



# Approximation Algorithm of Minimum All-Ones Problem for Arbitrary Graphs

Chen Wang<sup>a</sup>, Chao Wang<sup>b,\*</sup>, Gregory Z. Gutin<sup>c</sup>, Xiaoyan Zhang<sup>d</sup>

<sup>a</sup>Address: College of Software, Nankai University

<sup>b</sup>Address: College of Software, Nankai University

<sup>c</sup>Address: Royal Holloway, University of London

<sup>d</sup>School of Mathematical Science, Institute of Mathematics, Nanjing Normal University, Jiangsu 210023, China

## Abstract

Let  $G = (V, E)$  be a graph and let each vertex of  $G$  has a lamp and a button. Each button can be of  $\sigma^+$ -type or  $\sigma$ -type. Assume that initially some lamps are on and others are off. The button on vertex  $x$  is of  $\sigma^+$ -type ( $\sigma$ -type, respectively) if pressing the button changes the lamp states on  $x$  and on its neighbors in  $G$  (the lamp states on the neighbors of  $x$  only, respectively). Assume that there is a set  $X \subseteq V$  such that pressing buttons on vertices of  $X$  lights all lamps on vertices of  $G$ . In particular, it is known to hold when initially all lamps are off and all buttons are of  $\sigma^+$ -type.

Finding such a set  $X$  of the smallest size is NP-hard even if initially all lamps are off and all buttons are of  $\sigma^+$ -type. Using a linear algebraic approach we design a polynomial-time approximation algorithm for the problem such that for the set  $X$  constructed by the algorithm, we have  $|X| \leq \min\{r, (|V| + \text{opt})/2\}$ , where  $r$  is the rank of a (modified) adjacent matrix of  $G$  and  $\text{opt}$  is the size of an optimal solution to the problem. To the best of our knowledge, this is the first polynomial-time approximation algorithm for the problem with a nontrivial approximation guarantee.

© 2023 Published by Elsevier Ltd.

## Keywords:

lamp lighting problem, minimum all-ones problem, approximation algorithm, complexity

## 1. Introduction

The all-ones problem is a fundamental problem in applied mathematics, first proposed by Sutner in 1988 [17]. This problem has applications in linear cellular automata, as discussed in [18] and the references therein. To illustrate the problem, consider an  $n \times n$  grid with each area having a light lamp and a switch, and every lamp is initially off. Turning the switch on in some area lights the lamp in the area and the lamps in neighboring areas. Is there a set  $X$  of areas such that turning the switches on in  $X$  will turn on all the lamps? This problem can be extended to all graphs and we will call it the *all-ones problem*. Sutner [18] proved that a solution  $X$  exists for every graph. Later, several simple proofs of this result were given or rediscovered [3, 5, 7, 10, 13].

Many variants of the all-ones problem have been introduced and studied [1, 2, 6, 7, 11, 12, 19] over years. There are two important generalizations of the all-ones problem: (i) the initial state of lamps and switches can be arbitrary,

\*Corresponding author.

Email addresses: 2120220677@mail.nankai.edu.cn (Chen Wang), wangchao@nankai.edu.cn (Chao Wang), gutin@cs.rhul.ac.uk (Gregory Z. Gutin), xiaoyanice@aliyun.com (Xiaoyan Zhang)

i.e., some are on and the others are off, and (ii) every switch can be either of  $\sigma^+$ -type which changes the states of the lamp on its vertex and the lamps on the neighbors of its vertex or  $\sigma$ -type which changes the states of the lamps on the neighbors of its vertex only. As a result of these two generalizations, the *generalized all-ones problem* may not have a solution  $X$  which lights all lamps. This generalized problem is studied in this paper. Under the condition that such a solution  $X$  exists for the generalized all-ones problem, it is natural to ask for  $X$  of minimum size.

Unfortunately, this minimization problem is NP-hard even for all-ones problem [16]; we will call the minimization all-ones problem the *min all-ones problem*. Galvin and Lu both proved that the min all-ones problem of trees can be solved in linear time [9, 14]. Building on this, Chen proposed an algorithm for solving the min generalized all-ones problem on trees, with linear complexity [4]. Manuel et al. provided solutions for some of the widely studied architectures, such as binomial trees, butterfly, and benes networks [15]. Fleischer and Yu provided a detailed survey of the generalized all-ones problem [8]. More recently, Zhang extended the all-ones problem to the all-colors problem, in which each lamp had other states besides being on and off, and obtained additional findings on the all-colors problem [20].

Although significant research has been conducted on the all-ones problem on special graphs, such as trees, resulting in efficient algorithms, no polynomial-time approximation algorithms have been designed for the min all-ones problem on general graphs. Trees and cyclic graphs only represent a fraction of general graphs. In practical engineering scenarios, complex graphs are more common. In this paper, we design a polynomial-time approximation algorithm for the min generalized all-ones problem. If the problem has a solution, our algorithm outputs a solution  $X$  such that  $|X| \leq \min\{r, (|V| + \text{opt})/2\}$ , where the rank of a (modified) adjacent matrix of  $G$  and  $\text{opt}$  is the size of an optimal solution to the problem.

Apart from the introduction, this paper contains three sections. In Section 2, we introduce our approximation algorithm in detail. Section 3 shows the theoretical analysis and performance evaluation of this algorithm. Section 4 summarizes all the work of this paper and discusses future work.

