Combinatorial optimization problems in wireless switch design[*][†]

Renaud Sirdey[a]

[a] Service d'architecture BSC (PC 12A7), Nortel GSM Access R&D, Parc d'activités de Magny-Châteaufort, 78928 Yvelines Cedex 09, France.

Corresponding author: R. Sirdey (*renauds@nortel.com*, +33 (0)1 69 55 41 18).

Abstract

The purpose of this paper is to illustrate the diversity of combinatorial problems encountered in the design of wireless switching systems. This is done via a representative selection of examples of *real* problems along with their associated solution methods. It should be emphasized that all the solution methods presented in this paper are successfully operating in the field at the time of writing.

Keywords: Combinatorial optimization, OR in telecommunications.

Mathematics Subject Classification: 90C27, 90B18.

1 Introduction

The telecommunication industry is a major provider of challenging combinatorial problems to the operations research community, usually in the areas of networks design, dimensioning and operation. An additional and perhaps lesser known source of combinatorial problems can be found in the design of the machines which operate within the networks.

---

[*]Technical report Nortel GSM Access R&D PE/BSC/INF/20290 V01/EN, to appear in the industry section of *4OR—A Quarterly Journal of Operations Research*.

[†]Part of this work has been presented at the *Journées Franciliennes de la Recherche Opérationnelle*, June 2006.

Loosely speaking, a telecom switch is a system offering limited amounts of interrelated resources (circuits, ports, links, computing modules, disk storage, CPU, memory, etc.) which should be efficiently used. In such a context, it is not surprising for optimization problems to crop up. Additionally, many constraints such as on the execution time, on the size or on the maintainability of the software limit the spectrum of applicable solution methods, especially when dealing with NP-hard problems.

This paper illustrates the diversity of combinatorial problems encountered in the field of telecom switch design via a representative selection of *real* problems, real in the sense that all the problems presented in this paper have really occurred and that the solution methods discussed herein, which are essentially based on known algorithmic techniques, are successfully operating in the field at the time of writing. These problems have cropped up in the context of Nortel's GSM Base Station Subsystem (BSS), the subset of a GSM network mainly in charge of transmission and reception on the radio path, which is under the responsibility of the Base Transceiver Stations (BTS), as well as of radio resources and handovers management, under the responsibility of the Base Station Controllers (BSC). Loosely speaking, a BTS is an entity comprising radio transmission and reception devices (including the antennas) which can be seen as a kind of complex radio modem. The geographical areas which a BTS covers using different frequencies are referred to as radio cells. A BSC is essentially a small switch connected, on one side, to several BTS and, on the other side, to the core of the network known as the Network and Switching Subsystem (NSS). The BSC is in particular responsible for the management of the radio interface (radio channels allocation and release, handover management, etc.) through the remote command of the BTS and of the mobiles. The reader is referred to, e.g., Mouly and Pautet (1992); Lagrange et al. (2000) for more details regarding the GSM network architecture.

The paper is organized as follows. Section 2 describes our vision of the practice of operations research in the field of telecom switch design. Each of the sections 3 to 5 subsequently presents an application. The applications discussed in sections 3 and 4 concern the BSC while the application presented in Section 5 is related to the BTS. Section 3 discusses a multiobjective radio cell configuration problem which is reduced to an assignment problem and tackled using network flow techniques. Section 4 presents a problem related to the management of the interface between two switches which is equivalent to a variant of the bin-packing problem, the *extensible bin-packing* problem,

and dealt with using efficient approximate solution algorithms having quite satisfactory performance guarantees. Lastly, Section 5 is devoted to a problem which consists in minimizing the electrical consumption of a BTS during a power outage under a service constraint, this problem being a special case of a low-dimensional nonlinear knapsack problem which can be solved in polynomial time using dynamic programming.

It should be emphasized that this paper contains very few original results. Our goal has been to illustrate how some algorithms and methods published by the academic community have been adapted to effectively solve real problems as well as the surprising number of seemingly unrelated fields which have inspired our solution methods. It is our genuine belief that the applications discussed in sections 3 to 5, although far from being exhaustive, are representative.

2   Some thoughts on the practice of OR

Operations research and more particularly combinatorial optimization is, as in many other fields, of the uttermost practical relevance in the context of telecom switch design.

