**RESEARCH**　　　　　　　　　　　　　　　　　　　　　**Open Access**

# A two-level domain decomposition algorithm for linear complementarity problem

Shuilian Xie[1*], Zhe Sun[2] and Yuping Zeng[1]

*Correspondence:
shuilian6319@163.com
[1]School of Mathematics, Jiaying
Univiersity, Meizhou, Guangdong
514015, P.R. China
Full list of author information is
available at the end of the article

## Abstract

In this paper, a two-level domain decomposition algorithm for linear complementarity problem (LCP) is introduced. Inner and outer approximation sequences to the solution of LCP are generated by the proposed algorithm. The algorithm is proved to be convergent and can reach the solution of the problem within finite steps. Some simple numerical results are presented to show the effectiveness of the proposed algorithm.

## 1 Introduction

In this paper, we consider the following linear complementarity problem (LCP) of finding $u \in R^n$ such that

$$u \geq 0, \qquad F(u) \geq 0, \qquad u^T F(u) = 0, \tag{1.1}$$

where $F(u) = Au + b$, $A$ is an $M$-matrix, $b \in R^n$ is a given vector.

LCP is a wide class of problems and has many applications in such fields as physics, optimum control, economics, *etc.* As a result of their broad applications, the literature in this field has benefited from contributions made by mathematicians, computer scientists, engineers of many kinds, and economists of diverse expertise. There are many surveys and special volumes (see, *e.g.*, [1–3] and the references therein).

Domain decomposition techniques are widely used to solve PDEs since 1980's. This kind of techniques attracts much attention, since it is portable and easy to be parallelized on parallel machines. It has been applied to solve various linear and nonlinear variational inequality problems, and the numerical results show that it is efficient, see, for example, [4–6]. It contains many algorithms, such as classical additive Schwarz method (AS), multiplicative Schwarz method (MS), restricted additive Schwarz method (RAS), and so on. In [7], a variant of Schwarz algorithm, called two-level additive Schwarz algorithm (TLAS), was proposed for the solution of a kind of linear obstacle problem. This method can divide the original problem into subproblems in an 'efficient' way. In other words, the domain is decomposed in different way at each step and the dimensions of the subproblems we deal with are lower than that of the original problem. The numerical results show that the TLAS is significant. In [8], the TLAS is extended for the nonlinear complementarity problem with an $M$-function. The algorithm offers a possibility of making use of fast nonlinear solvers to the subproblems, and the choice of the initial is much easier than that of the TLAS. Another efficient way to solve problem (1.1) is given by semismooth Newton

Springer

methods (*e.g.*, see [9, 10]). This method is attractive, because it converges rapidly from any sufficiently good initial iterate, and the subproblems are also systems of equation. An active set strategy is also an efficient way to solve discrete obstacle problems, see, for example, [11–13]. Based on some kind of active set strategy, the discrete obstacle problem can be reduced to a sequence of linear problems, which are then solved by some efficient methods. In this paper, we combine the idea of the active set strategy with the thought of TLAS, *i.e.*, constructing inner and outer approximation sequences to the solution of LCP, and present a two-level domain decomposition algorithm. As we will see in the sequel, the main difference between the two-level domain decomposition algorithm (TLDD) and TLAS discussed in [7] lies in the way of constructing the outer approximation of the solution. What's more, with the idea of an active set strategy, the TLDD may be easier extended to other problems, such as bilateral obstacle problem.

The paper in the sequel is organized as follows. In Section 2, we give some preliminaries and present a two-level domain decomposition algorithm for problem (1.1). In Section 3, we discuss the convergence of the algorithm proposed in Section 2. In Section 4, we report some simple numerical results.

## 2 Preliminaries and two-level domain decomposition algorithm

In this section, we give some preliminaries and present a two-level domain decomposition algorithm for solving problem (1.1).

