



**HAL**  
open science

## A bilevel optimization model for load balancing in mobile networks through price incentives

Marianne Akian, Mustapha Bouhtou, Jean-Bernard Eytard, Stéphane Gaubert

### ► To cite this version:

Marianne Akian, Mustapha Bouhtou, Jean-Bernard Eytard, Stéphane Gaubert. A bilevel optimization model for load balancing in mobile networks through price incentives. 2018. hal-01972785

**HAL Id: hal-01972785**

**<https://inria.hal.science/hal-01972785v1>**

Preprint submitted on 7 Jan 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# A bilevel optimization model for load balancing in mobile networks through price incentives

Marianne Akian · Mustapha Bouhtou · Jean Bernard Eytard · Stéphane Gaubert

**Abstract** We propose a model of incentives for data pricing in large mobile networks, in which an operator wishes to balance the number of connections (active users) of different classes of users in the different cells and at different time instants, in order to ensure them a sufficient quality of service. We assume that each user has a given total demand per day for different types of applications, which he may assign to different time slots and locations, depending on his own mobility, on his preferences and on price discounts proposed by the operator. We show that this can be cast as a bilevel programming problem with a special structure allowing us to develop a polynomial time decomposition algorithm suitable for large networks. First, we determine the optimal number of connections (which maximizes a measure of balance); next, we solve an inverse problem and determine the prices generating this traffic. Our results exploit a recently developed application of tropical geometry methods to mixed auction problems, as well as algorithms in discrete convexity (minimization of discrete convex functions in the sense of Murota). We finally

---

M. Akian  
INRIA, CMAP, Ecole Polytechnique, CNRS  
Route de Saclay  
91128 Palaiseau Cedex, France  
Tel.: +331 69 33 46 39  
E-mail: marianne.akian@inria.fr

M. Bouhtou  
Orange Labs  
44, avenue de la République  
92320 Chatillon, France  
E-mail: mustapha.bouhtou@orange.com

J.B. Eytard  
INRIA, CMAP, Ecole Polytechnique, CNRS  
Route de Saclay  
91128 Palaiseau Cedex, France  
E-mail: jean-bernard.eytard@inria.fr

S. Gaubert  
INRIA, CMAP, Ecole Polytechnique, CNRS  
Route de Saclay  
91128 Palaiseau Cedex, France  
Tel.: +331 69 33 46 03  
E-mail: stephane.gaubert@inria.fr

present an application on real data provided by Orange and we show the efficiency of the model to reduce the peaks of congestion.

**Keywords** Bilevel programming · Mobile data networks · Tropical geometry · Discrete convexity · Graph algorithms

## 1 Introduction

With the development of new mobile data technologies (3G, 4G), the demand for using the Internet with mobile phones has increased rapidly. Mobile service providers (MSP) have to confront congestion problems in order to guarantee a sufficient quality of service (QoS).

Several approaches have been developed to improve the quality of service, coming from different fields of the telecommunication engineering and economics. For instance, one can refer to Bonald and Feuillet [5] for some models of performance analysis to optimize the network in order to improve the QoS. One of the promising alternatives to solve such problems consists in using efficient pricing schemes in order to encourage customers to shift their mobile data consumption. In [19], Maillé and Tuffin describe a mechanism of auctions based on game-theoretic methods for pricing an Internet network, see also [20]. In [1], Altman et al. study how to price different services by using a noncooperative game. These different approaches are based on congestion games. In the present work, we are interested in how a MSP can improve the QoS by balancing the traffic in the network. We wish to determine in which locations, and at which time instants, it is relevant to propose price incentives, and to evaluate the influence of these incentives on the quality of service.

This kind of problem belongs to smart data pricing. We refer the reader to the survey of Sen et al. [27] and also to the collection of articles [28]. Finding efficient pricing schemes is a revenue management issue. The first approach consists in usage-based pricing; the prices are fixed monthly by analysing the use of the former months. It is possible to improve this scheme by identifying peak hours and non-peak hours and proposing incentives in non-peak hours in order to decrease the demand at peak hours and to better use the network capacity at non-peak hours. This leads to time-dependent pricing. Such a scheme for mobile data is developed by Ha et al. in [14]. The prices are determined at different time slots and based on the usage of the previous day in order to maximize the utility of the customers and the revenue of the MSP. This pricing scheme was concretely implemented by AT&T, showing the relevance of such a model. In another approach, Tadrous et al. propose a model in which the MSP anticipates peak hours and determines incentives for proactive downloads [29].

The latter models concern only the time aspects. One must also take into account the spatial aspect in order to optimize the demand between the different locations. In [17], Ma, Liu and Huang present a model depending on time and location of the customers where the MSP proposes prices and optimizes his profit taking into account the utility of the customers.

Here, we assume (as in [17]) that the MSP proposes incentives at different time and places. Then, customers optimize their data consumption by knowing these incentives and the MSP optimizes a measure of the QoS. In this way, we introduce a bilevel model in which the provider proposes incentives in order to balance the traffic in the network and to avoid as much as possible the congestion (high level problem), and customers optimize their own consumption for the given incentives (low level problem).

Bilevel programs have been widely studied, see the surveys of Colson, Marcotte and Savard [8] and of Dempe [10]. They represent an important class of pricing problems in the sense that they model a leader wanting to maximize his profit and proposing prices to some followers who

maximize themselves their own utility. Most classes of bilevel programs are known to be NP-hard. Several methods have been introduced to solve such problems. For instance, if the low level program is convex, it can be replaced by its Karush-Kuhn-Tucker optimality conditions and the bilevel problem becomes a classical one-stage optimization problem, which is however generally non convex. If some variables are binary or discrete, and the objective function is linear, the global bilevel problem can be rewritten as a mixed integer program, as in Brotcorne et al. [6].

In the present work, we optimize the consumption of each customer in a large area (large urban agglomeration) during typically one day divided in time slots of one hour, taking into account the different types of customers and of applications that they use. Therefore, we have to confront both with the difficulties inherent to bilevel programming and with the large number of variables (around  $10^7$ ). Hence, we need to find polynomial time algorithms, or fast approximate methods, for classes of problems of a very large scale, which, if treated directly, would lead to mixed integer linear or nonlinear programming formulations beyond the capacities of current off-the-shelve solvers.

This motivated us to introduce a different approach, based on tropical geometry. Tropical geometry methods have been recently applied by Baldwin and Klemperer in [3] to an auction problem. This has been further developed by Yu and Tran [30]. In these approaches, the response of an agent to a price is represented by a certain polyhedral complex (arrangement of tropical hypersurfaces). This approach is intuitive since it allows one to visualize geometrically the behavior of the agents: each cell of the complex corresponds to the set of incentives leading to a given response. Then, we visualize the collective response of a group of customers by “superposing” (refining) the polyhedral complexes attached to every customer in this group. We apply here this idea to represent the response of the low-level optimizers in a bilevel problem. This leads to the following decomposition method: first we compute, among all the admissible consumptions of the customers, the one which maximizes a measure of balance of the network; then, we determine the price incentive which achieves this consumption. In this way, a bilevel problem is reduced to the minimization of a convex function over a certain Minkowski sum of sets. We identify situations in which the latter problem can be solved in polynomial time, by exploiting the discrete convexity results developed by Murota [22]. In this approach, a critical step is to check the membership of a vector to a certain Minkowski sum of sets of integer points of polytopes. In our present model, these polytopes, which represent the possible consumptions of one customer, have a remarkable combinatorial structure (they are hypersimplices). Exploiting this combinatorial structure, we show that this critical step can be performed quickly, by reduction to a shortest path problem in a graph. This leads to an exact solution method when there is only one type of contract and one type of application sensitive to price incentive, and to a fast approximate method in the general case.

We finally present the application of this model on real data from Orange and show how price incentives can improve the QoS by balancing the number of active customers in an urban agglomeration during one day. These results indicate that a price incentive mechanism can effectively improve the satisfaction of the users by displacing their consumption from the most loaded regions of the space-time domain to less loaded regions.

The paper is organized as follows. In Section 2, we present the bilevel model. In Section 3, we explain how a certain polyhedral complex can be used to represent the user’s responses, and we describe the decomposition method. In Section 4, we deal with the high level problem and identify special cases which are solvable in polynomial time. In Section 5, we develop accelerated algorithms which enable to solve bilevel problems with a large number of customers. In Section 6, we propose a general relaxation method. The application to the instance provided by Orange is presented in Section 7.

The first results of this article (without proofs) were published in the proceedings of the conference WiOpt 2017 [11].

## 2 A bilevel model

We consider a time horizon of one day, divided in  $T$  time slots numbered  $t \in [T] = \{1, \dots, T\}$ , and a network divided in  $L$  different cells numbered  $l \in [L]$ . We assume that  $K$  customers, numbered  $k \in [K]$ , are in the network. The customers have different types of contracts  $b \in [B]$  and they make requests for different types of applications  $a \in [A]$  (web/mail, streaming, download, ...). We denote by  $\mathcal{K}^b$  the set of customers with the contract  $b$ . A given customer  $k \in \mathcal{K}^b$  is characterized by the following data. We denote by  $L_t^k \in [L]$  the position of the customer  $k$  at each time  $t \in [T]$ , so that the sequence  $(L_1^k, \dots, L_T^k)$  represents the trajectory of this customer. We assume that this trajectory is deterministic, so we consider customers with a regular daily mobility (for example, the trip between home and work). We denote by  $\rho_k^a(t)$  the inclination of a customer  $k$  to make a request for an application of type  $a$  at time  $t \in [T]$ . We suppose that customer  $k$  wishes to make a fixed number of requests  $R_k^a \leq T$  using the application  $a$  during the day. We consider a set of time slots  $\mathcal{I}_k^a \subset [T]$  in which the customer  $k$  decides not to consume the application  $a$ .

We denote by  $u_k^a(t)$  the consumption of the customer  $k$  for the application  $a$  at time  $t$ , setting  $u_k^a(t) = 1$  if  $k$  is active at time  $t$  and makes a request of type  $a$  and  $u_k^a(t) = 0$  otherwise. Therefore, the number  $N^{a,b}(t, l)$  of active customers with contract  $b$  for the application  $a$  at time  $t$  and location  $l$  is given by  $N^{a,b}(t, l) = \sum_{k \in \mathcal{K}^b} u_k^a(t) \mathbb{1}(L_t^k = l)$ , where  $\mathbb{1}$  denotes the indicator function, and the total number of active customers  $N(t, l)$  at time  $t$  and location  $l$  is given by  $N(t, l) = \sum_a \sum_b N^{a,b}(t, l)$ .

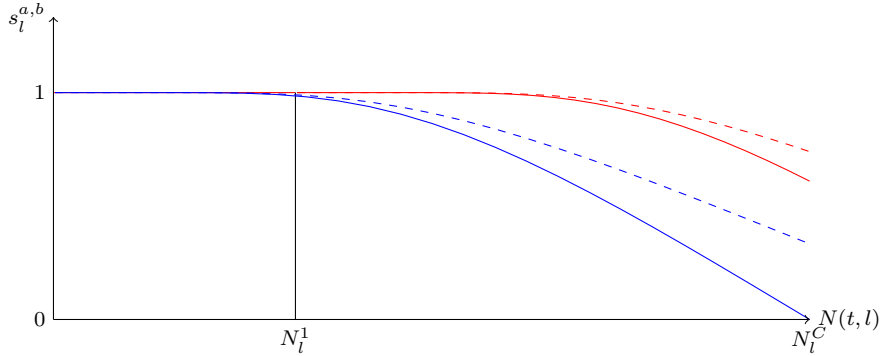
We consider the following two-stage model of price incentives. The first stage consists for the operator in announcing a discount  $y^{a,b}(t, l)$  at time  $t$  and location  $l$  for the customers of contract  $b$  making requests of type  $a$ . We consider only nonnegative discounts, so  $y^{a,b}(t, l) \geq 0$ . The second stage models the behavior of customers who modify their consumption by taking the discounts into account. We will assume the preference of a customer  $k$  of contract  $b$  for consuming at time  $t$  becomes  $\rho_k^a(t) + \alpha_k^a y^{a,b}(t, L_t^k)$ , where  $\alpha_k^a$  denotes the sensitivity of customer  $k$  to price incentives for the application  $a$ . It corresponds to classical linear utility functions, see e.g. [3]. We also assume that the customers cannot make more than one request at each time, that is  $\forall t \in [T], \sum_a u_k^a(t) \leq 1$ . Therefore, each customer  $k$  determines his consumptions  $u_k^a = (u_k^a(t))_{t \in [T]} \in \{0, 1\}^T$  for the applications, as an optimal solution of the linear program:

### Problem 1 (Low-level, customers)

$$\begin{aligned} & \max_{u_k^a \in \{0,1\}^T} \sum_{a \in [A]} \sum_{t=1}^T [\rho_k^a(t) + \alpha_k^a y^{a,b}(t, L_t^k)] u_k^a(t) & (1) \\ \text{s.t. } & \forall a \in [A], \sum_{t=1}^T u_k^a(t) = R_k^a, \quad \forall t \in [T], \sum_{a \in [A]} u_k^a(t) \leq 1 \\ & \forall t \in \mathcal{I}_k^a, \forall a \in [A], u_k^a(t) = 0 . \end{aligned}$$

Consequently, each price  $y^{a,b} = (y^{a,b}(t, l))_{t \in [T], l \in [L]}$  determines the possible individual consumptions  $u_k^a$  for the users with contract  $b$ , and so the possible cumulated traffic vectors  $N^{a,b} = (N^{a,b}(t, l))_{t \in [T], l \in [L]}$  and  $N = \sum_a \sum_b N^{a,b}$ . The aim of the operator is, through price incentives, to balance the load in the network into the different locations and time slots to improve the

quality of service perceived by each customer. We introduce a coefficient  $\gamma_b$  relative to the kind of contracts of the different customers in order to favor some classes of premium customers. In [16], Lee et al. suppose that the satisfaction of a customer depends on his perceived throughput, which can be considered as inversely proportional to the number of customers in the cell. Here, we assume that the satisfaction of each customer  $k$  in the cell  $l \in [L]$  is a nonincreasing function  $s_l^{a,b}$  of the total number of active customers in the cell  $N(t, l)$ , depending on the characteristics of the cell, of the type of application the user wants to do (some applications like streaming need a higher rate than others) and on the type of contract. We also assume that the satisfaction of all the customers with contract  $b$  using a given application  $a$  in a given cell is maximal until the number of active customers reaches a certain threshold  $N_l^{a,b}$ , then  $s_l^{a,b}(N(t, l)) = 1$  for  $N(t, l) \leq N_l^{a,b}$ . After this threshold, the satisfaction decreases until a critical value  $N_l^C$ . We add the constraint  $\forall t \in [T], \forall l \in [L], N(t, l) \leq N_l^C$  to prevent the congestion. For non-real time services like web, mail, download, the satisfaction function can be viewed as a concave function of the throughput, like  $1 - e^{-\delta/\delta_c}$  where  $\delta$  denotes the throughput, see Moety et al. [21]. Hence, we will consider that for contents like web, mail and download,  $N_l^{a,b} = N_l^1$ ,  $s_l^{a,b}(n) = 1$  for  $n \leq N_l^1$  and  $s_l^{a,b}(n) = 1 - \lambda_b \exp\left(-\frac{2N_l^C}{n - N_l^1}\right)$  for  $N_l^1 \leq n \leq N_l^C$  where  $\lambda_b$  is a positive parameter depending on the kind of contract of the customer. The more expensive the contract of the customer is, the larger is  $\lambda_b$ . We can prove that this function is concave for  $0 \leq n \leq N_l^C$ . For real time services like video streaming, the customers need a more important throughput to ensure a good QoS [16]. We will here consider the same type of functions  $s_l^{a,b}$  but with  $N_l^1$  replaced by  $N_l^{a,b} = 0$ , that is  $s_l^{a,b}(n) = 1 - \lambda_b \exp\left(-\frac{2N_l^C}{n}\right)$  for  $0 < n \leq N_l^C$ .



**Fig. 1** Different kind of satisfaction functions of the number of active customers in a cell. The blue ones are those for streaming contents whereas the red ones are those for web, mail and download contents. The dashed ones corresponds to the satisfaction of standard customers, the continuous ones to the satisfaction of premium customers.

So, the first stage consists in maximizing the global satisfaction function  $s$  which depends on the vectors  $N^{a,b} \in \mathbb{N}^{T \times L}$  and is defined by:

$$\begin{aligned}
s(N^{a,b}) &= \sum_{t=1}^T \sum_{a \in [A]} \sum_{b \in [B]} \sum_{k \in \mathcal{K}^b} \gamma_b s_{L_t^k}^{a,b}(N(t, L_t^k)) u_k^a(t) \\
&= \sum_{t=1}^T \sum_{a \in [A]} \sum_{b \in [B]} \sum_{k \in \mathcal{K}^b} \sum_{l=1}^L \gamma_b s_l^{a,b}(N(t, l)) \mathbf{1}(L_t^k = l) u_k^a(t) \\
&= \sum_{t=1}^T \sum_{l=1}^L \sum_{a \in [A]} \sum_{b \in [B]} \gamma_b N^{a,b}(t, l) s_l^{a,b}(N(t, l))
\end{aligned}$$

with  $\forall b \in [B], \gamma_b > 0$ . Our final model consists in solving the following bilevel program:

**Problem 2 (High-level, provider)**

$$\max_{y^{a,b} \in \mathbb{R}_+^{T \times L}} \sum_{t=1}^T \sum_{l=1}^L \sum_{a \in [A]} \sum_{b \in [B]} \gamma_b N^{a,b}(t, l) s_l^{a,b}(N(t, l)) \quad (2)$$

where  $\forall t \in [T], l \in [L], N(t, l) = \sum_{a=1}^A \sum_{b=1}^B N^{a,b}(t, l)$ , and  $N(t, l) \leq N_l^C, \forall t \in [T], l \in [L], a \in [A], b \in [B], N^{a,b}(t, l) = \sum_{k \in \mathcal{K}^b} u_k^a(t) \mathbf{1}(L_t^k = l)$ , and  $\forall k \in [K]$ , the vectors  $u_k^a$  are solutions of Problem 1.

