Nonlinear Dynamics and Systems Theory, **24** (4) (2024) 410–418

An Efficient DCA Algorithm for Solving Non-Monotone Affine Variational Inequality Problem

A. Noui [1*], Z. Kebaili [2] and M. Achache [2]

[1] *Geology Department, Setif 1 University, Setif 19000, Algeria.*
[2] *Fundamental and Numerical Mathematics Laboratory, Setif 1 University, Setif 19000, Algeria.*

Received: November 20, 2023;   Revised: July 5, 2024

Abstract: In this paper, we propose a method for solving a non monotone affine variational inequality problem (AVI). We consider an equivalent optimization model, which is formulated as a DC program, and we apply DCA for solving it. The process consists of solving a successive convex quadratic program. The effeciency of the proposed approach is illustrated by the numerical experiments on several test problems in terms of the quality of the obtained solutions and their convergence.

Keywords: *affine variational inequality problem; quadratic program; DC programming; DCA (Difference of Convex functions Algorithms).*

Mathematics Subject Classification (2010): 90C20, 90C30, 90C90, 49J40.

1 Introduction

The variational inequality problem (VIP) remains a prominent and highly sought-after research focus within the realm of numerical optimization. Our objective is to develop a more comprehensive and less restrictive theoretical framework while devising appropriate algorithms to address it. Both the affine variational inequality problem (AVI) and the standard linear complementarity problem bear a close connection to the Karush-Kuhn-Tucker conditions commonly encountered in quadratic programming. Let us define some essential notations: We denote by $\mathbb{R}^n$ and $\mathbb{R}^m$ the finite-dimensional Euclidean spaces, and $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{m\times n}$ represent, respectively, the spaces of $(n\times n)$-matrices and $(m\times n)$-matrices. The following constitutes an affine variational inequality problem (AVI):

$$\begin{cases} \text{Find } x \in C \\ \text{such that } \langle Mx+q, y-x\rangle \geq 0, \text{ for all } y \in C. \end{cases} \tag{1}$$

* Corresponding author: mailto:a.noui@univ-setif.dz

Here $(M, A, q, b) \in D = \mathbb{R}^{n\times n} \times \mathbb{R}^{m\times n} \times \mathbb{R}^n \times \mathbb{R}^m$ represents the data of the problem, the convex polyhedron $C = \{x \in \mathbb{R}^n : Ax \geq b\}$ is the constraint set of the problem and $\langle x, y\rangle = x^T y$ denotes the usual scalar product in $\mathbb{R}^n$. If $C = \mathbb{R}^n_+$, then the AVI becomes the linear complementarity problem

$$x^T (Mx + q) = 0, Mx + q \geq 0, x \geq 0. \tag{2}$$

This problem is often denoted as the Linear Complementarity Problem (LCP). Over time, a multitude of methods have been put forth to address the affine variational inequality problem (AVI), as documented in references [2, 4, 6, 7, 10, 13]. While the AVI is not inherently framed as an optimization problem, the literature contains several optimization models and methodologies designed to tackle it. Broadly speaking, the resultant optimization problem of the AVI is non-convex, rendering it extremely challenging to solve using global approaches, particularly in large-scale settings. In this paper, we delve into a novel and highly effective local optimization strategy for addressing the affine variational inequality problem (AVI). This approach leverages DC programming (Difference of Convex functions programming) and DCA (DC Algorithms). The foundations of DC programming and DCA were initially laid by Pham Dinh Tao in a preliminary form in 1985 and have since undergone extensive development, particularly in the works of Tao Pham Dinh, as documented in references [14, 15, 18–20], and in related works cited therein. These methods have now become classics and are widely adopted by many researchers, as evidenced by references [10, 11, 17, 21, 22]. The motivation behind our work stems from the successful application of DCA (DC Algorithms) to numerous large-scale non-convex programs, both smooth and non-smooth, across various domains in applied sciences. The DCA has proven itself to be a reliable approach, often yielding global solutions and demonstrating superior robustness and efficiency compared to standard methods.

In our study, we adopt an optimization perspective for the affine variational inequality problem (AVI). We demonstrate that this optimization problem can be cast as a DC (Difference of Convex functions) program, making it amenable to DCA.

A DC program involves the minimization of a DC function, typically expressed as $f = g - h$, over a convex set. Here, g and h are convex functions that constitute the DC components. It is noteworthy that the construction of DCA revolves around the convex DC components, namely g and h, "rather than the DC function f" itself.

