Vietnam J Comput Sci (2014) 1:15–28
DOI 10.1007/s40595-013-0010-5

REGULAR PAPER

DC programming in communication systems: challenging problems and methods

Hoai An Le Thi · Tao Pham Dinh

Received: 28 October 2013 / Accepted: 30 October 2013 / Published online: 28 November 2013

Abstract Nonconvex optimization becomes an indispensable and powerful tool for the analysis and design of Communication Systems (CS) since the last decade. As an innovative approach to nonconvex programming, Difference of Convex functions (DC) programming and DC Algorithms (DCA) are increasingly used by researchers in this field. The objective of this paper is to show that many challenging problems in CS can be modeled as DC programs and solved by DCA-based algorithms. We offer the community of researchers in CS promising approaches in a unified DC programming framework to tackle various applications, such as routing, power control, congestion control of the Internet, resource allocation in networks, etc.

Keywords Communication systems · DC programming · DCA · Exact penalty · DC constraint · Mixed 0–1 DC programs

1 Introduction

Optimization plays a key role in communication systems since most of issues of this domain are related to optimization problems. Convex programming has been studied for about a century. It has provided both a powerful tool and an intriguing mentality for the analysis and design of communication systems over several years in the past. During the last decade, an increasing amount of effort has been put into nonconvex optimization to deal with challenge problems appeared in many applications of this filed (in fact, most real-life problems are of nonconvex nature). The absence of convexity creates a source of difficulties of all kinds, in particular, the distinction between the local and global minima, the nonexistence of verifiable characterizations of global solutions, etc., that causes all the computational complexity while passing from convex to nonconvex programming. In general, unlike the convex programming, there is no iterative algorithm converging directly to a global solution of a nonconvex program. Finding a global solution of a nonconvex program, especially in the large-scale setting, is the holy grail of optimizers.

The special context of practical problems in CS, along with the dramatic progress of novel technologies, requires well-adapted optimization techniques. For example, solution methods to network management in the context of mobile service should take into account the following questions:

- The topology of networks is dynamic and real-time data transmissions are needed. Hence, real-time algorithms are expected.
- Self-organization and self-configuration require all protocols in mobile networks to be distributive and collaborative. By the way, distributed algorithms are necessary.
- Location/tracking management, in addition to the handover management and routing. In the case of hybrid communication networks, the choice of the best access gateway among a number of available access technologies becomes one of the important considerations. Routing in hybrid networks should be handled by finding a suitable mathematical model and efficient algorithms.
- Multi-user communications service involves large scale setting optimization problems. Therefore, the algorithms should be able to solve large-size problems.

H. A. Le Thi (✉)
Laboratory of Theoretical and Applied Computer Science, UFR MIM, University of Lorraine, Ile du Saulcy, 57045 Metz, France
e-mail: hoai-an.le-thi@univ-lorraine.fr

T. Pham Dinh
Laboratory of Mathematics, LMI,
National Institute for Applied Sciences, Rouen, Avenue de l'Université, 76801 Saint-Etienne-du-Rouvray Cedex, France
e-mail: pham@insa-rouen.fr

Many challenging issues arise from nonconvex optimization in communication systems, especially how to design suitable models and develop efficient, fast, scalable and distributed algorithms to tackle large-scale practical problems in the areas of wireless networking, internet engineering, mobile services in self-organized hybrid networks.

A variety of nonconvex optimization techniques have been recently developed by researchers in optimization on one side and in communications systems on another side for studying communication theory as well as for solving practical problems in CS (see e.g, [1–9,12,14–21,24–33,38–54]. Generally, there are two different, but complementary approaches: global approaches such as Cutting Plane (CP), Branch and Bound (B&B), Branch and Cut (B&C) can guarantee the globality of the solutions but they are very expensive, and cannot handle problems of high dimensionality; and local approaches, on the contrary, are much faster while only local minima are available. Many current approaches are not generally effective for practical large-scale problems. Finding efficient algorithms that realize a compromise to overcome these drawbacks is a challenge of nonconvex programming. Such algorithms must exploit domain-specific structures of the problems being considered.

As an innovative approach to nonconvex programming, Difference of Convex functions (DC) programming and DC Algorithms (DCA) are increasingly used by researchers in CS (see e.g. [1,2,16,22,45,54] and references therein). The objective of this paper is to show that many challenging problems in CS can be modeled as DC programs and solved by DCA-based algorithms. We offer the community of researchers in CS promising approaches in a unified DC programming framework to tackle various applications such as routing, power control, congestion control of the Internet, resource allocation in networks, etc.

1.1 Nonconvex optimization problems in CS

In terms of optimization, nonconvex problems appeared in CS can be divided into three classes:

- minimizing a nonconvex function on a convex set;
- minimizing a convex/nonconvex function on a nonconvex set;
- minimizing a convex/nonconvex function on a convex/nonconvex set with integer variables.

The reader will see that these classes of nonconvex programs in CS can be formulated or reformulated as DC programs and solved by DCA.

1.2 Why DC programming and DCA?

DC programming is an extension of convex programming which is sufficiently large to cover almost all nonconvex optimization problems, but not much to still allow using the arsenal of powerful tools in convex analysis and convex optimization. DC programming and DCA constitute the backbone of nonconvex programming and global optimization. The use of DCA for solving nonconvex optimization problems in CS is motivated by the following facts:

- DCA is a philosophy rather than an algorithm. For each problem, we can design a family of DCA-based algorithms. The flexibility of DCA on the choice of DC decomposition offers DCA schemes having the potential to outperform standard methods.
- By exploiting the nice effect of DC decomposition of the objective function we can build distributed algorithms. This issue is very important in communication networks that involve multi-users, in particular in the purpose of personalized mobile services.
- Convex analysis provides powerful tools to prove the convergence of DCA in a generic framework. Hence, any DCA-based algorithm enjoys (at least) general convergence properties of the generic DCA scheme that are already available.
- DCA is an efficient, fast and scalable method for smooth/nonsmooth nonconvex programming. To the best of our knowledge, DCA is actually one of the rare algorithms for nonsmooth nonconvex programming which allows to solve large-scale DC programs. DCA was successfully applied to a lot of different and various nonconvex optimization problems to which it quite often gave global solutions and proved to be more robust and more efficient than related standard methods. In particular, DCA has already efficiently solved large-scale DC programs in network optimization (see [1,2,24–27,29–31,45–51,54] and the list of references in [22]).

We will show how to solve these three classes of problems in CS by DC programming and DCA. For beginning, let us give in Sect. 2, a brief introduction of DCA programming and DCA. The solution methods of each class of problem will be presented in Sects. 3, 4, and 5 where, in addition to development of generic models and algorithms, methods for typical applications in CS will be illustrated. In Sect. 6, we mention another issue in CS for which DCA can also be investigated. Section 7 concludes the paper.

2 DC programming and DCA

2.1 A brief introduction

We are working with the space $X = \mathbb{R}^n$ which is equipped with the canonical inner product $\langle \cdot, \cdot \rangle$ and the corresponding

Euclidean norm $\|\cdot\|$, thus the dual space Y of X can be identified with X itself. We follow [13,37], for definitions of usual tools in modern convex analysis, where functions could take the infinite values $+\infty$. A function $\theta : X \to \mathbb{R} \cup \{+\infty\}$ is said to be proper if it is not identically equal to $+\infty$. The effective domain of θ, denoted by dom θ, is

$$\text{dom}\,\theta = \{x \in X : \theta(x) < +\infty\}.$$

The indicator function χ_C of a nonempty closed convex C set is defined by $\chi_C(x) = 0$ if $x \in C$, $+\infty$ otherwise. The set of all lower semicontinuous proper convex functions on X is denoted by $\Gamma_0(X)$. Let $\theta \in \Gamma_0(X)$, then the conjugate function of θ, denoted θ^*, is defined by

$$\theta^*(y) = \sup\{\langle x, y\rangle - \theta(x) : \ x \in X\}.$$

We have $\theta^* \in \Gamma_0(Y)$ and $\theta^{**} = \theta$.

Nonsmooth convex functions are handled using the concept of subdifferentials. For $\theta \in \Gamma_0(X)$ and $x_0 \in$ dom θ, $\partial\theta(x_0)$ denotes the subdifferential of θ at x_0, and is defined by

$$\partial\theta(x_0) := \{y \in Y : \theta(x) \geq \theta(x_0) + \langle x - x_0, y\rangle, \ \forall x \in X\}. \tag{1}$$

Each $y \in \partial\theta(x_0)$ is called a subgradient of θ at x_0. The subdifferential $\partial\theta(x_0)$ is a closed convex set in Y. It generalizes the derivative in the sense that θ is differentiable at x_0 if and only if $\partial\theta(x_0) \equiv \{\nabla\theta(x_0)\}$. Recall the well-known properties related to subdifferential calculus of $\theta \in \Gamma_0(X)$:

$$y_0 \in \partial\theta(x_0) \Leftrightarrow x_0 \in \partial\theta^*(y_0) \Leftrightarrow \langle x_0, y_0\rangle = \theta(x_0) + \theta^*(y_0); \tag{2}$$

$$\partial\theta^*(y_0) = \text{argmin}\{\theta(x) - \langle x, y_0\rangle : x \in X\}. \tag{3}$$

A function $\theta \in \Gamma_0(X)$ is said to be polyhedral convex if

$$\theta(x) = \max\{\langle a_i, x\rangle - \beta_i : i = 1, \ldots, m\} + \chi_C(x)\, \forall x \in X,$$

where C is a nonempty polyhedral convex set in X.