### 3 A decomposition approach for solving the first model

We will present a decomposition method for solving the previous bilevel problem. In this section, and in the next two ones, we suppose that there is only one kind of application and one kind of contract. This special case is already relevant in applications: it covers the case when, for instance, only the download requests are influenced by price incentives, whereas other requests like streaming or web are fixed. Whereas the analytical results of the present section carry over to the general model, the results of the next two sections (polynomial time solvability) are only valid under these restrictive assumptions. We shall return to the general case in Section 6, developing a fast approximate algorithm for the general model based on the present principles.

In the above special case, the bilevel model can be rewritten:

$$\max_{y \in \mathbb{R}_+^{T \times L}} \sum_{t=1}^T \sum_{l=1}^L N(t, l) s_l(N(t, l))$$

where  $\forall t, l N(t, l) \leq N_l^C$  and  $N(t, l) = \sum_{k \in [K]} u_k^*(t) \mathbf{1}(L_t^k = l)$ , and for each  $k \in [K]$  the vectors  $u_k^*$  are solutions of the problem:

$$\begin{aligned}
&\max_{u_k \in \{0,1\}^T} \sum_{t=1}^T [\rho_k(t) + \alpha_k y(t, L_t^k)] u_k(t) \\
&\text{s.t. } \sum_{t=1}^T u_k(t) = R_k, \quad \forall t \in \mathcal{I}_k, u_k(t) = 0,
\end{aligned}$$

In order to deal more abstractly with the bilevel model, we introduce the notation  $u_k(t, l) = u_k(t) \mathbf{1}(L_t^k = l)$ . Hence, we have  $u_k(t, l) = 0$  if  $L_t^k \neq l$ . By defining the set  $\mathcal{J}_k = \{(t, l) \mid t \in \mathcal{I}_k \text{ or}$

$L_t^k \neq l\}$ , we have that  $(t, l) \in \mathcal{J}_k$  implies that  $u_k(t, l) = 0$ . We can then define  $\rho_k(t, l) = \rho_k(t)/\alpha_k$  if  $(t, l) \notin \mathcal{J}_k$  and  $\rho_k(t, l) = -\infty$  otherwise. Then, we can rewrite each low-level problem as:

$$\max_{u_k \in F_k} \sum_{t,l} [\rho_k(t, l) + y(t, l)] u_k(t, l)$$

where  $F_k = \{u \in \{0, 1\}^{T \times L} \mid \sum_{t,l} u(t, l) = R_k \text{ and } \forall (t, l) \in \mathcal{J}_k, u(t, l) = 0\}$ , and the global bilevel problem becomes:

$$\max_{y \in \mathbb{R}_+^{T \times L}} \sum_{t,l} f_l(N(t, l)) \quad \text{s.t. } \forall (t, l) \quad N(t, l) \leq N_t^C, \quad N(t, l) = \sum_{k=1}^K u_k(t, l)$$

with  $f_l : x \in \mathbb{R}_+ \mapsto x s_l(x)$ . Notice that the set  $\mathcal{J}_k$  corresponds to the set of couples  $(t, l)$  such that  $\rho_k(t, l) = -\infty$ . It is possible to enumerate all the couples  $(t, l) \in [T] \times [L]$ . Let us define  $n = T \times L$  and associate each couple  $(t, l)$  to an integer  $i \in [n]$ . The quantities  $\rho_k(t, l)$ ,  $u_k(t, l)$ ,  $N(t, l)$  and  $y(t, l)$  can be respectively denoted by  $\rho_k(i)$ ,  $u_k(i)$ ,  $N_i$  and  $y_i$ . The function  $f_l$  and the integer  $N_t^C$  can be respectively denoted by  $f_i$  and  $N_i^C$ . It means that for two indices  $i$  and  $j$  associated to two couples  $(t, l)$  and  $(t', l)$  with the same  $l$ , we have  $f_i = f_j := f_l$  and  $N_i^C = N_j^C := N_l^C$ . The low-level problem can be rewritten:

**Problem 3 (Abstract low-level problem)**

$$\max_{u_k \in F_k} \sum_{i=1}^n [\rho_k(i) + y_i] u_k(i) \quad (3)$$

where  $F_k = \{u \in \{0, 1\}^n \mid \sum_{i=1}^n u(i) = R_k \text{ and } \forall i \in \mathcal{J}_k, u(i) = 0\}$ .

The global bilevel problem is:

**Problem 4 (Bilevel problem)**

$$\max_{y \in \mathbb{R}_+^n} \sum_{i=1}^n f_i(N_i) \quad \text{s.t. } \forall i, N_i \leq N_i^C, \quad N_i = \sum_{k=1}^K u_k^*(i) \quad (4)$$

with for all  $k \in [K]$ ,  $u_k^*$  solution of Problem 3.

**Lemma 1** *Suppose that the functions  $s_i$  are nonincreasing and concave on  $[0, N_i^C]$ . Then, the functions  $f_i$  are also concave on  $[0, N_i^C]$ .*

*Proof.* The result comes easily if we suppose that the functions  $s_i$  are twice differentiable, because we have:

$$\forall x \in [0, N_i^C], f_i''(x) = x s_i''(x) + 2s_i'(x) \leq 0 .$$

We could deduce that the same is true without the differentiability assumption by a density argument, writing a concave function as a pointwise limit of smooth concave functions. However, we prefer to provide the following elementary argument. Consider  $0 \leq x \leq y \leq N_i^C$  and  $t \in [0, 1]$ . Because  $s_i$  is nonincreasing, we have  $s_i(x) \geq s_i(y)$ . We have:

$$\begin{aligned} t f_i(x) + (1-t) f_i(y) &= t x s_i(x) + (1-t) y s_i(y) \\ &= (t x + (1-t) y) \left[ \frac{t x}{t x + (1-t) y} s_i(x) + \frac{(1-t) y}{t x + (1-t) y} s_i(y) \right] \\ &\leq (t x + (1-t) y) s_i \left( \frac{t x^2 + (1-t) y^2}{t x + (1-t) y} \right) \end{aligned}$$



Because of the well-known inequality  $2xy \leq x^2 + y^2$ , we have:

$$\begin{aligned} (tx + (1-t)y)^2 &= t^2x^2 + (1-t)^2y^2 + 2t(1-t)xy \\ &\leq tx^2 + (1-t)y^2 . \end{aligned}$$

Then, because  $s_i$  is nonincreasing, we have:

$$s_i \left( \frac{tx^2 + (1-t)y^2}{tx + (1-t)y} \right) \leq s_i(tx + (1-t)y) ,$$

so that:

$$tf_i(x) + (1-t)f_i(y) \leq (tx + (1-t)y)s_i(tx + (1-t)y) = f_i(tx + (1-t)y) ,$$

and  $f_i$  is concave. □

### 3.1 A tropical representation of customers' response

The lower-level component of our bilevel problem can be studied thanks to tropical techniques. Tropical mathematics refers to the study of the max-plus semifield  $\mathbb{R}_{\max}$ , that is the set  $\mathbb{R} \cup \{-\infty\}$  endowed with two laws  $\oplus$  and  $\odot$  defined by  $a \oplus b = \max(a, b)$  and  $a \odot b = a + b$ , see [2, 15, 7, 18] for background. We first consider the relaxation in which the price vector  $y$  can take any real value, i.e.  $y \in \mathbb{R}^n$ . Each customer  $k$  defines his consumption  $u_k^*$  by solving the problem:

$$\max_{u_k \in F_k} \sum_i [\rho_k(i) + y_i] u_k(i) = \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle . \quad (5)$$

The map  $P_k : y \mapsto \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle$  is convex, piecewise affine, and the gradients of its linear parts are integer valued. It can be thought of as a tropical polynomial function in the variable  $y$ . Indeed, with the tropical notation, we have

$$P_k(y) = \bigoplus_{u_k \in F_k} \left[ \bigodot_{i \in [n]} (\rho_k(i) \odot y_i)^{\odot u_k(i)} \right]$$

where  $z^{\odot p} := z \odot \dots \odot z = p \times z$  denotes the  $p$ th tropical power. In this way, we see that all the monomials of  $P_k$  have degree  $\sum_i u_k(i) = R_k$ , so that  $P_k$  is homogeneous of degree  $R_k$ , in the tropical sense. This remark leads to the following lemma:

**Lemma 2** *Denote by  $e = (1 \dots 1) \in \mathbb{R}^n$ . Let  $y$  be a solution of the relaxation  $y \in \mathbb{R}^n$  of Problem 4. Then, for all  $\beta \in \mathbb{R}$ ,  $y + \beta e$  is a solution of the relaxation  $y \in \mathbb{R}^n$  of Problem 4.*

*Proof.* Consider a solution  $y \in \mathbb{R}^n$  of the relaxed problem. Because  $P_k$  is homogeneous of degree  $R_k$ , we have for all  $\beta \in \mathbb{R}$ ,  $P_k(y + \beta e) = P_k(y) + \beta R_k$ . In particular:

$$u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle \Leftrightarrow u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y + \beta e, u_k \rangle$$

Hence,  $y + \beta e$  leads to the same repartition of the customers  $N^*$  and corresponds also to an optimal solution of the relaxed bilevel problem. □

**Corollary 1** *The bilevel problem 4 has the same value as its relaxation  $y \in \mathbb{R}^n$ .*

*Proof.* Consider a solution  $y^* \in \mathbb{R}^n$  of the relaxed problem, and take  $\beta \geq -\min_i y_i^*$ . Then, we have  $y^* + \beta e \in R_+^n$  and solution of the relaxed problem according to Lemma 2. Consequently,  $y^* + \beta e$  is a solution of Problem 4.  $\square$

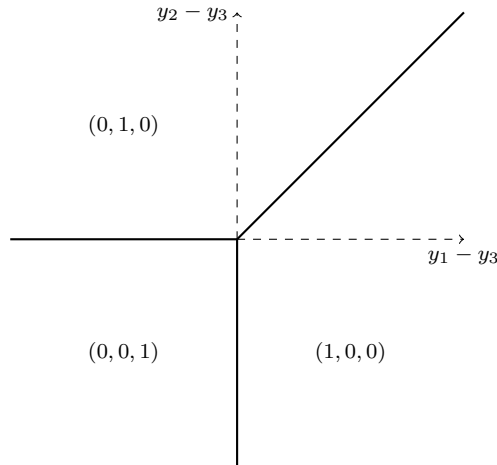
By definition, the *tropical hypersurface* associated to a tropical polynomial function is the nondifferentiability locus of this function. Since the monomial  $P_k$  is homogeneous, its associated tropical hypersurface is invariant by the translation by a constant vector. Therefore, it can be represented as a subset of the tropical projective space  $\mathbb{TP}^{n-1}$ . The latter is defined as the quotient of  $\mathbb{R}^n$  by the equivalence relation which identifies two vectors which differ by a constant vector, and it can be identified to  $\mathbb{R}^{n-1}$  by the map

$$\mathbb{TP}^{n-1} \rightarrow \mathbb{R}^{n-1}, y \mapsto (y_i - y_n)_{i \in [n-1]}.$$

*Example 1* Consider a simple example with  $T = 3$  time steps (for instance morning, afternoon and evening),  $L = 1$  (that is  $n = 3$ ),  $K = 5$  and  $\mathcal{J}_k = \emptyset$  for each  $k$ . The parameters of the customers are

$$\begin{aligned} \rho_1 &= [0, 0, 0], R_1 = 1, & \rho_2 &= [0, -1, 0], R_2 = 2, \\ \rho_3 &= [-1, 1, 0], R_3 = 1 & \rho_4 &= [1/2, 1/2, 0], R_4 = 2, \\ \rho_5 &= [1/2, 2, 0], R_5 = 1. \end{aligned}$$

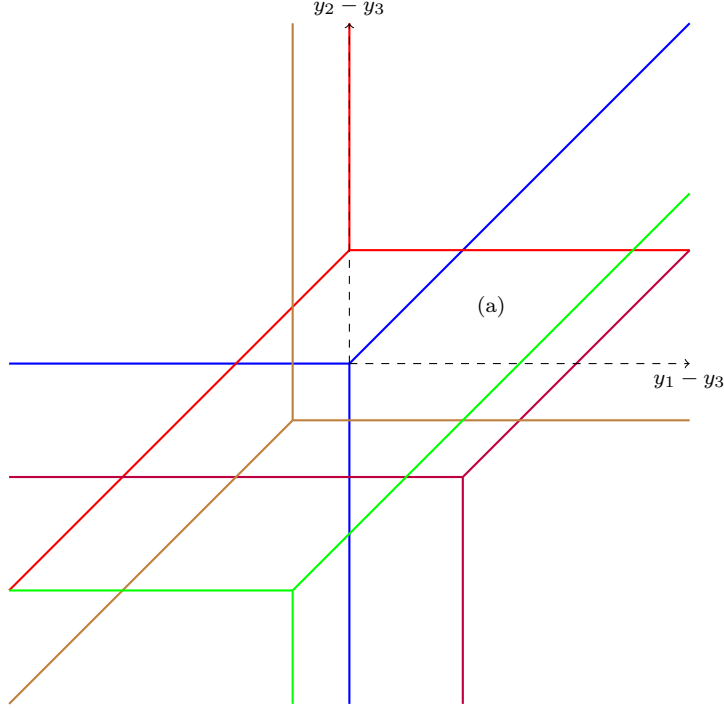
The tropical polynomial of the first customer is  $P_1(y) = \max(y_1, y_2, y_3)$ , meaning that this customer has no preference and consumes when the incentive is the best. Its associated tropical hypersurface is a tropical line (since  $P_1$  has degree 1), so it splits  $\mathbb{TP}^2$  in three different regions corresponding to a choice of the vector  $u_1$  among  $(1, 0, 0)$ ,  $(0, 1, 0)$  and  $(0, 0, 1)$ , see Figure 2. E.g., the cell labeled by  $(1, 0, 0)$  represents a consumption concentrated the morning, induced by a price  $y_1 > y_2$  and  $y_1 > y_3$ .



**Fig. 2** A customer response: a tropical line splits the projective space into three cells. Each cell corresponds to a possible customer response

To study jointly the responses of the five customers, we represent the arrangement of the tropical hypersurfaces associated to the  $P_k$ ,  $k \in [5]$  (see Figure 3), with

$$\begin{aligned} P_2(y) &= \max(y_1 + y_2 - 1, y_1 + y_3, y_2 + y_3 - 1), \\ P_3(y) &= \max(y_1 - 1, y_2 + 1, y_3), \\ P_4(y) &= \max(y_1 + y_2 + 1, y_1 + y_3 + 1/2, y_2 + y_3 + 1/2), \\ P_5(y) &= \max(y_1 + 1/2, y_2 + 2, y_3). \end{aligned}$$



**Fig. 3** Arrangement of tropical hypersurfaces: each tropical hypersurface corresponds to a customer response. For example, the cell (a) corresponds to discounts  $y$  with responses  $(1,0,0)$  for customer 1,  $(1,0,1)$  for customer 2,  $(0,1,0)$  for customer 3,  $(1,1,0)$  for customer 4 and  $(0,1,0)$  for customer 5. Hence, the total number of customers in the network with these discounts is  $(3,3,1)$ .

**Lemma 3 (Corollary of [30, §4, Lemma 3.1])** *Each cell of the arrangement of tropical hypersurfaces corresponds to a collection of customers responses  $(u_1, \dots, u_K)$  and to a unique traffic vector  $N$ , defined by  $N = \sum_k u_k$ .*

### 3.2 Decomposition theorem

We next show that the present bilevel problem can be solved by decomposition. We note that the function to optimize for the higher level problem, i.e. the optimization problem of the provider, depends only on  $N$ . The variables  $y_i$  allow one to generate the different possible vectors  $N$ .

**Definition 1** A vector  $N \in \mathbb{Z}^n$  is said to be *feasible* if there exists  $K$  vectors  $u_1^*, \dots, u_K^*$  such that  $N = \sum_{k=1}^K u_k^*$  and there exists  $y \in \mathbb{R}^n$  such that for each  $k \in [K]$ ,  $u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle$ .

So, we will characterize the feasible vectors  $N$  in order to optimize directly the satisfaction function on the set of feasible  $N$ . We define the relaxation of Problem 4 to the case  $y \in \mathbb{R}^n$ .

**Problem 5 (Bilevel problem with real discounts)**

$$\max_{y \in \mathbb{R}^n} \sum_{i=1}^n f_i(N_i) \quad \text{s.t. } \forall i, N_i \leq N_i^C,$$

with  $N = \sum_{k=1}^K u_k^*$  and for all  $k \in [K]$ ,  $u_k^*$  solution of:

$$\max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle.$$

According to Lemma 1, Problem 4 has the same value than the relaxation problem 5. Moreover, according to Lemma 2, if  $(y^*, N^*)$  is an optimal solution of Problem 5, then  $(y^* + \beta e, N^*)$  is also an optimal solution of Problem 5 for every  $\beta \in \mathbb{R}$ . We recall that  $e \in \mathbb{R}^n$  is a vector defined by  $e^T = (1, \dots, 1)$ . Then, if we find an optimal solution  $(y^*, N^*)$  of Problem 5, then  $(y^* + \beta e, N^*)$  with  $\beta = -\min_{i \in [n]} y_i^*$  is a solution of Problem 5 such that  $y^* + \beta e \in \mathbb{R}_+^n$ . Consequently,  $(y^* + \beta e, N^*)$  is a solution of Problem 4. Hence, a solution of Problem 5 (with real discounts) provides a solution of Problem 4 (with nonnegative discounts). In the sequel, we will study the bilevel problem 5.