As an introductory example, let us consider the theory of NP-hardness. As humorously illustrated in a famous cartoon by Garey and Johnson (1979), this theory allows to convince a customer (be it a validation team or a network operator) that some problems cannot, in practice, be solved exactly (though usually after a little bit of education as many people tend not to be familiar with the notion of computational complexity). That fact does not depend on the skills of the involved research and development teams.

Of course, the industry has a strong preference for problems which are tractable *in practice*[1]. Facing such a problem is the best scenario, not only an efficient solution procedure is likely to be found in a book or paper but, and perhaps more importantly, communication with the customer is eased. Indeed, as long as the constraints and objective function are agreed upon, criticism of the end results is almost pointless: best possible solutions are always provided, efficiently.

---

[1]I.e., excluding problems which are tractable only in theory that is, for example, excluding problems so far known to be polynomially solvable only using the ellipsoid algorithm (Grötschel et al., 1988; Schrijver, 2004).

Unfortunately, industrialists do not have the privilege of choosing their problems and intractable (i.e., NP-hard) problems often crop up due either to the intrinsic nature of the environment in which the system is to be embedded or simply to the fact that early high-level product specifications rarely take into account considerations such as the computational complexity of problems to be encountered downstream (often because only a small portion of these problems are foreseeable during the early stages of a system design).

Telecom switches are subject to real-time constraints and generally have a distributed architecture made up of many modules with limited resources. When facing intractable problems, these constraints limit the spectrum of applicable solution procedures to efficient approximate ones. Of course, this does not mean that exact solution procedures are useless: their primary role is to assist in the empirical assessment of the quality of the solutions provided by approximate solution ones, when appropriate (mainly in the absence of satisfactory performance guarantees).

Another point which should not be eluded is the simplicity requirement imposed on the solution methods (Sirdey, 2005). Nowadays, the maintenance of a product is more often than not under the responsibility of individuals other than those who designed it. As a consequence, in order not to jeopardize transfers of ownership, it is fundamental to achieve an appropriate balance between, on the one hand, the performances of a given method versus, on the other hand, the complexity of its implementation.

3   Optimum radio cell configuration

This section deals with the problem of optimally configuring a radio cell by assigning predefined groups of radio channels, also known as Time Division Multiple Access (TDMA) frames, to transceivers. A transceiver being an elementary device which can emit or receive continuously on a single frequency (at a given time).

3.1   Preliminaries

Let T denote a set of groups of radio channels, hereafter referred to as *groups*, and X denote a set of transceivers, hereafter referred to as *TRX*. Each group can be assigned to at most one TRX and each TRX can be assigned at most one group. Additionally, the set X is supplemented by a "dummy" TRX,

denoted d, which may be assigned an arbitrary number of groups. A group which is assigned to a dummy TRX is effectively unassigned and when at least one group is unassigned the cell is in *degraded mode*, as not all the configured service is provided.

Because of hardware constraints a group may not be assignable to all TRX. The set of TRX to which a group $t \in T$ can be assigned is denoted $X(t)$. Note that for all $t \in T$, d belongs to $X(t)$.

Additionally, a set S of services is given. Each groups $t \in T$ *demands* a set $S(t) \subseteq S$ of services and each TRX $x \in X$ *offers* a set of services denoted by $S(x) \subseteq S$. A group $t \in T$ may be assigned to a TRX $x \in X$ which does not support all of its service set, that is such that $S(t) \setminus S(x) \neq \emptyset$, when this happens the group, hence the cell, is also in degraded mode.

Lastly, each group is given a *criticality* $0, 1, 2, \ldots$ The value of this parameter may or may not be related to the group service set. For example, certain groups must be assigned to a TRX if the cell is to support any traffic, the criticality of this kind of groups is usually set to 0 (the most critical) and they must be assigned "at all cost".

Roughly speaking, the problem then consists in finding the "best" assignment of the groups to the TRX in terms of the objective function described in the next section.

3.2   Objective function

For each group $t \in T$ and each TRX $x \in X(t) \cup \{d\}$, the following variables are introduced

$$\delta_{tx} = \begin{cases} 1 & \text{if } t \text{ is to be assigned to } x, \\ 0 & \text{otherwise.} \end{cases}$$

We now consider a hierarchy of objective functions which captures the various facets of the problem.