Firstly, similarly to [7, 8], we introduce two operators, which will be useful in the construction of the algorithm in this paper. Let $N = \{1, 2, \ldots, n\}$. Let $I, J$ be a nonoverlapping decomposition of $N$. That is, $N = I \cup J$ and $I \cap J = \emptyset$. For any $v \in R^n$, we introduce the following linear problem of finding $w \in R^n$ such that

$$w_I = v_I, \qquad F_J(w) = 0, \tag{2.1}$$

where $v_I$ denotes the subvector of $v$ with elements $v_j$ ($j \in I$). Similar notation will be used in the sequel. We denote linear system (2.1) above by the operation form

$$w = G_J(v).$$

Similarly, we introduce the following problem of finding $w \in R^n$ such that

$$w_I = v_I, \qquad \min\{F_J(w), w_J\} = 0. \tag{2.2}$$

We denote nonlinear problem (2.2) above by the operation form

$$w = T_J(v).$$

**Theorem 2.1** [1] *Problem* (1.1) *is equivalent to the following variational inequality of finding* $u \in R^n$ *such that*

$$\big(F(u), v - u\big) \geq 0, \quad \forall v \in R^n. \tag{2.3}$$

**Theorem 2.2** [8] *The solution of problem* (1.1), *or equivalently* (2.3), *is unique and is the minimal element of S, where S is the supersolution set of problem* (1.1), *which is defined by*

$$S = \{v \in R^n : v \geq 0 \text{ and } F(v) \geq 0\}.$$

Similarly, we have the following theorem.

**Theorem 2.3** *The solution of problem* (1.1), *or equivalently* (2.3), *is unique and is the maximal element of U, where U is the subsolution set of problem* (1.1), *which is defined by*

$$U = \{v \in R^n : v \geq 0 \text{ and } \min\{v, F(v)\} \leq 0\}.$$

Based on Theorems 2.1 and 2.2, we can construct the following additive Schwarz algorithm for LCP (1.1).

**Algorithm 2.1** (Additive Schwarz algorithm with two subdomains) Let $I$ and $J$ be a decomposition of $N$, *i.e.*, $I \cup J = N$. Given $u^0 \in S$. For $k = 0, 1, \dots$, do the following two steps until convergence.

Step 1: Solve the following two subproblems in parallel

$$\begin{cases} \text{find } u^{k,1} \in K^{k,1} \text{ such that} \\ (F_I(u^{k,1}), v_I - u_I^{k,1}) \geq 0, \quad \forall v \in K^{k,1}, \end{cases}$$

$$\begin{cases} \text{find } u^{k,2} \in K^{k,2} \text{ such that} \\ (F_J(u^{k,2}), v_J - u_J^{k,2}) \geq 0, \quad \forall v \in K^{k,2}, \end{cases}$$

where

$$K^{k,1} = \{v \in R^n : (v - u^k)_{N \setminus I} = 0\},$$
$$K^{k,2} = \{v \in R^n : (v - u^k)_{N \setminus J} = 0\}.$$

Here we define $N \setminus I = \{j \in N : j \notin I\}$ for any subset $I$ of $N$.

Step 2: $u^{k+1} = \min(u^{k,1}, u^{k,2})$, where 'min' should be understood by componentwise.

Similar to the proof of Theorem 2.4 in [8], we have the following convergence theorem for Algorithm 2.1.

**Theorem 2.4** *Let the sequence* $\{u^k\}$ *be generated by Algorithm* 2.1. *For* $k = 0, 1, \dots$, *we have*

(a) $u^{k,i} \leq u^k, i = 1, 2$ *and then* $u^{k+1} \leq u^k$,

(b) $u^{k,i} \in S, i = 1, 2$ *and then* $u^{k+1} \in S$,

(c) $\lim_{k \to \infty} u^k = u$,

*where u is the solution of problem* (1.1).

In what follows, we let $N^0 = \{j \in N : u_j = 0\}$, $N^+ = \{j \in N : u_j > 0\}$, where $u$ is the solution of problem (1.1). If $u^0 \in S$, then the sequence $\{u^k\}$ generated by Algorithm 2.1 is

in $S$ and monotonically decreases and converges to the solution. Hence, if we define the coincidence set of $u^k$ as follows

$$I^k = \left\{ j \in N : u_j^k = 0 \right\},\tag{2.4}$$

we have by the monotonicity of $\{u^k\}$ such that

$$I^k \subseteq I^{k+1} \subseteq N^0, \quad k = 0, 1, \ldots.$$

Actually, this gives inner approximations for the coincidence set $N^0$.