A DC program is of the form

$$(P_{\text{dc}})\, \alpha = \inf\{f(x) := g(x) - h(x) : \ x \in X\}, \tag{4}$$

with $g, h \in \Gamma_0(X)$. Such a function f is called a DC function, and $g - h$, a DC decomposition of f, while the convex functions g and h are DC components of f. In (P_{dc}) the nonconvexity comes from the concavity of the function—h (except the case h is affine since (P_{dc}) then is a convex program). It should be noted that a convex constrained DC program can be expressed in the form (4) by using the indicator function on C, that is

$$\begin{aligned}\inf\{f(x) &:= g(x) - h(x) : \ x \in C\} \\ &= \inf\{\varphi(x) - h(x) \ : \ x \in X\}, \text{with } \varphi := g + \chi_C.\end{aligned}$$

Hence, throughout this paper, DC program of the form (4) is referred to as "standard DC program".

Polyhedral DC programs (P_{dc}) (i.e., when g or h are polyhedral convex) play a key role in nonconvex programming (see [25,34,35] and references therein), and enjoy interesting properties related to local optimality and DCA's convergence.

The DC duality is based on the conjugate functions and the fundamental characterization of a convex function $\theta \in \Gamma_0(X)$ as *the pointwise supremum of a collection of affine minorants:*

$$\theta(x) = \sup\{\langle x, y\rangle - \theta^*(y) : y \in Y\}, \qquad \forall x \in X. \tag{5}$$

That associates the primal DC program (4) (P_{dc}) with its dual DC program (D_{dc}) defined by

$$(D_{\text{dc}}) \quad \alpha = \inf\{h^*(y) - g^*(y) : y \in Y\}, \tag{6}$$

and investigates their mutual relations. We observe the perfect symmetry between primal and dual DC programs: the dual to (D_{dc}) is exactly (P_{dc}).

It is worth noting the wealth of the vector space DC$(X) = \Gamma_0(X) - \Gamma_0(X)$ spanned by the "convex cone" $\Gamma_0(X)$ [34, 35]: it contains most realistic objective functions and is closed under operations usually considered in optimization.

The complexity of DC programs resides, of course, in the lack of verifiable globality conditions. Lets recall the general local optimality conditions in DC programming (subdifferential's inclusion): if x^* is a local solution of (P_{dc}) then

$$\emptyset \neq \partial h(x^*) \subset \partial g(x^*). \tag{7}$$

The condition (7) is also sufficient (for local optimality) in many important classes of DC programs (see [34,35]).

A point x^* is said to be *a critical point* of g—h (or generalized KKT point for (P_{dc})) if

$$\partial h(x^*) \cap \partial g(x^*) \neq \emptyset. \tag{8}$$

Note that, by symmetry, the dual part of (7) and (8) are trivial.

DC Programming and DCA were introduced by Pham Dinh Tao in their preliminary form in 1985. These theoretical and algorithmic tools are extensively developed by Le Thi Hoai An and Pham Dinh Tao since 1994 to become now classic and increasingly popular. DCA is a continuous primal dual subgradient approach based on local optimality and duality in DC programming for solving standard DC programs (P_{dc}).

DCA's philosophy

The key idea behind DCA is to replace in (P_{dc}), at the current point x^k, the second component h with its affine minorant defined by

$$h_k(x) := h(x^k) + \langle x - x^k, y^k\rangle, \ y^k \in \partial h(x^k)$$

to give birth to the primal convex program of the form

$$(P_k) \quad \inf\{g(x) - h_k(x) : \; x \in X\}$$
$$\iff \inf\{g(x) - \langle x, y^k \rangle \; : x \in X\}$$

whose solution set is $\partial g^*(y^k)$.The next iterate x^{k+1} is taken in $\partial g^*(y^k)$.

Dually, a solution x^{k+1} of (P_k) is then used to define the dual convex program (D_{k+1}) obtained from (D_{dc}) by replacing g^* with its affine minorant

$$(g^*)_k(y) := g^*(y^k) + \langle y - y^k x^{k+1} \rangle$$

to obtain the convex program

$$(D_{k+1}) \; \inf\{h^*(y) - [g^*(y^k) + \langle y - y^k, x^{k+1}\rangle] : y \in Y\}$$
$$\Leftrightarrow \inf\{h^*(y) - \langle y, x^{k+1}\rangle \; : y \in Y\}$$

whose solution set is $\partial h(x^{k+1})$. The next iterate y^{k+1} is chosen in $\partial h(x^{k+1})$.

DCA scheme

Initialization: Let $x^0 \in \mathbb{R}^n$ be a guess, $k \leftarrow 0$.
Repeat

- Calculate $y^k \in \partial h(x^k)$
- Calculate $x^{k+1} \in \partial g^*(y^k)$, which is equivalent to

$$x^{k+1} \in \operatorname{argmin}\{g(x) - \langle x, y^k\rangle : x \in \mathbb{R}^n\} \quad (P_k)$$

- $k \leftarrow k + 1$

Until convergence of $\{x^k\}$.

DCA's convergence properties:

Convergence properties of DCA and its theoretical basis can be found in [25,34,35]. For instance, it is important to mention that

i) DCA is a descent method without linesearch but with global convergence: the sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing.
ii) If the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^k\}$ and $\{y^k\}$ are bounded, then every limit point x^* (resp. y^*) of the sequence $\{x^k\}$ (resp. $\{y^k\}$) is a critical point of $g - h$ (resp. $h^* - g^*$), i.e. $\partial h(x^*) \cap \partial g(x^*) \neq \emptyset$ (resp. $\partial h^*(y^*) \cap \partial g^*(y^*) \neq \emptyset$).
iii) DCA has a *linear convergence* for DC programs.
iv) DCA has a finite convergence for polyhedral DC programs.

For a complete study of DC programming and DCA, the reader is referred to [25,34,35] and the references therein. Without going into details, let us mention the key properties of DCA.

2.2 Key properties of DCA

a. *Flexibility* the construction of DCA is based on g and h but not on f itself, and there are as many DCA as there are DC decompositions. This is a crucial fact in DC programming. It is important to study various equivalent DC forms of a DC program, because each DC function f *has infinitely many DC decompositions which have crucial implications for the qualities* (speed of convergence, robustness, efficiency, globality of computed solutions, etc.) of DCA.
b. DCA is a *descent method without linesearch*, with global convergence (i.e. DCA converges from an arbitrary initial point).
c. *Return from DC programming to convex programming* DCA consists in an iterative approximation of a DC program by a sequence of convex programs that will be solved by appropriate convex optimization algorithms. This property is called successive convex approximation (SCA) in some recent works in CS.
d. *Versatility* With suitable DC decompositions *DCA generates most standard algorithms in convex and nonconvex optimization.* Hence DCA offers a wide framework for solving convex/nonconvex optimization problems. In particular, DCA is a global approach to convex programming, i.e., it converges to optimal solutions of convex programs reformulated as DC programs. Consequently, it can be used to build efficient customized algorithms for solving convex programs generated by DCA.

2.3 How to apply DCA for solving practical problems

It would be wrong to think that using DCA for solving a practical problem is a simple procedure. Indeed, the generic DCA scheme is an overall philosophical idea rather than a single algorithm. There is not only one DCA but *a family* of DCAs for a considered problem. While DC decompositions exist for a very large class of functions, there are no general procedure for determining such DC decompositions. The design of an efficient DCA for a concrete problem is an art which should be based on theoretical tools and on its special structure. It consists of the five steps :

a. Find a DC formulation of the considered optimization problem: this can be done if the feasible domain C is a convex set and the objective function f is DC, otherwise we must use the approximation or reformulation technique based on relevant theoretical tools.
b. Design a DCA scheme for (P_{dc}). This consists of i) computing a subgradient of h and ii) solving the convex program of the form (P_k). In the ideal case (what is not always possible, especially for nondifferentiable DC

programs) an optimal solution of (P_k) is explicitly determined, which corresponds to the explicit computation of a subgradient of g^*. Otherwise one should find efficient convex optimization algorithms suitably adapted to (P_k)'s specific structures in order to save computation time.

c. Search for "good" DC decompositions. If the computations in i) and ii) are not satisfactory (costly, computed solutions by DCA are not sufficiently good, ...) then one has to find more suitable DC decomposition. That is a difficult issue and should be done by exploiting distinctive features of the class of DC programs at hand. Here reformulation techniques play a key role to obtain suitable models. Reformulation techniques should be diversified and have recourse to good mathematical backgrounds in numerical analysis and optimization.
d. Search for "good" starting points. That can be done by combining with other approaches (heuristic or local search methods, solutions of convex relaxation problems in global optimization methods). Another efficient way in finding a convex minorant consists of the objective function on the feasible set C and solving the resulting convex program whose solution is used to initialize DCA. This strategy must be developed in depth and specifically, by exploiting the structure of the problem (P_{dc}).
e. Globalize DCA : to guarantee globality of sought solutions or to improve their quality it is advised to combine DCA with global optimization techniques.

