to prove that the constraint matrix is totally unimodular, which would imply that all extreme points are integer. However, we have solved numerous instances and in none of them did the LP-relaxation give a non-integral solution.

By inserting slack variables, a set partitioning problem is obtained. The feasible set of the LP-relaxation is a polytope, and in [Balas and Padberg, 1972], it is shown that the set partitioning polytope is quasi-integral. This means that any edge of the convex hull of the feasible integer points is also an edge of the polytope. For such an optimization problem, there exists a sequence of adjacent extreme points that are all integral, leading up to the integer optimum. This property must be seen as theoretical, since no one has yet been able to exploit it successfully in a practical solution method. However, in practice the property seems to imply that the probability of getting an integer solution when solving the LP-relaxation is very high. (This is the experience for the uncapacitated facility location problem, the uncapacitated network design problem, and other similar problems with quasi-integral polytopes.)

Let us consider the LP-relaxation of P1 further. An extreme point of the polyhedron can be represented as a basic solution, with $2n^2 + m$ basic variables. A basic solution is called degenerate if one or more of the basic variables have the value zero. A non-degenerate basic solution would need $2n^2 + m$ variables having strictly positive values. Let us assume that all variables that do not have the value zero have the value one. (This is the case in all the extreme points we have encountered in our computational tests.) Due to constraints (1.1) there will be exactly m variables equal to one (one for each connection), and the rest will be equal to zero.

There are $2n^2$ inequality constraints that may or may not be active, i.e. slack variables that may be basic variables. Connection l must pass exactly one switch in stage 1, namely switch k_l^1, so exactly one of constraints (1.2) will be active. It must also pass exactly one switch in stage 3, namely switch k_l^3, so exactly one of constraints (1.3) will be active. Thus $2m$ of these constraints will be active, and $2n^2 - 2m$ constraints will be inactive, i.e. have slack variables with positive values. Together this yields $2n^2 - m$ variables that are not equal to zero.

We conclude that $2m$ $(= (2n^2 + m) - (2n^2 - m))$ of the basic variables will be equal

l	o_l	d_l	k_l^1	k_l^3	k_l^2	Path
1	0	1	0	0	0	0-0, 0-0, 0-1
2	1	4	0	1	1	1-1, 3-4, 4-4
3	2	2	0	0	2	2-2, 6-6, 2-2
4	3	7	1	2	2	3-5, 7-8, 8-7
5	6	0	2	0	1	6-7, 5-3, 1-0
6	7	5	2	1	0	7-6, 2-1, 3-5

Table 1: A small example.

to zero, so the basic solutions are all massively degenerate. (This seems to be a frequent property of problems with quasi-integral feasible sets.)

If $m = n^2$, then P1 has n^3 variables and $3n^2$ constraints, and *all* constraints are active, i.e. all slack variables are equal to zero. Thus in this case, n^2 of the $3n^2$ basic variables will have positive values, so $2n^2$ basic variables will be equal to zero.

From a practical point of view, it is clear that the solution to P1 is never unique. Obviously there are *many* different ways of satisfying a certain number of demand requests. Simply exchanging two switches yields another solution which in practice is equivalent to the first one. So not only are there many different basic solutions representing the same actual solution, but there are also many different actual solutions that are equivalent (for P1).

2.3 Example

Let us give a small example for the case where $n = 3$. Assume that $m = 6$, and the origins and destinations are given in table 1. The table also gives a solution obtained by solving P1, in the form of k_l^2. The solution can be expanded into a full path for each connection. The paths are illustrated in figure 4.

In this example we have $L_0^1 = \{1, 2, 3\}$, $L_1^1 = \{4\}$, $L_2^1 = \{5, 6\}$, $L_0^3 = \{1, 3, 5\}$, $L_1^3 = \{2, 6\}$, $L_2^3 = \{4\}$. The x-solution is

$$x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix},$$

so we get $L_0^2 = \{1, 6\}$, $L_1^2 = \{2, 5\}$, $L_2^2 = \{3, 4\}$.

2.4 Matchings and edge colorings

In a companion paper, [Holmberg, 2007b], we discuss P1 in graph terms, and interpret it as matchings or edge colorings. This interpretation of the problem is well known, see for example [Hwang, 1983]. We will not go into details about this, but refer to [Holmberg, 2007b]. Here we only summarize some of the results, the first of which is the following. *Any feasible solution to P1 corresponds to the union of a number of matchings, one for each switch in the second stage.*

Minimizing the number of switches used means minimizing the number of matchings, which leads to the problem of finding an edge coloring of a bipartite graph with minimal number of colors. The number of colors will be equal to the maximal degree of a node in the graph, [Berge, 1973], which will be equal to $d = \max(\max_k |L_k^1|, \max_k |L_k^3|)$.

In [Cole and Hopcroft, 1982] it is shown how to find a minimal edge coloring in a bipartite graph in $O(|E| \log |V|)$. More recent methods yield $O(|E|d)$ in [Schrijver, 1999] and

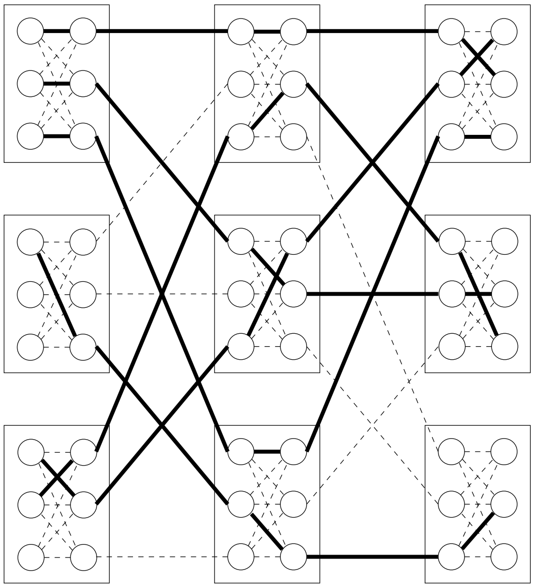

Figure 4: Connections in 3-stage switch.

$O(|E| \log d)$ in [Cole, Ost, and Schirra, 2001]. In our graph $|E| = m$ and $|V| = 2n$, which yields $O(m \log n)$, $O(md)$ and $O(m \log d)$. Furthermore $d \leq n$, so $O(md) = O(mn)$ and $O(m \log d) = O(m \log n)$. Assuming that $m = O(n^2)$, the complexity of the edge coloring method is $O(n^2 \log n)$. This proves that *simultaneous routing through a three stage network can be done in polynomial time.*

In other words, a solution to P1 is obtainable in polynomial time. One may note that here we have added the objective of minimizing the number of used switches.

2.5 Heuristic rearrangements

If we are given m connection requirements, i.e. the pairs (o_l, d_l) for $l = 1, \ldots, m$, (where $m \leq n^2$), at the same time, the routing could be established by simply solving P1. However, in practice this is not the most likely situation. Instead the requirements for connections would probably turn up one (or a few) at a time. Thus the most important situation from a practical point of view would be that there is a number of connections already routed, and one or several new requirements turn up.

We now assume that there are already $m - 1$ connections set up, and a new one is requested. We assume that $m \leq n^2$, since otherwise it will be impossible to route another connection. At this stage, we know that if all routings are removed and P1 solved, there will be a feasible solution. However, that could mean changing many of the existing routings, which would be burdensome. The goal in such a case is to satisfy the new request with as small changes as possible to the existing routings.

Let $\bar{x}$ be the solution of P1 without connection m. Then if $\sum_{l \in L_k^1} \bar{x}_{il} = 0$, then it is possible to route a new connection from switch k in the first stage to switch i in the second level (without rearranging any connections). Furthermore, if $\sum_{l \in L_k^3} \bar{x}_{il} = 0$, then it is possible to route a new connection from switch i in the second stage to switch k in the third level (without rearranging any connections).