Most of the following results are applications of classical notions of convex analysis which can be found in [25]. It is convenient to introduce the convex characteristic function  $\chi_A$  of a set  $A \subset \mathbb{R}^n$ , defined by  $\chi_A(x) = 0$  if  $x \in A$ , and  $\chi_A(x) = +\infty$  otherwise. If  $A$  is a convex set, then  $\chi_A$  is a convex function. We define also for every  $k$  the polytope  $\Delta_k$  as the convex hull of  $F_k$ , together with the convex function  $\varphi_k$  defined by  $\varphi_k(u) = -\langle \rho_k, u \rangle + \chi_{\Delta_k}(u)$ .

**Lemma 4**  $\Delta_k = \{u \in [0, 1]^n \mid \sum_{i=1}^n u(i) = R_k \text{ and } \forall i \in \mathcal{J}_k, u(i) = 0\}$  and  $F_k$  is exactly the set of vertices of  $\Delta_k$ .

*Proof.* Let us define the polytope  $\Delta'_k = \{u \in [0, 1]^n \mid \sum_{i=1}^n u(i) = R_k \text{ and } \forall (t, l) \in \mathcal{J}_k, u(i) = 0\}$ . Clearly,  $F_k \subset \Delta'_k$ . Then,  $\Delta_k \subset \Delta'_k$ .

Consider a point  $u$  of  $\Delta'_k$  which is not in  $F_k$ . There exists an index  $i$  such that  $0 < u(i) < 1$ . In particular  $u(i) \notin \mathbb{N}$ . However,  $\sum_i u(i) = R_k \in \mathbb{N}$ . So, there exists another index  $j$  such that  $0 < u(j) < 1$ . Hence, there exists  $\varepsilon > 0$  such that the points  $u^-$  and  $u^+$  defined by:

$$\begin{cases} u^-(i) = u(i) - \varepsilon \text{ and } u^+(i) = u(i) + \varepsilon \\ u^-(j) = u(j) + \varepsilon \text{ and } u^+(j) = u(j) - \varepsilon \\ u^-(k) = u^+(k) = u(k) \text{ otherwise} \end{cases}$$

are in  $\Delta'_k$ . Because  $x = \frac{x^- + x^+}{2}$  with  $x \neq x^-$  and  $x \neq x^+$ ,  $x$  is not a vertex of  $\Delta'_k$ . Consequently, the set of vertices of  $\Delta'_k$  is included in  $F_k$ . Because  $\Delta'_k$  is the convex hull of its vertices, we have  $\Delta'_k \subset \Delta_k$ .

The polytope  $\Delta_k$  is such that  $\Delta_k = \{u \in \mathbb{R}^n \mid 0 \leq u \leq a \text{ and } e^T u = R_k\}$ , with  $a(i) = 0$  if  $i \in \mathcal{J}_k$  and  $a(i) = 1$  otherwise, and  $e^T = (1, \dots, 1) \in \mathcal{M}_{1,n}(\mathbb{R})$ . Then, because  $e^T$  is a totally unimodular matrix, the vertices of  $\Delta_k$  are exactly its integer points, that is  $F_k$ .  $\square$

**Corollary 2** *The value of each low level problem 3 is the value of the Legendre-Fenchel transform of  $\varphi_k$  at point  $y$ , i.e.  $\varphi_k^*(y) = \sup_{u_k \in \Delta_k} [\langle y, u_k \rangle - \varphi_k(u_k)]$ .*

*Proof.* The vertices of  $\Delta_k$  are  $F_k$ . Hence:

$$\max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle = \sup_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle = \sup_{u_k \in \Delta_k} \langle y, u_k \rangle - \varphi_k(u_k)$$

$\square$

We want to characterize the feasible vectors. We have first the following result.

**Lemma 5** *Let  $N$  be a real vector. Then, there exists  $y \in \mathbb{R}^n$  and  $u_1^*, \dots, u_K^*$  such that  $N = \sum_{k \in [K]} u_k^*$  and for every  $k \in [K]$ ,  $u_k^* \in \arg \max_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle$  if and only if  $N \in \sum_{k \in [K]} \Delta_k$ .*

*Proof.* Such vectors  $u_k^*$  belong to  $\Delta_k$ , so  $N \in \sum_{k \in [K]} \Delta_k$ .

Let  $k \in [K]$  and  $y \in \mathbb{R}^n$ . A vector  $u_k^* \in \Delta_k$  is such that  $u_k^* \in \arg \max_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle$  if and only if  $u_k^* \in \partial \varphi_k^*(y)$ , where  $\partial \varphi_k^*$  denotes the subdifferential of the convex function  $\varphi_k^*$ . Then, a vector  $N = \sum_k u_k^*$  if and only if  $N \in \sum_k \partial \varphi_k^*(y)$ . By [25, Th. 23.8],  $\sum_k \partial \varphi_k^*(y) = \partial (\sum_k \varphi_k^*)(y) = \partial \psi^*(y)$ , where  $\psi = \square_k \varphi_k$  is the inf-convolution of the functions  $\varphi_k$ .

Let  $N$  be a real vector. Then, there exists  $y \in \mathbb{R}^n$  and  $u_1^*, \dots, u_K^*$  such that  $N = \sum_{k \in [K]} u_k^*$  and for every  $k \in [K]$ ,  $u_k^* \in \arg \max_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle$  if and only if  $N \in \partial \psi^*(y)$ , or equivalently  $y \in \partial \psi(N)$  (because  $\psi$  is convex), that is if and only if  $\partial \psi(N) \neq \emptyset$ . The function  $\psi$  is polyhedral (as the inf-convolution of polyhedral convex functions) and it is finite at every point in  $\sum_k \Delta_k$ . So,  $\forall N' \in \sum_k \Delta_k$ ,  $\partial \psi(N')$  is a non-empty polyhedral convex set [25, Th. 23.10]. The result comes straightforwardly.  $\square$

It is now possible to characterize the feasible vectors.

**Lemma 6** *A vector  $N \in \mathbb{Z}^n$  is feasible if and only if  $N \in \sum_k F_k$ .*

*Proof.* According to Definition 1, a vector  $N \in \mathbb{Z}^n$  is feasible if and only if there exists  $y \in \mathbb{R}^n$  and  $K$  vectors  $(u_k^*)_{k \in [K]}$  such that  $N = \sum_k u_k^*$  and  $u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle$ . As a consequence of Lemma 4,  $\arg \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle = \arg \max_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle$ . Then, by Lemma 5, a vector  $N \in \mathbb{Z}^n$  is feasible if and only if  $N \in (\sum_{k \in [K]} \Delta_k) \cap \mathbb{Z}^n$ . We have now to prove  $\sum_{k \in [K]} F_k = (\sum_{k \in [K]} \Delta_k) \cap \mathbb{Z}^n$ . Because  $F_k = \Delta_k \cap \mathbb{Z}^n$ , the inclusion  $\sum_{k \in [K]} F_k \subset (\sum_{k \in [K]} \Delta_k) \cap \mathbb{Z}^n$  is obvious. Conversely, consider  $N \in (\sum_{k \in [K]} \Delta_k) \cap \mathbb{Z}^n$ . Then, the set  $\Delta_N = \{(u_1, \dots, u_K) \in \Delta_1 \times \dots \times \Delta_K \mid \sum_{k=1}^K u_k = N\}$  is a non-empty polytope. A vector  $u = (u_1, \dots, u_K)$  belongs to  $\Delta_N$  if it satisfies the following constraints:

$$\begin{cases} \forall k, i, 0 \leq u_k(i) \leq \mathbb{1}_{i \in \mathcal{J}_k} , \\ \forall k, \sum_i u_k(i) = R_k , \\ \forall i, \sum_k u_k(i) = N_i . \end{cases}$$

, that is  $\Delta_N = \{u \in \mathbb{R}^{Kn} \mid 0 \leq u \leq a, Au = b\}$ , with  $a \in \mathbb{R}^{Kn}$  such that  $a_k(i) = \mathbb{1}_{i \in \mathcal{J}_k}$  for every  $i \in [n], k \in [K]$ , and  $A \in \mathcal{M}_{K+n, Kn}(\mathbb{Z})$  and  $b \in \mathbb{Z}^{K+n}$  defined by:

$$A = \begin{pmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & \dots & 0 & 0 & \dots & 0 \\ & & & & & & \dots & & & & & & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 1 & 1 & \dots & 1 \\ -1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 & \dots & -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 & 0 & -1 & \dots & 0 & \dots & 0 & -1 & \dots & 0 \\ & & & & & & \dots & & & & & & \\ 0 & 0 & \dots & -1 & 0 & 0 & \dots & -1 & \dots & 0 & 0 & \dots & -1 \end{pmatrix} \text{ and } b = \begin{pmatrix} R_1 \\ R_2 \\ \dots \\ R_K \\ -N_1 \\ -N_2 \\ \dots \\ -N_n \end{pmatrix}$$

By Poincaré's lemma,  $A$  is totally unimodular. In particular, the extreme points of  $\Delta_N$  are integer. Then, there exists  $(u_1^*, \dots, u_K^*)$  with for every  $k \in [K]$ ,  $u_k^* \in \Delta_k \cap \mathbb{Z}^n = F_k$  such that  $N = \sum_k u_k^*$ .  $\square$

Each vector  $N \in \sum_k F_k$  can be written as sum of vectors  $u_k^* \in F_k$  for  $k \in [K]$  such that there exists  $y \in \mathbb{R}^n$  with  $u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle$ . In order to determine such vectors  $u_k^*$ , we have the following lemma:

**Lemma 7** *Let  $N = \sum_k u_k^*$  with  $u_k^* \in \Delta_k \forall k$ . The following assertions are equivalent:*

1. *There exists  $y \in \mathbb{R}^n$  such that for each  $k \in [K]$ ,  $u_k^* \in \arg \max_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle$ .*
2. *The vectors  $u_1^*, \dots, u_K^*$  realize the minimum in the inf-convolution  $\psi$ , i.e.*

$$\psi(N) = - \sum_k \langle \rho_k, u_k^* \rangle$$

*Proof.* (1)  $\Rightarrow$  (2) : We have for every  $k$ :

$$\langle \rho_k + y, u_k^* \rangle = \sup_{u_k \in \Delta_k} [\langle \rho_k + y, u_k \rangle]$$

By summing those equalities, we have:

$$\begin{aligned} \langle y, N \rangle + \sum_k \langle \rho_k, u_k^* \rangle &= \sum_k \sup_{u_k \in \Delta_k} [\langle \rho_k + y, u_k \rangle] \\ &= \sup_{u_1 \in \Delta_1, \dots, u_K \in \Delta_K} \sum_k [\langle \rho_k + y, u_k \rangle] \end{aligned}$$

By considering only the vectors  $u_1 \in \Delta_1, \dots, u_K \in \Delta_K$  such that  $\sum_k u_k = N$ , we can write  $\sum_k \langle \rho_k, u_k^* \rangle = \sup_{\substack{u_1 \in \Delta_1, \dots, u_K \in \Delta_K \\ \sum_k u_k = N}} \sum_k \langle \rho_k, u_k \rangle$  which is exactly the second assertion.

(2)  $\Rightarrow$  (1): The set  $\partial\psi(N)$  is non-empty. Consider  $y \in \partial\psi(N)$ , that is  $N \in \partial\psi^*(y)$ . We can write:

$$\forall N' \in \sum_k \Delta_k, \langle y, N \rangle - \psi(N) \geq \langle y, N' \rangle - \psi(N')$$

So:

$$\begin{aligned} \sum_k \langle \rho_k + y, u_k^* \rangle &= \langle y, N \rangle + \sum_k \langle \rho_k, u_k^* \rangle = \langle y, N \rangle - \psi(N) \\ &= \sup_{N' \in \sum_k \Delta_k} [\langle y, N' \rangle - \psi(N')] \\ &= \sup_{N' \in \sum_k \Delta_k} \left[ \langle y, N' \rangle + \sup_{\substack{u_1 \in \Delta_1, \dots, u_K \in \Delta_K \\ \sum_k u_k = N'}} \sum_k \langle \rho_k, u_k \rangle \right] \\ &= \sup_{u_1 \in \Delta_1, \dots, u_K \in \Delta_K} \sum_k \langle \rho_k + y, u_k \rangle \\ &= \sum_k \sup_{u_k \in \Delta_k} \langle \rho_k + y, u_k \rangle \end{aligned}$$

Consequently, if one  $u_k^*$  is not an optimal solution of the low-level problem, the previous equality cannot be true.  $\square$

The high-level problem of Problem 5 consists in maximizing a function depending only on a vector  $N$  which has to be a feasible vector. It is now possible to write the main theorem of this section, which establishes a decomposition method for solving Problem 5.

**Theorem 1** (Decomposition) *The bilevel problem 5 can be solved as follows:*

1. Find an optimal solution  $N^*$  to the high level problem with unknown  $N$ :

$$\max_{N \in \sum_k F_k} \sum_{i=1}^n f_i(N_i) \quad \text{s.t. } \forall i, N_i \leq N_i^C. \quad (6)$$

2. Find vectors  $(u_1^*, \dots, u_K^*)$  solutions of the following problem:

$$\max_{\substack{u_1 \in F_1, \dots, u_K \in F_K \\ \sum_k u_k = N^*}} \sum_k \langle \rho_k, u_k \rangle.$$

3. Find a vector  $y^*$  such that  $\forall k, u_k^*$  is a solution of the low level problem.

*Proof.* The bilevel programming problem 5 can be rewritten  $\max_{N \text{ feasible}} \sum_i f_i(N_i)$  subject to  $\forall i \in [n], N_i \leq N_i^C$ . According to Lemma 6,  $N$  is feasible if and only if  $N \in \sum_k F_k$ . So, a necessary condition for a vector  $y^*$  to be an optimal solution of the bilevel problem is that for every  $k$ , there exists  $u_k^* \in \partial\varphi_k^*(y^*)$  such that  $N^* = \sum_k u_k^*$  is an optimal solution of the problem:

$$\begin{aligned} \max_{N \in \sum_k F_k} \sum_i f_i(N_i) \\ \text{s.t. } \forall i, N_i \leq N_i^C \end{aligned}$$

After finding  $N^*$ , it is possible to find  $u_k^* \in \partial\varphi_k^*(y^*)$  by solving the inf-convolution problem as a consequence of Lemma 7. Because  $u_k^* \in \partial\varphi_k^*(y^*)$  is equivalent to  $y^* \in \partial\varphi_k(u_k^*)$ , each point of  $\bigcap_k \partial\varphi_k(u_k^*)$  is an optimal solution of the bilevel problem.  $\square$

The second step of this theorem consists in solving a linear program. We next show that the third step reduces to a linear feasibility problem.

**Lemma 8** *Let  $N \in \mathbb{Z}^n$  be a feasible vector and  $u_k^* \in F_k$  ( $k \in [K]$ ) be vectors such that  $N = \sum_k u_k^*$  and  $\psi(N) = -\sum_k \langle \rho_k, u_k^* \rangle$ . Then, the set of vectors  $y^* \in \mathbb{R}^n$  such that  $\forall k \in [K], u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y^*, u_k \rangle$  is non-empty and is the polytope defined by the following inequalities:*

$$\forall k \in [K], \forall i, j \notin \mathcal{J}_k, \text{ such that } u_k^*(i) = 1, u_k^*(j) = 0, \rho_k(i) + y_i^* \geq \rho_k(j) + y_j^*$$

*Proof.* According to Lemma 7, there exists  $y^* \in \bigcap_k \partial\varphi_k(u_k^*)$ . Hence, we have  $\forall u_k \in F_k, \langle \rho_k + y^*, u_k^* \rangle \geq \langle \rho_k + y^*, u_k \rangle$ .

Consider indices  $i, j \notin \mathcal{J}_k$  with  $u_k^*(i) = 1, u_k^*(j) = 0$ , and the vector  $u_k$  defined by  $u_k(i) = 0, u_k(j) = 1$  and  $\forall l \neq i, j, u_k(l) = u_k^*(l)$ . We verify easily  $u_k \in F_k$ , so that the condition  $\langle \rho_k + y^*, u_k^* \rangle \geq \langle \rho_k + y^*, u_k \rangle$ , which can be rewritten  $\rho_k(i) + y_i^* \geq \rho_k(j) + y_j^*$ , is satisfied.

Moreover, this condition is sufficient. Consider  $y^*$  such that  $\forall i, j \notin \mathcal{J}_k$  with  $u_k^*(i) = 1, u_k^*(j) = 0$ , we have  $\rho_k(i) + y^*(i) \geq \rho_k(j) + y^*(j)$ . Consider  $u_k \in F_k$ . By definition of  $F_k$ , the

quantity  $\langle \rho_k + y^*, u_k \rangle$  corresponds to the sum of  $R_k$  coordinates of  $\rho_k + y^*$  for which the index is not in  $\mathcal{J}_k$ . Hence,

$$\begin{aligned} \langle \rho_k + y^*, u_k \rangle &= \sum_{i, u_k(i)=1, u_k^*(i)=1} (\rho_k(i) + y_i^*) + \sum_{j, u_k(j)=1, u_k^*(j)=0} (\rho_k(j) + y_j^*) \\ &\leq \sum_{i, u_k(i)=1, u_k^*(i)=1} (\rho_k(i) + y_i^*) + \sum_{j, u_k(j)=0, u_k^*(j)=1} (\rho_k(j) + y_j^*) = \langle \rho_k + y^*, u_k \rangle \end{aligned}$$

because of the lemma hypothesis and because  $\#\{j | u_k(j) = 1, u_k^*(j) = 0\} = \#\{j | u_k(j) = 0, u_k^*(j) = 1\}$

□

For every  $k$ , the latter inequalities define a polytope, and we have to find  $y^*$  in the intersection of all these polytopes.

#### 4 A first algorithm

In this section, we explain how the decomposition method provided by Theorem 1 leads to a polynomial time algorithm for solving Problem 5. We will use some elements of discrete convexity developed by Danilov, Koshevoy [9] and Murota [22], that we recall first. We next explain how to solve Problem 5.

An integer set  $B \subset \mathbb{Z}^n$  is *M-convex* [22, Ch. 4, p.101] if  $\forall x, y \in B, \forall i \in [n]$  such that  $x_i > y_i, \exists j \in [n]$  such that  $x_j < y_j, x - e_i + e_j \in B$  and  $y + e_i - e_j \in B$ , where  $e_i$  is the  $i$ -th vector of the canonical basis in  $\mathbb{R}^n$ .