In this work, we propose a suitable DC formulation for optimizing the AVI and investigate the implications of this DC decomposition in terms of the mentioned properties. To evaluate the efficiency of both the optimization model and the corresponding DC formulation, as well as the resulting DCA schemes, we conduct tests using several benchmark datasets.

The paper is structured as follows.

Section 2: This section delves into the relationship between a quadratic program and the affine variational inequality (AVI).

Section 3: This section provides a brief introduction to DC programming and DCA, offering insights into the solution methods for addressing the optimization problem associated with the AVI using DC programming and DCA.

Section 4: In this section, we present the numerical results obtained from experiments conducted on several test problems, providing insights into the practical performance of the proposed methods.

2 Optimization and AVI

The relationship between optimization and the Affine Variational Inequality (AVI) lies in the fact that the AVI can be formulated as an optimization problem, while the AVI is not explicitly an optimization problem, it can often be transformed into a quadratic optimization problem when the matrix M is symmetric (see [1,5,6,9]), and this opens the door for applying various optimization techniques to find solutions efficiently. Researchers leverage optimization principles and methodologies to tackle the AVI and obtain solutions that meet certain criteria or objectives. The AVI problem is equivalent to the first-order conditions of the following quadratic program:

$$\min_{x\in C} f(x) = \frac{1}{2}x^T Mx + q^T x. \tag{3}$$

3 Overview of DC Programs and DCA

DC programming and DCA indeed serve as fundamental tools in both smooth and non-smooth non-convex programming, including global optimization. These methods are particularly valuable when dealing with optimization problems where the objective function is expressed as the difference between two convex functions, either over the entire space $\mathbb{R}^n$ or within a specified set $C \subset \mathbb{R}^n$. In a general sense, a DC program can be represented by the following mathematical form:

$$\alpha = \min_{x\in\mathbb{R}^n} \left(f(x) = g(x) - h(x)\right), \tag{4}$$

where both $g(x)$ and $h(x)$ are convex functions. The necessary local optimality condition for the primal DC program (4) is

$$\partial h(x^*) \subset \partial g(x^*), \tag{5}$$

where $\partial h(x)$ and $\partial g(x)$ denote the sub-differential of $h(x)$ and $g(x)$ at the point x.

A point x^* satisfies the generalized Kuhn-Tucker condition (a critical point of $g - h$) if

$$\partial g(x^*) \cap \partial h(x^*) = \emptyset. \tag{6}$$

The DCA (DC Algorithm) is the algorithmic framework employed to address these DC programming problems. It iteratively approximates the non-convex objective function using convex components, optimizes the convexified subproblems, and updates the solution until convergence is achieved. The DCA has demonstrated its utility in handling challenging non-convex optimization problems, making it a valuable tool in various fields of applied mathematics and optimization. Based on optimality conditions and duality in the DCA, the idea of the DCA is quite simple: each iteration k of the DCA approximates the concave part $-h$ by its affine majorization, which corresponds to taking

$$y^k \in \partial h(x^k),$$

and minimizes the resulting convex function, which is equivalent to determining the unique solution of the problem (4):

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

with g^* being the conjugate function of g. The generic form of a DC algorithm is stated as follows.

Algorithm 3.1 ***Initialization:*** *Let $x \in \mathbb{R}^n$ be the best guess, $k \longrightarrow 0$.*
Repeat.
Calculate $y^k \in \partial h(x^k)$.
Calculate $x^{k+1} \in \arg\min(g(x) - h(x^k) - \langle x - x^k, y^k \rangle : x \in \mathbb{R}^n)$.
$k \longrightarrow k+1$
Until *convergence of $\{x^k\}$.*

The convergence characteristics of the DCA and its underlying theoretical foundation can be found in the following cited sources [14,16,19,20].

3.1 DCA for addressing AVI

In the context of a general non-convex quadratic program, various DC formulations have been introduced in [14,16]. In this paper, we employ the following DC decomposition, which appears to be the most intuitive: $f(x) = g(x) - h(x)$, where

$$\begin{cases} g(x) = \varkappa_C(x) + \frac{\rho}{2}\|x\|^2 + \langle q, x\rangle, \\ h(x) = \frac{\rho}{2}\|x\|^2 - \frac{1}{2}\langle Mx, x\rangle, \end{cases} \quad \text{and } \rho \geq \lambda_{\max}.$$