There are many algorithms based on active set strategy. Based on some kind of criterion, the index set is divided into two parts: active set and inactive set. We only need to calculate the simplified linear system related to the inactive set. We also draw on the experience of active set strategy to derive the outer approximations for the coincidence set. To be precise, we define

$$O^k = \left\{ j \in N : w_j^k = 0 \text{ and } F_j(w^k) \geq 0 \right\}, \qquad L^k = N \setminus O^k, \quad k = 0, 1, \ldots,\tag{2.5}$$

and define $C^k$ as

$$C^k = N \setminus \left( I^k \cup L^k \right).\tag{2.6}$$

$C^k$ may contain both elements of $N^0$ and $N^+$. So, it is called the critical subsets. Let

$$\hat{C}^k = C^k \cup H^k,\tag{2.7}$$

where $H^k$ is a subset of $N$ corresponding to an overlapping of the subsets associated with $L^k$ and $C^k$. That is $H^k \subset L^k$ and $H^k = \hat{C}^k \cap L^k$.

Now, we are ready to present two-level domain decomposition algorithm for problem (1.1).

**Algorithm 2.2** (Two-level domain decomposition algorithm)

1. Initialization. $k := 0$:
   (a) Choose an initial $u^0$, $w$ such that $u^0 \in S$ and $w \in U$. Define the coincidence set $I^0$ according to (2.4).
   (b) Solve $w^0$ such that

$$\begin{cases} w_i^0 = 0, & i \in I^0 \text{ or } w_i = 0, F_i(w) \geq 0, \\ F_i(w^0) = 0, & \text{otherwise,} \end{cases}\tag{2.8}$$

and define $L^0$, $C^0$ and $\hat{C}^0$ according to (2.5), (2.6) and (2.7), respectively.

2. Iteration step:
   (a) Inner approximation (additive Schwarz algorithm with two subdomains). Solve the following two subproblems in parallel:
      (i) The subproblem defined by the following obstacle problem

$$u^{k,1} = T_{\hat{C}^k}(u^k).$$

(ii) The subproblem defined by the following linear equation

$$u^{k,2} = G_{L^k}(u^k). \tag{2.9}$$

Let $u^{k+1} = \min(u^{k,1}, u^{k,2})$ and define the coincidence set $I^{k+1}$ according to (2.4).

(b) Outer approximation. Solve the linear system

$$\begin{cases} w_i^{k+1} = 0, & i \in I^{k+1} \text{ or } w_i^k = 0, F_i(w^k) \geq 0, \\ F_i(w^{k+1}) = 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

If $F(w^{k+1}) \geq 0$, then stop; $w^{k+1}$ is the solution. Otherwise, define $L^{k+1}$ and $\hat{C}^{k+1}$ according to (2.5) and (2.7), respectively, and let $k := k + 1$ and return to step 2.

**Remark 2.1** The subproblems in PSOR method and classical Schwarz algorithm are obstacle problems, while the subproblems (2.9) and (2.10) can be solved by the use of fast linear solvers.

**Remark 2.2** The difference between Algorithm 2.2 and Algorithm 3.3 in [7] lies on the way of generating the outer approximation sequence. Algorithm 3.3 in [7] seems difficult to extend to other problems, such as bilateral obstacle problem, while the idea of Algorithm 2.2 may be easier to be applied to other problems.

## 3 The convergence of Algorithm 2.2

In this section, we analyze the convergence of Algorithm 2.2. First, we introduce some lemmas.