If there is no i that allows a path from k_m^1 to k_m^3, one or more existing connections must be rerouted, in order to accommodate the new connection requirement. If the first part of the path is free, the connection using the second part of the path must be rerouted. This connection is given by $L_i^2 \cap L_k^3$.

This can be developed into a heuristic search for free paths. Route the new connection on a path that is almost free, find the connection that blocks the path, and try to reroute it. Repeat this until a free path is found. For more details, see [Holmberg, 2007a].

We may compare to heuristics such as the one described in [Lee et al., 1996], we find that the following. There one works with partly infeasible solutions, where paths may end up at the wrong switch in stage 3. One then applies a swap heuristic that iteratively corrects this. Thus our heuristic can be put into the same framework. A fairly elaborate scheme seems necessary in order to avoid cycling, see [Carpinelli and Oruç, 1993]. (A number of proposed methods of this type have subsequently been shown to fail in certain circumstances.) Our conclusion is that we may adopt the same strategies as is used in for example [Lee et al., 1996], in order to make the heuristic converge.

These methods are probably quite quick, but are not guaranteed to give the minimal number of reroutings.

2.6 Minimal rearrangements

Since heuristics may fail yielding the minimal number of reroutings, one might want to use an approach that is certain to succeed. Therefore we formulate a mathematical model that, based on a given solution and new connection requirements, finds a feasible solution with a minimum of rerouting.

Let us generalize somewhat by allowing more than one new connection. Let $l \in C_O$ denote the "old" connections, already routed, and $l \in C_N$ the new connections, to be routed. One might allow $|C_O| + |C_N| > n^2$ and decide which of the new ones to route, based on the number of reroutings needed. However, here we will not do that, since it requires a comparison of the disadvantages of not routing different connections. Also one might then wish to compare this to the disadvantages of rerouting. Instead, we will here assume that $|C_O| + |C_N| \leq n^2$, which means that all connections can be routed, and will also require that they are all routed.

That is, we require that all (new and old) connections shall be routed, and wish to find the solution with minimal number of reroutings. To this end, we need to introduce costs for rerouting. Not making any changes to the present solution should incur no cost at all.

Given a binary solution, $\bar{x}$, for $l \in C_O$ there should be a cost if $\bar{x}_{il} = 0$ and we set $x_{il} = 1$, or if $\bar{x}_{il} = 1$ and we set $x_{il} = 0$, but not if $\bar{x}_{il} = 0$ and we set $x_{il} = 0$ or if $\bar{x}_{il} = 1$ and we set $x_{il} = 1$. Furthermore, there are no costs on x_{il} for $l \in C_N$. Now let $A_1 = \{(i,l), l \in C_O : \bar{x}_{il} = 1\}$ and $A_0 = \{(i,l), l \in C_O : \bar{x}_{il} = 0\}$. The cost should thus occur if we set $x_{il} = 0$ for any $(i,l) \in A_1$, or set $x_{il} = 1$ for any $(i,l) \in A_0$, and can be expressed as

$$\sum_{(i,l)\in A_1} (1 - x_{il}) + \sum_{(i,l)\in A_0} x_{il}$$

The cost will be equal to zero if no rearrangements are made, and equal to $|C_O|$ if all (old) routes are changed. We now get the objective function

$$\sum_{(i,l)\in A_1} (1 - x_{il}) + \sum_{(i,l)\in A_0} x_{il} = |A_1| - \sum_{(i,l)\in A_1} x_{il} + \sum_{(i,l)\in A_0} x_{il} = \sum_{(i,l)} c_{il} x_{il} + |C_o|$$

where $c_{il} = -1$ for $(i,l) \in A_1$, $c_{il} = 1$ for $(i,l) \in A_0$ and $c_{il} = 0$ for all i and $l \in C_N$. The mathematical model for minimizing the number of reroutings is thus as follows.

$$
\begin{aligned}
v^* = \quad & \min \quad \sum_{(i,l)} c_{il} x_{il} \\
& \text{s.t.} \quad (1.1), (1.2), (1.3), x_{il} \in \{0,1\} \quad \forall i, l
\end{aligned}
\qquad \text{[P1r]}
$$

We know that $-|C_O| \le v^* \le |C_O|$. The number of reroutings will be equal to $n_r = (v^* + |C_O|)/2$, since we must adjust for the constant $|C_O|$, and a rerouting makes two changes, one path is added and another one is removed.

A feasible solution to P1r can be found in polynomial time exactly as for P1, but that does not ensure minimal rerouting, unless it is optimal. In [Holmberg, 2007b], we describe a polynomial and practically efficient method based on edge coloring for finding the routing of a new connection so that minimal rearrangements of other connections are done.

2.7 Lagrangean duality

For the heuristic approaches, it is a disadvantage not to know if the number of rearrangements is minimal or close to minimal. If we have a feasible binary solution to P1r, its objective function value is an upper bound on the minimal number of rearrangements that are needed. In order to estimate how close this is to the minimum, we would like to have a lower bound on v^* to compare with.

Let us therefore consider using Lagrangean duality as follows. We relax constraints 1.2 and 1.3 with multipliers β and γ. We get the following dual problem.

$$\max g(\beta, \gamma) \text{ s.t. } \beta \ge 0,\ \gamma \ge 0$$

where

$$
\begin{aligned}
g(\beta,\gamma) = \quad & \min \quad \sum_i \sum_l c_{il} x_{il} + \sum_i \sum_k \beta_{ik} (\sum_{l \in L_k^1} x_{il} - 1) + \sum_i \sum_k \gamma_{ik} (\sum_{l \in L_k^3} x_{il} - 1) \\
& \text{s.t.} \quad \sum_{i \in N} x_{il} = 1 \ \forall l \\
& \qquad\ \ x_{il} \in \{0,1\} \ \forall i, l
\end{aligned}
\qquad \text{[LR1]}
$$

LR1 is separable into one problem for each connection l. Letting $\hat{c}_{il} = c_{il} + \beta_{ik_l^1} + \gamma_{ik_l^3}$, we have $g(\beta,\gamma) = \sum_l g_l(\beta,\gamma) - \sum_i \sum_k \beta_{ik} - \sum_i \sum_k \gamma_{ik}$, where, for each l,

$$
\begin{aligned}
g_l(\beta,\gamma) = \quad & \min \quad \sum_i \hat{c}_{il} x_{il} \\
& \text{s.t.} \quad \sum_{i \in N} x_{il} = 1 \\
& \qquad\ \ x_{il} \in \{0,1\} \ \forall i
\end{aligned}
$$

Letting $\hat{c}_{\hat{i}l} = \min_i \hat{c}_{il}$, the optimal solution is $x_{\hat{i}l} = 1$ and $x_{il} = 0$ for all $i \neq \hat{i}$, i.e. the subproblem is trivially solvable.

The dual problem can now be solved by searching in β and γ with subgradient optimization, [Poljak, 1967], [Poljak, 1969], [Held, Wolfe, and Crowder, 1974]. It is well-known that $g(\beta,\gamma) \le v*$ for any nonnegative β and γ. In order to prove optimality of the primal solution, it is sufficient to find β and γ such that $g(\beta,\gamma) \ge \hat{v} - 2$, where $\hat{v}$ is the objective function value of P1r, since the objective function value must be an even integer. If the primal solution is not optimal, an estimation of the error may be obtained by finding a near optimal solution to the dual problem. For more details, and an algorithmic description, see [Holmberg, 2007a].

We may also note that solving LR1 yields an x-solution that satisfies (1.1) but probably not (1.2) and (1.3). In any case this solution might give an indication of how to change the x-solution. One such solution is obtained every time LR1 is solved, so an iterative dual search method may yields many such indications. Occasionally we may even get a feasible solution.

3 Five stages

3.1 Mathematical model

Let us now turn our attention to the case with five stages of switches. We replace the center stage in a three stage network with n blocks of separate three stage units. The first and the fifth stages distribute the connections between the different three stage units.