First, criticality must be taken into account. The most critical groups must be assigned even if in doing so all the groups of lesser criticality cannot be assigned. This is achieved by minimizing the following objective function,

$$f_1(\delta) = \sum_{t \in T} W(C(t))\delta_{tx}, \tag{1}$$

where $C(t)$ denote the criticality of group $t \in T$ and, H denoting the largest criticality value, where weight function W is defined recursively by $W(H) = 1$

and, for $c = H - 1, \ldots, 0$,

$$W(c) = W(c+1)(|T(c+1)| + 1),$$

with $T(c) = \{t \in T : C(t) = c\}$.

Second, service must be taken into account. This is done via the minimization of the following objective function

$$f_2(\delta) = \sum_{t \in T} \sum_{x \in X(t) \setminus \{d\}} |S(t) \setminus S(x)| \delta_{tx}. \tag{2}$$

Note that the term $|S(t) \setminus S(x)|$ may be replaced by a weighted sum over the set $S(t) \setminus S(x)$, when services have different impacts.

The third objective allows to deal with TRX failures, additions or removals. For example, TRX may fail at any time and, upon TRX failure, the system is required to reconfigure the cell so as to ensure optimal operation with one less TRX. The new optimal assignment should then be reached via a sequence of group reassignments of lowest impact. Note that all calls carried by a group are lost upon reassignment of the group and that the group radio channels are unavailable during the reassignment. Let $R(t)$ denote the cost of reassigning group t (e.g., a function of its criticality, as for the first objective) and let

$$\gamma_{tx} = \begin{cases} 1 & \text{if } t \text{ is } \textit{presently} \text{ assigned to } x, \\ 0 & \text{otherwise.} \end{cases}$$

Then, $t \in T$ is reassigned if and only if $\sum_{x \in X(t) \setminus \{d\}} (1 - \gamma_{tx}) \delta_{tx} = 1$. Hence, the lowest impact is achieved by minimizing

$$f_3(\delta) = \sum_{t \in T} \sum_{x \in X(t) \setminus \{d\}} R(t)(1 - \gamma_{tx}) \delta_{tx}. \tag{3}$$

Lastly, service is again taken into account by minimizing the following objective function

$$f_4(\delta) = \sum_{t \in T} \sum_{x \in X(t) \setminus \{d\}} (H - C(t) + 1)|S(x) \setminus S(t)| \delta_{tx}. \tag{4}$$

This latter objective ensures that the more critical groups are given their best fitting TRX, if possible. As a consequence, the more critical groups are less likely to be reassigned (e.g., upon failure of another TRX).

Finally, functions (1), (2), (3) and (4) are aggregated into one linear objective function

$$f(\delta) = \sum_{i=1}^{4} \lambda_i f_i(\delta), \qquad (5)$$

where the λ_i have been chosen so as to enforce a strict criteria hierarchy, i.e., so as to enforce that an arbitrary improvement in any of the criteria $j > i$ does not justify a degradation of criterion i.

3.3   Solution method

Since the objective function (5) is linear, it is easy to see that the problem can be modelled using a network flow and dealt with any algorithm solving the minimum cost flow problem (Ahuja et al., 1993).

Figure 1 provides an example of network for an instance with 3 groups and 4 TRX. The nodes s, t and d respectively denote the *source*, the *sink* and the *dummy* TRX. Only the capacity of the arcs are indicated (the cost of an arc which either links s to a vertex or links a vertex to t is zero, for the other arcs the cost is given as specified in the previous section). The existence of a solution is guaranteed by the fact that the capacity of the $\{d, t\}$ arc is equal to the number of groups. On this example, $X(t_1) = \{x_1, x_3, x_4, d\}$ and $X(t_2) = X(t_3) = \{x_2, d\}$ (recall that d always belongs to $X(t_i)$), as a consequence, either t_2 or t_3 will be assigned to d, i.e., not assigned. Given a minimum cost s-t-flow on the above network, the only arc $\{t_i, x_j\}$ or $\{t_i, d\}$ which carries one unit out of vertex t_i indicates the TRX, x_j or d, to which t_i is assigned.

Our implementation was based on the Successive Shortest Path algorithm (Ahuja et al., 1993; Korte and Vygen, 2000) using 64 bits integer arithmetic so as to cope with the large coefficients generally present in the objective function.