**Lemma 3.1** *Let $u^0 \in S$, $w \in U$ and $w^0$ be defined by (2.8). Then, we have $0 \leq w \leq w^0$ and $w^0 \in U$.*

*Proof* Let $\hat{I} = \hat{I}^1 \cup \hat{I}^2$, where $\hat{I}^1 = \{i | i \in I^0\}$, $\hat{I}^2 = \{i | w_i = 0 \text{ and } F_i(w) \geq 0\}$ and $\hat{J} = N \setminus \hat{I}$. By the definition of $w^0$, we have $w_{\hat{I}^2}^0 = w_{\hat{I}^2} = 0$. By Theorems 2.2 and 2.3, we have $u^0 \geq w \geq 0$. Hence if $i \in \hat{I}^1$, we have $w_i = 0$. Then $w_{\hat{I}}^0 = w_{\hat{I}} = 0$. Since $w \in U$, we have $F_{\hat{J}}(w) \leq F_{\hat{J}}(w^0) = 0$. Hence, noting that $F(u) = Au + b$, and $A$ is an $M$-matrix, we have $0 \leq w \leq w^0$. This completes the proof. □

**Lemma 3.2** *Let $u^0 \in S$, let subsets $L^0$ and $\hat{C}^0$ be defined by (2.5) and (2.7), respectively, then*

$$u^{0,1} = T_{\hat{C}^k}(u^0) \in S, \quad u^{0,1} \leq u^0, \tag{3.1}$$

$$u^{0,2} = G_{L^0}(u^0) \in S, \quad u^{0,2} \leq u^0, \tag{3.2}$$

$$u^1 = \min(u^{0,1}, u^{0,2}) \in S, \tag{3.3}$$

$$u \leq u^1 \leq u^0. \tag{3.4}$$

*Proof* Equation (3.1) can be directly obtained by Theorem 2.4. By (2.9), we have

$$F_{L^0}(u^{0,2}) = 0, \quad u_{N \setminus L^0}^{0,2} = u_{N \setminus L^0}^0. \tag{3.5}$$

Since $u^0 \in S$, we have

$$F_{L^0}(u^0) \geq 0.$$

Noticing that $F(u) = Au + b$, and $A$ is an $M$-matrix, (3.5) concludes

$$u^{0,2} \leq u^0. \tag{3.6}$$

We have by (3.5), (3.6) that

$$F_{N \setminus L^0}(u^{0,2}) \geq F_{N \setminus L^0}(u^0) \geq 0. \tag{3.7}$$

Let $L^0 = L_1^0 \cup L_2^0$, where $L_1^0 = \{i \in L^0 | w_i^0 = 0, F_i(w^0) < 0\}$, and $L_2^0 = \{i \in L^0 | w_i^0 > 0\}$. Since $w^0 \in U$, $w^0 \leq u$, we have $u_i > 0$ for $i \in L_2^0$, and then $F_{L_2^0}(u) = 0$. For $i \in L_1^0$, we have $i \in I^0$, and then $u_i = 0$. It follows then from (3.5) and $u^0 \in S$ that

$$u_{N \setminus L_2^0}^{0,2} \geq u_{N \setminus L_2^0}, \qquad F_{L_2^0}(u^{0,2}) = F_{L_2^0}(u) = 0, \tag{3.8}$$

which means $u^{0,2} \geq u \geq 0$. This, together with (3.7) and (3.8), implies that $u^{0,2} \in S$. Therefore, (3.2) holds. Similar to the proof in Theorem 2.4, we have (3.3) and (3.4). The proof is then completed. □

By Lemmas 3.1 and 3.2, and the principle of induction, we have $w^{k+1} \geq w^k \geq 0$, $u^k \geq u^{k+1} \geq 0$, $k = 1, 2, \ldots$.

**Lemma 3.3** $O^{k+1} \subseteq O^k$ *and* $I^k \subseteq I^{k+1} \subseteq N^0$, $k = 0, 1, \ldots$. *If* $F(w^{k+1}) \geq 0$, $w^{k+1}$ *is the solution.*

*Proof* If $j \in O^{k+1}$, by the definition of $O^{k+1}$, we have $w_j^{k+1} = 0$, $F_j(w^{k+1}) \geq 0$. Noting that $w^{k+1} \geq w^k \geq 0$, we have $w_j^k = 0$. If $F_j(w^k) < 0$, notice that $A$ is an $M$-matrix, we have $F_j(w^{k+1}) \leq F_j(w^k) < 0$, which is a contradiction. Hence, $j \in O^k$ and $O^{k+1} \subseteq O^k$. $I^k \subseteq I^{k+1} \subseteq N^0$, $k = 0, 1, \ldots$ is obvious. Noting (2.10), it is obvious that if for some $k$ such that $F(w^k) \geq 0$, $w^k$ is the solution. □