It goes without saying that the two steps (c) and (d) and solution methods for convex programs (P_k) (if necessary) constitute *the key issues* for successful applications of DC programming and DCA.

We show below how to use DCA for solving the three classes of nonconvex problems in CS.

3 Minimizing a nonconvex function under a convex constraint set

The mathematical formulation of this class of problem is given by

$$\text{(P)}\ \ \inf\{f(x) : x \in \text{C}\} \tag{9}$$

where $C \subset \mathbb{R}^n$ is a closed nonempty convex set and f is a nonconvex function on $\mathbb{R}^n$. Many applications in CS can be formulated in the form of (P) whose typical examples are Network Utility Maximization (NUM) [9,30], power control problem [2,6,48,54]), dynamic spectrum management in DSL systems [27,53] MIMO relay optimization [15], sum-rate maximization, proportional-fairness and max–min optimization of SISO/MISO/MIMO ad hoc networks [14,17,43,44].

We will consider two examples of applications of DCA on NUM [30] and DSL [27]. Complete works on NUM and power control using DCA can be found in [24,48,31].

3.1 Can one get a DC formulation for any problem in this class?

The answer is "yes", one can always formulate (P) as a standard DC program of the form (P_{dc}). In fact, as indicated above, the vector space $\mathcal{DC}(X)$ contains most realistic objective functions. In the rare cases where f is not DC (for example, when f is a discontinous function), one can approximate f by a DC function or use reformulation techniques to get an equivalent DC program.

3.2 Useful DC decompositions and corresponding DCA

To illustrate the way to construct DC decompositions and design the resulting DCA, let us show the two useful DC decompositions of the problem (P) in (9) and discuss on their effectiveness.

Assume that there exists a nonnegative number η (resp. ρ) such that the function $\frac{1}{2}\eta\|x\|^2 + f(x)$ (resp. $\frac{1}{2}\rho\|x\|^2 - f(x)$) is convex. We can now write (P) in the form of DC program (P$_{\text{dc}}$) with, for example, two following DC decompositions:

$$g(x) := \chi_C(x) + \frac{1}{2}\eta\|x\|^2 + f(x); \quad h(x) := \frac{1}{2}\eta\|x\|^2 \tag{10}$$

and

$$g(x) := \chi_C(x) + \frac{1}{2}\rho\|x\|^2; \quad h(x) := \frac{1}{2}\rho\|x\|^2 - f(x). \tag{11}$$

The DCA applied to (P) with decomposition (10) and/or (11) can be described as follows.

```
Algorithm DCAP1 Let $x^0$ be given in $\mathbb{R}^n$. Set $k \leftarrow 0$.
  Repeat
    – Calculate $x^{k+1}$ by solving the convex program
        $\min\left\{\frac{1}{2}\eta\|x\|^2 + f(x) - \langle x, \eta x^k\rangle : x \in C\right\},$   (12)
    – $k \leftarrow k+1$
  Until convergence of $\{x^k\}$.
```

```
Algorithm DCAP2: Let $x^0$ be given in $\mathbb{R}^n$. Set $k \leftarrow 0$.
Repeat
    – Calculate $y^k \in \partial\left(\frac{1}{2}\rho\|.\|^2 - f(.)\right)(x^k)$
```

– Calculate x^{k+1} by solving the convex program

$$\min\left\{\frac{1}{2}\rho\|x\|^2 - \langle x, y^k\rangle : x \in C\right\}, \tag{13}$$

i.e. $x^{k+1} = \mathrm{Proj}_C(y^k/\rho)$.

– $k \leftarrow k+1$

Until convergence of $\{x^k\}$.

Here, Proj_C stands for the orthogonal projection on C.

As indicated above, we are greatly interested in the choice of DC decompositions: what is "the best' among (10) and (11)? The answer depends on C and f. In fact, the performance of the DCA depends upon that of the algorithm for solving convex programs (12) and (13). For certain problems, for example, box constrained quadratic programming and ball constrained quadratic programming, Algorithm DCAP2 is greatly less expensive than Algorithm DCAP1, because the orthogonal projection onto C in these cases is given in explicit form (see for example [35]). In practice, when f is differentiable and the computation of its gradient is not difficult, and the projection on C can be inexpensively determined, the use of DCAP2 is very recommended.

For using the above DC decompositions the crucial question is how to determine a nonnegative number η (resp. ρ) such that the function $\frac{1}{2}\eta\|x\|^2 + f(x)$ (resp. $\frac{1}{2}\rho\|x\|^2 - f(x)$) is convex. In many practical problems such η and ρ exist and can be computed according to the properties of the function f. For example, when f is a *smooth function with Lipschitz continuous gradient* ρ is nothing but the Lipschitz constant of ∇f.

3.3 DCA for network utility maximization

Network utility maximization (NUM) has many applications in network rate allocation algorithms and Internet congestion control protocols. Consider a communication network with L links, each with a fixed capacity of c_l bps, and S sources (i.e., end users), each transmitting at a source rate of x_s bps. Each source s emits one flow, using a fixed set $L(s)$ of links in its path, and has a utility function $U_s(x_s)$. Each link l is shared by a set of sources denoted $S(l)$, (the set of users using link l). NUM, in its basic version, consists of maximizing the total utility of the network $\sum_s U_s(x_s)$ over the source rates x, subject to linear flow constraints for all links l:

$$\begin{aligned}&\text{maximize} && \sum_{s\in\mathcal{S}} U_s(x_s)\\ &\text{s.t} && \sum_{s\in S(l)} x_s \le c_l \quad \forall l,\ x_s \ge 0 \quad \forall s \in \mathcal{S},\end{aligned} \tag{14}$$

where $\mathcal{S}$ denotes the set of users. Here the vector variable is $x = (x_s)_{s\in\mathcal{S}} \in \mathbb{R}^S$ and the constraint set is a well-defined convex polytope. There are many nice properties of the basic NUM model due to several simplifying assumptions of the utility functions and flow constraints, which provide the mathematical tractability of problem (14) but also limit its applicability. In particular, the utility functions U_s are usually assumed to be concave increasing. In such a case, the optimization problem (14) is a convex program, and so far is easy to solve. In the past, maximization of concave utility functions and the resulting distributed rate allocation for elastic traffic have gained extensive attention. Based on the concavity and continuity assumptions on utility functions and the elasticity assumption on application traffic, rigorous mathematical frameworks for standard price-based distributed algorithms have been investigated.

However, it is known that for many multimedia applications, user satisfaction may assume nonconcave shape as a function of the allocated rate. Furthermore, in some other models of utility functions, the concavity assumption on $U_s(x_s)$ is also related to the elasticity assumption on rate demands by users. When demands for x_s are not perfectly elastic, $U_s(x_s)$ may not be concave. In this case, the resulting NUM becomes nonconvex and significantly harder to be analyzed and solved. Since inelastic flows with nonconcave utility functions represent important applications in practice, today solving the NUM problem with nonconcave utility function is a *challenge* of the analysis and design of communication systems by nonconvex optimization techniques.

As an illustrative example, we consider the NUM problem with Sigmoidal-like utility functions [30] that are used in many multimedia applications and Internet congestion control (for example, the utility for voice applications is modeled by a Sigmoidal function with a convex part at low rate and a concave part at high rate). Other useful utility functions can also be solved by DCA-based algorithms (see [24]).

Consider Sigmoidal utilities in a standard form:

$$U_s(x_s) = \frac{1}{1+\mathrm{e}^{-(a_s x_s + b_s)}},$$

where $a_s > 0$, $b_s < 0$ and a_s, b_s are integers. The Sigmoidal function is neither convex nor concave, but it is DC (difference of convex functions). Then the resulting NUM problem is a DC program. We are going to present a DC decomposition for the Sigmoidal function.