At the time of writing, this method successfully operates in the field. Also, it is important to stress that the method was implemented in less than 200 lines of code and successfully replaced the few thousands lines of code of an ad hoc algorithm implemented in an earlier version of the system which, on top of being hardly maintainable, was known not to operate satisfactorily.
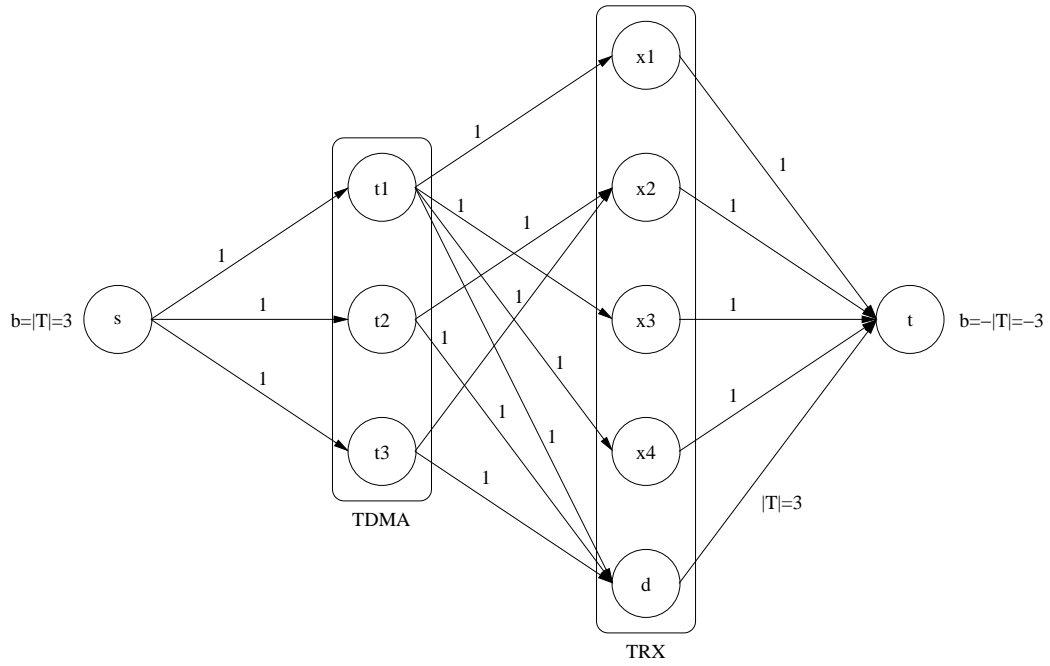
Figure 1: Example of network for an instance of the radio cell configuration problem of Section 3 with 3 groups and 4 TRX.

4 PCM interface management

This section is devoted to a problem related to the management of the PCM links[2] between two switches. The links offer some circuits which have to be allocated to a set of radio cells under the constraint that all the circuits allocated to a given cell must be on the same link (the cell is then said to be *assigned* to that link). Generally, be it due do the configuration itself or due to the unavailability of some of the links, the total circuit demand exceeds the total circuit capacity of the links and, hence, some cells will be allocated less circuits than required. The objective is thus to minimize the unsatisfied circuit demand.

4.1 Preliminaries

Let C denote a set of radio cells, hereafter referred to as *cells*, and let P denote a set of PCM links, hereafter referred to as *links*, available on the interface between two telecom switches. For a cell $c \in C$, let $w(c)$ denote the cell circuit requirement and for a link $p \in P$, let $K(p)$ denote the link circuit capacity.

Generally, the total circuit demand exceeds the total circuit offer that is

$$\sum_{c \in C} w(c) > \sum_{p \in P} K(p),$$

and the problem consists in finding an assignment $f : C \longrightarrow P$ of the cells to the links so as to minimize the total overflow defined as

$$\sum_{p \in P} \max \left(0, \sum_{c \in C : f(c) = p} w(c) - K(p) \right).$$

It turns out that this problem is equivalent to the so-called *extensible bin-packing problem* (Dell'Olmo et al., 1998) which (paraphrasing the above), given a list of n positive numbers $\alpha_1, \ldots, \alpha_n$ and m bins $b_1, \ldots, b_m$ of capacities $k_1, \ldots, k_m$, asks for an assignment $f : \{1, \ldots, n\} \longrightarrow \{1, \ldots, m\}$ such that

$$\sum_{i=1}^{m} \max(k_i, \ell(b_i)),$$

---

[2]A PCM (Pulse Code Modulation) link is a link carrying either 31 (European standard) or 24 (North-American standard) 64 kbit/s channels.

where

$$\ell(b_i) = \sum_{j=1\ldots n : f(j)=i} \alpha_j,$$

is minimum.