In this context, $\lambda_{\max}$ represents the largest eigenvalue of the matrix M, while $\varkappa_C(.)$ represents the indicator function associated with the set C. It is evident that both g and h are convex functions, rendering problem (1) a DC program in the standard form

$$\min_{x\in\mathbb{R}^n} \{g(x) - h(x)\}. \tag{7}$$

Following the generic DCA scheme, along with its properties and theoretical foundation detailed in [9,12,15,16], at each step $k \geq 0$, the computation of y^k is performed as

$$y^k = \left(\nabla h\left(x^k\right)\right)^T = (\rho I - M)\, x^k \tag{8}$$

and subsequently, the unique solution denoted as x^{k+1} is determined for the convex minimization problem

$$\min_{x\in\mathbb{R}^n} \left\{g(x) - \left[h\left(x^k\right) + \left\langle x - x^k, y^k\right\rangle\right]\right\}.$$

So, at every step $k \geq 0$, the point is computed as follows:

$$x^{k+1} = P_c\left(x^k - \frac{1}{\rho}\left(Mx^k - q\right)\right).$$

This is the unique solution corresponding to the problem (7). The latter can be equivalently expressed as

$$\min\left\{\left\|x - \frac{1}{\rho}\left(y^k - q\right)\right\|^2,\ Ax \geq b\right\}$$

with $y^k = (\rho I - M)\, x^k$.

The primary operation at each iteration of the algorithm involves solving a quadratic program. The application of the DCA to (7) can be outlined as follows.

Algorithm 3.2 ***Step 0***. *Let $\epsilon > 0$ be a sufficiently small positive number,*
Let $x^0 \in C$ be a starting point, set $k = 0$,
For $k = 0, 1, \ldots$
Step 1. *Compute $y^k = (\rho I - M)\, x^k$,*
Step 2. *Compute x^{k+1} as an optimal solution of the following optimization program:*
$$\min \left\{ \left\| x - \frac{1}{\rho}\left(y^k - q\right) \right\|^2,\ Ax \geq b \right\}$$
If *either $\left\| x^{k+1} - x^k \right\| \leq \varepsilon$ or $\| f(x^{k+1}) - f(x^k) \| \leq \varepsilon$* , ***then*** *stop,*
otherwise, *set $k = k + 1$ and go to* ***Step 1***.

The main operation at each iteration of the algorithm consists of solving one quadratic program.

3.2 Convergence of the algorithm

Theorem 3.1 *[14]*
1) The DCA generates the sequence $\{x_k\}$ in C such that the sequence $\{f(x_k)\}$ is decreasing.
2) If the optimal value of problem (7) is finite, then the sequence $\{x_k\}$ converges to x^* satisfying the necessary local optimality condition $\partial h\,(x^*) \subset \partial g\,(x^*)$.

4 Computational Results

To provide a better understanding of our algorithm's performance, we implemented it in Matlab and applied it to a set of examples that have been previously studied in the literature. We designate the initial point as x^0. These examples were tested with various values of ρ and x^0. In our implementation, we set the tolerance parameter to $\epsilon = 10^{-6}$.

Example 4.1 [12] Consider the following AVI problem, where its data is given by
$$M = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}, \ q = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \ b = \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix}.$$
The optimal solution set of Example 4.1 is $Sol(M, A, q, b) = \left\{ (0, 2)^T,\ (1, 0)^T \right\}$.

In the implementation, we take $\rho_{\max} = 1$ and we use two starting points such as $x_1^0 = (0.5, 0.5)^T$ and $x_2^0 = (-10, -10)^T$. The numerical results obtained by the algorithm are summarized in Table 1.

Example 4.2 [12] The data of the AVIs is given by
$$M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 0 \end{bmatrix}, \ q = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \ b = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}.$$
The optimal solution set is given by $Sol(M, A, q, b) = \left\{ (2, -1)^T,\ (2, 1)^T, (2, 0)^T \right\}$. We take $\rho_{\max} = 1$ and the initial points $x_1^0 = (4, 0)^T$ and $x_2^0 = (-0.5, -10)^T$. The obtained numerical results are then summarized in Table 2.