We have

$$U_s(x_s) = \mathrm{e}^{(a_s x_s + b_s)} - \frac{\mathrm{e}^{2(a_s x_s + b_s)}}{1+\mathrm{e}^{(a_s x_s + b_s)}}.$$

It is easy to verify that the functions

$$h_s(x_s) := \mathrm{e}^{(a_s x_s + b_s)} \quad \text{and} \quad g_s(x_s) := \frac{\mathrm{e}^{2(a_s x_s + b_s)}}{1+\mathrm{e}^{(a_s x_s + b_s)}}$$

are convex (their derivative is increasing). Therefore, U_s is a DC function, and so is $-U_s$.

Let K be the set defined by the constraints of Problem (14), say

$$K := \left\{ \sum_{s \in S(l)} x_s \le c_l \quad \forall l \in \{1 \dots L\},\ x_s \ge 0 \quad \forall s \in \mathcal{S} \right\}$$

and denote by χ_K the indicator function on K. Then the Sigmoidal NUM problem can be expressed as

$$\begin{aligned} &\max_{x \in K} \left\{ U(x) := \sum_{s \in \mathcal{S}} U_s(x_s) \right\} \\ &= -\min_{x \in K} \left\{ \sum_{s \in \mathcal{S}} \frac{e^{2(a_s x_s + b_s)}}{1 + e^{(a_s x_s + b_s)}} - e^{(a_s x_s + b_s)} \right\} \\ &= -\min \{ g(x) - h(x) : x \in K \} \text{ where } g(x) \\ &:= \sum_{s \in \mathcal{S}} g_s(x_s),\ h(x) := \sum_{s \in \mathcal{S}} h_s(x_s). \end{aligned}$$

Since g_s and h_s are convex functions, the function g and h are convex too (note also that g and h are differentiable). Hence the Sigmoidal NUM problem is a DC program that can be written in the standard form as

$$\min \left\{ [\chi_K(x) + g(x)] - h(x) : x \in \mathbb{R}^{\mathbb{S}} \right\}. \tag{15}$$

According to the general DCA scheme, applying DCA to (15) amounts to computing two sequences $\{y^k\}$ and $\{x^k\}$ in the way that

$$\begin{aligned} &y^k = \nabla h(x^k), \\ &x^{k+1} \in \operatorname{argmin} \left\{ [\chi_K(x) + g(x)] - \langle x, y^k \rangle : x \in \mathbb{R}^{\mathbb{S}} \right\}. \end{aligned}$$

Hence the algorithm can be described as follows.

Algorithm: DCA for Sigmoidal utility maximization:

```
1. Choose $x^0 \in \mathbb{R}^{\mathbb{S}}$ as the initial point. Let $\epsilon > 0$ be sufficiently small, $k \leftarrow 0$.
2. **Repeat**
Set $y^k = (a_1 e^{a_1 x_1^k + b_1}, \dots, a_s e^{a_s x_1^k + b_1})$.
Set $x^{k+1}$ as an optimal solution of the convex program
    $$\min \left\{ \sum_{s \in \mathcal{S}} \frac{e^{2(a_s x_s + b_s)}}{1 + e^{(a_s x_s + b_s)}} - \langle x, y^k \rangle : x \in K \right\} \tag{16}$$
  $k \leftarrow k + 1$
  **until** $||x^{k+1} - x^k|| \le \epsilon(1 + ||x^k||)$.
```

Note that the convex problem (16) can be distributedly implemented using the Lagrangian dual decomposition method as shown in [30].

3.4 Spectrum management problem (SMP)

Discrete multitone (DMT) [42] has been adopted as standard in various DSL applications such as asymmetric DSL (ADSL) and more recently for very-high-bit-rate digital subscriber line (VDSL) by International Telecommunication Union (ITU). For a sufficiently large number of subcarriers, DMT transmission over a frequency-selective fading channel can be modeled as a set of K parallel independent flat-fading sub-carrier AWGN channels (Additive white Gaussian noise). Under this Gaussian assumption, the achievable bit-loading rate of user n on tone k is

$$\begin{aligned} r_k^n &= \log_2 \left(1 + \frac{1}{\Gamma} \frac{|g_k^{n,n}|^2 p_k^n}{\sum_{m \ne n} |g_k^{n,m}|^2 p_k^m + \omega_k^n} \right) \\ &= \log_2 \left(1 + \frac{1}{\Gamma} \frac{p_k^n}{\sum_{m \ne n} h_k^{n,m} p_k^m + \sigma_k^n} \right), \end{aligned}$$

where p_k^n denotes user n's transmit power spectral density (PSD) on tone k; ω_k^n denotes user n's transmit noise power on tone k; $g_k^{n,m}$ is the channel path gain from user m to user n on tone k; $h_k^{n,m} = \frac{|g_k^{n,m}|^2}{|g_k^{n,n}|^2}$ is the normalized interference path power gain from user m to user n on tone k ; $\sigma_k^n = \frac{\omega_k^n}{|g_k^{n,n}|^2}$ is the noise variance of user n on tone k, and Γ is the SNR-gap to capacity. The data rate of user n is $R_n = f_s \sum_{k=1}^{K} r_k^n$, where f_s is the DMT symbol rate.

The goal of spectrum management problem is to achieve best possible user rates tradeoff among users in the network, i.e., to find the boundary of rate region. Assume that each user is subject to an individual total transmission power constraint. One way to define the SMP in the literature is consider the following optimization problem

$$\begin{aligned} \max_{p_1, p_2, \dots, p_K} &\left\{ R_1 : R_n \ge T_n, \quad \forall n \ge 1; \sum_k p_k^n \le P_n, \forall n, k; \right. \\ &\left. p_k^n \le p_k^{n,\text{mask}}, \quad \forall n, k \right\}, \end{aligned} \tag{17}$$

where T_n is minimum target rates of user n and P_n is maximum total transmission power of user n. The SMP (17) aims to maximize the rate of user 1 while guarantees the achievable rates of other users higher than their required minimum target rates T_n. P_n denotes the maximum total transmission power of user n. Spectral mask constraints $p_k^{n,\text{mask}}$ may also be applied if needed.

Among various Dynamic Spectrum Management (DSM) techniques, centralized Optimal Spectrum Balancing (OSB) achieves the maximum data rates by computing the optimal PSDs (power spectral density) for all modems in DSL systems. The centralized algorithm based on dual decomposition for OSB, proposed in [3], decouples joint optimization across all tones to make the problem solvable per-tone basis. If the rate region is convex (the assumption that the rate region is convex is justified in [3] for two-user in DSL system, and the same logic for two-user can be applied to justify the convexity of rate region for multiple-user case), solving the problem (17) amounts to solving the following weighted sum rate optimization problem [5]:

$$\max_{p_1,\ldots,p_K} \left\{ \sum_n \omega_n R_n : \sum_k p_k^n \le P_n \quad \forall n;\ 0 \le p_k^n \le p_k^{n,\text{mask}} \quad \forall k,n \right\}, \tag{18}$$

where the weight for user 1, ω_1, is set to unity, resulting in the maximization of the rate of user 1; whereas $\omega_n \ge 0,\ n \ne 1$ can be adjusted to guarantee the target rate of user n. In [27], we have investigated DC programming and DCA for solving (18).

3.4.1 A nice DC formulation of SMP (18)

First, we write Problem (18) in the form of a minimization program

$$\min_{p=(p_1,\ldots,p_K)} \left\{ f(p) := -\sum_{n=1}^{N} \omega_n R_n : \sum_{k=1}^{K} p_k^n \le P_n\ \forall n;\ 0 \le p_k^n \le p_k^{n,\text{mask}}\ \forall k=1\ldots K, n=1\ldots N \right\}. \tag{19}$$

A natural DC decomposition of f (easily deduced from the definition of r_k^n) has been given in [27]. However, as indicated in [27], from numerical point of views, the DCA scheme corresponding to this DC decomposition is not interesting because it requires an iterative algorithm for solving a convex program at each iteration. In an elegant way we introduced in [27] a nice DC reformulation of the problem (19) (based on the second DC decomposition discussed in Sect. 2) for which the resulting DCA is explicitly determined via a very simple formula. Such a DC decomposition of f is inspired by the following result.

Theorem 1 *There exists $\rho > 0$ such that the function*

$$h(p) := \frac{1}{2}\rho||p||^2 - f(p) \tag{20}$$

is convex on C, the feasible set of (19), say $C := \{p \in \mathbb{R}^{K\times N} \mid \sum_{k=1}^{K} p_k^n \le P_n, 0 \le p_k^n \le p_k^{n,\text{mask}}\ \forall n = 1,2,\ldots,N;\ k = 1,2,\ldots,K\}$.

Proof See [27]. □

Using the theorem above, we get the next DC decomposition of f:

$$g(p) = \frac{1}{2}\rho||p||^2, \quad h(p) = \frac{1}{2}\rho||p||^2 - f(p), \tag{21}$$

and Problem (19) can be now written in the form

$$\min\{\ f(p) := g(p) - h(p) \mid p\ \in C\}$$

or again, in the standard form of DC program:

$$\min\{\chi_C(x) + \ f(p)\ ) \mid p\ \in \mathbf{R}^{N\times K}\}.$$

Then, DCA apply to Problem (19) is described as follows.