**Lemma 9** *The feasible domain of the high-level program*

$$B = \{N \in \sum_k F_k | \forall i N_i \leq N_i^C\}$$

is a *M-convex set* of  $\mathbb{Z}^n$ .

*Proof.* We can check easily that  $\forall k$ , the set  $F_k$  is *M-convex*. Taking two different vectors  $u_k$  and  $v_k$  in  $F_k$ , there exist  $i, j$  such that  $u_k(i) = 1, v_k(i) = 0$  and  $u_k(j) = 0, v_k(j) = 1$ . These indices  $i, j$  do not belong to  $\mathcal{J}_k$ . The vectors  $u_k - e_i + e_j$  and  $v_k + e_i - e_j$  have coordinates in  $\{0, 1\}$  with a sum equal to  $R_k$  and all coordinates in  $\mathcal{J}_k$  equal to 0.

It is known that a Minkowski sum of *M-convex* sets is *M-convex* [22, Th. 4.23, p.115], and so the set  $\sum_k F_k$  is *M-convex*.

Finally, consider two vectors  $N$  and  $N'$  of  $B$ . They belong to  $\sum_k F_k$ , so for each  $i$  with  $N_i > N'_i$ , we can find  $j$  with  $N_j < N'_j$  such that  $N - e_i + e_j$  and  $N' + e_i - e_j$  are in  $\sum_k F_k$ . The  $i$ -th coordinate of  $N - e_i + e_j$  is  $N_i - 1 < N_i \leq N_i^C$  and the  $j$ -th coordinate of  $N - e_i + e_j$  is  $N_j + 1 \leq N'_j \leq N_j^C$ . So  $N - e_i + e_j \in B$  and similarly  $N' + e_i - e_j \in B$ , which proves the *M-convexity* of  $B$ .

□

A function  $g : \mathbb{Z}^n \mapsto \mathbb{R}$  is *M-convex* [22, ch. 6.1, p.133] if  $\forall x, y \in \mathbb{Z}^n$  such that  $g(x)$  and  $g(y)$  are finite real values,  $\forall i \in [n]$  such that  $x_i > y_i, \exists j \in [n]$  such that  $x_j < y_j$  and the following condition holds true:

$$g(x) + g(y) \geq g(x - e_i + e_j) + g(y + e_i - e_j)$$



A function  $g$  is  $M$ -concave if  $-g$  is  $M$ -convex. It follows from this definition that if  $B$  is a  $M$ -convex set, then  $\chi_B$  is a  $M$ -convex function (we recall that  $\chi_B : \mathbb{Z}^n \mapsto \mathbb{R}$  is defined by  $\chi_B(x) = 0$  if  $x \in B$  and  $\chi_B(x) = +\infty$  otherwise). An important property of  $M$ -convex functions is that local optimality guarantees global optimality [22, Th. 6.26, p.148] in the following sense. Let  $g$  be a  $M$ -convex function and  $x \in \mathbb{Z}^n$ . Then  $g(x) = \min_{y \in \mathbb{Z}^n} g(y)$  if and only if  $\forall i, j \in [n]$ ,  $g(x) \leq g(x - e_i + e_j)$ .

According to Theorem 1, we have to solve  $\max_{N \in \mathbb{Z}^n} f(N) - \chi_B(N)$ , where  $f : N \mapsto \sum_i f_i(N_i)$  is a separable concave function, and  $B$  is the  $M$ -convex set introduced in Lemma 9. The function  $f - \chi_B$  is  $M$ -concave [22, Th. 6.13.(4), p.143]. Then, we have the following result as a direct consequence of [22, Th. 6.26, p.148] :

**Theorem 2** *Let  $N^* \in B$ . Then,  $N^*$  is a maximum point of  $f$  over  $B$  if and only if  $\forall i, j \in [n]$  such that  $N^* - e_i + e_j \in B$ ,  $f(N^* - e_i + e_j) \leq f(N^*)$ .*

Moreover, Murota ([22], ch.10, p.281) gives an algorithm which runs in pseudo-polynomial time to minimize  $M$ -convex functions (see Algorithm 1).

---

**Algorithm 1** Murota's greedy algorithm to minimize a  $M$ -convex function  $g$ .

---

1. Find  $x \in \mathbb{Z}^n$  such that  $g(x) < +\infty$ ;
  2. Find  $i, j \in \arg \min_{k, l \in [n]} g(x - e_k + e_l)$ ;
  3. If  $g(x - e_i + e_j) \geq g(x)$  then stop ( $x$  is a global minimizer of  $g$ );
  4. Else  $x := x - e_i + e_j$  and go back to Step 2;
- 

By adding a priority rule in Step 2 of Algorithm 1 in the case where  $\arg \min_{k, l \in [n]} f(x - e_k + e_l)$  is not reduced to a single point, a global minimizer of  $f$  is obtained by Algorithm 1 in pseudo-polynomial time.

**Proposition 1 ([22], Prop.10.2)** *Assume that  $\text{dom } f$  is bounded. Let  $F$  be the number of arithmetic operations needed to evaluate  $f$  and  $K_1 = \max(\|x - y\|_1 \mid x, y \in \text{dom } f)$ . Then, if a vector in  $\text{dom } f$  is given, Algorithm 1 finds a global minimizer of  $f$  in  $O(Fn^2K_1)$  time.*

However, the minimization of a  $M$ -convex function can be achieved in polynomial time.

**Proposition 2 ([22], Prop.10.4)** *Assume that  $\text{dom } f$  is bounded. Let  $F$  be the number of arithmetic operations needed to evaluate  $f$  and  $K_\infty = \max(\|x - y\|_\infty \mid x, y \in \text{dom } f)$ . Then, if a vector in  $\text{dom } f$  is given, a global minimizer of  $f$  can be found in  $O(Fn^3 \log^2(K_\infty/n))$  time.*

The different algorithms developed by Murota [22, Section 10.1] provide a minimizer of a  $M$ -convex function in polynomial time, if an initial point is given and if the domain of the function is bounded. Whereas it is trivial to find a vector of  $\mathbb{Z}^n$  such that  $\forall_i, N_i \leq N_i^C$  or a vector  $N$  belonging to  $\sum_k F_k$ , it is not obvious to find one satisfying both conditions. In fact, such a point can be obtained by solving the minimization problem:

$$\min_{N \in \sum_k F_k} \sum_i \max(N_i - N_i^C, 0)$$

The condition  $N \in B$  is equivalent to  $N \in \arg \min_{N \in \sum_k F_k} \sum_i \max(N_i - N_i^C, 0)$  if  $B$  is non-empty. The function  $N \mapsto \sum_i \max(N_i - N_i^C, 0)$  is separable convex. Then, the function  $N \mapsto \sum_i \max(N_i - N_i^C, 0) + \chi_{\sum_k F_k}$  is  $M$ -convex according to [22, Th. 6.13.(4), p.148]. Because  $\sum_k F_k$  is bounded and a point in  $\sum_k F_k$  can be obtained in  $O(Kn)$  operations by summing vectors taken

in each set  $F_k$ , it is possible to find a point  $N^0 \in \arg \min_{N \in \sum_k F_k} \sum_i \max(N_i - N_i^C, 0) = B$  in polynomial time, by Proposition 2.

We can finally write the following result about the complexity of the decomposition method given by Theorem 1.

**Theorem 3** *Let  $R = \sum_k R_k$ , for every  $k \in [K]$ ,  $n_k = n - \#\mathcal{J}_k$  and  $\bar{R} = \sum_k R_k(n_k - R_k)$ . An optimal solution of Problem 5 can be obtained in  $O((Kn)^{3.5}Ln^3 \log^2(K/n) + (n + \bar{R})^{3.5}L)$  arithmetic operations, where  $L$  is the input size of the bilevel problem.*

*Proof.* The first step of Theorem 1 is a maximization of a  $M$ -concave function over a bounded domain  $B$ . Finding a point in  $B$  can be done by solving the  $M$ -convex minimization problem:

$$\min_{N \in \sum_k F_k} \sum_i \max(N_i - N_i^C, 0)$$

The domain of the function  $N \mapsto \sum_i \max(N_i - N_i^C, 0) + \chi_{\sum_k F_k}$  is  $\sum_k F_k$ . We define  $K_\infty^1$  by:

$$K_\infty^1 = \max\{\|N - N'\|_\infty \mid N, N' \in \sum_k F_k\}$$

For every  $N \in \sum_k F_k$ , the entries of  $N$  are sum of  $K$  binary values. Then,  $K_\infty^1 \leq K$ . We have to estimate the number of operations  $F^1$  needed to evaluate the function  $N \mapsto \sum_i \max(N_i - N_i^C, 0) + \chi_{\sum_k F_k}$ . The function  $N \mapsto \sum_i \max(N_i - N_i^C, 0)$  can be evaluated in  $O(n)$  operations. As a consequence of Lemma 6 and Lemma 7,  $\sum_k F_k = (\sum_k \Delta_k) \cap \mathbb{Z}^n$ . Hence, for any vector  $N$ , the conditions  $N \in \sum_k F_k$  is equivalent to  $N \in (\sum_k \Delta_k) \cap \mathbb{Z}^n$ . A vector  $N$  belongs to  $\sum_k \Delta_k$  if there exists for every  $k \in [K]$  a vector  $u_k \in \Delta_k$  such that  $\sum_k u_k = N$ . Hence, to know whether  $N$  belongs to  $\sum_k \Delta_k$  or not is a linear feasibility problem in dimension  $Kn$ . It can be solved in  $O((Kn)^{3.5}L)$  arithmetic operations by an interior point method ([24]). Here  $L$  is the input size of the linear program. Consequently,  $F^1 = O((Kn)^{3.5}L)$ , and a point in  $B$  can be obtained in  $O((Kn)^{3.5}Ln^3 \log^2(K/n))$  by Theorem 2.

After obtaining a point in  $B$ , the first step of Theorem 1 consists in solving the  $M$ -concave maximization problem:

$$\max_{N \in B} \sum_i f_i(N_i).$$

The domain of the function  $N \mapsto \sum_i f_i(N_i) - \chi_B(N)$  is bounded and equal to  $B$ . We define  $K_\infty^2$  by:

$$K_\infty^2 = \max\{\|N - N'\|_\infty \mid N, N' \in B\}$$

For every  $N \in \sum_k F_k$ , the entries of  $N$  are sum of  $K$  binary values. Then, for every  $i \in [n]$ , we have  $N_i \leq \min(K, N_i^C)$ . Then,  $K_\infty^2 \leq \min(K, \bar{N}^C)$ , with  $\bar{N}^C = \max_{i \in [n]} N_i^C$ . The number of operations  $F^2$  needed to evaluate the function  $N \mapsto \sum_i \max(N_i - N_i^C, 0) - \chi_B(N)$  is  $O((Kn)^{3.5}L)$  like previously. Hence, a point  $N^* \in \arg \max_{N \in B} \sum_i f_i(N_i)$  can be obtained in  $O((Kn)^{3.5}Ln^3 \log^2(\min(K, \bar{N}^C)/n))$  by Theorem 2.

According to the proof of Lemma 6, the second step of Theorem 1 is a linear program in dimension  $Kn$ . In fact, we have:

$$\max_{\substack{u_1 \in F_1, \dots, u_K \in F_K \\ \sum_k u_k = N^*}} \sum_k \langle \rho_k, u_k \rangle = \max_{\substack{u_1 \in \Delta_1, \dots, u_K \in \Delta_K \\ \sum_k u_k = N^*}} \sum_k \langle \rho_k, u_k \rangle,$$

and the extreme points of the polyhedron  $\Delta_N$  defined by:

$$\Delta_N = \{(u_1, \dots, u_K) \in \Delta_1 \times \Delta_K \mid \sum_k u_k = N\}$$

are integer. Hence, the second step of Theorem 1 can be solved in  $O((Kn)^{3.5}L)$  arithmetic operations.

The third step of Theorem 1 is a linear program in  $n$  variables. For some  $u_k^* \in F_k$ , the constraints of this program are:

$$\forall k \in [K], \forall i, j \notin \mathcal{J}_k, \text{ such that } u_k^*(i) = 1, u_k^*(j) = 0, \rho_k(i) + y_i^* \geq \rho_k(j) + y_j^*.$$

For every  $k \in [K]$ , the number of entries of  $u_k^*$  equal to 1 is  $R_k$ , and the number of entries of  $u_k^*$  equal to 0 and which do not belong to  $\mathcal{J}_k$  is  $n_k$ . Hence, the number of inequality constraints of this linear program is  $\sum_k R_k(n_k - R_k) = \bar{R}$ . Hence, a solution of this linear program can be found in  $O((n + \bar{R})^{3.5}L)$  by interior-point methods.  $\square$

## 5 A faster algorithm for solving the bilevel problem

### 5.1 A polynomial time algorithm for the bilevel problem

Algorithm 1 can be applied to solve problem (6) of Theorem 1, that is maximizing the  $M$ -concave function  $f - \chi_B$ , or equivalently minimizing the  $M$ -convex function  $-f + \chi_B$ .

Step 1 consists in finding an initial vector  $N \in B$ . As explained in Section 4, this can be done by solving a  $M$ -convex minimization problem. Another approach consists in replacing the function  $f - \chi_B$  by  $g : N \mapsto f(N) - \chi_{\sum_k F_k}(N) - M \sum_i \max(N_i - N_i^C, 0)$ , where  $M > 0$  is an integer. If  $N \in B$ , then  $g(N) = f(N)$ . If  $M$  is sufficiently large, then  $M \sum_i \max(N_i - N_i^C, 0) \geq M$  if  $N \notin B$ , and the maximum of the function  $g$  is attained for  $N \in B$ . Moreover  $N \mapsto M \sum_i \max(N_i - N_i^C, 0)$  is separable convex, then  $g$  is  $M$ -concave according to [22, Th. 6.13.(4), p.148]. Then, both problems  $\max_{N \in B} f(N)$  and  $\max_{N \in \mathbb{Z}^n} g(N)$  are equivalent, and we can apply Algorithm 1 to solve the problem  $\max_{N \in \mathbb{Z}^n} g(N)$ . An initial point is obtained by taking any point in  $\sum_k F_k$ .

We need first part is to determine the number  $F$  of operations to evaluate  $g$ . Because the different functions  $f_i$  are known, we have to determine the number of operations to decide whether a vector  $N$  belongs to  $\sum_k F_k$  or not. More precisely, the different evaluations of  $f - \chi_B$  are done in Step 2. Hence, the question is the following: given a vector  $N \in \sum_k F_k$ , how many operations are needed to check whether  $N - e_i + e_j$  (for  $i, j \in [n]$ ) belongs to  $\sum_k F_k$ . We next show that this problem can be studied as a shortest path problem in a graph. Consider  $N \in \sum_k F_k$  and let us define  $u_k^* \in F_k$  for  $k \in [K]$  such that  $\psi(N) = \sum_k \langle \rho_k, u_k^* \rangle$ , that is an optimal decomposition of  $N$  in Theorem 1. For each  $k \in [K]$  and  $\alpha, \beta \in [n]$ , we define by  $w_{\alpha\beta}^k$  the following quantity:  $w_{\alpha\beta}^k = \rho_k(\alpha) - \rho_k(\beta)$  if  $u_k^*(\alpha) = 1$  and  $u_k^*(\beta) = 0$ , and  $w_{\alpha\beta}^k = +\infty$  otherwise. Then, we define for each  $\alpha, \beta \in [n]$ ,  $w_{\alpha\beta} = \min_{k \in [K]} w_{\alpha\beta}^k$ . We consider the oriented valuated graph  $G = (V, E)$  where the set of vertices  $V = [n]$  and there is an oriented edge between each vertices  $\alpha, \beta \in V$  of value  $w_{\alpha\beta}$ .

**Theorem 4** *Let  $i, j \in [n]$ . Suppose that there exists a path in  $G$  with finite valuation between the vertices  $i, j \in V$ . Then  $N - e_i + e_j \in \sum_k F_k$ . Moreover, there are no negative cycles and there is a shortest path between  $i$  and  $j$ . Let  $(\alpha_u)_{0 \leq u \leq p}$  be any sequence such that  $\alpha_0 = i$ ,  $\alpha_p = j$  and let  $\alpha_0 \rightarrow \alpha_1 \dots \alpha_{p-1} \rightarrow \alpha_p$  be a shortest path between  $i$  and  $j$ . Let also  $(k_u)_{0 \leq u \leq p-1}$  be any sequence such that  $w_{\alpha_u \alpha_{u+1}}^{k_u} = w_{\alpha_u \alpha_{u+1}}$  for all  $0 \leq u \leq p-1$ . Let us finally define the vectors  $v_k^*$ ,  $k \in [K]$  such that  $v_{k_u}^* = u_{k_u}^* - e_{\alpha_u} + e_{\alpha_{u+1}}$  for each  $0 \leq u \leq p-1$  and  $v_k^* = u_k^*$  for each  $k \notin \{k_0, \dots, k_{p-1}\}$ . Then,  $\psi(N - e_i + e_j) = \sum_k \langle \rho_k, v_k^* \rangle$ .*

*Proof.* By Lemma 6 and 7, we know that  $N - e_i + e_j \in \sum_k F_k$  if and only if there exists  $K$  vectors  $v_k^*$  such that  $N - e_i + e_j = \sum_k v_k^*$  with each  $v_k^* \in F_k$  and  $\psi(N - e_i + e_j) = -\sum_k \langle \rho_k, v_k^* \rangle$ .