The coupling is just an up-scaling of the arrangement of the three stage unit. The first (and third) stage now consists of n units containing n switches each. The n blocks of three stage units between them is called the intermediate stage. The first output of the first switch in the first unit of the first stage is coupled to the first input of the first switch in the first block in the intermediate stage, the second output of the first switch in the first unit of the first stage is coupled to the first input of the first switch in the second block of the intermediate stage, and so on. The first output of the second switch in the first unit of the first stage is coupled to the second input of the first switch in the second block of the intermediate stage, and so on. The first output of the first switch in the second unit of the first stage is coupled to the first input of the second switch in the first block of the intermediate stage, and so on. This arrangement is pictured in figure 5. In this case, the network can accommodate at most n^3 connections.

Note that the couplings between the first and second stage are not symmetric. The couplings between the fourth and fifth stages are similar to those between the first and the second stages, but reversed (mirrored) so that the whole network is symmetrical. One may also note that there are never two connections between a switch in the first (or fifth) stage and a block in the center stage.

We could use the multicommodity flow model, as in the three stage case. The only difference is that the network will be larger, as there will be $10n^3$ nodes, $5n^4 + 4n^3$ arcs and n^3 commodities, yielding $5n^7 + 4n^6$ variables. (For $n = 20$ this is 80000 nodes, 832000 arcs, 8000 commodities, and 6 656 000 000 variables.) Again it is doubtful if this is the most efficient way of solving the problem.

The mathematical modeling is here somewhat more complicated. Let us start by defining all the parameters, as in the previous section. There is an additional index for blocks. We need to specify which input/output (i/j), which switch (k) and which block (p) and connection is to use.

In addition to the previous notation, connection l will use block p_l^t in stage t, for $t = 1, 2, 3$, where the second stage contains the three stages for switches.

From o_l, we calculate $p_l^1 = \lfloor o_l/n^2 \rfloor$, $k_l^1 = \lfloor (o_l - p_l^1 n)/n^2 \rfloor$ and $i_l^1 = o_l - k_l^1 n - p_l^1 n^2$.

From d_l, we calculate $p_l^3 = \lfloor d_l/n^2 \rfloor$, $k_l^5 = \lfloor (d_l - p_l^3 n)/n^2 \rfloor$ and $j_l^5 = d_l - k_l^5 n - p_l^3 n^2$.

We also get $k_l^2 = p_l^1$ and $k_l^4 = p_l^3$, $i_l^2 = k_l^1$ and $j_l^4 = k_l^5$, as well as $i_l^3 = k_l^2$ and $j_l^3 = k_l^4$.

If we choose to use block p_l^2 in the second stage, we get $j_l^1 = i_l^5 = p_l^2$. If we choose to use switch k_l^3 in the third stage, we get $j_l^2 = i_l^4 = k_l^3$.

Thus p_l^2 and k_l^3 are the only choices we have to make for connection l. All the other parameters are determined by this choice. Let $p_l^2 = w$ and $k_l^3 = v$.

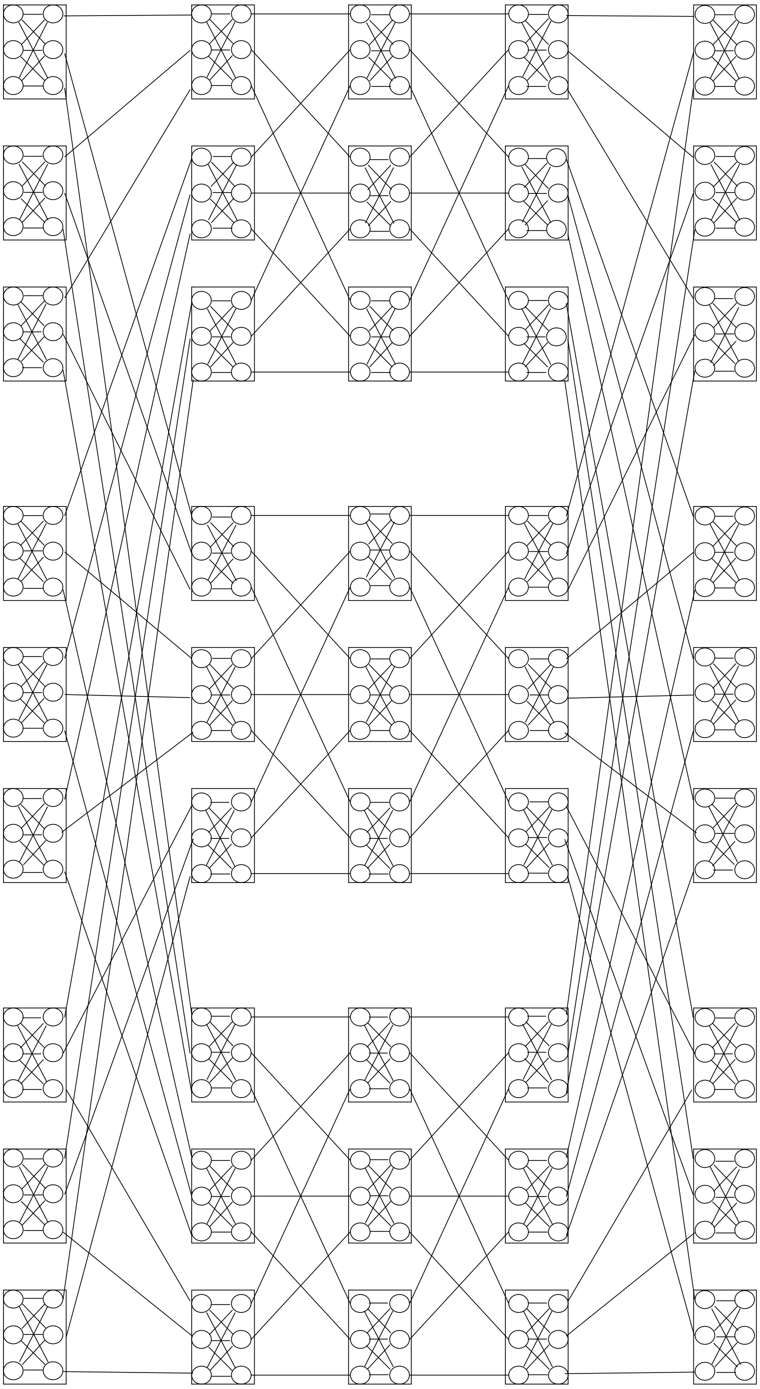

Figure 5: 5-stage switch.

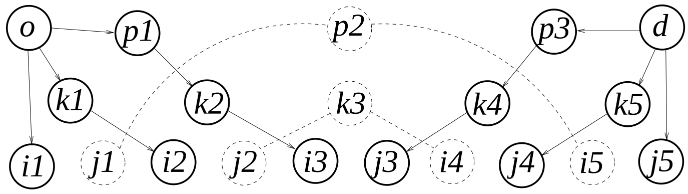

Figure 6: Dependencies between parameters in the 5-stage case.

Stage	Block	Switch	Input	Output
1	$p_l^1 = \lfloor o_l/n^2 \rfloor$	$k_l^1 = \lfloor (o_l - p_l^1 n)/n^2 \rfloor$	$i_l^1 = o_l - k_l^1 n - p_l^1 n^2$	$j_l^1 = w$
2	-	$k_l^2 = p_l^1$	$i_l^2 = k_l^1$	$j_l^2 = v$
3	$p_l^2 = w$	$k_l^3 = v$	$i_l^3 = k_l^2$	$j_l^3 = k_l^4$
4	-	$k_l^4 = p_l^3$	$i_l^4 = v$	$j_l^4 = k_l^5$
5	$p_l^3 = \lfloor d_l/n^2 \rfloor$	$k_l^5 = \lfloor (d_l - p_l^3 n)/n^2 \rfloor$	$i_l^5 = w$	$j_l^5 = d_l - k_l^5 n - p_l^3 n^2$

We illustrate the dependencies in figure 6. Solid circles indicate parameters that are directly calculated from o_l and d_l and solid lines indicate the dependencies. Dashed circles indicate parameters not fixed and dashed lines indicate dependencies between parameters not fixed.