DCA-SMP

Initialization: Let $\epsilon > 0$ be given, $\mathbf{p}^{(0)} \in C$ be an initial point, set $r := 0$;

Repeat: set

$$\mathbf{q}^{(r)} = \nabla h(\mathbf{p}^{(r)}) = \rho\ \mathbf{p}^{(r)} - \nabla f(\mathbf{p}^{(r)})$$

and calculate $\mathbf{p}^{(r+1)} \in \partial(g+\chi_C)^*(\mathbf{q}^{(r)})$ by solving the linear constrained quadratic program

$$\min\left\{\frac{1}{2}\rho||p||^2 - \langle p, \mathbf{q}^{(r)}\rangle | p \in C\right\} \tag{22}$$

Set $r+1 \leftarrow r$

until either $||\mathbf{p}^{(r+1)} - \mathbf{p}^{(r)}|| \le \epsilon(||\mathbf{p}^{(r)}||+1)$ or $|f(\mathbf{p}^{(r+1)}) - f(\mathbf{p}^{(r)})| \le \epsilon(|f(\mathbf{p}^{(r)})| + 1)$.

The advantage of the DC decomposition (21) is that the resulting DCA-SMP requires, at each iteration, the computation of the projection of a point on the set C (having very specific structure) for which efficient algorithms are available (see [27]).

4 Nonconvex constraint set

A typical application of this class of problems is *Internet routing* (see for example [5]). Mathematically, the nonconvex constraint set is expressed as

$$E := \{x \in C,\ f_i(x) := g_i(x) - h_i(x) \le 0,\ i = 1,\ldots,m\}$$

where C is a closed nonempty convex set in $\mathbb{R}^n$, $g_i, h_i \in \Gamma_0(\mathbb{R}^n)$, $i = 0,\ldots,m$, and $f_i(x) \le 0$ are called DC constraints. The generic formulation of this class of problems takes the form

$$\alpha = \inf\{f_0(x) := g_0(x) - h_0(x)\ : x \in E \quad (P_{\text{dcg}}) \tag{23}$$

where E is assumed to be nonempty.

This class of nonconvex programs (called general DC programs) is the most general in DC Programming and, a fortiori, more difficult to treat than that of standard DC programs (P_{dc}) because of the nonconvexity of the constraints. It is not new and has been addressed in [34]. Its renewed interests is due to the fact that this class appears, increasingly, in many models of nonconvex variational approaches.

We can solve (P_{dcg}) by DCA via penalty techniques. First, we transform (P_{dcg})-(23) into (P_{dc}) via penalty techniques in DC programming.

Let the functions p and p^+ be defined by

$$p(x) := \max\{f_i(x) : i = 1, \dots, m\};$$
$$I(x) := \{i \in \{1, \dots, m\} : \ f_i(x) = p(x)\}$$
$$p^+(x) := \max\{0, p(x)\},$$

which are DC functions with the following DC decompositions (in case g_i, h_i are finite on C for $i = 1, \dots, m$,(see, e.g., [34]), obtained directly from those of f_i, $i = 1, \dots, m$.

$$p(x) = \max_{i=1,\dots,m} \left\{ g_i(x) + \sum_{j=1, j\neq i}^{m} h_j(x) \right\} - \sum_{j=1}^{m} h_j(x) \tag{24}$$

$$p^+(x) = \max_{i=1,\dots,m} \left\{ \sum_{j=1}^{m} h_j(x), g_i(x) + \sum_{j=1, j\neq i}^{m} h_j(x) \right\} - \sum_{j=1}^{m} h_j(x). \tag{25}$$

The general DC program (P_{dcg})-(23) can then be formulated as

$$\alpha = \inf\{f_0(x) := g_0(x) - h_0(x) \ : \ x \in C, \ p^+(x) \le 0\} \tag{26}$$

and its penalized is a standard DC program

$$\alpha(\tau) = \inf\{\varphi_\tau(x) := f(x) + \ \tau p^+(x) : \ x \in C\}. \ (P_\tau) \tag{27}$$

Let DC decompositions of f_0 and p^+ be given by

$$f_0(x) = g_0(x) - h_0(x); \tag{28}$$
$$p^+(x) = p_1(x) - p_2(x), \tag{29}$$

where g_0, h_0 , p_1, p_2 are convex functions defined on the whole space. Then, we have the following DC decomposition for φ_τ

$$\varphi_\tau(x) = g_\tau(x) - h_\tau(x), \tag{30}$$

where,

$$g_\tau(x) := g_0(x) + \tau p_1(x); \quad h_\tau(x) := h_0(x) + \tau p_2(x). \tag{31}$$

Exact penalty (relative the constraint $p^+(x) \le 0$) for (26) means that there is $\tau_0 \ge 0$ such that for all $\tau > \tau_0$ both DC programs (P_{dcg})-(23) and (P_τ)-(27) are equivalent in the sense that $\alpha(\tau) = \alpha$ and (P_{dcg})-(23) et (P_τ)-(27) have the same (global) solution set. In this case, the solution of (P_{dcg})-(23) can be achieved by applying DCA to a standard DC program (P_τ)-(27) with $\tau > \tau_0$.

Exact penalty techniques in DC programming have been widely investigated in our works [25,28,34]. However, from a computational point of view, an inconvenience of this exact penalty method is that the penalty parameter is generally unknown. Moreover, there are practical optimization problems for which the exact penalization is not satisfied. In [28], we proposed to develop the DCA for solving general DC program (P_{dcg})-(23) by using a penalty technique with updated parameter.

The generalized DCA can be deduced from DCA as follows: instead of fixing the penalty parameter τ, DCA is applied to the sequence of (P_{τ_k}) with an increasing sequence of penalty parameters $\{\tau_k\}$ given by a updating rule from the current iteration x^k such that x^{k+1} is the next iteration of DCA applied to (P_{τ_k}) from x^k. Our work consists in the statement of appropriate updating rules for the sequence $\{\tau_k\}$ and the refinement of constraint qualifications used, in order to ensure global convergence (to a critical point of (P_{dcg})-(23)) and efficiency of DCA1. It is also important that the sequence $\{\tau_k\}$ is constant after a certain rank. The penalty introduced uses l_∞- norm, but we can also consider the l_1 -norm where $q(x) := \sum_{i=1}^{m} f_i^+(x)$ replaces $p^+(x)$.

Some other DCA-based algorithms for (P_{dcg})-(23) have been developed in [36].

5 Integer variables

Several applications in CS can be formulated as an optimization problem with (mixed) integer variables. Here we mention some classes of problems which have been successfully solved by DCA.

5.1 Cross-layer optimization in multi-hop time division multiple access (TDMA) networks

Efficient design of wireless networks is a challenging task due to the interference nature of shared wireless medium. Recently, the concept of cross-layer design has been investigated extensively. In [26,29] a cross-layer optimization framework, i.e., joint rate control, routing, link scheduling and power control for multi-hop TDMA networks, has been considered. Particularly, we studied a centralized controller that coordinates the routing process and transmissions of links such that the network lifetime is maximized [29] and the quality of-service (QoS) constraints on the minimum source rates are satisfied. Alternatively, the energy consumption is an important design criterion for a multi-hop wireless network. In [26], we considered the energy minimization-based cross-layer design problem. We will show below that the aforementioned problems can be formulated as mixed integer-linear programs (MILP) and then efficiently solved by DCA.

In the considered TDMA network, time is partitioned into fixed-length frames, and each frame is further divided into J time slots with unit duration. Since the resource allocation is the same in all frames, we concentrate our design on a single frame. A node may need to transmit in one or more slots for its own traffic and/or relay traffic from other nodes. If a node transmits in a slot, while its transmission power can be varied from $[0, P_{\max}]$, its transmission rate is fixed at a unit rate. In the TDMA-based network, a channel is specified by two elements (j, l), $j \in \mathcal{J}$, $l \in \mathcal{L}$, where $\mathcal{J} = \{1, 2, \ldots, J\}$. For the channel, the resource allocation is denoted by (s_j^l, P_j^l), where $s_j^l = 1$ means link l is active at slot j while $s_j^l = 0$ otherwise, and $P_j^l > 0$ denotes the transmission power of link l at slot j if $s_j^l = 1$, $P_j^l = 0$ otherwise.

At each node, the difference of its outgoing traffic and its incoming traffic should be the traffic generated by itself, i.e.,

$$\sum_{l \in \mathcal{O}(n)} \sum_{j=1}^{J} s_j^l - \sum_{l \in \mathcal{I}(n)} \sum_{j=1}^{J} s_j^l = r_n, \quad n \in \mathcal{N} \tag{32}$$

where $\mathcal{O}(n)$ and $\mathcal{I}(n)$ are the set of outgoing links and incoming links at node n, respectively. The values of s_n for the non-source nodes are set to zero, or equivalently all the traffic entering such nodes must be routed.