4.2 Solution method

Although the extensible bin-packing problem is NP-hard in the strong sense, there exist simple approximate solution algorithms with fairly reasonable performance guarantees.

Indeed, in the case where the bins have equal capacity Dell'Olmo et al. (1998) have established that the well-known LPT heuristic, which sorts the items in non-increasing order into a list and assigns, at each iteration, the first item of the list to a least loaded bin so far, is such that for any instance I of the problem

$$\frac{\text{LPT}(I)}{\text{OPT}(I)} \leq \frac{13}{12} \approx 1.0833. \tag{6}$$

Additionally, when the bins have different capacities, Dell'Olmo and Speranza (1999) have established that under the (reasonable) assumption that

$$\max_{i=1,\ldots,n} \alpha_i \leq \min_{i=1,\ldots,m} k_i, \tag{7}$$

the LPT heuristic (modified so as to consider the bin with the biggest idle space rather than the least load) is such that for any instance I of the problem[3]

$$\frac{\text{LPT}(I)}{\text{OPT}(I)} < 2(2 - \sqrt{2}) \approx 1.1716. \tag{8}$$

The above motivates the following algorithm: sort the cells in order of decreasing circuit requirements $w_{c_1} \geq \ldots \geq w_{c_{|C|}}$, for $i = 1$ to $|C|$ if there exist a link $p \in P$ with positive remaining capacity, i.e., such that $K(p) > \sum_{j=1:f(c_j)=p}^{i-1} w_{c_j}$, then assign c_i to the link with the biggest remaining capacity otherwise assign c_i to the proportionally least loaded link, i.e., the link which minimizes

$$\frac{1}{K(p)} \sum_{j=1:f(c_j)=p}^{i-1} w_{c_j}.$$

---

[3]A tight bound of $\frac{8}{7} \approx 1.1429$ was conjectured in that paper.

It is clear that the above algorithm is equivalent to LPT when the links have the same capacity and, hence, that it provides ratio (6) as well as when the links have different capacities (something fairly rare, though possible, in our application) in which case it guarantees only ratio (8) as long as assumption (7) is satisfied, this assumption being more than reasonable for our application.

Although the algorithm which was finally implemented was slightly more complicated than the above, so as to cope with a few additional requirements which are not detailed here, the fact that such simple (and fast) heuristics provided acceptable performance guarantees was highly beneficial. In particular, it allowed to cut short some discussions with the validation team when coming with hand-generated examples of instances for which they were able to find solutions of better quality than those provided by the algorithm. Indeed, no trouble was found as long as the two solutions were within the tolerance derived from the performance guarantees and this allowed to elude a certain number of illegitimate enhancement requests, thereby avoiding adding undue complexity to the resulting software.

4.3 The parliament analogy

We now consider the problem of fairly allocating the circuits of a link to the cells which have been assigned to it, when demand exceeds offer.

Consider a country made up of a set of districts. Under the assumption that the country is democratic, it has a parliament with a small number of seats, small compared to overall country population.

A central problem in electoral systems theory is to find a fair repartition of the parliament seats between the districts, under some constraints such as that each district must be allocated at least one seat.

It is generally admitted that a fair repartition algorithm must satisfy the following requirements (Balinski and Young, 2001):

1. If the number of seats increases then each district should be allocated at least the same number of seats.

2. If the population of a given district increases whereas the population of another decreases, then the latter should not be allocated more seats to the detriment of the earlier.

3. The repartition should be identical if one considers only a subset of the districts and the number of seats which have been allocated to them.

4. Perfect proportionality should be achieved when it is achievable.

It turns out that a very simple algorithm, known as the Sainte-Laguë method, fulfils all of the above requirements (Balinski and Young, 2001).

Returning to our per-link circuit allocation problem, the parliament analogy simply consists in seeing the set of cells as a set of districts, the circuit demand of a cell as a district population, the PCM as a parliament and the PCM circuit capacity as the number of seats in that parliament.

More generally, the analogy works with any set of consumers demanding certain amounts of a given scarce resource, when demand exceeds offer.

5 Battery life maximization

This section is devoted to a problem which consists in minimizing the electrical consumption of a Base Transceiver Station (BTS) during a power outage, under a service constraint. The BTS is made up of cabinets containing radio modules themselves containing TRX. Each TRX is assigned to one of the cells managed by the BTS. The service constraint is defined, for each cell, as the number of TRX which must remain operational, for as long as possible, during the power outage. As the electrical consumption of the site is proportional to the number of operational modules, the minimum consumption is achieved, in the monocabinet case, when the service constraint is satisfied with the smallest possible number of operational modules. The multicabinet case gives rise to a min-max problem: as each cabinet has its own battery, the time during which the service constraint is satisfied is limited by the lifespan of the battery of the cabinet hosting the biggest number of operational modules.