We consider  $\psi(N) = -\sum_k \langle \rho_k, u_k^* \rangle$  with each  $u_k^* \in F_k$ . Hence,  $\psi(N - e_i + e_j) - \psi(N)$  is equal to:

$$\min_{v_k \in F_k \text{ and } \sum_k v_k = N - e_i + e_j} \sum_k \langle \rho_k, u_k^* - v_k \rangle.$$

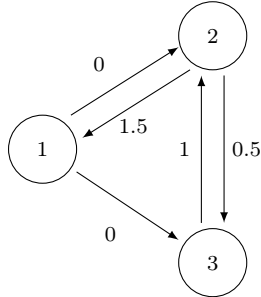
We have  $\sum_k (u_k^* - v_k) = e_i - e_j$ . When  $v_k$  describes  $F_k$ , the possible  $u_k^* - v_k$  are the vectors  $x_k$  with the following properties:

$$\begin{aligned} \sum_{\alpha=1}^n x_k(\alpha) &= 0, \quad \forall \alpha \text{ s.t. } u_k^*(\alpha) = 1, x_k(\alpha) \in \{0; 1\}, \\ \forall \alpha \in \mathcal{J}_k, x_k(\alpha) &= 0, \quad \forall \alpha \text{ s.t. } u_k^*(\alpha) = 0, x_k(\alpha) \in \{-1; 0\} \end{aligned}$$

Hence,  $\psi(N - e_i + e_j) - \psi(N) = \sum_k \langle \rho_k, x_k^* \rangle$ , where  $x_k^*$  is such that  $\#\{\alpha \mid x_k^*(\alpha) = 1\} = \#\{\alpha \mid x_k^*(\alpha) = -1\}$ . Consequently,  $\psi(N - e_i + e_j) - \psi(N)$  can be written as a sum of  $w_{\alpha\beta}^k$  for certain  $\alpha, \beta$ . Because of the condition  $\sum_k u_k^* - v_k = e_i - e_j$ , we have  $\psi(N - e_i + e_j) - \psi(N) = w_{\alpha_0\alpha_1}^{k_0} + w_{\alpha_1\alpha_2}^{k_1} + \dots + w_{\alpha_{p-1}\alpha_p}^{k_{p-1}}$ , with the notations introduced in Theorem 4.

Consider now the graph defined in Theorem 4. If there exists a path between  $i$  and  $j$ , then its value can be written  $w_{\beta_0\beta_1}^{l_0} + w_{\beta_1\beta_2}^{l_1} + \dots + w_{\beta_{q-1}\beta_q}^{l_{q-1}}$  (with the convention  $\beta_0 = i$  and  $\beta_q = j$ ). By defining  $v_k = u_k^*$  if  $k \notin \{l_0, \dots, l_{q-1}\}$  and  $v_{l_u} = u_{l_u}^* - e_{\beta_u} + e_{\beta_{u+1}}$  for  $0 \leq u \leq q-1$ , the value of the path is equal to  $\sum_k \langle \rho_k, u_k^* - v_k \rangle$ . Because  $w_{\beta_u\beta_{u+1}}^{l_u} < +\infty$ , we have  $u_{l_u}^*(\beta_u) = 1$  and  $u_{l_u}^*(\beta_{u+1}) = 0$ . Then, each  $v_k \in F_k$ . Consequently, the value  $\min_{v_k \in F_k \text{ and } \sum_k v_k = N - e_i + e_j} \sum_k \langle \rho_k, u_k^* - v_k \rangle$  is finite and  $N - e_i + e_j \in \sum_k F_k$ . Moreover, the value  $\psi(N - e_i + e_j) - \psi(N)$  corresponds to the minimal values of the path between  $i$  and  $j$  in  $G$ , that is the shortest path. Hence, if the value of the shortest path is  $\sum_{u=0}^{p-1} w_{\alpha_u\alpha_{u+1}}^{l_u}$ , we have  $\psi(N - e_i + e_j) - \psi(N) = \sum_k \langle \rho_k, u_k^* - v_k \rangle$ , with  $v_k^*$  defined as in the statement of Theorem 4. Moreover, we can prove that there exists no cycle with negative weight in this graph. Suppose that such a cycle exists. It can be written  $w_{\gamma_0\gamma_1}^{l_0} + w_{\gamma_1\gamma_2}^{l_1} + \dots + w_{\gamma_r\gamma_0}^{l_r} < 0$ . For all  $i \in \{0 \dots r\}$ , we have  $u_{l_i}(\gamma_i) = 1$  and  $u_{l_i}(\gamma_{i+1}) = 0$ . We consider for  $k \in [K]$  the vectors  $v_k$  defined by  $v_{l_i} = u_{l_i}^* - e_{\gamma_i} + e_{\gamma_{i+1}}$ , and  $v_k = u_k^*$  for  $k \notin \{l_0, \dots, l_r\}$ . We have  $\sum_k u_k^* - v_k = 0$  and so  $\sum_k \langle \rho_k, u_k \rangle = \sum_k \langle \rho_k, v_k \rangle + w_{\alpha_1\alpha_2}^{k_1} + w_{\alpha_2\alpha_3}^{k_2} + \dots + w_{\alpha_p\alpha_1}^{k_p} < \sum_k \langle \rho_k, v_k \rangle$  which refutes the optimality of the vectors  $u_k^*$  in the definition of  $\psi(N)$ .  $\square$

*Example 2* We consider the cell (a) of Figure 3. We build the graph associated to  $N = (3, 3, 1)$  (see Figure 4).



**Fig. 4** Graph  $G$  associated to the vector  $N = (3, 3, 1)$

Consider  $N' = N - e_1 + e_2 = (2, 4, 1)$ . The shortest path in  $G$  is  $1 \rightarrow 2$  with  $w_{12} = 0 = w_{12}^1$ . Then, according to Theorem 4, the optimal decomposition of  $(2, 4, 1)$  is  $v_1^* = (0, 1, 0)$ ,  $v_2^* = (1, 0, 1)$ ,  $v_3^* = (0, 1, 0)$ ,  $v_4^* = (1, 1, 0)$  and  $v_5^* = (0, 1, 0)$ .

Thanks to Theorem 4, if we know that a vector  $N$  belongs to  $\sum_k F_k$ , it is possible to check whether a vector  $N - e_i + e_j$  belongs to  $F_k$  by checking if there exists a path between  $i$  and  $j$  in the graph  $G = (V, E)$ . Generally,  $G$  has  $n$  vertices and  $n^2$  edges. From each vertex  $i \in V$ , it is possible to find if there exists a path between  $i$  and  $j$  by using a depth-first or breadth first search algorithm in  $O(n^2)$  operations. Consequently, the number of operations needed to evaluate  $g$  is  $O(n^3)$ .

According to Theorem 4, by checking if  $N - e_i + e_j \in B$ , we obtain the optimal decomposition of  $N - e_i + e_j = \sum_k v_k^*$  such that  $\psi(N - e_i + e_j) = -\sum_k \langle \rho_k, v_k^* \rangle$  by solving a shortest path problem between two vertices. This can be done in  $O(n^3)$  operations thanks to Ford-Bellman algorithm ([4], [12]), because the graph  $G$  has  $n$  vertices and at most  $n^2$  edges. Hence, according to Theorem 1, it suffices to solve the bilevel problem 5 to solve the linear feasibility problem of Lemma 8. Moreover, this problem can also be viewed as a shortest path problem in  $G$ , according to the following result.

**Theorem 5** Consider  $K$  vectors  $u_k^* \in F_k$  for each  $k \in [K]$  such that, if we define  $N = \sum_k u_k^*$ , we have  $\psi(N) = -\sum_k \langle \rho_k, u_k^* \rangle$ . Consider the graph  $G$  associated to  $N$ . Consider an index  $s \in [n]$ . Let  $M > 0$  be any real scalar such that  $M \geq n \max_{i,j \in [n]} w_{ij}$  and let us modify  $G$  such that for all  $t \in [n]$  with  $t \neq s$  and  $w_{st} = +\infty$ , we have  $w_{st} = M$ . Let us define a vector  $y^* \in \mathbb{R}^n$  by  $y_s^* = 0$  and for each  $t \in [n]$  with  $t \neq s$ ,  $y_t^*$  is the length of the shortest path between  $s$  and  $t$  in  $G$ . Then, for  $M$  sufficiently large and for each  $k \in [K]$ ,  $u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y^*, u_k \rangle$ .

*Proof.* According to Lemma 8, a vector  $y \in \mathbb{R}^n$  is such that for every  $k \in [K]$ ,

$$u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y, u_k \rangle$$

if and only if the following inequalities are satisfied:

$$\forall k \in [K], \forall i, j \notin \mathcal{J}_k, \text{ such that } u_k^*(i) = 1, u_k^*(j) = 0, \rho_k(i) + y_i \geq \rho_k(j) + y_j.$$

Consider such a vector  $y$ . Consider also the graph  $G$  associated to  $N$ . The previous inequalities can be rewritten  $\forall k \in [K], \forall i, j \in [n], y_j - y_i \leq w_{ij}^k$ , or equivalently :  $\forall i, j \in [n], y_j - y_i \leq w_{ij}$ . For each  $\delta \in \mathbb{R}$ ,  $y + \delta e$  is also a solution. Consequently, it is possible to fix a coordinate to 0. Take a coordinate  $s$  such that  $y_s = 0$ . Consider  $M > 0$  such that  $M \geq n \max_{i,j} w_{ij}$  and modify the graph  $G$  as in the statement of the theorem. Consider an elementary cycle (that is a cycle containing no smaller cycle) of the modified graph. The cycle has no more than  $n - 1$  edges. Suppose that exactly  $q$  edges have a modified weight, with  $0 \leq q \leq n - 1$ . If  $q = 0$ , then no edge has a modified weight, and this cycle is a cycle of  $G$ . So, its weight is nonnegative. If  $q \geq 1$ , then the total weight of the cycle is bigger than  $qM + (n - 1 - q) \min_{i,j} w_{ij} \geq n(\max_{i,j} w_{ij} - \min_{i,j} w_{ij}) \geq 0$ . Consequently, the modified graph has no negative cycles.

For each  $t \in [n]$ , with  $t \neq s$ , there exists a path between  $s$  and  $t$ . Let us define  $y^*$  such that  $y_s^* = 0$  and for each  $t \in [n]$  with  $t \neq s$ ,  $y_t^*$  corresponds to the length of the shortest path between  $s$  and  $t$ . Consider  $i, j \in [n]$ . Then  $y_i^* + w_{ij}$  is the length of a path between  $s$  and  $j$  defined as the concatenation of the shortest path between  $s$  and  $i$  and the edge  $i \rightarrow j$ . So  $y_i^* + w_{ij} \geq y_j^*$ . Hence, according to Lemma 8, we have for each  $k \in [K]$ ,  $u_k^* \in \arg \max_{u_k \in F_k} \langle \rho_k + y^*, u_k^* \rangle$ .  $\square$

These different results lead to Algorithm 2 to solve the bilevel problem 5. First, we have to find an initial point  $N$  in  $\sum_k F_k$ , with its optimal decomposition  $\sum_k u_k^*$ . We can calculate for each  $k \in [K]$  and for each  $i, j \notin \mathcal{J}_k$  the value  $w_{ij}^k$ , store them, and then define the graph  $G$  associated to  $N$ . Hence, with a graph search algorithm, we know for each  $i, j \in [n]$  whether  $N - e_i + e_j \in \sum_k F_k$  or not, and can calculate  $g(N - e_k + e_l)$  for each  $k, l \in [n]$  and find  $i, j \in \arg \max_{k,l} g(N - e_k + e_l)$ . By finding the shortest path between  $i$  and  $j$  in  $G$ , we obtain

the optimal decomposition  $N - e_i + e_j = \sum_k v_k^*$ . Like in Algorithm 1, if  $g(N - e_i + e_j) \leq g(N)$ , then  $N^* = N$  is the maximum value of  $g$  over  $\sum_k F_k$ . Else, we take  $N := N - e_i + e_j$ . For all the indices  $k$  such that  $u_k^* \neq v_k^*$ , we evaluate the new value of  $w_{ij}^k$  and we define the graph  $G$  associated to  $N - e_i + e_j$  and restart the algorithm. Notice that the number of indices  $k$  such that  $u_k^* \neq v_k^*$  is bounded by the length of the shortest path in  $G$ ; it means that this number is less than  $n$ . After finding the optimal  $N^*$  and having its optimal decomposition  $N^* = \sum_k u_k^*$ , we can redefine the graph associated to  $N^*$  and return an optimal  $y^*$  defines as in the statement of Theorem 5.

Algorithm 2 can be written as follows. We take in input a function `GraphSearch`, which associate to a graph  $G$  (defined by the weight vector  $w$  of its edges) a Boolean vector  $b$  such that  $b_{ij} = 1$  if there is an edge between  $i$  and  $j$  and 0 otherwise. We also take a function `ShortestPath`, which associate to a graph  $G$  (also defined by the weight vector  $w$ ) and two vertices  $i$  and  $j$ , the value  $v$  of the shortest path and a vector `path` with the indices of this shortest path. Finally, we consider the function `ShortestPath2`, which associate to  $w$  and a vertex  $s$  a vector corresponding to the values of the shortest path between  $s$  and all other vertices in  $G$ . For much ease, we denote by  $f^*$  the function  $f^* : N \mapsto f(N) + M \sum_{i=1}^n \max(N_i - N_i^C, 0)$ .

Note that the pseudo-polynomial time bound for Murota's greedy algorithm 1 given by Proposition 1 leads in this special case to a polynomial time bound, as explained in the following result.

**Theorem 6** *Let us define  $R = \sum_k R_k$ , for each  $k \in [K]$   $n_k = n - \#\mathcal{J}_k$  (that is the number of possible non-zero entries of the vectors of  $F_k$ ) and  $\bar{R} = \sum_k R_k(n_k - R_k)$ . Algorithm 2 returns a global optimizer with a time complexity of  $O(R(n^3 + \bar{R}))$  and a space complexity of  $O(\bar{R})$ .*

*Proof.* The vector returned by the algorithm is a global optimizer according to Algorithm 1 and Theorem 4. The initialization consists in taking vectors in each  $F_k$  and in adding them; it can be done in  $O(K)$  operations. Then, to define the graph  $G$ , we have to calculate  $w_{ij}^k$  for each  $i, j \notin \mathcal{J}_k$  and each  $k \in [K]$ , and to store the values. Let us define for each  $k \in [K]$   $n_k = n - \#\mathcal{J}_k$ . For each  $k \in [K]$ , we have  $R_k \leq n_k$ , and there are precisely  $R_k$  coordinates of  $u_k^*$  equal to 1 for each  $u_k^* \in F_k$ . Then, for each  $k \in [K]$ , there are exactly  $R_k(n_k - R_k)$  finite values of  $w_{ij}^k$  to store. Then, by defining  $\bar{R} = \sum_k R_k(n_k - R_k)$ , we need  $O(\bar{R})$  operations to define  $w_{ij}$  and  $k_{ij}$ . The function `GraphSearch` needs  $O(n^3)$  operations by a depth-first or breadth-first algorithm to know if there is a path between  $i$  and  $j$ . The function `ShortestPath` needs also  $O(n^3)$  operations to calculate the shortest path between  $i$  and  $j$  with Ford-Bellman algorithm. The length of the path is bounded by  $n$ . Consequently, there is less than  $n$  vectors  $u_k^*$  which have to be updated; and then less than  $2nn_k$  values  $w_{\alpha\beta}^k$  to update.  $\bar{R}$  operations are needed to calculate the new values of  $w_{ij}$  and  $k_{ij}$ . So, the number of operations in each step of the "while" loop is  $O(n^3 + n\bar{R})$ . The number of iterations of the loop is the same as in Algorithm 1, and is bounded by  $K_1$  where  $K_1 = \max(\|x - y\|_1, x, y \in \sum_k F_k)$ . For each  $x, y \in \sum_k F_k$ , we have:

$$\|x - y\|_1 = \sum_{i=1}^n |x_i - y_i| \leq \sum_{i=1}^n (x_i + y_i) = 2R$$

by defining  $R = \sum_{k=1}^K R_k$ . Finally, to find the optimal  $y^*$ ,  $n^2$  operations are needed to find  $M$ , and  $O(n^3)$  operations are needed to evaluate the function `ShortestPath2` by using again the Ford-Bellman algorithm. Step 7 consists in calculating the shortest path between a vertex  $s$  and the other ones in a graph with  $n$  vertices and  $n^2$  edges. Then, Step 7 can be obtained in  $O(n^3)$  thanks to Ford-Bellman algorithm. Hence, the global time complexity of Algorithm 2 is  $O(R(n^3 + \bar{R}))$  and space complexity is  $O(\bar{R})$ .  $\square$

**Algorithm 2** Solving the bi-level problem, for one application and one type of contract

---

**Require:**  $u_k^* \in F_k, \forall k \in [K], \rho_k, \forall k \in [K], f^*, \text{GraphSearch}, \text{ShortestPath}, \text{ShortestPath2}, s \in [n]$

**Ensure:**  $N^*$  optimal number of customers,  $y^*$  optimal discount vector

```

 $N \leftarrow \sum_{k=1}^K u_k^*$ 
for all  $k \in [K]$  do
  for all  $i, j \notin \mathcal{J}_k$  do
    if  $u_k(i) = 1$  and  $u_k(j) = 0$  then
       $w_{ij}^k \leftarrow \rho_k(i) - \rho_k(j)$ 
    end if
  end for
end for
for all  $i, j \in [n]$  do
   $w_{ij} \leftarrow \min_{k \in [K]} w_{ij}^k; \quad k_{ij} \in \arg \min_{k \in [K]} w_{ij}^k$ 
end for
 $stop \leftarrow 0$ 
while  $stop = 0$  do
   $b \leftarrow \text{GraphSearch}(w)$ 
   $g_N \leftarrow f^*(N); \quad g^* \leftarrow \max_{u, v \in [n], b_{uv}=1} f^*(N - e_u + e_v); \quad i, j \in \arg \max_{u, v \in [n], b_{uv}=1} f^*(N - e_u + e_v)$ 
  if  $g^* \leq g_N$  then
     $stop \leftarrow 1$ 
  else
     $(v, path) \leftarrow \text{Shortestpath}(w, i, j); \quad N \leftarrow N - e_i + e_j$ 
    for  $q = 1$  to  $\text{Length}(path) - 1$  do
       $\alpha \leftarrow path(q); \quad \beta \leftarrow path(q + 1); \quad k \leftarrow k_{\alpha\beta}; \quad u_k^*(\alpha) = 0; \quad u_k^*(\beta) = 1$ 
      for all  $\gamma \notin \mathcal{J}_k$  do
         $w_{\alpha\gamma}^k \leftarrow +\infty; \quad w_{\gamma\beta}^k \leftarrow +\infty$ 
        if  $u_k^*(\gamma) = 1$  then
           $w_{\gamma\alpha}^k \leftarrow \rho_k(\gamma) - \rho_k(\alpha)$ 
        else
           $w_{\beta\gamma}^k \leftarrow \rho_k(\beta) - \rho_k(\gamma)$ 
        end if
      end for
    end for
    for all  $i, j \in [n]$  do
       $w_{ij} \leftarrow \min_{k \in [K]} w_{ij}^k; \quad k_{ij} \in \arg \min_{k \in [K]} w_{ij}^k$ 
    end for
  end if
end while
 $M \leftarrow 1 + n \max_{i, j \in [n]} w_{ij}$ 
for all  $t \in [n]$  do
  if  $t \neq s$  AND  $w_{st} = +\infty$  then
     $w_{st} = M$ 
  end if
   $y^* \leftarrow \text{Shortestpath2}(w, s)$ 
   $y_s^* \leftarrow 0$ 
end for

```

---

Notice that for each  $k \in [K]$ ,  $n_k \leq n$  and  $1 \leq R_k \leq n_k$ . Then  $K \leq R \leq nK$  and  $0 \leq \bar{R} \leq Kn^2$ . Therefore, the time complexity of Algorithm 2 is  $O(Kn^3(K+n))$  in the worst case, whereas the space complexity is  $O(Kn^2)$ .