The energy consumption at node $n \in \mathcal{N}$ can be written as

$$\mathcal{E}_n = \sum_{l \in \mathcal{O}(n)} \sum_{j=1}^{J} P_j^l + \sum_{l \in \mathcal{O}(n)} \sum_{j=1}^{J} \epsilon_l s_j^l + \sum_{l \in \mathcal{I}(n)} \sum_{j=1}^{J} \varepsilon_l s_j^l, \quad n \in \mathcal{N} \tag{33}$$

where ϵ_l and ε_l denote the energy needed to transmit and receive a unit of traffic over link l, respectively. Note that ϵ_l, ε_l include the energy consumed by the signal processing blocks at the link ends.

Interference Model

Wireless channel is a shared medium and interference-limited where links contend with each other for channel use. Moreover, interference relations among the nodes and/or links can be modeled in various ways, for example, by using the signal-to-interference-plus-noise-ratio (SINR)-based model [32,52]. Specifically, if the link $l \in \mathcal{L}$ is active at slot j (i.e., $s_j^l = 1$), the following inequality should hold so as to guarantee the transmission quality of the link

$$\mathrm{SINR}_j^l = \frac{P_j^l h_{ll}}{\sum_{k \neq l} P_j^k h_{kl} + \eta_l} \geq \gamma^{\mathrm{th}} \tag{34}$$

where SINR_j^l is the SINR for link l at slot j, h_{kl} is the path gain from the transmitter of link k to the receiver of link l, η_l is the noise power at receiver of link l, and γth is the required SINR threshold for accurate information transmission.

We assume that all wireless nodes are low-mobility devices and/or the topology of the network is static or changes slowly allowing enough time for computing the new scheduler. An example of such networks is a wireless sensor network for environmental monitoring with fixed sensor locations. In this case, the need for distributed implementation is not necessary.

From the preceding discussions, the energy minimization-based cross-layer design, i.e., joint rate control, routing, link scheduling, and power allocation problem can be mathematically formulated as

$$\min_{r_n, P_j^l, s_j^l} \quad \sum_{n \in \mathcal{N}} \mathcal{E}_n \tag{35a}$$

subject to:

$$\sum_{l \in \mathcal{O}(n)} \sum_{j=1}^{J} s_j^l - \sum_{l \in \mathcal{I}(n)} \sum_{j=1}^{J} s_j^l = r_n, \; n \in \mathcal{N} \tag{35b}$$

$$r_n \geq r_n^{\min}, \; n \in \mathcal{N} \tag{35c}$$

$$\sum_{l \in \mathcal{I}(\hat{n})} \sum_{j=1}^{J} s_j^l = \sum_{n \in \mathcal{N}} r_n \tag{35d}$$

$$\sum_{l \in \mathcal{O}(n)} s_j^l + \sum_{l \in \mathcal{I}(n)} s_j^l \leq 1, \quad \forall n \in \{\mathcal{N} \cup \hat{n}\}, \; j=1, \ldots, J \tag{35e}$$

$$h_{ll} P_j^l \geq \gamma^{\mathrm{th}} \sum_{k \neq l} P_j^k h_{kl} + \gamma^{\mathrm{th}} \eta_l + D(s_j^l - 1), \quad \forall l \in \mathcal{L}, \; j = 1, \ldots, J \tag{35f}$$

$$0 \leq P_j^l \leq P_{\max} s_j^l, \quad \forall l \in \mathcal{L}, \; j = 1, \ldots, J \tag{35g}$$

$$s_j^l \in \{0, 1\}, \quad \forall l \in \mathcal{L}, \; j = 1, \ldots, J \tag{35h}$$

where $\hat{n}$ denotes the common sink node for all data generated in the network, D is a very large positive constant. The objective function is the energy consumption in the network.[1] Constraints (35b) ensure that the data generated by source nodes are routed properly. Constraints (35c) guarantee that the rate for each node is no less than a minimum rate. The minimum rates are possibly different for nodes and are usually determined by the network QoS. Nodes which do not generate traffic have $r_n = r_n^{\min} = 0$. Constraint (35d) is the flow conservation at the traffic destination for all the sources. Constraints (35e) state that a node can not receive and transmit simultaneously in one particular time slot. Constraints (35f) make sure the SINR requirement is met: if a link l is active in time slot j, then the SINR at receiver of link l must be larger than the given threshold γ^{th} which also depends on the system implementation. Constraint (35f) is automatically satisfied if link l is not scheduled in time slot j. Constraint (35g) states that if a link l is scheduled for time slot j, i.e., $s_j^l = 1$, then the corresponding power value P_j^l must be less than $P_{\max}$. Otherwise, P_j^l obviously equals to zero. We also impose binary integer constraints on s_j^l.

It can be seen that the cross-layer optimization problem (35a)–(35h) belongs to a class of well-known mixed-integer

[1] The traffic sink node consumes a fixed amount of energy within a frame for receiving data.

linear programs (MILPs). The combinatorial nature of the optimization (35a)–(35h) is not surprising and it has been shown in some previous works, albeit with different objective functions and formulations [7,32,52]. Theoretically, MILPs are NP-hard which is clearly inviable for practical scenarios when the dimension is large. It has been shown in [26] that, at optimality, the source rate constraints (35c) must be met with equalities for all sources.

Note that by considering (35a), one aims to minimize the total energy consumption, it may cause some particular nodes spending more energy than the other nodes, and thus, running out of energy quicker. Therefore, equal energy distribution among nodes is not optimal. Another design objective which may help to prevent such situation is as follows

$$\min_{r_n, P_j^l, s_j^l} \quad \max_{n\in\mathcal{N}} \mathcal{E}_n \tag{36a}$$

$$\text{subject to:} \quad \text{The constraints (35b)–(35h).} \tag{36b}$$

The optimization problem (36a)–(36b) aims at minimizing the maximum energy consumed at nodes(s). As a result, more nodes are likely to be involved in the routing algorithm, i.e., relaying information for other nodes. For simplicity, the optimization problem (35a)–(35h) is often considered in the literature.

The cross-layer optimization problem (35a)–(35h) has worst case exponential complexity when BnB methods are used to compute the solution. Moreover, when modeling practical networks and depending on the number of links, nodes and time slots, problem with large sizes may arise. As a result, it is extremely difficult to schedule links optimally. Most research in literature is based on heuristic at the cost of performance degradation, for example, see [7,8,52]. In [26] , we investigated a DCA scheme to solve the mixed 0–1 linear program (35a)–(35h) efficiently.

The network lifetime maximization problem [29] is similar to (35a)–(35h) in which (35a) is replaced by network lifetime maximization.

5.2 Quality of service (QoS) routing problems

The Unicast (resp. Multicast) QoS routing emphasizes to find paths (resp. a set of paths) from a source node to a destination node (resp. a set of destination nodes) satisfying the QoS requirements. The Routing problems become more complex as far as we consider mobile networks or hybrid networks, because of dynamic topology and real time routing procedure. As an example, we consider a scenario in Multicast routing problem, such as we are staying in a car parking place. The mobile services are provided in each moving car, equipped with a mobile device, via a car service center likes in the car parking place. There are m cars sending their requests to a mobile car service center, they need help to find the route to go to their destinations under the travel time constraint, the less latency traffic jam, the jitter time delay constraint, the travel cost (same sources, different destinations, considering local constraints to each mobile vehicle). Therefore, based on the temporary update data of the network state, the mobile car service system has to calculate the route and given the answer for each car in a few seconds. In this context, we need a centralized and efficient algorithm to calculate the routes.

The problem of finding a path in network with multiple constraints (the MCP problem) is NP-complete. We reformulated the MCP [46] and MCOP (multi-constrained optimal path problem) [47,51] problem as Binary Integer Linear Programs (BILP) and investigated DCA-based algorithms for solving them. The DCA is fast and furnished an optimal solution in almost all cases, and a near-optimal solution in the remaining cases. For large scale problems we investigated the proximal decomposition technique to solve convex subprograms at each iteration of DCA. Computational results show that this approach is efficient, especially for large-scale settings where the powerful CPLEX fails to be applicable.

5.3 The partitioning-hub location-routing problem

The Partitioning-Hub Location-Routing Problem (PHLRP) is a hub location problem involving graph partitioning and routing features. PHLRP consists of partitioning a given network into sub-networks, locating at least one hub in each sub-network and routing the traffic within the network at minimum cost. There are various important applications of PHLRP, such as the deployment of network routing protocol problems and the planning of freight distribution problems. In [50] we formulated this problem as an Binary Integer Linear Programming (BILP) and then investigate DCA for solving it. Preliminary numerical results are compared with the well-known commercial solver CPLEX, they show the efficiency and the superiority of DCA.