Example 4.3 [18] The data of the AVIs is deduced from a non-convex quadratic program, where

	x_1^0		x_2^0	
ρ	4 iter	CPU (s)	iter	CPU (s)
-1	7	0.0389	8	0.0422
0.5	3	0.0155	12	0.0630
1	3	0.0178	13	0.0701
1.5	4	0.0289	16	0.0773
2	4	0.0184	19	0.0724
4	6	0.0525	30	0.2047
10	12	0.0853	60	0.3669
100	103	0.5227	476	3.5156

Table 1: Example 4.1.

	x_1^0		x_2^0	
ρ	4 iter	CPU (s)	iter	CPU (s)
0.01	2	0.0224	3	0.0278
0.1	2	0.0246	3	0.0583
0.5	2	0.0095	3	0.01992
1	2	0.0093	5	0.0390
2	3	0.1899	8	0.0457
10	12	0.1395	26	0.1926
100	111	0.6902	210	0.7789

Table 2: Example 4.2.

$$M = \begin{pmatrix} 263 & -97 & 62 & 217 & 52 & 621 & 935 & 258 & -61 & -10 \\ -97 & 299 & -17 & 9 & 4 & -123 & -17 & -40 & -3 & 37 \\ 62 & -17 & 178 & 71 & -118 & -83 & -110 & 9 & -56 & 42 \\ 217 & 9 & 71 & 143 & -5 & 842 & 228 & 42 & 58 & -41 \\ 52 & -4 & -118 & -5 & 177 & 102 & -15 & 120 & 13 & -52 \\ 621 & -123 & -83 & 842 & 102 & 219 & 574 & 22 & 73 & -53 \\ 935 & -17 & -110 & 228 & -15 & 574 & 457 & 154 & -25 & 84 \\ 258 & -40 & 9 & 42 & 120 & 22 & 154 & 473 & 18 & -29 \\ -61 & -3 & -56 & 58 & 13 & 73 & -25 & 18 & -4 & -79 \\ -10 & 37 & 42 & -41 & -52 & -53 & 84 & -29 & -79 & 224 \end{pmatrix},$$

$$A = \begin{pmatrix} I \\ -I \end{pmatrix}, \; b = (0,0,0,0,0,-1,-1,-1,-1,-1)^T,$$

and

$$q = (-20, -314, 46, -83.45, -128.7, 41.3, 43.85, 341.8, 34.05, -34.6)^T.$$

The optimal solution set of Example 3 is given by
$Sol(M, A, q, b) = \left\{(0, 1, 0.5, 0, 0.75, 0, 0, 0.6, 1, 0.49)^T\right\}$.
Here $\rho_{\max} = 2081.7$ and the initial starting points are given by
$x_1^0 = (-1, \dots, -1)^T$ and $x_2^0 = (0.5, \dots, 0.5)^T$. The obtained numerical results for this problem are summarized in Table 3.

	x_1^0		x_2^0	
ρ	4 iter	CPU (s)	iter	CPU (s)
500	40	0.02951	97	0.06440
1000	81	0.2974	262	0.0308
2000	4	0.0646	3	0.0545
2081.7	4	0.0652	3	0.0581
2500	7	0.1137	4	0.0711
3000	7	0.1178	7	0.1216

Table 3: Example 4.3.

Example 4.4 [8] (Variable Size Example). Consider the following quadratic program:

$$\begin{cases} \min\left[-\sum_{i=1}^{n} x_i^2\right] \\ \text{Such that} \\ \sum_{i=1}^{n} x \geq j,\ j = 1,\ 2, ..., n, \\ x_i \geq 0,\ i = 1,\ 2, ..., n. \end{cases}$$

The data of the corresponding variational problem are

$$M = \begin{pmatrix} -2 & 0 & 0 & . & . & . & 0 \\ 0 & -2 & 0 & . & . & . & 0 \\ 0 & 0 & -2 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & -2 \end{pmatrix},\ q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix},$$

$$A = \begin{pmatrix} -1 & 0 & 0 & . & . & . & 0 \\ -1 & -2 & 0 & . & . & . & 0 \\ -1 & -2 & -3 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ -1 & -2 & -3 & . & . & . & n \\ 1 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & 1 \end{pmatrix},\ b = \begin{pmatrix} -1 \\ -2 \\ -3 \\ . \\ . \\ . \\ -n \\ 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}.$$

The results of this example are quoted in the following table for various values of the dimension n.

n	iteration number	CPU (s)
5	2	0.01026
10	2	0.0292
50	2	0.0751
100	2	0.1048
300	2	0.4333
700	2	2.2656
1000	2	6.1662
1500	2	19.3652
3000	2	207.7703

Table 4: Example 4.4.