Comparing to the three stage case, we find that we must first choose the block in the intermediate stage, and then, within this block, choose the switch in the middle stage (stage 3). For the mathematical model, we therefore define the following variables.

$x_{qil} = 1$ if connection l uses switch i and block q.

(The relation to the previously used notation is that $x_{qil} = 1$ for $q = p_l^2$, $k = k_l^3$, while $x_{qil} = 0$ for all $i \neq k_l^3$ and $q \neq p_l^2$.)

Again, for each switch, each input can be used by at most one connection, and each output can be used by at most one connection. Assuming that the overall inputs, o_l, and outputs, d_l, obey this, takes care of i_l^1, i_l^2 (via k_l^1) and i_l^3 (via p_l^1 and k_l^2). It also takes care of j_l^5, j_l^4 (via k_l^5) and j_l^3 (via p_l^3 and k_l^4).

This leaves j_l^1, j_l^2, i_l^4 and i_l^5, but $j_l^1 = i_l^5 = p_l^2$ and $j_l^2 = i_l^4 = k_l^3$, so in the choice of p_l^2 and k_l^3, this should be taken into account.

We introduce the following sets.

$L_k^1 = \{l : k_l^1 = k\}$, $L_k^2 = \{l : k_l^2 = k\}$, $L_k^4 = \{l : k_l^4 = k\}$ and $L_k^5 = \{l : k_l^5 = k\}$.

Thus L_k^1 is the set of connections that use switch k in the first stage, L_k^2 is the set of connections that use switch k in the second stage, L_k^4 is the set of connections that use switch k in the fourth stage, and L_k^5 is the set of connections that use switch k in the fifth stage. Again we note that $\bigcup\limits_k L_k^1 = \bigcup\limits_k L_k^2 = \bigcup\limits_k L_k^4 = \bigcup\limits_k L_k^5 = \{1, \ldots, m\}$.

We can calculate k_l^1, k_l^2, k_l^4 and k_l^5 from the given o_l and d_l, so these sets are given by the indata. We also introduce the following sets.

$P_p^1 = \{l : p_l^1 = p\}$ and $P_p^3 = \{l : p_l^3 = p\}$,

i.e. P_p^1 is the set of connections that use block p in the first stage, and P_p^3 is the set of connections that use block p in the last stage. Here $\bigcup\limits_p P_p^1 = \bigcup\limits_p P_p^3 = \{1, \ldots, m\}$.

Since $k_l^2 = p_l^1$, we know the following. If connection l uses block p in the first stage, i.e. $l \in P_p^1$, then the connection l also uses switch p in the second stage, i.e. $l \in L_p^2$. This means that $P_p^1 = L_p^2$. Similarly, since $k_l^4 = p_l^3$, we have $P_p^3 = L_p^4$.

We also introduce

$L_k^3 = \{l : k_l^3 = k\}$ and $P_p^2 = \{l : p_l^2 = p\}$

as an alternate way of representing a solution.

Since a block in the intermediate stage is a three stage unit, we can use the same type of constraints.

$$\sum_{l \in L_k^2} x_{qil} \leq 1 \qquad q \in N, i \in N, k \in N$$

i.e. at most one connection may use switch k in the second stage and switch i in the third stage, in any block.

$$\sum_{l \in L_k^4} x_{qil} \leq 1 \qquad q \in N, i \in N, k \in N$$

i.e. at most one connection may use switch k in the fourth stage and switch i in the third stage, in any block. This takes care of the inputs and outputs within the intermediate blocks, i.e. j_l^2 and i_l^4.

We must also take the connections between the intermediate stage and the first and last stages into account. Between a switch, k, in a block, p, in the first stage and a block in the second stage, q, at most one connection can be used. The connections using switch k in the first stage are those in L_k^1, and the connections using block p in the first stage are those in P_p^1, so we consider the connections in $P_p^1 \cap L_k^1$. However, $P_p^1 = L_p^2$, so we use the set $L_k^1 \cap L_p^2$.

$$\sum_{i \in N} \sum_{l \in L_k^1 \cap L_p^2} x_{qil} \leq 1 \qquad q \in N, k \in N, p \in N$$

The same holds between the second and last stages.

$$\sum_{i \in N} \sum_{l \in L_k^5 \cap L_p^4} x_{qil} \leq 1 \qquad q \in N, k \in N, p \in N$$

This takes care of j_l^1 and i_l^5.

Here we note that $\bigcup_p \bigcup_k (L_k^1 \cap L_p^2) = \bigcup_p \bigcup_k (L_k^5 \cap L_p^4) = \{1, \ldots, m\}$.

One might consider only connections using a certain switch, m, in the second stage. Then we would consider the set $P_p^1 \cap L_k^1 \cap L_m^2$. However, we previously noticed that $L_m^2 = P_m^1$, so the set would be $P_p^1 \cap L_k^1 \cap P_m^1$, which obviously is empty if $m \neq p$ (no connection uses two different blocks in the first stage), and equal to $P_p^1 \cap L_k^1$ if $p = m$. This is an example of constraints that are automatically satisfied by the definition of our variables and sets.

We now wish to find a feasible solution to the following model.

$$\begin{array}{rcll}
\sum_{q \in N} \sum_{i \in N} x_{qil} & = & 1 & \forall l \quad (2.1) \\
\sum_{l \in L_k^2} x_{qil} & \leq & 1 & q \in N, i \in N, k \in N \quad (2.2) \\
\sum_{l \in L_k^4} x_{qil} & \leq & 1 & q \in N, i \in N, k \in N \quad (2.3) \\
\sum_{i \in N} \sum_{l \in L_k^1 \cap L_p^2} x_{qil} & \leq & 1 & q \in N, k \in N, p \in N \quad (2.4) \\
\sum_{i \in N} \sum_{l \in L_k^5 \cap L_p^4} x_{qil} & \leq & 1 & q \in N, k \in N, p \in N \quad (2.5) \\
x_{qil} & \in & \{0, 1\} & \forall q, i, l
\end{array} \qquad [\text{P2}]$$

Summing up the roles of the constraints ensuring that each input/output is used by at most one connection, we find that (2.2) takes care of j_l^2, (2.3) takes care of i_l^4, (2.4) takes care of j_l^1, and (2.5) takes care of i_l^5.

P2 has mn^2 variables and $4n^3 + m$ constraints (not counting the binary requirements). For $m = 20$ and $m = n^3$, this is 3 200 000 variables and 40000 constraints. It can be solved with a general MIP-code. There are similarities and differences to P1. Just as P1, P2 will be massively degenerated. However, it is easy to find instances where the LP-relaxation does not give an integer solution. In [Holmberg, 2007a], we give an example of P2 and its solution.

If we do not specify k, the switch in the first stage, in constraints (2.4), we get aggregated constraints that are summations of (2.4).

$$\sum_{i \in N} \sum_{l \in P_p^1} x_{qil} \leq n \qquad q \in N, p \in N \qquad (2.6)$$

The same holds between blocks and the second and last stages.

$$\sum_{i \in N} \sum_{l \in P_p^3} x_{qil} \leq n \qquad q \in N, p \in N \qquad (2.7)$$

Here it is more difficult to see the relations to matchings. Clearly the center stage, which is three stage units, can be modeled by matchings. This is ensured by constraints (2.2) and (2.3). Aggregating everything to blocks, we find that constraints (2.6) and (2.7) define what we may call n-matchings, i.e. multigraphs where not more than n edges may be adjacent to the same node. Constraints (2.4) and (2.5) are however harder to interpret in matching terms.

We could construct a bipartite multigraph, called the p-graph, with the edges (p_l^1, p_l^3), and extract matchings from it in polynomial time. One such matching represents connections that may use the same switch and the same block in the center stages. However, we are not sure that two such matchings don't interfere with each other. In other words, we are not sure to satisfy constraints (2.4) and (2.5).