The purpose of such a functionality is to maximize the time during which a GSM network still covers a geographical area, though offering limited service, after a catastrophic power breakdown, e.g., after a natural disaster in which context the network can play a crucial role by still providing emergency services or even by helping locating trapped individuals when the wireline networks are damaged.

5.1   Preliminaries

Consider a BTS made up of a set K of cabinets as well as of a set R of radio modules, hereafter referred to as *modules*, each module containing up to 3 TRX. Also the BTS manages a set of cells, denoted by C.

Let W denote a $|R| \times |C|$ integer valued matrix which rows and columns are respectively indexed by the elements of R and C and such that given a module r and a cell c, W_{rc} denotes the number of TRX on r dedicated to cell c. Also, let L be a $|R| \times |K|$ binary valued matrix which rows and columns are respectively indexed by the element of R and K and such that given an module r and a cabinet k, $L_{rk} = 1$ if and only if module r is in cabinet k. Lastly, for each cell, a number N_c which specifies the TRX requirement during the power outage is given.

To each module is associated a binary variable x_r such that

$$x_r = \begin{cases} 1 & \text{if module } r \text{ is powered during the outage,} \\ 0 & \text{otherwise.} \end{cases}$$

The problem can then be modelled using the following nonlinear mathematical program

$$\begin{cases} \text{Minimize } \max_{k \in K} \sum_{r \in R} L_{rk} x_r, & (9) \\[2mm] \text{s. t.} \\ \sum_{r \in R} W_{rc} x_r \geq N_c & \forall c \in C, \\[2mm] x_r \in \{0, 1\} & \forall r \in R. \end{cases}$$

Assuming that $\sum_{r \in R} L_{rk} = L$ (i.e., that all the cabinets contain the same number of modules[4]), letting $x_r = 1 - y_r$ and rearranging leads to the following mathematical program

$$\begin{cases} \text{Maximize } \min_{k \in K} \sum_{r \in R} L_{rk} y_r, & (10) \\[2mm] \text{s. t.} \\ \sum_{r \in R} W_{rc} y_r \leq \sum_{r \in R} W_{rc} - N_c & \forall c \in C, & (11) \\[2mm] y_r \in \{0, 1\} & \forall r \in R. \end{cases}$$

---

[4]When this is not true, dummy modules can be added. For such a modules, $W_{rc} = 0$ for each cell c.

Stated as above, the problem is a Multidimensional Max-Min Knapsack Problem.

Given an optimal solution to the above mathematical program one has to switch off all modules for which $y_r = 1$ and for each cell to switch off $\sum_{r \in R} W_{rc} x_r - N_c$ TRX (no combination of TRX can improve the objective function, otherwise the solution would not be optimal).

5.2   The max-min knapsack problem

The Monodimensional Max-Min Knapsack Problem has been relatively recently introduced and studied by Yu (1996), see also Kellerer et al. (2004). This problem arises when one has to fill a knapsack with a selected set of items so that the minimum total profit gained under a set of scenarios is maximized, each item having a scenario-dependent profit (e.g., the profit of taking an umbrella for travel depends on the weather at destination). This problem has applications in the area of robust optimization under uncertainty, for example in robust capital budgeting.

The Monodimensional Max-Min Knapsack Problem can be stated as follows

$$
\begin{cases}
\text{Maximize} \ \min_{k=1,\ldots,t} \sum_{j=1}^{n} p_{kj} x_j, \\
\text{s. t.} \\
\sum_{j=1}^{n} w_j x_j \leq c, \\
x_j \in \{0,1\} \qquad\qquad\qquad j = 1, \ldots, n,
\end{cases}
\tag{12}
$$

When t is fixed, the problem is only weakly NP-hard and can be solved in pseudopolynomial time by dynamic programming. The solution method considers the following family of problems

$$
\begin{cases}
z_j(d; v_1, \ldots, v_t) = \text{Maximize} \ \min_{k=1,\ldots,t} \sum_{i=1}^{j} (p_{ki} x_i + v_k), \\
\text{s. t.} \\
\sum_{i=1}^{j} w_i x_i \leq d, \\
x_i \in \{0,1\} \qquad\qquad\qquad i = 1, \ldots, j,
\end{cases}
\tag{13}
$$