**Theorem 3.4** *The sequence generated by two-level domain decomposition method* (Algorithm 2.2) *converges to the solution $u$ of problem* (1.1) *after a finite number of iterations.*

*Proof* If for some $k$, $F(w^k) \geq 0$, by Lemma 3.1, $w^k$ is the solution and $w^k = u$ since the problem (1.1) has only one solution. Otherwise, since $u^{k+1} = \min(u^{k,1}, u^{k,2})$, we have $I^{k,1} \subseteq I^{k+1}$ and hence $I^{k+1} \setminus I^k \neq \emptyset$. Noting that $I^k \subseteq I^{k+1}$ and that $N$ is an index set with finite elements, $I^{k+1} \setminus I^k \neq \emptyset$ can only occur in finite steps. By Lemma 3.3, we have $O^{k+1} \subseteq O^k$, and $O^k \setminus O^{k+1} \neq \emptyset$ also can only occur in finite steps. In this case, after some finite steps, we have $I^{k+1} = I^k$, $O^{k+1} = O^k$. By the definition of $w^{k+1}$, we have $F_i(w^{k+1}) \geq 0$, $\forall i \in N$. Hence, by Lemma 3.3, $w^{k+1}$ is the solution. This completes the proof. □

## 4 Numerical experiment

In this section, we present numerical experiments in order to investigate the efficiency of Algorithm 2.2. The programs are coded in Visual C++ 6.0 and run on a computer with 2.0 GHz CPU. In the tests, we consider the following LCP:

$$u \geq 0, \qquad A(u) - b \geq 0, \qquad u^T \big( A(u) - b \big) = 0, \tag{4.1}$$

where

$$A = \frac{1}{h^2} \begin{pmatrix} H & -I & & \\ -I & H & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & H \end{pmatrix}$$

and

$$H = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{pmatrix},$$

$h = \frac{1}{\sqrt{n+1}}$, $b = (b_1, b_2, \ldots, b_n)^T$ is a given vector. In our test, we set $b_i = -0.5$, $i = 1, 2, \ldots, n/2$ and $b_i = 0.5$, $i = n/2 + 1, n/2 + 2, \ldots, n$.

The matrix $A$ may be obtained by discretizing the operator $-\triangle u$ by using five-point difference scheme with a constant mesh step size $h = 1/(m+1)$, where $m$ denotes the number of mesh nodes in $x$- or $y$-direction ($n = m^2$ is the total number of unknowns).

We compare different algorithms from the point of view of iteration numbers and CPU times. Here, we consider three algorithms: classical additive Schwarz algorithm (*i.e.*, Algorithm 2.1, denoted by AS), Newton's method proposed in [9] (denoted by SSN), and Algorithm 2.2 (denoted by TLDD). In the AS, we decompose $N$ into two equal parts with the overlapping size $O(\frac{1}{10})$. In the algorithms we considered, all subproblems relating to obstacle problems are solved by PSOR with the same relaxation parameter $\omega = 1.4$, and the initial point is $u^0 = A^{-1}e$ with $e = (1, 1, \ldots, 1)^T$. The tolerance in the subproblems of the algorithms is chosen to be equal to $10^{-3}$ in $\| \cdot \|_2$-norm, while in the outer iterative processes, it is chosen to be equal to $10^{-6}$ in $\| \cdot \|_2$-norm. In the TLDD, we choose initial $w = 0$. The tolerance in the subproblems of the algorithms is chosen to be equal to $10^{-4}$ in $\| \cdot \|_2$-norm, while in the outer iterative processes, it is chosen to be equal to $10^{-6}$ in $\| \cdot \|_2$-norm. In the SSN, we choose $\epsilon = 10^{-6}$, $p = 3$, $\rho = 0.5$, $\beta = 0.3$, which is defined by the procedure proposed in ([9], Section 7). We choose the initial point $u^0 = 0$.