5.4 The car pooling problem

Car pooling is a well-known transport solution that consists of sharing a car between a driver and passengers sharing the same route, or part of it. The challenge is to minimize both the number of required cars and the additional cost in terms of time for the drivers. To solve the problem, several tasks should be performed: choosing drivers and passengers, allocating passengers to cars, computing an optimal route for the cars. As such, the car pooling transport problem may be described as some kind of fleet management problem. In [49], we formulated this problem as a Mixed Integer Linear Program for which DCA has been efficiently applied. In order to globally solve the problem, we combine DCA with classical Branch and Bound algorithm. DCA is used to calculate upper bound while lower bound is obtained from a linear relaxation problem. Preliminary numerical results are compared with CPLEX. They show the efficiency and the superiority of DCA-based algorithms.

5.5 The minimum m-dominating set problem

Let $G = (V, E)$ be a graph, where V is the set of nodes and E is the set of edges of G. A dominating set D of a graph $G = (V, E)$ is a subset of nodes $D \subseteq V$ such that every vertex not in D is joined to at least one member of D by some edge. The domination number $\gamma(G)$ is the number of vertices in a smallest dominating set for G. The dominating set problem is a classical NP-complete decision problem [10] and has various applications in CS. A classical network application for this problem would be to choose a set of locations to install relay antennas. In ad hoc networks, creating a dominating set is a way to organize the network and is generally used as a first step for generating a connected dominating set [11]. This problem is formulated as a BILP for which DCA is investigated in [39]. Numerical results show that the DCA is efficient even for very large instances of problem. Moreover, our algorithm obtained better solution in significantly less time than CPLEX.

In a general framework, we show below how to solve optimization problems with integer variables by DCA.

5.6 From combinatorial optimization to DC programming: reformulation

5.6.1 Mixed zero-one concave minimization programming problem

Let $D \neq \emptyset$ be a bounded polyhedral convex set in $\mathbb{R}^n$ and let $J \subset \{1, \ldots, n\}$.

$$\min\{f(x) : x \in D,\ x_i \in \{0,1\},\quad \forall i \in J\}, \tag{37}$$

where f is a finite concave function on D.
Let $K := \{x \in D : 0 \le x_i \le 1,\quad \forall i \in J\}$ and define $p(x) = \sum_{i\in J} x_i(1-x_i)$. Clearly, p is a concave function with nonnegative values on K and

$$\{x \in D,\ x_i \in \{0,1\}\} = \{x \in K : p(x) = 0\}$$
$$= \{x \in K : p(x) \le 0\}.$$

Hence (37) can be reformulated as (by Theorem below)

$$\min\{f(x) : x \in K,\ p(x) \le 0\}$$

which is equivalent to, for any $t > t_o$

$$\min\{f(x) + tp(x) : x \in K\}. \tag{38}$$

Theorem 2 *[23] Let K be a nonempty-bounded polyhedral convex set in $\mathbb{R}^n$ and f, p be finite concave on K. Assume the feasible set of (P) be nonempty and p be nonnegative on K. Then there exists $t_o \ge 0$ such that for every $t > t_o$ the following problems have the same solution sets:*

$$(P_t)\quad \alpha(t) = \inf\{f(x) + tp(x) : x \in K\},$$
$$(P)\quad \alpha = \inf\{f(x) : x \in K,\ p(x) \le 0\}.$$

Furthermore

(i) *if the vertex set of K, denoted by $V(K)$, is contained in $\{x \in K : p(x) \le 0\}$, then $t_o = 0$.*
(ii) *if $V(K)$ is not contained in $\{x \in K : p(x) \le 0\}$, then $t_o \le \frac{f(x^o)-\alpha(0)}{S}$ for every $x^o \in K$, $p(x^o) \le 0$, where $S := \min\{p(x) : x \in V(K),\ p(x) > 0\}$.*

DCA for solving (P$_t$)

$$(\mathrm{P}_t)\quad \alpha(t) = \inf_{x\in K}\{f(x) + tp(x)\}$$
$$= \inf_{x\in\mathbb{R}^n}\{F_t(x) := \chi_K(x) + f(x) + tp(x)\}.$$

Assuming that a subgradient of $-f$ is computable. One DC decomposition of F_t can be chosen as

$$F_t(x) := g(x) - h(x) \text{ with } g(x) := \chi_K(x),$$
$$h(x) := -f(x) - tp(x). \tag{39}$$

In this case, (P_t) is a polyhedral DC program because χ_K is a polyhedral convex function, and the general DCA scheme becomes:

$$y^k \in \partial(-f(x^k) - tp(x^k));$$
$$x^{k+1} \in \operatorname{argmin}\left\{-\langle x, y^k\rangle : x \in K\right\}. \tag{40}$$

Besides the computation of subgradients of $-f$ and set $\nabla(-p)(x) = \sum_{i\in J}(2x_i - 1)$, the algorithm requires one linear program at each iteration The convergence properties can be stated as follows:

Theorem 3 i) *DCA generates a finite sequence $x^1, \ldots, x^{k_*}$ contained in V(K) such that $f(x^{k+1}) + tp(x^{k+1}) \le f(x^k) + tp(x^k)$, $p(x^{k+1}) \le p(x^k)$ for each k , and x^{k_*} is a critical point of $g - h$.*
ii) *If, in addition, h is differentiable at x^{k_*}, then x^{k_*} is actually a local minimizer to (P_t).*
iii) *Let $t > t_1 := \max\left\{\frac{f(x)-\alpha(0)}{S} : x \in V(K) \cap \{x \in K : p(x) = 0\}\right\}$. If at an iteration q one has $p(x^q) = 0$, then $p(x^k)$=0 and $f(x^{k+1}) \le f(x^k)\ \forall k \ge q$.*

5.6.2 Extension cases

Based on new results related to exact penalty and error bounds in DC programming [28], the same reformulation technique via exact penalty can be used for

– Linear constrained mixed zero-one DC programming problems.
– Linear constrained mixed integer DC programming problems.

6 Another issue: solving convex programs by DCA

Another issue which is also important in CS but were not discussed in this paper is how to solve large-size convex programs. Although convex programming has been studied for about a century, an increasing amount of effort has been put recently into developing fast and scalable algorithms to deal with large scale problems. While some convex regularizations involve convex quadratic programs (QP) for which standard QP solvers can be certainly used, many first-order methods have been developed in the last years for large scale convex problems. Since DC programming and DCA encompass convex programming and convex programs can be recast as (infinitely many) DC programs to which DCAs become global, (i.e. providing optimal solutions), one can make use of these theoretical and algorithmic tools to better reformulate and solve convex programs.

7 Conclusion

We have presented DC programming and DCA for modeling and solving three challenging classes that cover most nonconvex programs in communication systems. These theoretical and algorithmic tools have been outlined in an appropriate way to make them understandable to the reader. They highlight the distinctive features (flexibility, versatility, inexpensiveness, scalability, efficiency and globality) of DC programming and DCA. It is desirable that our approaches will help researchers and practitioners tackle efficiently their nonconvex programs, especially in the large-scale setting.