*Example 3* Consider again Example 1 together with the concave function  $f$  defined by

$$f : N \mapsto - \sum_{t,l} N(t, l)^2.$$

We suppose that  $\forall k, \mathcal{J}_k = \emptyset$ . Hence, we can prove that  $\sum_k F_k = \{N \in \mathbb{N}^3 \mid \sum_{i=1}^3 N_i = 7 \text{ and } \max(N_i) \leq 5\}$ . First, we want to solve  $\max_{N \in \sum_k F_k} -(N_1^2 + N_2^2 + N_3^2)$ . We start from  $N^{(0)} =$

$(5, 2, 0)$ , a feasible point. Following Algorithm 1, we compute  $N^{(1)} = (4, 2, 1)$  and  $N^{(2)} = (3, 2, 2)$  which is a minimizer. We take  $N^* = (3, 2, 2)$ . Now, we solve  $\max_{u_1 \in F_1, \dots, u_5 \in F_5, \sum_{k=1}^5 u_k = N^*} \sum_k \langle \rho_k, u_k \rangle$ . We obtain  $u_1^* = [1, 0, 0]$ ,  $u_2^* = [1, 0, 1]$ ,  $u_3^* = [0, 1, 0]$ ,  $u_4^* = [1, 0, 1]$ ,  $u_5^* = [0, 1, 0]$ . Applying Lemma 8, we obtain the linear inequalities  $y_1^* - y_2^* \leq 3/2$ ,  $0 \leq y_1^* - y_3^*$  and  $-1 \leq y_2^* - y_3^* \leq -1/2$ . In particular,  $y^* = (3/4, 0, 3/4)$  is an optimal solution.

## 5.2 A particular case : theory of majorization

Algorithm 2 can be accelerated in the particular case  $\forall k \in [K]$ ,  $\mathcal{J}_k = \emptyset$ , that is  $F_k = \{u_k \in \{0; 1\}^n \mid \sum_{i=1}^n u_k(i) = R_k\}$ .

As previously, an important step of the maximization of the function  $g$  consists in being able to know whether a point belongs to  $\sum_k F_k$  or not. In this particular case, we can use the *majorization order* [23]. For every  $x \in \mathbb{R}^n$ , denote by  $x_{[1]} \geq \dots \geq x_{[n]}$  the coordinates of  $x$  arranged in nonincreasing order. A vector  $x \in \mathbb{R}^n$  is said to be *majorized* by another vector  $y \in \mathbb{R}^n$ , denoted  $x \prec y$ , if  $\sum_{i=1}^n x_i = \sum_{i=1}^n y_i$  and  $\forall 1 \leq k \leq n-1$ ,  $\sum_{i=1}^k x_{[i]} \leq \sum_{i=1}^k y_{[i]}$ .

We have the following result.

**Theorem 7 (Gale-Ryser , see [23, Th. 7.C.1])** *Let  $a \in \mathbb{N}^k$  and  $b \in \mathbb{N}^n$  be two integer vectors with nonnegative values. Let  $a^* \in \mathbb{N}^n$  defined by  $a_i^* = \#\{j \mid a_j \geq i\}$ . Then, the following assertions are equivalent:*

1.  $b \prec a^*$
2. *There exists a matrix  $U \in \mathbb{Z}^{k \times n}$  such that for each  $i, j$ ,  $u_{ij} \in \{0; 1\}$ ,  $\forall 1 \leq i \leq k$ ,  $\sum_{j=1}^n u_{ij} = a_i$  and  $\forall 1 \leq j \leq n$ ,  $\sum_{i=1}^k u_{ij} = b_j$*

**Corollary 3** *Denoting by  $f_r = (1, \dots, 1, 0, \dots, 0)$  the vector with exactly  $r$  1 and by  $p_r = \#\{k \mid R_k = r\}$ , for  $1 \leq r \leq n$ , we have  $\sum_k F_k = \{N \in \mathbb{N}^n \mid N \prec \sum_{r=1}^n p_r f_r\}$ .*

*Proof.* A vector  $N$  belongs to  $\sum_k F_k$  if and only if for each  $i \in [n]$ ,  $N_i$  corresponds to the sum of the coefficients of the  $i$ -th column of a matrix of size  $K \times n$  with coefficients in  $\{0; 1\}$  and such that the sum of the coefficients of the  $k$ -th line is  $R_k$ . We conclude by 7.  $\square$

*Example 4* Consider Example 1. We have  $p_1 = 3$ ,  $p_2 = 2$  and  $p_3 = 0$ . So  $N$  is feasible iff  $N$  verifies  $N \prec (5, 2, 0)$ .

Like for Algorithm 2, we need to know for a given  $N \in \sum_k F_k$  whether  $N - e_i + e_j \in \sum_k F_k$  for each  $i, j \in [n]$ . It is possible to answer to this question in polynomial time in  $n$  by sorting  $N - e_i + e_j$  for each  $i, j$  and by checking the condition  $N - e_i + e_j \prec N^{\max}$ . The time complexity of such a procedure is  $O(n^3 \log(n))$ . However, it can be accelerated thanks to the following result.

**Lemma 10** *Let  $N \in \sum_k F_k$ , and  $i, j \in [n]$ . Let  $S$  be the function defined on  $\mathbb{R}^n \times [n]$  such that  $\forall x \in \mathbb{R}^n, \forall k \in [n]$ ,  $S(x, k)$  is the sum of the  $k$  largest values of the coordinates of  $x$ . Suppose finally that  $N_j$  is the  $k_j$ -th largest value of the coordinates of  $N$  (if  $k_j > 1$ , then we suppose that the  $k_j - 1$ -th largest value of  $N$  is strictly bigger than  $N_j$ ), and that  $N_i$  is the  $k_i$ -th largest value of the coordinates of  $N$  (if  $k_i < n$ , then we suppose that the  $k_i + 1$ -th largest value of  $N$  is strictly smaller than  $N_j$ ). Then  $N - e_i + e_j \in \sum_k F_k$  if and only if  $N_i > 0$  and, either  $N_i > N_j$  or  $\forall k_j \leq k \leq k_i, S(N, k) < S(N^{\max}, k)$ .*



*Proof.* Suppose  $N - e_i + e_j \in \sum_k F_k$ . Then  $N_i - 1 \geq 0$  and  $N_i > 0$ . Moreover, suppose  $N_i \leq N_j$ . Then,  $N_i - 1 < N_j + 1$  and  $S(N, k) = S(N, k) + 1$ . Then,  $S(N, k) < S(N, k) + 1 = S(N^{\max}, k)$ .

Conversely, if  $N_i > 0$ , then all the coordinates of  $N - e_i + e_j$  are nonnegative integers. If  $N_i > N_j$ , then we easily see that  $N - e_i + e_j \prec N$ . So  $N - e_i + e_j \prec N^{\max}$  and  $N - e_i + e_j \in \sum_k F_k$ . Suppose that  $N_i \leq N_j$ . Because we suppose that the  $k - 1$ -th largest value of  $N$  is strictly bigger than  $N_j$ , then  $k_i > k_j$ . We also suppose that  $\forall k_j \leq k \leq k_i, S(N, k) < S(N^{\max}, k)$ . The  $k - 1$ -th largest value of  $N$  is strictly bigger than  $N_j$ , so it is bigger than  $N_j + 1$ . Consequently, we have for all  $1 \leq l \leq k - 1, S(N - e_i + e_j, l) = S(N, l) \leq S(N^{\max}, l)$  (because  $N \prec N^{\max}$ ). Moreover,  $\forall k_j \leq k \leq k_i - 1, S(N, k) < S(N^{\max}, k)$ . Because the  $k_i + 1$ -th larger coordinate of  $N$  is strictly smaller than  $N_i$ , then it is smaller than  $N_i + 1$  and we have  $S(N - e_i + e_j, k_i) = S(N, k_i) \leq S(N^{\max}, k_i)$  and  $\forall l \geq k_i + 1, S(N - e_i + e_j, l) = S(N, l) \leq S(N^{\max}, l)$ . Hence,  $N - e_i + e_j \prec N^{\max}$  and  $N - e_i + e_j \in \sum_k F_k$ .  $\square$

To solve the bilevel problem 5 in this specific case, we need to find  $u_1^* \in F_1, \dots, u_K^* \in F_K$  such that  $\psi(N^*) = -\sum_k \langle \rho_k, u_k^* \rangle$ . In Algorithm 2, such vectors  $(u_k)^*$  are found in the same time as  $N^*$ . Then, to accelerate Algorithm 2, we need to be able to solve this problem rapidly. In particular, to use a classical linear programming approach leads to a  $O((Kn)^{3,5})$  time complexity, which is not acceptable. The problem to solve can be written:

### Problem 6

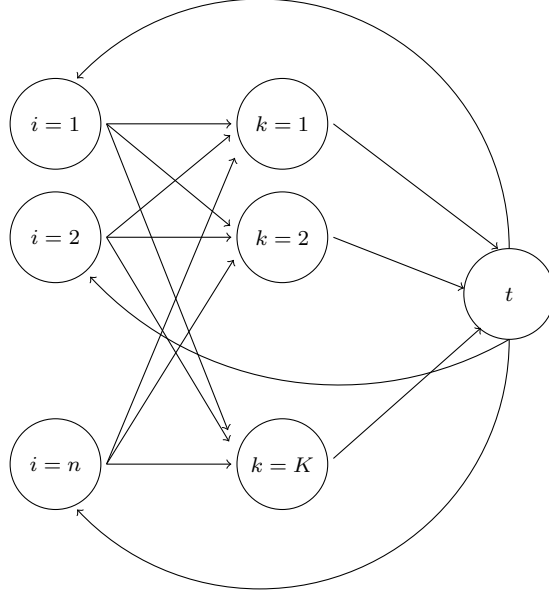
$$\max_{\substack{u_1, \dots, u_K \in \{0;1\}^n \\ \forall k, \sum_{i=1}^n u_k(i) = R_k \\ \forall i, \sum_{k=1}^K u_k(i) = N_i}} \sum_{k=1}^K \langle \rho_k, u_k \rangle$$

We already mentioned in the proof of Theorem 7 that the constraints of this linear program can be written  $0 \leq u \leq 1, Au = b$ , where  $A$  is a totally unimodular matrix. Therefore, the value of this problem is equal to the value of its continuous relaxation. Moreover, it can be interpreted as a minimum cost flow problem (see [26, Ch. 12] for background). We define a bipartite graphs with vertices  $i \in [n]$  and  $k \in [K]$ , and edges between each  $i \in [n]$  and each  $k \in [K]$ . Each vertex  $i \in [n]$  has an incoming flow equal to  $N_i$ , whereas each vertex  $k \in [K]$  has an outgoing flow equal to  $R_k$ . Moreover, the capacity of each edge is 1, meaning that each flow  $u_k(i)$  satisfies  $0 \leq u_k(i) \leq 1$ , and a cost  $-\rho_k(i)$  is associated to each edge. Hence, the problem consists in finding the flow  $u$  minimizing the total cost in this graph. Plenty of algorithms exist to solve such a problem. In our case, we have  $K \gg n$ . According to Theorem 6, Algorithm 2 needs  $O(Rn^2(K+n))$  operations to solve Problem 5. Notice that  $K \leq R \leq nK$ . Therefore, in order to accelerate Algorithm 2 in the studied case, we need an algorithm solving the flow problem with a complexity depending on  $K$  in  $K^\alpha$  with  $\alpha < 2$ .

We can interpret the minimum cost flow problem as a minimum cost circulation problem, as presented in [26, Ch. 12]. We introduce a sink  $t$ . We define an edge between each  $k \in [K]$  and  $t$  of cost equal to 0, with a lower-bound for the flow equal to  $R_k$  and a capacity of  $R_k$ . We also define an edge between  $t$  and each  $i \in [n]$  of cost equal to 0, with a lower-bound for the flow equal to  $N_i^*$  and a capacity of  $N_i^*$ . Such a graph is represented on Figure 5.

Such a graph has  $|V| = K + n + 1$  vertices and  $|E| = Kn + K + n$  edges. The sum of the capacities of the different edges is  $2R + Kn$ . In [13, Sec. 3.3], an algorithm is proposed to solve such a problem. Different complexity bounds of such an algorithm are given in [13, Th. 3.5]. In the case  $K \gg n$ , the optimal vectors  $u_1^*, \dots, u_K^*$  can be found in  $O((Kn)^{3/2} \log((K+n)\|\rho\|_\infty))$ .

We can now write an algorithm for solving the bilevel problem in this specific case. We need first to calculate  $N^{\max} = \sum_{r=0}^n p_r f_r$ , where  $p_r$  is defined as in the statement of Theorem 7, and to find an initial point  $N \in \sum_k F_k$ . We apply the same method as in Algorithm 1. In order to calculate  $g(N - e_i + e_j)$  for each  $i, j \in [n]$ , we sort the coordinate of  $N$  in the decreasing order,



**Fig. 5** Minimum cost flow problem transformed in a minimum cost circulation problem. The flow in the edges between each  $i$  and  $k$  is in  $[0, 1]$ , the flow in the edges between each  $k$  and  $t$  is equal to  $R_k$ , and the flow in the edges between  $t$  and each  $i$  is  $N_i^*$ .

and we use Lemma 10 to decide whether  $N - e_i + e_j \in F_k$  for all  $i, j$ . We use the same loop as in Algorithm 1 to compute an  $N^*$  such that  $g(N^*)$  is the maximal value of  $g$  over  $\sum_k F_k$ . Then, we solve the minimum cost flow problem 6, as described previously, to find the optimal  $u_k^*$  and then we use Lemma 5 to determine an optimal  $y^*$ . It leads to Algorithm 3. The function *Sort* associates to a vector  $x \in \mathbb{R}^n$  a couple  $(y, ind)$ , where  $y$  is a permutation of  $x$  such that  $y_1 \geq \dots \geq y_n$  and  $ind$  is such that  $x_i = y_{ind(i)}$  for each  $i \in [n]$ . The function  $S$  is defined by  $S(x, k) = \sum_{i=1}^n x_i$ . The function *MinCostFlow* associates to the different vectors  $(\rho_k)_{k \in [K]}$  the vectors  $(u_k^*)_{k \in [K]}$  solving the minimum cost flow problem 6. The functions  $f^*$  and *ShortestPath2* are defined as for Algorithm 2.

