DC Programming and DCA for Nonconvex Optimization: Theory, Algorithms and Applications

Le Thi Hoai An\textsuperscript{1} and Pham Dinh Tao\textsuperscript{2}

\textsuperscript{1}Laboratory of Theoretical and Applied Computer Science
UFR MIM Paul Verlaine University Metz, Ile de Saulcy, 57045 Metz, France
e-mail: lethi@univ-metz.fr

\textsuperscript{2}Laboratory of Modelling, Optimization & Operations Research
INSF-ROUEN, Place Emile Blondel, BP 8, 76131 Mont Saint Aignan cedex, France
e-mail: pham@insa-rouen.fr

Keywords: DC programming, DCA.

Abstract. We present in this work an innovative approach in nonconvex optimization called DC (Difference of Convex functions) programming and its applications in various fiels of sciences, in particular in biomathematic and bioinformatic.

DC programming and DCA (DC Algorithms) introduced by Pham Dinh Tao in 1985 and extensively developed in our joint works since 1994 is an efficient approach for nonconvex continuous optimization. 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, especially in the large scale setting.

1 Introduction

In recent years there has been a very active research in nonconvex programming, because most real life optimization problems are nonconvex. DC programming constitutes the backbone of smooth/nonsmooth nonconvex programming and global optimization. There are two different but complementary approaches (we can say two schools) in DC programming. A great deal of work involves global optimization (which is concerned with finding global solutions to nonconvex programs) whose main tools and solution methods are developed according to the spirit of the combinatorial optimization, but with the difference that one works in the continuous framework. The second approach, called DCA, is founded on convex analysis tools. DCA is a continuous approach based on local optimality and the duality in DC programming for solving DC programs. DCA has been introduced by Pham Dinh Tao in 1985 as an extension of his subgradient algorithms (for convex maximization programming) to DC programming. Crucial developments and improvements for DCA from both theoretical and computational aspects have been completed since 1993 throughout our joint works.
Generally speaking, DCA aims to solve a DC program that takes the form:

$$\beta_p = \inf \{F(x) := g(x) - h(x) : x \in \mathbb{R}^n\} \quad (P_{dc})$$

where $g, h$ are lower semicontinuous proper convex functions on $\mathbb{R}^n$. Such a function $F$ is called DC function, and $g - h$, DC decomposition of $F$ while $g$ and $h$ are DC components of $F$.

A constrained DC program whose feasible set $C$ is convex can always be transformed into an unconstrained DC program by adding the indicator function of $C$.

Let $g^*(y) := \sup \{\langle x, y \rangle - g(x) : x \in \mathbb{R}^n\}$ be the conjugate function of $g$. Then, the following program is called the dual program of $(P_{dc})$:

$$(D_{dc}) \quad \beta_d = \inf \{h^*(y) - g^*(y) : y \in \mathbb{R}^n\}.$$  

Under the natural convention in DC programming that is $+\infty - (+\infty) = +\infty$, it can be proved that $\beta_p = \beta_d$. There is a perfect symmetry between primal and dual DC programs: the dual of $(D_{dc})$ is $(P_{dc})$.

## 2 How DCA works?

Based on local optimality conditions and duality in DC programming, the DCA consists in the construction of two sequences $\{x^k\}$ and $\{y^k\}$, candidates to be optimal solutions of primal and dual programs respectively, such that the sequences $\{g(x^k) - h(x^k)\}$ and $\{h^*(y^k) - g^*(y^k)\}$ are decreasing and $\{x^k\}$ (resp. $\{y^k\}$) converges to a primal feasible solution $x^*$ (resp. a dual feasible solution $y^*$) verifying local optimality conditions and

$$x^* \in \partial g^*(y^*); y^* \in \partial g^*(x^*).$$

These two sequences $\{x^k\}$ and $\{y^k\}$ are determined in the way that $x^{k+1}$ (resp. $y^{k+1}$) is a solution to the convex program $(P_k)$ (resp. $(D_{k+1})$) defined by

$$\inf \{g(x) - h(x^k) - \langle x - x^k, y^k \rangle : x \in \mathbb{R}^n\}, \quad (P_k)$$

$$\inf \{h^*(y) - g^*(y^k) - \langle y - y^k, x^{k+1} \rangle : y \in \mathbb{R}^n\}, \quad (D_{k+1}).$$