14

where $z_n(c; 0, \ldots, 0)$ is the solution of problem (12). The solution method then uses the following recurrence formula

$$z_{j+1}(d; v_1, \ldots, v_t) =$$
$$\begin{cases} z_j(d; v_1, \ldots, v_t) & \text{if } d < w_{j+1}, \\ \max\left\{ z_j(d; v_1, \ldots, v_t), \right. & \\ \left. \quad z_j(d - w_{j+1}; v_1 + p_{1,j+1}, \ldots, v_t + p_{t,j+1}) \right\} & \text{otherwise.} \end{cases} \quad (14)$$

with

$$z_0(d; v_1, \ldots, v_t) = \min_{k=1,\ldots,t} v_k.$$

The resulting algorithm runs in $O(ncP^t)$, with $P = \max_{k=1,\ldots,t} \sum_{j=1}^{n} p_{kj}$.

Problem (12) can easily be generalized to multiple dimensions yielding the Multidimensional Max-Min Knapsack Problem, that is the following mathematical program:

$$\begin{cases} \text{Maximize} \quad \min_{k=1,\ldots,t} \sum_{j=1}^{n} p_{kj} x_j, \\ \text{s. t.} \\ \sum_{j=1}^{n} w_{ij} x_j \leq c_i & i = 1, \ldots, m, \\ x_j \in \{0, 1\} & j = 1, \ldots, n, \end{cases}$$

where all the inputs (p_{kj}, w_{ij} and c_i) are nonnegative integers.

5.3 Solution method

As in the case of the classical knapsack problem (Minoux, 1983; Kellerer et al., 2004), d (in equations (13) and (14)) above can be interpreted as a vector and the above formulas can then be used to derive a

$$O\left(n \left(\prod_{i=1}^{m} c_i \right) P^t \right)$$

pseudopolynomial algorithm solving the Multidimensional Max-Min Knapsack Problem, when both t and m are fixed.

However, provided that $n = |R|$, $c_i \leq O(|R|)$ and $0 \leq v_k \leq \sum_{j=1}^{n} p_{kj} \leq |R|$, the complexity of such an algorithm as far as our problem is concerned is

$$O\left(|R|^{|C|+|K|+1} \right),$$

hence polynomial for fixed $|C|$ and fixed $|K|$. This leads to a $O(|R|^7)$ algorithm for our application as 3 is an upper bound for both $|C|$ and $|K|$. Of course, this is not satisfactory in practice and rules out a direct dynamic programming implementation as such an implementation always realizes the worst case in terms of both time and space. Instead, we use a vector version of the above recurrence relation within a depth-first search algorithm relying on a data structure, referred to as the *cache*, which allows to filter out all of the search tree redundancy. The cache is keyed using integers uniquely associated to integer-valued vectors of the form $(j; d_1, \ldots, d_m; v_1, \ldots, v_t)$ $(0 \leq j \leq n,$ $0 \leq d_i \leq c_i$ and $0 \leq v_k \leq \sum_{j=1}^{n} p_{kj})$. A cache item is defined as a pair

$$\{z_j(d_1, \ldots, d_m; v_1, \ldots, v_t), x_j\}$$

where x_j is a boolean variable which indicates whether or not item j belongs to the optimal solution of subproblem $z_j(d_1, \ldots, d_m; v_1, \ldots, v_t)$, this latter field allowing to build the solution once the data structure has been filled. Thus, the subtree rooted at node $(j; d_1, \ldots, d_m; v_1, \ldots, v_t)$ of the search tree is explored if and only if no cache item is associated to that node, a cache item is then added after exploring the subtree. Also note that the cache is assumed to provide logarithmic access time, i.e., it is assumed to be some kind of balanced binary search tree (Knuth, 1998).

This algorithm may be considered a reinterpretation of a dynamic programming recurrence formula as a dominance relationship for use in a tree search algorithm. This point is further discussed by Ibaraki (1977).

In practice, it turns out that an implementation of this algorithm is able to solve the biggest practical instances (3 cabinets, 3 cells and 24 TRX per cell) in a bit more than 1 second on a standard laptop PC, a duration which is acceptable as power outages of less than a minute are filtered out. Table 1 illustrates the performances of the algorithm, note that for all instances $\sum_r W_{r1} = \sum_r W_{r2} = \sum_r W_{r3} = 24$.