**Theorem 8** *Let us define  $\|\rho\|_\infty = \max_{k \in [K], i \in [n]} |\rho_k(i)|$ ,  $R = \sum_k R_k$ , for each  $k \in [K]$   $n_k = n - \#\mathcal{J}_k$  and  $\bar{R} = \sum_k R_k(n_k - R_k)$ . Algorithm 3 is correct and returns a global optimizer in  $O(Rn^2 + (Kn)^{3/2} \log((K+n)\|\rho\|_\infty) + \bar{R} + n^3)$  time and  $O(Kn + n^2)$  space.*

*Proof.* According to Theorem 1, Theorem 7, Lemma 10 and Algorithm 1, this algorithm returns an optimal solution  $N^*$  of the high-level problem and an optimal discount vector  $y^*$ . Similarly as in the proof of Algorithm 2, the number of calls of the "while" loop is bounded by  $R$ . The function *Sort* needs  $O(n \log(n))$  time and space operations.  $O(n^2)$  operations are needed to evaluate the vector  $b$ , then the global time complexity of the "while" loop is  $O(Rn^2)$  whereas the space complexity is  $O(n^2)$ . Then, the optimal vectors  $u_1^*, \dots, u_K^*$  can be obtained in  $O((Kn)^{3/2} \log((K+n)\|\rho\|_\infty))$  time and  $O(Kn)$  space. By calculating only the finite values of  $w_{ij}^k$  (which are not necessary stored here), the number of operations needed to determine each  $w_{ij}$  and  $k_{ij}$  is  $O(\bar{R})$ , with  $\bar{R} = \sum_k R_k(n_k - R_k)$  and for each  $k \in [K]$ ,  $n_k = n - \#\mathcal{J}_k$ . We need only  $O(n^2)$  space to store the values  $w_{ij}$  and  $k_{ij}$ . Finally, the vector  $y^*$  can be found by using the Ford-Bellman algorithm in a graph of  $n$  vertices and  $n^2$  edges, that is in time complexity of  $O(n^3)$ .  $\square$

**Algorithm 3** Solving the bilevel problem, in the case of majorization

---

**Require:**  $N \in \sum_k F_k, N^{\max}, \rho_k, \forall k \in [K], f^*, S, Sort, MinCostFlow, ShortestPath2, s \in [n]$   
**Ensure:**  $N^*$  optimal number of customers,  $y^*$  optimal discount vector

```

 $s_1^{\max} \leftarrow 0$ 
for  $i = 1$  to  $n$  do
   $s_i^{\max} \leftarrow s_i^{\max} + N_i^{\max}$ 
end for
 $stop \leftarrow 0$ 
while  $stop = 0$  do
   $(N^{sort}, ind) \leftarrow Sort(N); \quad s_1 = 0$ 
  for  $i = 1$  to  $n$  do
     $s_i \leftarrow s_i + N^{sort}(i)$ 
    for all  $j \in [n]$  do
      if  $N(i) = 0$  then
         $b(i, j) \leftarrow 0$ 
      else
         $b(i, j) \leftarrow 1$ 
      end if
    end for
    if  $s_i = s_i^{\max}$  then
      for  $j = 1$  to  $i$  do
         $b(ind(i), ind(j)) \leftarrow 0$ 
      end for
      for  $j = i$  to  $n$  do
         $b(ind(j), ind(i)) \leftarrow 0$ 
      end for
    end if
  end for
   $g_N \leftarrow f^*(N); \quad g^* \leftarrow \max_{u, v \in [n]} b(u, v) f^*(N - e_u + e_v); \quad i, j \in \arg \max_{u, v \in [n]} b(u, v) f^*(N - e_u + e_v)$ 
  if  $g^* \leq g_N$  then
     $stop \leftarrow 1$ 
  end if
end while
 $(u_1^*, \dots, u_K^*) \leftarrow MinCostFlow((\rho_k)_{k \in [K]})$ 
for all  $k \in [K]$  do
  for all  $i, j \notin \mathcal{J}_k$  do
    if  $u_k^*(i) = 1$  and  $u_k^*(j) = 0$  then
       $w_{ij}^k \leftarrow \rho_k(i) - \rho_k(j)$ 
    end if
  end for
end for
for all  $i, j \in [n]$  do
   $w_{ij} \leftarrow \min_{k \in [K]} w_{ij}^k; \quad k_{ij} \in \arg \min_{k \in [K]} w_{ij}^k$ 
end for
 $M \leftarrow 1 + n \max_{i, j \in [n]} w_{ij}$ 
for all  $t \in [n]$  do
  if  $t \neq s$  AND  $w_{st} = +\infty$  then
     $w_{st} = M$ 
  end if
   $y^* \leftarrow Shortestpath2(w, s)$ 
   $y_s^* \leftarrow 0$ 
end for

```

---

In the worst case, we have  $R = Kn$  and  $\bar{R} = Kn^2$ . Then, the time complexity of Algorithm 3 is  $O(Kn^3 + (Kn)^{3/2} \log((K+n)\|\rho\|_\infty))$ . If the number of bits needed to write  $\|\rho\|_\infty$  is polynomial in  $n$  and if  $K \gg n$ , then Algorithm 3 is faster than Algorithm 2. We finally notice that a minimum cost flow problem is strongly polynomial time solvable, and it is then possible to adapt

Algorithm 3 to return an optimal  $y^*$  in strongly polynomial time. However, Algorithm 3 does not go faster than Algorithm 2 in this case.

## 6 The general algorithm

In this section, we come back to the general bilevel problem 2 proposed in Section 2, and extend Algorithm 2 to it. In the low level problem of each customer, the consumptions for different contents verify the constraints  $\forall a \in [A], \sum_{t=1}^T u_k^a(t) = R_k^a, \forall t \in \mathcal{I}_k^a, a \in [A], u_k^a(t) = 0$  and  $\forall t \in [T], \sum_{a \in [A]} u_k^a(t) \leq 1$ . We make the assumption that for each customer  $k$ , the sets of possible instants at which this customer makes a request for the different applications are disjoint, meaning that for any two applications  $a \neq a'$ , the complements of  $\mathcal{I}_k^a$  and  $\mathcal{I}_k^{a'}$  in  $[T]$  have an empty intersection. Then the constraint  $\forall t \in [T], \sum_{a \in [A]} u_k^a(t) \leq 1$  is automatically verified and the low-level problem of each customer can be separated into different optimization problems corresponding to the consumption vector  $u_k^a$  of each customer  $k$  for each application  $a$ . Each of these problems takes the following form:

### Problem 7

$$\begin{aligned} & \max_{u_k^a \in \{0,1\}^T} \sum_{t=1}^T [\rho_k^a(t) + \alpha_k^a y^{a,b}(t, L_t^k)] u_k^a(t) \\ \text{s.t. } & \sum_{t=1}^T u_k^a(t) = R_k^a, \quad \forall t \in \mathcal{I}_k^a, a \in [A], u_k^a(t) = 0 . \end{aligned} \quad (7)$$

We denote by  $F_k^a$  the feasible set of this problem. The above assumption (that the complements of  $\mathcal{I}_k^a$  and  $\mathcal{I}_k^{a'}$  have an empty intersection) is relevant in particular if only one kind of application is sensitive to price incentives. For instance, requests for downloading data can be anticipated (see [29]) and it makes sense to assume that customers are only sensitive to incentives for this kind of contents. In this case, the assumption means that customers wanting to download data can shift their consumption only at instants when they do not request another kind of content.

Moreover, under this assumption, the decomposition theorem is still valid and Problem 2 can be solved with the following method:

**Theorem 9 (Decomposition (general case))** *The bilevel problem 2 can be solved as follows:*

1. Find an optimal solution  $(N^{a,b})^*$  to the high level problem with unknown  $N^{a,b}$  for each  $a \in [A]$ ,  $b \in [B]$ :

### Problem 8

$$\begin{aligned} & \max_{N^{a,b} \in \sum_k F_k^a} \sum_{t,l} \left( \sum_{a \in [A]} \sum_{b \in [B]} \gamma_b N^{a,b}(t,l) s_l^{a,b}(N(t,l)) \right) \\ \text{s.t. } & \forall t, l, N(t,l) = \sum_{a \in [A]} \sum_{b \in [B]} N^{a,b}(t,l) \quad \text{and} \quad \forall t, l, N(t,l) \leq N_l^C . \end{aligned}$$

2. For each  $a \in [A]$  and  $b \in [B]$ , find vectors  $((u_k^a)^*)_{k \in \mathcal{K}_b}$  solutions of the following problem:

$$\max_{\substack{(u_k^a \in F_k^a)_{k \in \mathcal{K}_b} \\ \sum_{k \in \mathcal{K}_b} u_k^a = (N^{a,b})^*}} \sum_{k \in \mathcal{K}_b} \langle \rho_k^a, u_k^a \rangle .$$

3. Find for each  $a \in [A]$  and  $b \in [B]$  a vector  $y_{a,b}^*$  such that  $\forall k \in \mathcal{K}_b$ ,

$$(u_k^a)^* \in \arg \max_{u_k^a \in F_k^a} \langle \rho_k^a, u_k^a \rangle$$

*Proof.* The different problems corresponding for each  $a \in [A]$ , for each  $b \in [B]$  and for each  $k \in \mathcal{K}_b$  to Problem 7 are independent. Thus, according to Lemma 6, the global bilevel program consists in solving Problem 8. Moreover, the optimal decomposition of  $(N^{a,b})^*$  and the optimal price vector  $(y^{a,b})^*$  are totally independent for each  $a \in [A]$  and  $b \in [B]$ . Then, the proof of the last two parts in the theorem is the same as in Theorem 1.  $\square$

The last two parts of Theorem 9 are independent for each  $a \in [A]$  and  $b \in [B]$ . Thus, they can be solved similarly as in the case of one kind of application and one kind of contracts, studied in Section 3. We need to solve Problem 8. The function to optimize is separable (it can be written as a sum of function depending only of one coordinate), but these functions are not concave in  $(N^{1,1}, \dots, N^{A,B}) \in \mathbb{R}^{n_{AB}}$ . However, because each function  $s_l^{a,b}$  is concave nonincreasing and each  $N^{a,b}(t, l)$  is positive, we notice that  $\forall a' \in [A], b' \in [B]$ , the function which sends  $N^{a',b'}(t, l)$  to  $\sum_{a \in [A]} \sum_{b \in [B]} \gamma_b N^{a,b}(t, l) s_l^{a,b}(N(t, l))$  is still concave. Consequently, the function to optimize in Problem 8 is  $M$ -concave in each vector  $N^{a,b} \in \mathbb{Z}^{T \times L}$  considered separately, the other one being fixed. This leads to a block descent method, in which we use the same scheme as in Algorithm 1, successively, to maximize the objective function over every vector  $N^{a,b}$ . We denote by  $f(N^{1,1}, \dots, N^{A,B})$  the objective function of the high-level problem. We consider for each  $a, b$  a vector  $N^{a,b} \in \sum_{k \in \mathcal{K}_b} F_k^a$ . For each couple  $(a, b)$  taken successively, we find  $(i^{a,b}, j^{a,b})$  belonging to:

$$\arg \max_{(k,l) \text{ s.t. } N^{a,b} - e_k + e_l \in \sum_{k \in \mathcal{K}_b} F_k^a} f(N^{1,1}, \dots, N^{a,b} - e_k + e_l, \dots, N^{A,B})$$

If  $f(N^{1,1} - e_{i^{a,b}} + e_{j^{a,b}}, \dots, N^{A,B} - e_{i^{a,b}} + e_{j^{a,b}}) \leq f(N^{1,1}, \dots, N^{A,B})$ , then the algorithm stops and returns  $(N^{1,1}, \dots, N^{A,B})$ . Otherwise, we take for each  $a, b$ ,  $N^{a,b} := N^{a,b} - e_{i^{a,b}} + e_{j^{a,b}}$  and begin again. Consequently, Algorithm 2 can be modified to solve the bilevel problem 6 in the general case. It leads to Algorithm 4. The function *GraphSearch*, *ShortestPath* and *ShortestPath2* are the same as for Algorithm 2. The function  $f^*$  is here defined by:

$$f^* : (N^{1,1}, \dots, N^{A,B}) \mapsto \sum_t \sum_l \left[ \sum_{a \in [A]} \sum_{b \in [B]} \gamma_b N^{a,b}(t, l) s_l^{a,b}(N(t, l)) - M \max(N(t, l) - N_l^C, 0) \right]$$

with  $N(t, l) = \sum_{a \in [A]} \sum_{b \in [B]} N^{a,b}(t, l)$ .

Because the objective function of Problem 8 is not  $M$ -convex in  $(N^{1,1}, \dots, N^{A,B})$ , we have no guarantee of convergence of Algorithm 4 to a global optimal of the function  $f^*$ . However, we can characterize the nature of the optimum returned by Algorithm 4. In order to estimate the complexity of Algorithm 4, we define the function  $\Delta f^*$  by:

$$\begin{aligned} \Delta f^*(N^{1,1}, \dots, N^{A,B}) &= -f^*(N^{1,1}, \dots, N^{A,B}) \\ &+ \max_{\substack{u^{a,b}, v^{a,b} \in [T] \times [L] \\ N^{a,b} - e_{u^{a,b}} + e_{v^{a,b}} \in \sum_{k \in \mathcal{K}_b} F_k^a}} f^*(N^{1,1} - e_{u^{a,b}} + e_{v^{a,b}}, \dots, N^{A,B} - e_{u^{a,b}} + e_{v^{a,b}}) \end{aligned}$$

If for each  $a, b$  we have  $u^{a,b} = v^{a,b}$ , then  $\Delta f^*(N^{1,1}, \dots, N^{A,B}) = 0$ . Thus, we have

$$\Delta f^*(N^{1,1}, \dots, N^{A,B}) \geq 0$$

**Algorithm 4** Solving the bilevel problem for an arbitrary number of types of contracts.

**Require:**  $(u_k^a)^* \in F_k^a, \forall a \in [A], \forall k \in [K], \rho_k^a, \forall a \in [A], \forall k \in [K], f^*, \text{GraphSearch}, \text{ShortestPath}, \text{ShortestPath2}, s \in [n]$

**Ensure:**  $N^*$  optimal number of customers,  $y^*$  optimal discount vector

```

for all  $a \in [A], b \in [B]$  do
   $N^{a,b} \leftarrow \sum_{k \in \mathcal{K}_b} (u_k^a)^*$ 
  for all  $k \in \mathcal{K}_b$  do
    for all  $t, t' \notin \mathcal{I}_k^a$  do
       $i = (t, L_k(t)) \quad j = (t', L_k(t'))$ ,
      if  $u_k^a(i) = 1$  and  $u_k^a(j) = 0$  then
         $w_{ij}^{k,a} \leftarrow \rho_k^a(i) - \rho_k^a(j)$ 
      end if
    end for
  end for
  for all  $t, t' \in [T], l, l' \in [L]$  do
     $i = (t, l) \quad j = (t', l')$ 
     $w_{ij}^{a,b} \leftarrow \min_{k \in \mathcal{K}_b} w_{ij}^{k,a}; \quad k_{ij}^{a,b} \in \arg \min_{k \in \mathcal{K}_b} w_{ij}^{k,a}$ 
  end for
end for
 $stop \leftarrow 0$ 
while  $stop = 0$  do
   $g_N \leftarrow f^*(N^{1,1}, \dots, N^{A,B});$ 
  for all  $a \in [A], b \in [B]$  do
     $c^{a,b} \leftarrow \text{GraphSearch}(w^{a,b})$ 
     $g^* \leftarrow \max_{u,v \in [T] \times [L]} c_{uv}^{a,b} f^*(N^{1,1}, \dots, N^{a,b} - e_u + e_v, \dots, N^{A,B}); \quad i^{a,b}, j^{a,b} \in$ 
     $\arg \max_{u,v \in [T] \times [L]} c_{uv}^{a,b} f^*(N^{1,1}, \dots, N^{a,b} - e_u + e_v, \dots, N^{A,B})$ 
  end for
  if  $g^* \leq g_N$  then
     $stop \leftarrow 1$ 
  else
    for all  $a \in [A], b \in [B]$  do
       $(v, path) \leftarrow \text{Shortestpath}(w^{a,b}, i, j); \quad N^{a,b} \leftarrow N^{a,b} - e_{i,a,b} + e_{j,a,b}$ 
      for  $q = 1$  to  $\text{Length}(path) - 1$  do
         $\alpha \leftarrow path(q); \quad \beta \leftarrow path(q+1); \quad k \leftarrow k_{\alpha\beta}^{a,b}; \quad (u_k^a)^*(\alpha) = 0; \quad (u_k^a)^*(\beta) = 1$ 
        for all  $\gamma \notin \mathcal{I}_k^a$  do
           $w_{\alpha\gamma}^{k,a} \leftarrow +\infty; \quad w_{\gamma\beta}^{k,a} \leftarrow +\infty$ 
          if  $(u_k^a)^*(\gamma) = 1$  then
             $w_{\gamma\alpha}^{k,a} \leftarrow \rho_k^a(\gamma) - \rho_k^a(\alpha)$ 
          else
             $w_{\beta\gamma}^{k,a} \leftarrow \rho_k^a(\beta) - \rho_k^a(\gamma)$ 
          end if
        end for
      end for
    end for
    for all  $t, t' \in [T], l, l' \in [L]$  do
       $i = (t, l) \quad j = (t', l')$ 
       $w_{ij}^{a,b} \leftarrow \min_{k \in \mathcal{K}_b} w_{ij}^{k,a}; \quad k_{ij}^{a,b} \in \arg \min_{k \in \mathcal{K}_b} w_{ij}^{k,a}$ 
    end for
  end for
end while
 $M \leftarrow 1 + \text{ABTL} \max_{i,j \in [T] \times [L], a \in [A], b \in [B]} w_{ij}^{a,b}$ 
for all  $t \in [n]$  do
  if  $t \neq s$  AND  $w_{st} = +\infty$  then
     $w_{st} = M$ 
  end if
   $y^* \leftarrow \text{Shortestpath2}(w, s)$ 
   $y_s^* \leftarrow 0$ 
end for

```

. Because the set  $\prod_{a,b}(\sum_{k \in \mathcal{K}_b} F_k^a)$  is finite, we can define the value  $\delta g$  by:

$$\delta g = \min_{\substack{N^{a,b} \in \sum_{k \in \mathcal{K}_b} F_k^a \\ \Delta f^*(N^{1,1}, \dots, N^{A,B}) > 0}} \Delta f^*(N^{1,1}, \dots, N^{A,B})$$

because  $f^*$  has not a constant value.