Clearly (2.6) and (2.7) follows from (2.4) and (2.5), since they are obtained by simply summing the constraints. However, summing (2.2) and (2.3) over i also yields (2.6) and (2.7). Interestingly enough, we furthermore find that König's theorem on bipartite edge coloring tells us that a bipartite graph with maximal degree d can be colored by d colors. Applying this result to our case, we see that constraints (2.6) and (2.7) yields a bipartite graph with degree at most n. Therefore (2.6) and (2.7) tell us that there exists at most n matchings, i.e. implies (2.2) and (2.3). That seems to indicate that (2.2) and (2.3) are equivalent to (2.6) and (2.7), which is somewhat surprising.

However, (2.4) and (2.5) do not follow from (2.6) and (2.7). Trying to interpret these constraints as matchings, we find that in an extended bipartite multigraph, where there is one node for each block-switch combination, these constraints indeed imply matchings.

Consider a small example, with $n = 3$ and two connections, one with $o_1 = 0$ and $d_1 = 0$ and one with $o_2 = 1$ and $d_2 = 1$. This means that $p_1^1 = 0$, $p_1^3 = 0$, $p_2^1 = 0$ and $p_2^3 = 0$, $k_1^1 = 0$, $k_1^5 = 0$, $k_2^1 = 0$, and $k_2^5 = 0$, and also $k_1^2 = 0$, $k_1^4 = 0$, $k_2^2 = 0$, and $k_2^4 = 0$. We have $L_0^1 = L_0^2 = L_0^4 = L_0^5 = P_0^1 = P_0^3 = \{1, 2\}$.

The p-graph will simply have two parallel edges between the two top nodes. Since the degree is less then 3, one block should be sufficient. The multigraph can then be separated into two identical matchings, corresponding to switches 0 and 1 in block 0. We thus get $k_1^3 = 0$, $p_1^2 = 0$, $k_2^3 = 1$ and $p_2^2 = 0$.

This corresponds to the solution $x_{001} = 1$ and $x_{012} = 1$, which satisfies constraints (2.2), (2.3), (2.6) and (2.7), but not (2.4) and (2.5).

Thus our conclusion is that the solutions must be matchings in two different graphs. In other words we need to find selections of matchings satisfying additional constraints, and this is not likely to be possible in polynomial time.

Unfortunately this means that the polynomial methods using matching and edge coloring can not be used for the five stage problem. As of now, we do not know if P2 can be solved in polynomial time. We may recall a few *NP*-complete problems with some similarities, namely minimal edge coloring in general (not restricted to bipartite graphs), the maximum disjoint connecting paths problem (and its variation the minimum path coloring problem) and the maximum 3-dimensional matching problem, [Garey and Johnson, 1979].

3.2 Minimal rearrangements

If there already is a set connections, C_O, set up, and a set of new ones, C_N, is requested, we let $\bar{x}$ be the solution of P2 without the new connections. Then we search for a $\hat{i}$ and $\hat{p}$ that allows the routing of each new connection. As in the previous case, we can use a heuristic based on the slack in the constraints. We first decide $\hat{p}$ with the help of 2.4 and 2.5, and then decide $\hat{i}$ with the help of $\hat{p}$ inserted in 2.2 and 2.3.

For the problem for exactly minimizing the rearranging, let $A_1 = \{(q,i,l) : \bar{x}_{qil} = 1, l \in C_O\}$, $A_0 = \{(q,i,l) : \bar{x}_{qil} = 0, l \in C_O\}$, and let $c_{qil} = -1$ for all $(q,i,l) \in A_1$, $c_{qil} = 1$ for all $(q,i,l) \in A_0$ and $c_{qil} = 0$ for all $l \in C_N$. Then we wish to solve

$$
\begin{aligned}
v^* = \quad & \min \quad \sum_{q\in N}\sum_{i\in N}\sum_{l=1}^{m} c_{qil}x_{qil} \\
& \text{s.t.} \quad (2.1),(2.2),(2.3),(2.4),(2.5), x_{qil} \in \{0,1\} \quad \forall q,i,l
\end{aligned}
\qquad \text{[P2r]}
$$

Clearly P2r is more difficult to solve than P1r. The polynomial method in [Holmberg, 2007b] for the three stage case can not, as far as we see, be used here.

3.3 Lagrangean duality

We can apply Lagrangean duality on P2r by relaxing constraints 2.2, 2.3, 2.4 and 2.5 with multipliers β^1, γ^1, β^2 and γ^2. Letting $\hat{c}_{qil} = c_{qil} + \beta^1_{qik^2_l} + \gamma^1_{qik^4_l} + \beta^2_{qk^1_l k^2_l} + \gamma^2_{qk^5_l k^4_l}$, we get the following subproblem, for each l.

$$
\begin{aligned}
g_l(\beta^1,\gamma^1,\beta^2,\gamma^2) = \quad & \min \quad \sum_{q\in N}\sum_{i\in N} \hat{c}_{qil}x_{qil} \\
& \text{s.t.} \quad \sum_{q\in N}\sum_{i\in N} x_{qil} = 1 \\
& \qquad\quad x_{qil} \in \{0,1\} \forall q,i
\end{aligned}
$$

Letting $\hat{c}_{\hat{q}\hat{i}l} = \min_{q,i} \hat{c}_{qil}$, the optimal solution is $x_{\hat{q}\hat{i}l} = 1$ and $x_{qil} = 0$ for all $q \neq \hat{q}$ and $i \neq \hat{i}$. Again $g(\beta^1,\gamma^1,\beta^2,\gamma^2) \leq v*$ for any nonnegative β^1, γ^1, β^2 and γ^2, and the maximal lower bound can be found by using subgradient optimization,

Another possibility of applying Lagrangean relaxation is to relax only constraints (2.4) and (2.5), while keeping the constraints (2.2) and (2.3) in the subproblem. In that case the subproblem is not separable. However, the subproblem will be of the same type as P1r, and can thus be solved with the same techniques. In order to see this clearly, the indices q and i should be replaced with one index. This way we get a subgradient optimization procedure with polynomially solvable subproblems.

4 Seven stages

We will now consider the case with seven stages, which is even more complicated. We do a similar up-scaling of the network, by replacing the center stage in a three stage network by a n separate 5-stage units. This configuration can handle up to n^4 simultaneous connections.

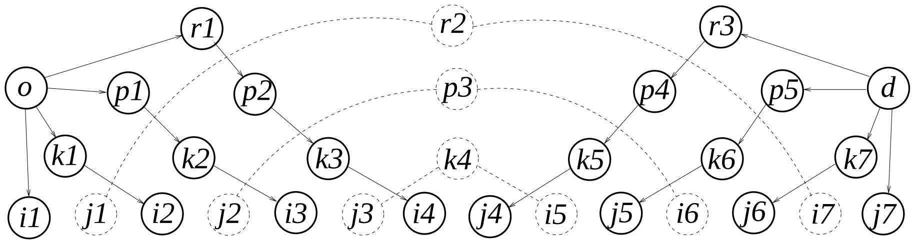

Figure 7: Dependencies between parameters in the 7-stage case.

Here we need to use all similarities to the smaller cases. We introduce another index, r, for the larger groups, and let connection l use group r_l^t in stage t.

From o_l, we calculate $r_l^1 = \lfloor o_l/n^3 \rfloor$, $p_l^1 = \lfloor (o_l - r_l^1 n^3)/n^2 \rfloor$, $k_l^1 = \lfloor (o_l - r_l^1 n^2 - p_l^1 n)/n \rfloor$ and $i_l^1 = o_l - k_l^1 n - p_l^1 n^2 - r_l^1 n^3$. From d_l, we calculate $r_l^3 = \lfloor d_l/n^3 \rfloor$, $p_l^5 = \lfloor (d_l - r_l^3 n^3)/n^2 \rfloor$, $k_l^7 = \lfloor (d_l - r_l^3 n^2 - p_l^5 n)/n \rfloor$ and $j_l^7 = d_l - k_l^7 n - p_l^5 n^2 - r_l^3 n^3$.