The first interpretation of DCA is simple: at each iteration one replaces in the primal DC program $(P_{dc})$ the second component $h$ by its affine minorization $h_k(x) := h(x^k) + \langle x - x^k, y^k \rangle$ at a neighbourhood of $x^k$ to give birth to the convex program $(P_k)$ whose the solution set is nothing but $\partial g(y^k)$. Likewise, the second DC component $g$ of the dual DC program $(D_{dc})$ is replaced by its affine minorization $(g^*)_k(y) := g^*(y^k) + \langle y - y^k, x^{k+1} \rangle$ at a neighbourhood of $y^k$ to obtain the convex program $(D_k)$ whose $\partial h(x^{k+1})$ is the solution set. DCA performs so a double linearization with the help of the subgradients of $h$ and $g$ and the DCA then yields the next scheme:

$$y^k \in \partial h(x^k); \quad x^{k+1} \in \partial g^*(y^k).$$  

## 3 Why DCA?

- Global optimization approaches such as Branch and Bound, Cutting plan for DC programs do not work in large-scale DC programs that we are often faced in real world problems!
- DCA is a robust and efficient method for nonsmooth nonconvex programming which allow to solve large-scale DC programs.
• DCA is simple to use and easy to implement.
• 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, especially in the large scale setting.

It is worth noting that for suitable DC decompositions, DCA generates almost standard algorithms in convex and nonconvex programming.

4 Key properties of DCA

• DCA is constructed from DC components and their conjugates but not from the DC function itself.
• A DC function has infinitely many DC decompositions and there are as many DCA as there are DC decompositions.
• DCA is a descent method without linesearch which has a linear convergence for general DC programs.
• DCA has a finite convergence for polyhedral DC programs.

5 Important questions in the development of DCA

• How to find a ”good” DC decomposition ?
• How to find a ”good” starting point ?

From the theoretical point of view, the question of ”good” DC decompositions is still open. Of course, this depends strongly on the very specific structure of the problem being considered. In order to tackle the large scale setting, one tries in practice to choose \( g \) and \( h \) such that sequences \( \{x^k\} \) and \( \{y^k\} \) can be easily calculated, i.e. either they are in explicit form or their computations are inexpensive. As for the degree of dependence on initial points: the larger the set (made up of starting points which ensure convergence of the algorithm to a global solution) is, the better the algorithm will be.

6 Globalizing DCA

To guarantee globality of sought solutions or to improve their quality it is advised to combine DCA with global optimization techniques, the most popular of which are Branch-and-Bound, SDP and cutting plane techniques..., in a deeper and efficient way. Note that for a DC function \( f = g - h \), a good convex minorization of \( f \) can be taken as \( g + \text{co}(-h) \) where \( \text{co} \) stands for convex envelope. Knowing that the convex envelope of a concave function on a bounded polyhedral convex set can be easily computed.

7 Applications

DCA was successfully applied to a lot of different and various nonconvex optimization problems (see references in http://lita.sciences.univ-metz.fr/lethi/DCA.html).

Hard problems in nonconvex programming and combinatorial optimization
1. The Trust Region subproblems
2. Nonconvex Quadratic Programs
3. Quadratic Zero-One Programming problems / Mixed Zero-One Programming problems
4. Minimizing a quadratic function under convex quadratic constraints
5. Multiple Criteria Optimization: Optimization over the Efficient and Weakly Efficient Sets
6. Linear Complementarity Problem
7. Nonlinear Bilevel Programming Problems

**Economie and Finance**
9. Portfolio optimization under DC transaction costs and minimal transaction unit constraints
10. Portfolio selection problem under buy-in threshold constraints
11. Downside risk portfolio selection problem under cardinality constraint
12. Worst-case Robust Investment Strategies
13. Collusive game solutions via optimization
14. The Value-At-Risk.

**Transport-Logistic, Supply chain, Management**
15. The strategic supply chain design problem from qualified partner set
16. The concave cost supply chain problem
17. Strategic Capacity Planning in Supply Chain Design for a New Market Opportunity
18. The Multimodal transport problem
19. Container Allocation on trains in a rapid transhipment shunting yard

**Telecommunication**
20. Nonconvex Multicommodity Network Optimization problems
21. Capacity and flow assignment problems
22. Cross-layer Design in Wireless Networks
23. Optimal Spectrum Balancing in Multi-user DSL Networks, ...

**Machine Learning and Data mining**
24. Multidimensional Scaling Problems (MDS)
25. Hard Clustering
26. Fuzzy Clustering
27. Multilevel hierarchique clustering and its application to multicast structures
28. Support Vector Machine
29. Large margin classification to psi-learning
30. Multicategory psi-learning
31. Transductive support vector machines
32. Marge-margin semi-supervised learning
33. Combined feature selection and classification.

**Biology, Bioinformatic**
34. Molecular Optimization via the Distance Geometry Problems
35. Molecular Optimization by minimizing the Lennard-Jones Potential Energy
36. Minimizing Morse potential energy
37. Multiple alignment of sequences
38. Phylogenetic analysis

**Image analysis**
40. Images restoration
41. Image segmentation
42. Discrete tomography

**Security, Safety**
43. Generating Bent Functions in Cryptography
44. Generating highly nonlinear balanced Boolean Functions in Cryptography

*Authentification, identification:* 45. Perceptron Problem, Permutation Perceptron Problem