**Theorem 10** *Let us define  $\gamma_{\max} = \max_{b \in [B]} \gamma_b$ . Let us also define  $R = \sum_{a \in [A]} \sum_{k \in [K]} R_k^a$ , for each  $a \in [A]$  and  $k \in [K]$   $n_k^a = TL - \#\mathcal{J}_k^a$  (that is the number of possible non-zero coordinates of the vectors of  $F_k^a$ ) and  $\bar{R} = \sum_a \sum_k R_k^a (n_k^a - R_k^a)$ . Algorithm 4 terminates in  $O(\frac{\gamma_{\max} R}{\delta g} (AB(TL)^3 + \bar{R}))$  time and  $O(\bar{R})$  space, and returns vectors  $(y^{a,b})_{a \in [A], b \in [B]}^*$  and  $(N^{a,b})_{a \in [A], b \in [B]}^*$  such that  $\forall a \in [A], b \in [B], \forall N^{a,b} \in \sum_k \mathcal{K}_b F_k^a$ :*

$$f^*((N^{1,1})^*, \dots, (N^{a,b})^*, \dots, (N^{A,B})^*) \geq f^*((N^{1,1})^*, \dots, N^{a,b}, \dots, (N^{A,B})^*)$$

*Proof.* Algorithm 4 continues while the value  $g^*$  is strictly larger than  $g_N$ . Because the set  $\prod_{a,b}(\sum_{k \in \mathcal{K}_b} F_k^a)$  is finite, the algorithm terminates. When it stops, the vector  $(N^{a,b})_{a \in [A], b \in [B]}$  is such that  $\forall a \in [A], b \in [B], \forall u, v \in [T] \times [L]$ :

$$f(N^{1,1}, \dots, N^{a,b} - e_u + e_v, \dots, N^{A,B}) \leq f(N^{1,1}, \dots, N^{a,b}, \dots, N^{A,B})$$

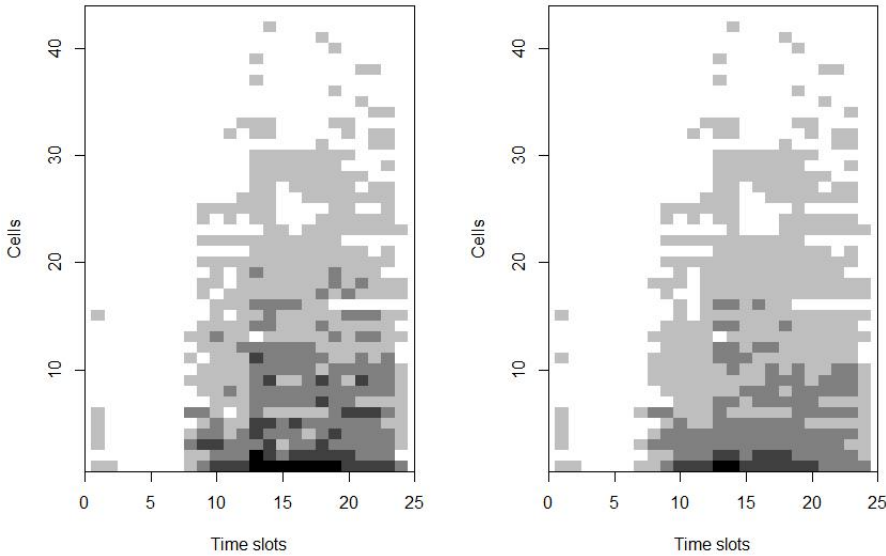
For each  $a, b$ , the function  $N^{a,b} \mapsto f(N^{1,1}, \dots, N^{a,b}, \dots, N^{A,B})$  is  $M$ -concave. The statement of the theorem comes straightforwardly from the equivalence between local and global optimality for  $M$ -concave functions.

Algorithm 4 differs from Algorithm 2 by the different applications and kind of contracts and by the number of iterations of the loop. The set  $[K]$  of customers is split following the different kind of contracts  $b \in [B]$ . Thus, we have to define the parameters  $w_{ij}^{k,a}$  for each  $k \in [K]$  and  $a \in [A]$  and the global space complexity becomes  $\sum_a \sum_k R_k^a (n_k^a - R_k^a) = \bar{R}$ . The number of iterations of the loop can be estimated with a pseudo-polynomial bound. The algorithm continues while  $g^* > g_N$ . Then, the new value of  $g^*$  is  $f^*(N^{1,1} - e_{i_1,1} + e_{j_1,1}, \dots, N^{A,B} - e_{i_{A,B}} + e_{j_{A,B}})$ . Consequently, at each iteration of the loop, the value of  $g^*$  increases of at least  $\delta g$  until the algorithm stops. The finite values of  $f^*$  are nonnegative, and an upper bound is  $(\max_{b \in [B]} \gamma_b) (\sum_a \sum_{k \in [K]} R_k^a) = \gamma_{\max} R$  because each function  $s_l$  takes values between 0 and 1. In each loop, the number of operations is  $O(\bar{R} + AB(TL)^3)$  to calculate the new values of  $w_{ij}^{a,b}$  and to solve a shortest path problem for each  $a$  and  $b$  in the graph  $G^{a,b}$  with nodes corresponding to all couples in  $[T] \times [L]$  and edges with values  $w_{ij}^{a,b}$  between vertices  $i, j \in [T] \times [L]$ .  $\square$

## 7 Experimental results

We consider an application based on real data provided by Orange. It involves the data consumptions in an area of  $L = 43$  cells, during one day divided in time slots of one hour, that is  $T = 24$  time slots. We will focus here our study on price incentives only for download contents. During this day, a number  $K$  of more than 2500 customers make some requests for downloading data in this area and we are interested in balancing the number of active customers in the network. Even though they are insensitive to price incentives, other kind of requests (web, mail, etc.) have to be satisfied and they are taken into account in the high level optimization problem. We consider two classes of users: standard and premium customers. The premium ones demand a better quality of service. Hence, they are less satisfied than the standard customers if they share their cell with a given number of active customers. We therefore define the satisfaction function as in Section

2. The provider wants to favor the premium customers. Hence, we take  $\gamma_b = 2$  for the latter ones and  $\gamma_b = 1$  for the standard customers, in the high-level optimization problem. We also assume that the premium customers are less sensitive to the incentives, and thus take  $\alpha_k^a = 1/2$  for all standard customers and  $\alpha_k^a = 1$  for all premium customers in the low-level problem 1. We estimate very simply the parameters  $\rho_k$ . We take  $\rho_k(t) = 1$  when the customer  $k$  consumes download at time  $t$  without incentives,  $\rho_k(t) = 0$  when he does not make any request without incentives but makes a request for download at times  $t - 1$  or  $t + 1$  (we assume he could shift his consumption of one hour) and  $\rho_k(t) = -\infty$  otherwise.

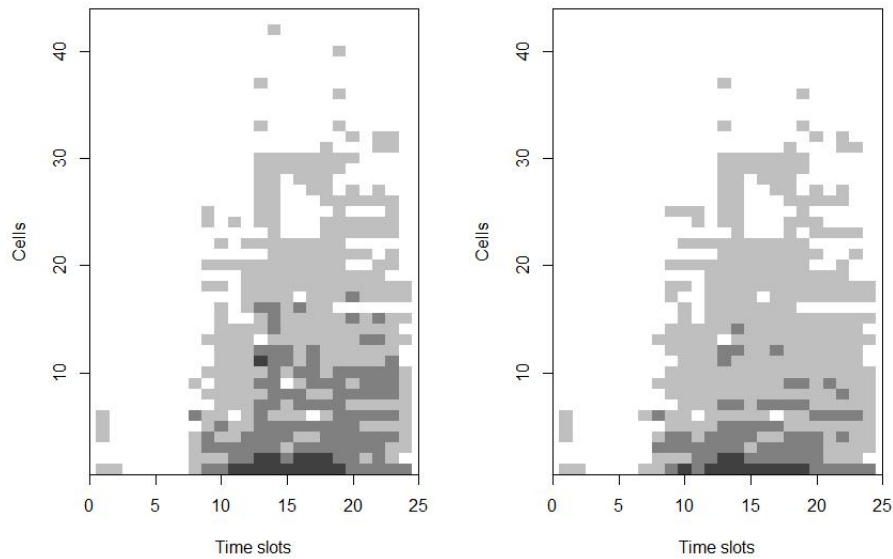


**Fig. 6** Satisfaction of premium customers for streaming without (left) and with (right) incentives. The grey level indicates the satisfaction: critical unsatisfaction,  $s < 0.3$  (black),  $0.3 < s < 0.7$  (dark grey),  $0.7 < s < 0.9$  (grey),  $0.9 < s < 0.99$  (light grey) and complete satisfaction  $0.99 < s$  (white).

We solve the bilevel problem using Algorithm 4, implemented in `Scilab`. The computation took 9526 seconds on a single core of an Intel i5-4690 processor @ 3.5 GHz.

On Figures 6–9, we show the evolution of the satisfaction of different kind of customers for different kind of contents without and with incentives. These results show that price incentives have an effective influence on the load, especially in the most loaded cells (the number of black regions in the space-time coordinates, in which the unsatisfaction of the users is critical, is considerably reduced). Moreover, Figure 10 reveals that the consumption of users is not only moved in time, but also in space: not only some consumption is moved from the peak hour to the night (off peak), but the surface of the dark grey region, representing the total download consumption in the cell over the whole day, is decreased, indicating that some part of the consumption has been shifted to other cells.





**Fig. 7** Satisfaction of standard customers for streaming without (left) and with (right) incentives

## 8 Conclusion

We presented here a bilevel model for price incentives in data mobile networks. We solved this problem by a decomposition method based on discrete convexity and tropical geometry. We finally applied our results to real data. In further work, we shall consider more general models: unfixed number of requests, nonlinear preferences of the customers, satisfaction functions of the provider taking into account the profit. Stochastic models shall also be considered in particular to take into account the partial information of the provider about the customers preferences and trajectories.

## 9 Acknowledgments

We thank the reviewers of our earlier work [11] for their remarks and comments. We also thank Orange for providing us real data for our experimental results.

## References

1. Altman, E., Barman, D., El Azouzi, R., Ros, D., Tuffin, B.: Pricing differentiated services: A game-theoretic approach. *Computer Networks* **50**(7), 982–1002 (2006)
2. Baccelli, F., Cohen, G., Olsder, G., Quadrat, J.: *Synchronization and Linearity*. Wiley (1992)
3. Baldwin, E., Klemperer, P.: Tropical geometry to analyse demand. Tech. rep., Working paper, Oxford University (2012)
4. Bellman, R.: On a routing problem. *Quarterly of applied mathematics* **16**(1), 87–90 (1958)
5. Bonald, T., Feuillet, M.: *Network performance analysis*. John Wiley & Sons (2013)

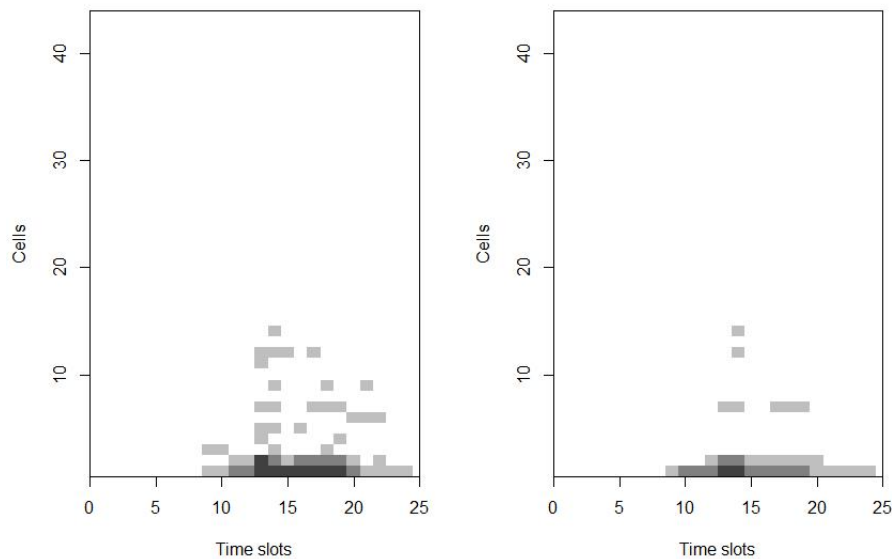
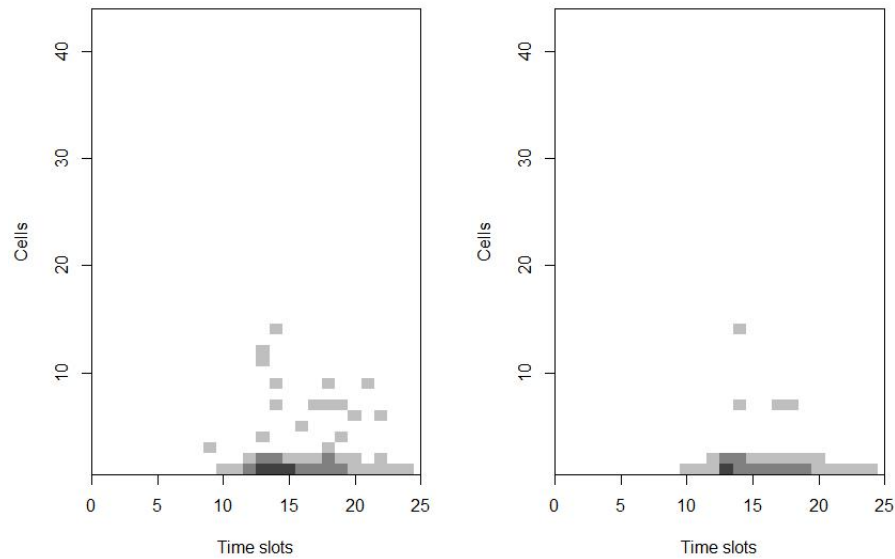
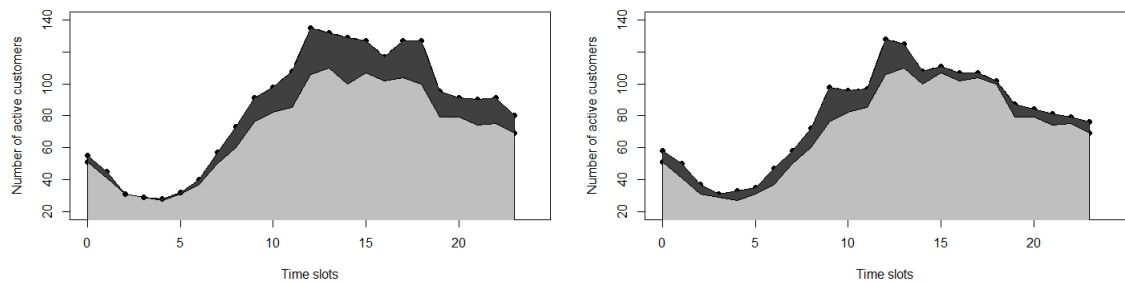


Fig. 8 Satisfaction of premium customers for web, mail or download without (left) and with (right) incentives

6. Brotcorne, L., Labbé, M., Marcotte, P., Savard, G.: A bilevel model and solution algorithm for a freight tariff-setting problem. *Transportation Science* **34**(3), 289–302 (2000)
7. Butkovič, P.: *Max-linear systems : theory and algorithms*. Springer monographs in mathematics. Springer (2010)
8. Colson, B., Marcotte, P., Savard, G.: An overview of bilevel optimization. *Annals of operations research* **153**(1), 235–256 (2007)
9. Danilov, V.I., Koshevoy, G.A.: Discrete convexity and unimodularity—i. *Advances in Mathematics* **189**(2), 301–324 (2004)
10. Dempe, S.: *Bilevel programming: A survey*. Dekan der Fak. für Mathematik und Informatik (2003)
11. Eytard, J.B., Akian, M., Bouhtou, M., Gaubert, S.: A bilevel optimization model for load balancing in mobile networks through price incentives. In: *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2017 15th International Symposium on, pp. 1–8. IEEE (2017)
12. Ford Jr, L.R.: *Network flow theory*. Tech. rep., RAND CORP SANTA MONICA CA (1956)
13. Gabow, H.N., Tarjan, R.E.: Faster scaling algorithms for network problems. *SIAM Journal on Computing* **18**(5), 1013–1036 (1989)
14. Ha, S., Sen, S., Joe-Wong, C., Im, Y., Chiang, M.: Tube: time-dependent pricing for mobile data. *ACM SIGCOMM Computer Communication Review* **42**(4), 247–258 (2012)
15. Itenberg, I., Mikhalkin, G., Shustin, E.I.: *Tropical algebraic geometry*, vol. 35. Springer Science & Business Media (2009)
16. Lee, J.W., Mazumdar, R.R., Shroff, N.B.: Non-convex optimization and rate control for multi-class services in the internet. *IEEE/ACM transactions on networking* **13**(4), 827–840 (2005)
17. Ma, Q., Liu, Y.F., Huang, J.: Time and location aware mobile data pricing. In: *Communications (ICC)*, 2014 IEEE International Conference on, pp. 3235–3240. IEEE (2014)
18. Maclagan, D., Sturmfels, B.: *Introduction to Tropical Geometry*, *Graduate Studies in Mathematics*, vol. 161. American Mathematical Society, Providence, RI (2015)
19. Maillé, P., Tuffin, B.: Pricing the internet with multibid auctions. *IEEE/ACM transactions on networking* **14**(5), 992–1004 (2006)
20. Maillé, P., Tuffin, B.: *Telecommunication network economics: from theory to applications*. Cambridge University Press (2014)
21. Moety, F., Bouhtou, M., En-Najjary, T., Nasri, R.: Joint optimization of user association and user satisfaction in heterogeneous cellular network. In: *28th International Teletraffic Congress* (2016)



**Fig. 9** Satisfaction of standard customers for web, mail or download without (left) and with (right) incentives



**Fig. 10** Traffic in the most loaded cell. The light grey part represents the web, mail and streaming customers who have no incentives and are fixed. The dark grey part corresponds to the download customers in the cell without (left) and with (right) incentives

22. Murota, K.: Discrete convex analysis. SIAM (2003)
23. Olkin, I., Marshall, A.W.: Inequalities: theory of majorization and its applications, vol. 143. Academic press (1979)
24. Renegar, J.: A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical Programming* **40**(1-3), 59–93 (1988)
25. Rockafellar, R.T.: Convex analysis. Princeton university press (1970)
26. Schrijver, A.: Combinatorial optimization: polyhedra and efficiency, vol. 24. Springer Science & Business Media (2003)
27. Sen, S., Joe-Wong, C., Ha, S., Chiang, M.: A survey of smart data pricing: Past proposals, current plans, and future trends. *ACM Computing Surveys (CSUR)* **46**(2), 15 (2013)
28. Sen, S., Joe-Wong, C., Ha, S., Chiang, M.: Smart Data Pricing. John Wiley & Sons (2014)

- 
29. Tadrous, J., Eryilmaz, A., El Gamal, H.: Pricing for demand shaping and proactive download in smart data networks. In: INFOCOM, 2013 Proceedings IEEE, pp. 3189–3194. IEEE (2013)
  30. Tran, N.M., Yu, J.: Product-mix auctions and tropical geometry. arXiv preprint arXiv:1505.05737 (2015)