References

1. Alvarado, A., Scutari, G., Pang, J.S.: A New Decomposition Method for Multiuser DC-Programming and its Applications, arXiv:1308.3521v2 [cs.IT] (2013). http://arxiv.org/pdf/1308.3521.pdf
2. Al-Shatri, H., Weber, T.: Achieving the maximum sum rate using D.C. programming in cellular networks. IEEE Trans. Signal Process. **60**(3), 1331–1341 (2012)
3. Cendrillon, R., Yu, W., Moonen, M., Verlinden, J., Bostoen, T.: Optimal multiuser spectrum management for digital subscriber lines. IEEE Trans. Comm. **54**, 922–933 (2006)
4. Chiang, M.: Geometric programming for communication systems. Found. Trends Commun. Inf. Theory **2**, 1–154 (2005)
5. Chiang, M.: Nonconvex optimization of communication systems. In: Gao, D., Sherali, H. (eds.) Advances in Mechanics and Mathematics, Special Volume on Strang's 70th Birthday, vol. 3, pp. 136–196. Springer, Berlin (2008)
6. Chiang, M., Hande, P., Lan, T., Tan, C.W.: Power control in wireless cellular networks. Found. Trends Netw. **2**, 1–156 (2008)
7. Commander, C.W., Pardalos, P.M.: A combinatorial algorithm for the TDMA message scheduling problem. Comput. Optim. Appl. **43**(3), 449–463 (2009)
8. ElBatt, T., Ephremides, A.: Joint scheduling and power control for wireless ad hoc networks. In: Proc. IEEE INFOCOM'02, pp. 976–984, New York, USA (2002)
9. Fazel, M., Chiang, M.: Nonconcave network utility maximization through sum of squares method. Proc. IEEE Control and Decision Conference, Seville, Spain (2005)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman Co, San Franciso (1990)
11. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. Algorithmica Number **4**(20), 374–387 (1998)
12. Hande, P., Zhang, S., Chiang, M.: Distributed Rate Allocation for Inelastic Flows. IEEE/ACM Transactions on Networking, vol. 15, No 6 (2007)
13. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Parts I&II. Springer, Berlin (1991)
14. Hong, M., Luo, Z.-Q.: Signal processing and optimal resource allocation for the interference channel. Elsevier e-Reference-Signal Processing (2013). http://arxiv.org/pdf/1206.5144v1.pdf
15. Khabbazibasmenj, A., Roemer, F., Vorobyov, S., Haardt, M.: Sum-rate maximization in two-way AF MIMO relaying: polynomial time solutions to a class of DC programming problems. IEEE Trans. Signal Process. **60**(10), 5478–5493 (2012)
16. Kha, H.H., Tuan, H.D., Nguyen, H.H.: Fast global optimal power allocation in wireless networks by local D.C. programming. IEEE Trans. Wireless Commun. **11**(2), 510–512 (2012)
17. Kim, S.-J., Giannakis, G.B.: Optimal resource allocat ion for MIMO ad hoc cognitive radio networks. IEEE Trans. Inf. Theory **57**(5), 3117–3131 (2011)
18. Kelly, F.P., Maulloo, A., Tan, D.: Rate control for communication networks: shadow prices, proportional fairness and stability. J. Op. Res. Soc. **49**(3), 237–252 (1998)
19. Julian, D., Chiang, M., ONeill, D., Boyd, S.: QoS and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. Proc. IEEE INFOCOM (2002)
20. La, R.J., Anantharam, V.: Utility-based rate control in the Internet for elastic trafic. IEEE/ACM Trans. Netw. **10**(2), 272–286 (2002)
21. Lee, J.W., Mazumdar, R.R., Shroff, N.: Non-convex optimization and rate control for multi-class services in the Internet. Proc. IEEE Infocom, Hong Kong, China (2004)
22. Le Thi, H.A., DC Programming and DCA. http://lita.sciences.univ-metz.fr/~lethi
23. Le Thi, H.A., Pham Dinh, T., Le, D.M.: Exact penalty in DC programming. Vietnam J. Math. **27**(2), 169–178 (1999)
24. Le Thi, H.A.: Network Utility Maximisation: a unified DC programming approach. Technical Report, LITA (2012)
25. Le Thi, H.A., Pham Dinh, T.: The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems. Ann. Oper. Res. **133**, 23–46 (2005)
26. Le Thi, H.A., Nguyen, T.K., Phan, T.K., Pham Dinh, T.: Energy minimization-based cross-layer design in wireless networks. In The

Proceedings of the 2008 High Performance Computing & Simulation Conference (HPCS 2008) Nicosia, Cyprus, June 3–6, pp 283–289 (2008)
27. Le Thi, H.A., Ta, A.S., Pham Dinh, T., Le, N.T.: Optimal Spectrum Balancing in Multi-User DSL Network by DC programming and DCA, Technical report, LITA-UPV-M (2009)
28. Le Thi, H.A.: Pham Dinh, T., Huynh, V.N.: Exact penalty and error bounds in DC programming. J. Global Optim. **52**(3), 509–535 (2012)
29. Le Thi, H.A., Nguyen, Q.T., Phan, T.K., Pham Dinh, T.: DC Programming and DCA Based Cross-layer Optimization in Multi-hop TDMA Networks. Intelligent Information and Database Systems. Lecture Notes in Artificial Intelligence LNCS/LNAI (2013, to appear)
30. Le Thi, H.A., Pham Dinh, T.: Network Utility Maximisation: a DC programming approach for Sigmoidal Utility function. Proceedings of IEEE conference Advance Techonologies for Communications ATC'13 Ho Chi Minh city october 16–18, 2013, 978-1-4799-1089-2/13/31.00. IEEE, pp. 50–54 (2013)
31. Le Thi, H.A., Ta, A.S., Pham Dinh, T.: DC programming and DCA for some resource allocation problems in Communication, Networks (2013, submitted)
32. Madan, R., Cui, S., Lall, S., Goldsmith, A.: Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. IEEE Trans. Wireless Commun. **5**(11), 3142–3152 (2006)
33. Palomar, D.P., Chiang, M.: A tutorial on decomposition methods for network utility maximization. IEEE J. Select. Areas Commun. 24(8) (2006)
34. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to dc programming: theory, algorithms and applications. Acta Math. Vietnam. **22**(1), 289–357 (1997)
35. Pham Dinh, T., Le Thi, H.A.: DC optimization algorithms for solving the trust region subproblem. SIAM J. Optim. **8**, 476–505 (1998)
36. Pham Dinh, T., Le Thi, H.A.: Recent advances on DC programming and DCA. Transactions on Computational Collective Intelligence, Springer, Berlin (2013, to appear)
37. Rockafellar, R.T.: Convex Analysis. Princeton University Press, N.J. (1970)
38. Schleich, J., Bouvry, P., Le Thi, H.A.: Decentralized Fault-tolerant Connected Dominating Set Algorithm for Mobile Ad hoc Networks. Proceedings of the: International Conference on Wireless Networks, World Congress in Computer Science Computer Engineering, and Applied Computing, July 13–16, 2009. Las Vegas, USA, ICWN 2009, 354–360 (2009)
39. Schleich, J., Le Thi, H.A., Bouvry, P.: Solving the Minimum m-Dominating Set problem by a Continuous Optimization Approach based on DC Programming and DCA. J. Combin. Optim. **24**(4), 397–412 (2012)
40. Srikant, R.: The Mathematics of Internet Congestion Control. Birkhauser, Basel (2004)
41. Song, K.B., Cheung, S.T., Ginis, G., Cioffi, J.M.: Dynamic spectrum management for next-generation dsl systems. IEEE Commun. Mag. **40**, 101–109 (2002)
42. Starr, T., Cioffi, J.M., Silverman, P.: Understanding Digital Subscriber Line Technology. Prentice Hall, Upper Saddle River (1999)
43. Schmidt, D., Shi, C., Berry, R., Honig, M., Utschick, W.: Distributed resource allocation schemes: Pricing algorithms for power control and beamformer design in interference networks. IEEE Signal Process. Mag. **26**(5), 53–63 (2009)
44. Scutari, G., Facchinei, F., Song, P., Palomar, D., Pang, J.: Decomposition by partial linearization: Parallel optimization of multi-agent systems, IEEE Trans. Signal Process (2013, submitted). http://arxiv.org/pdf/1302.0756v1.pdf
45. Ta, A.S.: Contributions aux développements des nouvelles technologies de communication en transport multimodal par des techniques d'optimisation, thèse de doctorat soutenue au LMI-INSA de Rouen Juin (2012)
46. Ta, A.S., Le Thi, H.A., Khadraoui, D., Pham Dinh, T.: Solving QoS routing problems by DCA. In Intelligent Information and Database Systems. Lecture Notes in Artificial Intelligence (LNAI), vol. 5991, pp. 460–470. Springer, Berlin (2010)
47. Ta, A.S., Le Thi, H.A., Khadraoui, D., Pham Dinh, T.: Solving Multicast QoS Routing Problem in the context V2I Communication Services using DCA, 9th IEEE/ACIS ICIS: August 18–20, 2010, pp. 471–476. Yamagata, Japan (2010)
48. Ta, A.S., Le Thi, H.A., Pham Dinh, T.: Power control by DC programming and DCA. In: The proceedings of International Conference on Industrial Engineering and Systems Management IESM (2011)
49. Ta, A.S., Le Thi, H.A., Arnould, G., Khadraoui, D., Pham Dinh, T.: Solving car pooling problem using DCA. Proceedings of IEEE conference Global Information Infrastructure Symposium (GIIS 2011), Danang 4–6 (2011) (published by IEEE Xplore)
50. Ta, A.S., Le Thi, H.A., Khadraoui, D., Pham Dinh, T.: Solving Partitioning-Hub Location-Routing Problem using DCA. J. Ind. Manage. Optim. **8**(1), 87–102 (2012)
51. Ta, A.S., Le Thi, H.A., Pham Dinh, T., Khadraoui, D.: Solving many to many multicast QoS routing problem using dca and proximal decomposition technique, In Proc. IEEE International Conference on Computing, Networking and Communications, pages 809–814, Hawaii, American, 30 January-2 February (2012) (published by IEEEXplore)
52. Tang, J., Xue, G., Chandler, C., Zhang, W.: Link scheduling with power control for throughput enhancement in multihop wireless networks. IEEE Trans. Vehicular Tech. **55**(3), 733–742 (2006)
53. Tsiaflakis, P., Diehl, M., Moonen, M.: Distributed spectrum management algorithms for multiuser dsl networks **56**(10), 4825–4843 (2008)
54. Vucic, N., Shi, S., Schubert, M.: DC programming approach for resource allocation in wireless networks, pp. 380–386. In International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) (2010)