This yields $p_l^2 = r_l^1$, $k_l^2 = p_l^1$ and $i_l^2 = k_l^1$, followed by $k_l^3 = p_l^2$, $i_l^3 = k_l^2$, and finally $i_l^4 = k_l^3$. We also get $p_l^4 = r_l^3$, $k_l^6 = p_l^5$ and $j_l^6 = k_l^7$, followed by $k_l^5 = p_l^4$, $j_l^5 = k_l^6$, and finally $j_l^4 = k_l^5$.

If we choose to use group r_l^2 in the intermediate stage, we get $j_l^1 = i_l^7 = r_l^2$. If we choose to use block p_l^3 in the third stage, we get $j_l^2 = i_l^6 = p_l^3$. If we choose to use switch k_l^4 in the fourth stage, we get $j_l^3 = i_l^5 = k_l^4$. Thus only r_l^2, p_l^3 and k_l^4 remain to be chosen.

In figure 7 solid circles indicate parameters that are directly calculated from o_l and d_l and solid lines indicate the dependencies. Dashed circles indicate parameters not fixed and dashed lines indicate dependencies between parameters not fixed.

We define the following variables.

$x_{rqil} = 1$ if connection l uses switch i, block q and group r.

We introduce the following sets. $L_k^1 = \{l : k_l^1 = k\}$, $L_k^2 = \{l : k_l^2 = k\}$, $L_k^3 = \{l : k_l^3 = k\}$, $L_k^5 = \{l : k_l^5 = k\}$, $L_k^6 = \{l : k_l^6 = k\}$ and $L_k^7 = \{l : k_l^7 = k\}$. L_k^1 is the set of connections that use switch k in the first stage, L_k^2 is the set of connections that use switch k in the second stage, L_k^3 is the set of connections that use switch k in the third stage, L_k^5 is the set of connections that use switch k in the fifth stage, L_k^6 is the set of connections that use switch k in the sixth stage, and L_k^7 is the set of connections that use switch k in the seventh stage.

We could also introduce the following sets. $P_p^1 = \{l : p_l^1 = p\}$, $P_p^2 = \{l : p_l^2 = p\}$, $P_p^4 = \{l : p_l^4 = p\}$, and $P_p^5 = \{l : p_l^5 = p\}$, i.e. P_p^1 is the set of connections that use block p in the first stage, P_p^2 is the set of connections that use block p in the second stage, P_p^4 is the set of connections that use block p in the third stage, and P_p^5 is the set of connections that use block p in the last stage. However, $P_p^1 = L_p^2$, $P_p^2 = L_p^3$, $P_p^4 = L_p^5$ and $P_p^5 = L_p^6$.

We could also define $R_r^1 = \{l : r_l^1 = r\}$ and $R_r^3 = \{l : r_l^3 = r\}$, but $R_r^1 = P_r^2 = L_r^3$ and $R_r^3 = P_r^4 = L_r^5$.

The mathematical model is scaled up as follows, following the earlier principles.

$$
\begin{array}{rcllr}
\sum_{r\in N}\sum_{q\in N}\sum_{i\in N} x_{rqil} & = & 1 & \forall l & (3.1)\\
\sum_{l\in L_k^3} x_{rqil} & \leq & 1 & r\in N, q\in N, i\in N, k\in N & (3.2)\\
\sum_{l\in L_k^5} x_{rqil} & \leq & 1 & r\in N, q\in N, i\in N, k\in N & (3.3)\\
\sum_{i\in N}\sum_{l\in L_k^2\cap L_p^3} x_{rqil} & \leq & 1 & r\in N, q\in N, k\in N, p\in N & (3.4)\\
\sum_{i\in N}\sum_{l\in L_k^6\cap L_p^5} x_{rqil} & \leq & 1 & r\in N, q\in N, k\in N, p\in N & (3.5)\\
\sum_{q\in N}\sum_{i\in N}\sum_{l\in L_k^1\cap L_p^2\cap L_s^3} x_{rqil} & \leq & 1 & r\in N, k\in N, p\in N, s\in N & (3.6)\\
\sum_{q\in N}\sum_{i\in N}\sum_{l\in L_k^7\cap L_p^6\cap L_s^5} x_{rqil} & \leq & 1 & r\in N, k\in N, p\in N, s\in N & (3.7)\\
x_{rqil} & \in & \{0,1\} & \forall r,q,i,l &
\end{array}
\qquad [\text{P3}]
$$

This problem has $n^3 m$ variables and $6n^4 + m$ constraints. A model for minimal rearrangements can be constructed as in the previous sections.

5 Nine stages and more

5.1 Mathematical model

Considering nine stages, i.e. replacing the center stage in a three stage network by n separate 7-stage units, we may handle up to n^5 connections. We have now identified a structure in how the dependencies and the mathematical model grows. There is an additional index for "collections of groups", t.

$$
\begin{array}{rcllr}
\sum_{t\in N}\sum_{r\in N}\sum_{q\in N}\sum_{i\in N} x_{trqil} & = & 1 & \forall l & (4.1)\\
\sum_{l\in L_k^4} x_{trqil} & \leq & 1 & t\in N, r\in N, q\in N, i\in N, k\in N & (4.2)\\
\sum_{l\in L_k^6} x_{trqil} & \leq & 1 & t\in N, r\in N, q\in N, i\in N, k\in N & (4.3)\\
\sum_{i\in N}\sum_{l\in L_k^3\cap L_p^4} x_{trqil} & \leq & 1 & t\in N, r\in N, q\in N, k\in N, p\in N & (4.4)\\
\sum_{i\in N}\sum_{l\in L_k^7\cap L_p^6} x_{trqil} & \leq & 1 & t\in N, r\in N, q\in N, k\in N, p\in N & (4.5)\\
\sum_{q\in N}\sum_{i\in N}\sum_{l\in L_k^2\cap L_p^3\cap L_s^4} x_{trqil} & \leq & 1 & t\in N, r\in N, k\in N, p\in N, s\in N & (4.6)\\
\sum_{q\in N}\sum_{i\in N}\sum_{l\in L_k^8\cap L_p^7\cap L_s^6} x_{trqil} & \leq & 1 & t\in N, r\in N, k\in N, p\in N, s\in N & (4.7)\\
\sum_{r\in N}\sum_{q\in N}\sum_{i\in N}\sum_{l\in L_k^1\cap L_p^2\cap L_s^3\cap L_v^4} x_{trqil} & \leq & 1 & t\in N, k\in N, p\in N, s\in N, v\in N & (4.8)\\
\sum_{r\in N}\sum_{q\in N}\sum_{i\in N}\sum_{l\in L_k^9\cap L_p^8\cap L_s^7\cap L_v^6} x_{trqil} & \leq & 1 & t\in N, k\in N, p\in N, s\in N, v\in N & (4.9)\\
x_{trqil} \in \{0,1\} & & & \forall t,r,q,i,l &
\end{array}
\qquad [\text{P4}]
$$

Now we have identified a structure that enables us to construct models for even larger numbers of stages, if the networks are constructed by the same principles, namely the following. A $2K+1$-stage network is constructed by letting n $2K-1$-stage networks replace the center stage in a three stage network, and connecting the stages accordingly.

For a $2K+1$-stage network, there will be K decisions for each connection. For $K=1$, we get three stage networks, and the decision is which switch (in the second stage) to use. For $K=2$, we get five stage networks, and the two decisions are which switch and which block to use. For $K=3$, we get seven stage networks, and the three decisions are which switch, which block and which group to use.

A $2K+1$-stage network can handle up to n^{K+1} origin-destination pairs. The mathematical model will have $n^K m$ variables and $2Kn^{K+1}+m$ constraints in $2K+1$ groups.