5 Conclusion

Our investigation is centered around a nonconvex programming approach that relies on DC programming and DCA algorithms to tackle a non-monotone Affine Variational Inequality problem (AVIP). To address the AVIP, we formulated an optimization model and leveraged DCA algorithms for its resolution. Employing a suitable decomposition for this model, we devised a straightforward DCA algorithmic scheme, involving the successive solution of convex quadratic programs. Our numerical experiments, conducted on a variety of test problems, provided compelling evidence for the efficiency and effectiveness of the proposed approach. We successfully applied our approach to solve the model presented above, and it has demonstrated its capability to be employed in addressing large-scale mathematical problems.

References

[1] M. Achache. A new primal-dual path-following method for convex quadratic programming. *Computational & Applied Mathematics* **25** (2006) 97–110.

[2] M. Achache. Complexity analysis and numerical implementation of a short-step primal-dual algorithm for linear complementarity problems. *Applied Mathematics and computation* **216** (2010) 1889–1895.

[3] N. Anane, Z. Kebaili and M. Achache. A DC Algorithm for Solving non-Uniquely Solvable Absolute Value Equations. *Nonlinear Dynamics and Systems Theory* **23** (2) (2023) 119–128.

[4] Y. Censor, A.N. Iusem and A.S. Zenios. An interior-point method with Bregman functions for the variational inequality problem with paramonotone operators. *Mathematical Programming* **81** (1998) 373–400.

[5] D. den Hertog, B. Roos and T. Terlaky. A Polynomial Method of Weighted Centers for Convex Quadratic Programming. *Journal of Information and Optimization Sciences* **12** (1991) 187–205.

[6] F. Facchinei and J.S. Pang. Finite-dimensional Variational Inequalities and Complementarity Problems, vol. I, II. In: Springer Series in Operations Research. Springer-Verlag, New York, 2003.

[7] F. Fukushima. Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical Programming* **53** (3) (1992) 99–110.

[8] Li Ge and Sanyang Liu. An accelerating algorithm for globally solving non convex quadratic programming. *Journal of Inequalities and Applications* **178** (2018).

[9] P.T. Harker and B. Xiao. A polynomial-time algorithm for affine variational inequalities. *Applied Mathematical Letters.* **3** (4) (1991) 31–34.

[10] Z. Kebaili and M. Achache. On solving non monotone affine variational inequalities problem by DC programmingand DCA. *Asian-European Journal of Mathematics* **1** (1) (2010) 1–8.

[11] N. Krause and Y. Singer. Leveraging the margin more carefully In: Proceedings of the 21st International Conference on Machine Learning, ICML, 2004, Banff, Alberta, Canada, (2004) p. 63. ISBN:1-58113-828-5.

[12] G.M. Lee and N.N. Tam. Continuity of the Solution Map in Parametric Affine Variational Inequalities. *Set-Valued Analysis* **15** (2007) 105–123.

[13] Gui-Hua Lin and Zun-Quan Xia. Some improved convergence results for variational inequality problems. *Journal of Interdisciplinary Mathematics* **2** (1) (1999) 81–88.

[14] H.A. Le Thi and T. Pham Dinh. On solving linear complementarity problems by DC programming and DCA. *Computational Optimization and Applications* **50** (3) (2011) 507–524.

[15] H.A. Le Thi and T. Pham Dinh. Solving a Class of Linearly Constrained Indefinite Quadratic Problems by D.C. Algorithms. *Journal of global optimization* **11** (3) (1997) 253–285.

[16] H.A. Le Thi and T. Pham Dinh. The DC (Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems. *Annals of Operations Research* **133** (3) (2005) 23–46.

[17] Y. Liu, X. Shen and H. Doss. Multicategory ψ-learning and support vector machine: Computational tools. *Journal of Computational and Graphical Statistics* **14** (2005) 219–236.

[18] A. Malek and N. Hosseinipour-Mahani. Solving a class of non-convex quadratic problems based on generalized KKT conditions and neurodynamic optimization technique. *Kybernetika* **51** (5) (2015) 890–908.

[19] T. Pham Dinh and H.A. Le Thi. Convex analysis approach to dc programming. Theory, algorithms and applications. *Acta Math. Vietnam* **22** (1) (1997) 289–355.

[20] T. Pham Dinh and H.A. Le Thi. A DC optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization* **8** (2) (1998) 476–505.

[21] C. Ronan, S. Fabian, W. Jason and B. Léon. Trading convexity for scalability. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, Pittsburgh, Pennsylvania, 2006. 201–208. ISBN:1-59593-383-2.

[22] T. Schüle, C. Schnörr, S. Weber and J. Hornegger. Discrete tomography by convex-concave regularization and D.C. programming. *Discrete Applied Mathematics* **151** (2005) 151–229.