6   Conclusion

In this paper, our intent has been to share some insights on the practical relevancy of using OR methods in the field of telecom switch design. In order to do so, we have provided a representative selection of *real* problems along with their associated solution methods:

N_1	N_2	N_3	# mod.	# TRX	CPU (s)	cache size
12	12	12	24	72	0.510	91 022
5	7	11	24	72	0.971	161 178
6	6	6	24	72	1.201	200 190
7	6	4	24	72	1.221	199 422
2	2	2	24	72	1.331	223 463

Table 1: Illustration of the performance of the depth-first search algorithm for solving the battery life maximization problem of Section 5 on a set of instances of biggest practical size.

- A polynomial radio cell configuration problem dealt with using network flow techniques.

- A strongly NP-hard radio-cell-to-link assignment problem tackled with simple approximation algorithms with acceptable performance guarantees (as well as a touch of electoral systems theory).

- A BTS battery life maximization problem which ends up being a polynomial special case of a nonlinear knapsack problem dealt with using a depth-first search algorithm supplemented with dominance relationships derived from a dynamic programming recurrence relation.

At the time of writing, all these solution methods are successfully operating in the field.

Although it is far from being exhaustive, other problems are discussed in Sirdey et al. (2003) and Sirdey et al. (2005), we believe that this list of problems is fairly representative of the kind of combinatorial problems encountered in telecom switch design. We also believe that the solution methods presented herein are representative of the spectrum of techniques which are usable under the constraints of the field (real-time, maintainability, etc.).

Lastly, we hope that we have been able to convincingly illustrate how research from academia can be almost directly and successfully applied to concrete telecom problems, even when the research comes from seemingly unrelated fields.

References

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows. Theory, algorithms and applications.* Prentice Hall, 1993.

M. L. Balinski and H. P. Young. *Fair representation: meeting the ideal of one man, one vote.* Brookings Institution Press, second edition, 2001.

P. Dell'Olmo, H. Kellerer, M. G. Speranza, and Z. Tuza. A $\frac{13}{12}$ approximation algorithm for bin packing with extendable bins. *Information Processing Letters*, 65:229–233, 1998.

P. Dell'Olmo and M. G. Speranza. Approximation algorithms for partitioning small items in unequal bins to minimize the total size. *Discrete Applied Mathematics*, 94:181–191, 1999.

M. R. Garey and D. S. Johnson. *Computers and intractability—A guide to the theory of NP-completeness.* W. H. Freeman and Company, 1979.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics.* Springer, 1988.

T. Ibaraki. The power of dominance relations in branch-and-bound algorithms. *Journal of the ACM*, 24(2):264–279, 1977.

H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems.* Springer, 2004.

D. E. Knuth. *Sorting and searching*, volume 3 of *The Art of Computer Programming.* Addison-Wesley, second edition, 1998.

B. Korte and J. Vygen. *Combinatorial optimization—Theory and algorithms*, volume 21 of *Algorithms and Combinatorics.* Springer, 2000.

X. Lagrange, P. Godlewski, and S. Tabbane. *Réseaux GSM, des principes à la norme.* Réseaux et télécommunications. Hermès Science Publications, cinquième edition, 2000.

M. Minoux. *Programmation mathématique. Théorie et algorithmes (tome 2).* Collection Technique et Scientifique des Télécommunications. Dunod, 1983.

M. Mouly and M.-B. Pautet. *The GSM System for Mobile Commnunications—A comprehensive overview of the European Digital Cellular Systems*. Telecom Publishing, 1992.

A. Schrijver. *Combinatorial optimization—Polyhedra and efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2004.

R. Sirdey. BSC software engineering guidelines. Technical Report PE/BSC/APP/16486 V01/EN, Service d'architecture BSC, Nortel GSM Access R&D, 2005.

R. Sirdey, J. Carlier, H. Kerivin, and D. Nace. On a resource-constrained scheduling problem with application to distributed systems reconfiguration. Technical Report PE/BSC/INF/015593 V01/EN, Service d'architecture BSC, Nortel GSM Access R&D, France (submitted to the European Journal of Operational Research), 2005.

R. Sirdey, D. Plainfossé, and J.-P. Gauthier. A practical approach to combinatorial optimization problems encountered in the design of a high availability distributed system. In *Proceedings of INOC*, pages 532–539, 2003.

G. Yu. On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44:407–415, 1996.