In a $2K+1$-stage network, stage $K+1$ is the center stage. The following constraints will either concern stages before the center stage, K, $K-1$ etc, or concern stages after the center stage, $K+2$, $K+3$ etc. The second and third sets of constraints will be sums over L_k^K and L_k^{K+2}, handling the inputs/outputs closest to the center stage. The following two sets of constraints will be sums over i and over $L_k^{K-1} \cap L_p^K$ and $L_k^{K+3} \cap L_p^{K+2}$. The following two sets of constraints will be sums over q and i and over $L_k^{K-2} \cap L_p^{K-1} \cap L_r^K$ and $L_k^{K+4} \cap L_p^{K+3} \cap L_r^{K+2}$. Each new two groups of constraints handle the inputs/outputs one stage more away from the center stage.

Now let switch index i be denoted by i_K, block index q by i_{K-1}, group index r by i_{K-2} and so on. When enumerating constraints, we denote k with j_1, p with j_2, s with j_3 and so on. This means that i_1 will denote the largest units, i.e. the first decision, i_2 the second largest unit, i.e. the second decision, and so on up to i_K, which denotes the smallest unit, which is the choice of switch in the center stage. On the other hand, j_1 denotes the smallest unit when it comes to constraints. (This discrepancy is motivated by a much needed symmetry in the following parts of the paper.)

We define the following variables.

$x_{i_1 \cdots i_K l} = 1$ if connection l uses switch i_K, block i_{K-1}, group i_{K-2}, etc.

The mathematical model, [P5], will now be as follows.

$$\sum_{i_1\in N}\cdots\sum_{i_K\in N} x_{i_1\cdots i_K l} = 1 \quad \forall l \tag{5.1}$$

$$\sum_{l\in L_{j_1}^K} x_{i_1\cdots i_K l} \le 1 \quad i_1\in N,\dots,i_K\in N, j_1\in N \tag{5.2}$$

$$\sum_{l\in L_{j_1}^{K+2}} x_{i_1\cdots i_K l} \le 1 \quad i_1\in N,\dots,i_K\in N, j_1\in N \tag{5.3}$$

$$\sum_{i_1\in N}\sum_{l\in L_{j_1}^{K-1}\cap L_{j_2}^K} x_{i_1\cdots i_K l} \le 1 \quad i_2\in N,\dots,i_K\in N, j_1\in N, j_2\in N \tag{5.4}$$

$$\sum_{i_1\in N}\sum_{l\in L_{j_1}^{K+3}\cap L_{j_2}^{K+2}} x_{i_1\cdots i_K l} \le 1 \quad i_2\in N,\dots,i_K\in N, j_1\in N, j_2\in N \tag{5.5}$$

$$\sum_{i_2\in N}\sum_{i_1\in N}\sum_{l\in L_{j_1}^{K-2}\cap L_{j_2}^{K-1}\cap L_{j_3}^K} x_{i_1\cdots i_K l} \le 1 \quad i_3\in N,\dots,i_K\in N, j_1\in N,\dots,j_3\in N \tag{5.6}$$

$$\sum_{i_2\in N}\sum_{i_1\in N}\sum_{l\in L_{j_1}^{K+4}\cap L_{j_2}^{K+3}\cap L_{j_3}^{K+2}} x_{i_1\cdots i_K l} \le 1 \quad i_3\in N,\dots,i_K\in N, j_1\in N,\dots,j_3\in N \tag{5.7}$$

$$\vdots$$

$$\sum_{i_{K-1}\in N}\cdots\sum_{i_1\in N}\sum_{l\in\bigcap_{m=1}^K L_{j_m}^m} x_{i_1\cdots i_K l} \le 1 \quad i_K\in N, j_1\in N,\dots,j_K\in N \tag{5.8}$$

$$\sum_{i_{K-1}\in N}\cdots\sum_{i_1\in N}\sum_{l\in\bigcap_{m=1}^K L_{j_m}^{2K+2-m}} x_{i_1\cdots i_K l} \le 1 \quad i_K\in N, j_1\in N,\dots,j_K\in N \tag{5.9}$$

$$x_{i_1\cdots i_K l}\in\{0,1\} \qquad \forall i_1,\dots,i_K,l$$

Here $L_{j_1}^K$ is the set of connections that use switch j_1 in stage K, $L_{j_1}^{K-1}\cap L_{j_2}^K$ the set of connections that use switch j_1 in stage $K-1$ and switch j_2 in stage K, etc.

Since each connection must pass each stage once, each l is represented in one the left-hand-sides of constraints (5.2), in one the left-hand-sides of constraints (5.3), in one the left-hand-sides of constraints (5.4), in one the left-hand-sides of constraints (5.5), etc. This means that taking the union of all the sets L in the left-hand-sides yields the whole set of connections (each once). More formally we have

$$\bigcup_{j_1} L_{j_1}^K = \bigcup_{j_1}\bigcup_{j_2}(L_{j_1}^{K-1}\cap L_{j_2}^K) = \cdots = \bigcup_{j_1}\cdots\bigcup_{j_K}(\bigcap_{m=1}^K L_{j_n}^m) = \{1,\dots,m\}.$$

We have K groups of constraints, each containing two groups, one containing constraints for stages before the center stage and one containing constraints for stages after the center stage Let us use the parameter v for these groups. The first group is a bit special, but for $v=2,3,\dots,K$, the constraints will be the following.

$$\sum_{i_{v-1}\in N}\cdots\sum_{i_1\in N}\sum_{l\in\bigcap_{m=1}^v L_{j_m}^{K-v+m}} x_{i_1\cdots i_K l} \le 1 \quad i_v\in N,\dots,i_K\in N, j_1\in N,\dots,j_v\in N$$

$$\sum_{i_{v-1}\in N}\cdots\sum_{i_1\in N}\sum_{l\in\bigcap_{m=1}^v L_{j_m}^{K+2+v-m}} x_{i_1\cdots i_K l} \le 1 \quad i_v\in N,\dots,i_K\in N, j_1\in N,\dots,j_v\in N$$

In order to simplify notation somewhat, let

$$M_v^1(j_1,\dots,j_v) = \bigcap_{m=1}^v L_{j_m}^{K-v+m} \quad\text{and}\quad M_v^2(j_1,\dots,j_v) = \bigcap_{m=1}^v L_{j_m}^{K+v+2-m}.$$

We have $\bigcup_{j_1}\cdots\bigcup_{j_K} M_v^1(j_1,\dots,j_v) = \bigcup_{j_1}\cdots\bigcup_{j_K} M_v^2(j_1,\dots,j_v) = \{1,\dots,m\}$.

Now $M_v^1(j_1,\dots,j_v)$ is the set of connections that use switch j_m in stage $K-v+m$ for

$m = 1$ to v, i.e. switch j_1 in stage $K - v + 1$ and switch j_2 in stage $K - v + 2$ and so on, up to switch j_v in stage K.

The model, [P6], now becomes as follows.

$$\begin{array}{rcll}
\sum_{i_1 \in N} \cdots \sum_{i_K \in N} x_{i_1 \cdots i_K l} & = & 1 & \forall l \qquad (6.1)\\
\sum_{l \in L^K_{j_1}} x_{i_1 \cdots i_K l} & \leq & 1 & i_1 \in N, \ldots, i_K \in N, j_1 \in N \qquad (6.2)\\
\sum_{l \in L^{K+2}_{j_1}} x_{i_1 \cdots i_K l} & \leq & 1 & i_1 \in N, \ldots, i_K \in N, j_1 \in N \qquad (6.3)\\
\sum_{i_{v-1} \in N} \cdots \sum_{i_1 \in N} \sum_{l \in M^1_v(j_1, \ldots, j_v)} x_{i_1 \cdots i_K l} & \leq & 1 & \forall i_v, \ldots, i_K, j_1, \ldots, j_v, v = 2, \ldots, K \qquad (6.4)\\
\sum_{i_{v-1} \in N} \cdots \sum_{i_1 \in N} \sum_{l \in M^2_v(j_1, \ldots, j_v)} x_{i_1 \cdots i_K l} & \leq & 1 & \forall i_v, \ldots, i_K, j_1, \ldots, j_v, v = 2, \ldots, K \qquad (6.5)\\
x_{i_1 \cdots i_K l} \in \{0, 1\} & & & \forall i_1, \ldots, i_K, l
\end{array}$$

For any given K, this model is well stated. The number of indices, and thus also number of sums, variables and constraints, are then fixed and given.