We investigate the performances of each algorithm with different dimensions. Table 1 gives the iteration numbers and CPU times for the above-mentioned algorithms. From the table, we can easily see that the iteration numbers of TLDD are fewest among the algorithms we considered. The subproblems in AS are solved by PSOR, and it takes very little time to find an approximate solution to the obstacle subproblems. Nevertheless, in order to find the exact solution of subproblems, SSN and TLDD spent much more time to

**Table 1 Comparisons of iteration numbers and cpu times**

| $n^2$ | AS | | SSN | | TLDD | |
|---|---|---|---|---|---|---|
| | iter. | cpu | iter. | cpu | iter. | cpu |
| 100 | 51 | 0.015 | 5 | 0.015 | 3 | 0.047 |
| 400 | 155 | 0.64 | 10 | 1.997 | 6 | 1.154 |
| 900 | 318 | 6.567 | 12 | 27.378 | 9 | 17.755 |
| 1,600 | 546 | 34.564 | 14 | 178.101 | 11 | 112.117 |

solve the related linear equations at each iteration step. This may explain why these two algorithms did not perform as well as we expected.

**Concluding remark** In this paper, we propose a new kind of domain decomposition method for linear complementarity problem and establish its convergence. From the numerical result, we can see that this method needs less iteration number to converge to the solution rapidly than the additive Schwarz method and SSN. There are still some interesting future works that need to be done. For example, as we can see from TLDD, the main work is calculating the linear equations; we can discuss the affect of inexact solution for related linear subproblems. It is also interesting for us to extend the new method to some other problems, such as nonlinear complementarity problem and bilateral obstacle problem. We leave it as a possible future research topic.

**Competing interests**
The authors declare that they have no competing interests.

**Authors' contributions**
All authors jointly worked on the results and they read and approved the final manuscript.

**Author details**
[1]School of Mathematics, Jiaying Univiersity, Meizhou, Guangdong 514015, P.R. China. [2]College of Mathematics and Information Science, Jiangxi Normal University, Nanchang, Jiangxi 330022, P.R. China.

**References**
 1. Harker, PT, Pang, JS: Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. Math. Program. **48**, 161-220 (1990)
 2. Billups, SC, Murty, KG: Complementarity problems. J. Comput. Appl. Math. **124**, 303-318 (2000)
 3. Ferris, MC, Mangasarian, OL, Pang, JS (eds.): Complementarity: Applications, Algorithms and Extensions. Kluwer Academic, Dordrecht (2001)
 4. Badea, L, Wang, JP: An additive Schwarz method for variational inequalities. Math. Comput. **69**, 1341-1354 (1999)
 5. Zeng, JP, Zhou, SZ: On monotone and geometric convergence of Schwarz methods for two-sided obstacle problems. SIAM J. Numer. Anal. **35**, 600-616 (1998)
 6. Jiang, YJ, Zeng, JP: Additive Schwarz algorithm for the nonlinear complementarity problem with $M$-function. Appl. Math. Comput. **190**, 1007-1019 (2007)
 7. Tarvainen, P: Two-level Schwarz method for unilateral variational inequalities. IMA J. Numer. Anal. **19**, 273-290 (1999)
 8. Xu, HR, Zeng, JP, Sun, Z: Two-level additive Schwarz algorithms for nonlinear complementarity problem with an $M$-funcion. Numer. Linear Algebra Appl. **17**, 599-613 (2010)
 9. Luca, TD, Facchinei, F, Kanzow, C: A theoretical and numerical comparison of some semismooth algorithms for complementarity problems. Comput. Optim. Appl. **16**, 173-205 (2000)
10. Li, DH, Li, Q, Xu, HR: An almost smooth equation reformulation to the nonlinear complementarity problem and Newton's method. Optim. Methods Softw. **27**, 969-981 (2012)
11. Hintermüller, M, Ito, K, Kunisch, K: The primal-dual active set strategy as a semismooth Newton method. SIAM J. Optim. **13**, 865-888 (2002)
12. Kunisch, K, Rösch, A: Primal-dual active set strategy for a general class of constrained optimal control problems. SIAM J. Optim. **13**, 321-334 (2002)

13.  Kärkkäinen, T, Kunisch, K, Tarvainen, P: Augmented Lagrangian active set methods for obstacle problems. J. Optim. Theory Appl. **119**, 499-533 (2003)