5.2 Rearrangements

Again let $\bar{x}$ be the solution of P6 without the new connections. Then we search for $\hat{i}_K$, $\hat{i}_{K-1}$, $\hat{i}_{K-2}$ ... $\hat{i}_1$, that allows the routing of each new connection. We can device a sequential heuristic that first decides $\hat{i}_1$ with the help of 6.4 and 6.5 for $v = K$, then $\hat{i}_2$ with the help of $\hat{i}_1$ and 6.4 and 6.5 for $v = K - 1$, and so on, until finally deciding $\hat{i}_K$ with the help of $i_1 \cdots i_{K-1}$ together with 6.2 and 6.3.

The model for minimal rearrangements will be as follows. Let $A_1 = \{(i_1, \ldots, i_k, l) : \bar{x}_{i_1 \cdots i_K l} = 1, l \in C_O\}$, $A_0 = \{(i_1, \ldots, i_k, l) : \bar{x}_{i_1 \cdots i_K l} = 0, l \in C_O\}$, and let $c_{i_1 \cdots i_k l} = -1$ for all $(i_1, \ldots, i_k, l) \in A_1$, $c_{i_1 \cdots i_k l} = 1$ for all $(i_1, \ldots, i_k, l) \in A_0$ and $c_{i_1 \cdots i_k l} = 0$ for all $l \in C_N$.

$$\begin{array}{rll}
v^* = & \min & \sum_{l=1}^{m} \sum_{i_1 \in N} \cdots \sum_{i_K \in N} c_{i_1 \cdots i_K l} x_{i_1 \cdots i_K l} \\
& \text{s.t.} & (6.1), (6.2), (6.3), (6.4), (6.5), x_{i_1 \cdots i_K l} \in \{0, 1\} \quad \forall i_1, \ldots, i_K, l
\end{array} \qquad \text{[P6r]}$$

The number of variables in this linear integer programming problem is mn^K and the number of constraints is $m + 2Kn^{K+1}$.

5.3 Lagrangean relaxation

Applying Lagrangean relaxation, we use multipliers β for constraints (6.2) and (6.4) and γ for (6.3) and (6.5). We get the following dual problem.

$$\max g(\beta, \gamma) = \sum_l g_l(\beta, \gamma) - C(\beta, \gamma) \text{ s.t. } \beta \geq 0, \gamma \geq 0$$

where the Lagrangean subproblem is separable into one problem for each connection l.

$$\begin{array}{rll}
g_l(\beta, \gamma) = & \min & \sum_{i_1} \cdots \sum_{i_K} \hat{c}_{i_1 \cdots i_K l} x_{i_1 \cdots i_K l} \\
& \text{s.t.} & \sum_{i_1 \in N} \cdots \sum_{i_K \in N} x_{i_1 \cdots i_K l} = 1 \ \forall l \\
& & x_{i_1 \cdots i_K l} \in \{0, 1\} \ \forall i_1, \ldots, i_K, l
\end{array} \qquad [\text{LR6}_l]$$

Letting

$$\hat{c}_{\hat{i}_1\cdots\hat{i}_K l} = \min_{i_1\cdots i_K l} \hat{c}_{i_1\cdots i_K l},$$

the optimal solution is $x_{\hat{i}_1\cdots\hat{i}_K l} = 1$. This yields

$$g_l(\beta,\gamma) = \sum_{i_1}\cdots\sum_{i_K} \min_{i_1\cdots i_K l} \hat{c}_{i_1\cdots i_K l} x_{i_1\cdots i_K l}$$

So again the subproblem is trivially solvable. The dual problem can now be attacked with subgradient optimization. This approach will yield a sequence of lower bounds that can be compared with upper bounds from a constructive heuristic. One should, however, bear in mind that in order to solve the dual problem to maximize $g(\beta,\gamma)$, one needs to find the optimal values of all the multipliers, β and γ. There are Kn^{K+1} β-variables, and the same number of γ-variables. More details are found in [Holmberg, 2007a].

Another possibility is to apply Lagrangean relaxation to all constraint groups except the first three, 6.1, 6.2 and 6.3. This yields a subproblem of the same type as P1r. The details are very similar to those discussed earlier.

6 Conclusions

We have constructed mathematical models for finding the routing through symmetrical Clos networks, for three stages, five stages and so on up to any number of stages. We give models for simultaneous routing of connections, and for routing of additional connections, with the objective of doing minimal rearrangement of the previously routed connections. The models can be used to analyze the problem, and to discuss both exact and heuristic solution methods. We mention the possibility of using a primal heuristic together with a dual Lagrangean relaxation scheme for finding bounds on the optimal number of rearrangements.

References

Balas, E., and Padberg, M. (1972), On the set-covering problem. *Operations Research* 20, 1152–1161.

Beneš, V. (1965), *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic press.

Berge, C. (1973), *Graphs and Hypergraphs*, North-Holland, Amsterdam.

Carpinelli, J. D., and Oruç, A. Y. (1993), A nonbacktracking matrix decomposition algorithm for routing on Clos networks. *IEEE Transactions on Communications* 41, 1245–1251.

Clos, C. (1953), A study of nonblocking switching networks. *Bell Syst. Tech. J.* 32, 406–424.

Cole, R., and Hopcroft, J. (1982), On edge coloring bipartite graphs. *SIAM Journal on Computing* 11, 540–546.

Cole, R., Ost, K., and Schirra, S. (2001), Edge-coloring bipartite multigraphs in $O(E \log D)$ time. *Combinatorica* 21, 5–12.

Franaszek, P. A., Georgiou, C. J., and Li, C.-S. (1995), Adaptive routing in Clos networks. in: *Proceedings of the International Conference on Computer Design*, 266–270.

Garey, M. R., and Johnson, D. S. (1979), *Computers and Intractbility: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York.

Gordon, J., and Srikanthan, S. (1990), Novel algorithm for Clos-type networks. *Electronics Letters* 26, 1772–1774.

Held, M., Wolfe, P., and Crowder, H. P. (1974), Validation of subgradient optimization. *Mathematical Programming* 6, 62–88.

Holmberg, K. (2007a), Optimization models for switching networks of Clos type with many stages. Research report LiTH-MAT-R-2007-11, Department of Mathematics, Linköping Institute of Technology, Sweden.

Holmberg, K. (2007b), Graph optimization approaches for minimal rerouting in symmetric three stage Clos networks. Research report LiTH-MAT-R-2007-12, Department of Mathematics, Linköping Institute of Technology, Sweden.

Hwang, F. K. (1983), Control algoritmhs for rearrangeable Clos networks. *IEEE Transactions on Communications* 31, 952–954.

Hwang, F. K. (1997), A modification to a decomposition algorithm of Gordon and Srikanthan. *IEEE Transactions on Computers* 46, 958–960.

Jajszczyk, A. (1985), A simple algorithm for the control of rearrangeable switching networks. *IEEE Transactions on Communications* COM-33, 169–171.

Lee, H. Y., Hwang, F. K., and Carpinelli, J. D. (1996), A new decomposition algorithm for rearrangeable Clos interconnection networks. *IEEE Transactions on Communications* 44, 1572–1578.

Poljak, B. T. (1967), A general method of solving extremum problems. *Soviet Mathematics Doklady* 8, 593–597.

Poljak, B. T. (1969), Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics* 9, 14–29.

Schrijver, A. (1999), Bipartite edge-coloring in $O(\Delta m)$ time. *SIAM Journal on Computing* 28, 841–846.