## 2. Approximation algorithm of min generalized all-ones problem

### 2.1. Linear algebraic formulation of min generalized all-ones problem

It is not hard to see that the min generalized all-ones problem can be described as the following linear integer program over  $\mathbb{F}_2$ . For an arbitrary graph  $G = (V, E)$  with  $V = \{v_1, \dots, v_n\}$  we can get its modified adjacency matrix  $A = (a_{ij})_{n \times n}$  such that for all  $i \neq j$ ,  $a_{ij} = 1$  if  $v_i v_j \in E$  and  $a_{ij} = 0$  otherwise, and for all  $i \in \{1, 2, \dots, n\}$ ,  $a_{ii} = 1$  ( $a_{ii} = 0$ , respectively) if the switch on  $v_i$  is of  $\sigma^+$ -type (of  $\sigma$ -type, respectively). Combined with the initial state  $B = (b_1, b_2, \dots, b_n)$ , where  $b_i = 0$  if the lamp on vertex  $v_i$  is initially on and  $b_i = 1$  if the lamp is initially off, we can construct a system of linear equations  $AU = B$  over  $\mathbb{F}_2$ . The solution to this problem is the minimum of  $\sum U = \sum_{i=1}^n u_i$ .

Suppose the rank of  $A$  is  $r$  and the corank is  $m$  so that  $m + r = n$ . If  $a_{ii} = 1$  for all  $i \in \{1, 2, \dots, n\}$ , the system of equations  $AU = B$  must have a solution, but if  $a_{ii} = 0$ , the system may not necessarily have a solution. However, as long as the system has at least one solution  $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)^T$ , we can find all solutions of the system using the following system combining  $\gamma$  with the fundamental solution set  $\eta = (\eta_1, \eta_2, \dots, \eta_m)$  within time  $O(n^3)$ . Here  $x_i$  is the coefficient of the column vector  $\eta_i = (\eta_{1i}, \dots, \eta_{ni})^T$ .

$$\eta X + \gamma = \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1m} \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2m} \\ \eta_{31} & \eta_{32} & \cdots & \eta_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{n1} & \eta_{n2} & \cdots & \eta_{nm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} + \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_n \end{pmatrix} \quad (1)$$

The problem is how to find the appropriate column vector  $X$  to minimize  $\sum U$ , under the condition that  $X$  has a total of  $2^m$  values. This problem was proven to be an NP-complete [16]. Therefore, the next subsection provides an approximation algorithm running in polynomial time.

## 2.2. Approximation algorithm

Firstly, it can be observed that the polynomial time complexity (not exceeding  $O(n^3)$ ) of finding the matrix  $(\eta_1, \eta_2, \dots, \eta_m)$  and the special solution  $\gamma$  makes this process cost-effective in solving NP-complete problems. Secondly, it is challenging to identify alternative methods capable of directly computing the optimal solution without obtaining all the solutions. Even if such a solution is obtained, verification is often infeasible. When  $\eta$  and  $\gamma$  are known, we need to find the  $X$  that minimizes  $\sum U$ .

$$\begin{aligned} \eta X + \gamma &= \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1m} \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2m} \\ \eta_{31} & \eta_{32} & \cdots & \eta_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{n1} & \eta_{n2} & \cdots & \eta_{nm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} + \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_n \end{pmatrix} \\ &= \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \vdots \\ \delta_n \end{pmatrix} + \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_n \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{pmatrix} \end{aligned} \quad (2)$$

**Proposition 2.1.** Row exchanges of matrix  $\eta$  do not change  $\sum U$ .

*Proof.* Multiply both sides of Equation 2 by matrix  $P$  as shown in Equation 3. Matrix  $P$  is a product of elementary matrices that perform row exchanges. This operation essentially reorders the elements of vector  $U$ , but does not change the  $\sum U$ .

$$P(\eta X + \gamma) = P(\delta + \gamma) = PU \quad (3)$$

□

**Proposition 2.2.** Column transformation of matrix  $\eta$  does not change  $\sum U$ .

*Proof.* Let  $Q_{m \times m}$  be a full rank matrix, and  $QZ = X$ , with the following equation.

$$\begin{aligned} \eta X + \gamma &= \eta QZ + \gamma \\ &= (\eta Q)Z + \gamma \\ &= \epsilon Z + \gamma \\ &= \delta + \gamma \\ &= U \end{aligned} \quad (4)$$

$Q$  is the transition matrix between  $X$  and  $Z$ , and  $Q$  is full rank. When we find that  $X$  makes  $\sum U$  the smallest, we can definitely find the corresponding  $Z$ , so that the obtained  $U$  is the same. □

We can transform the  $\eta$  column into an echelon form using row exchanges and column transformations, as shown in the following equation, with a complexity of  $O(m^2n)$ . The question mark indicates that the value of the number is uncertain, which may be 0 or 1. We can divide the matrix into  $m + 1$  parts based on the echelon and assume the last line of the  $i$ -th part is line  $k_i$  ( $i = 0, 1, \dots, m$ ) for the rank of matrix  $\eta$  is always  $m$ . Part 0 is the most special, with all

elements in each row being 0. To ensure that Equation 4 holds, there should be  $(u_1, u_2, \dots, u_{k_0}) = (\gamma_1, \gamma_2, \dots, \gamma_{k_0})$ .

$$\eta Q = \epsilon = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_{(k_1+1)1} & 1 & 0 & 0 & \cdots & 0 \\ \epsilon_{(k_1+2)1} & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_{(k_2+1)1} & \epsilon_{(k_2+1)2} & 1 & 0 & \cdots & 0 \\ \epsilon_{(k_2+2)1} & \epsilon_{(k_2+2)2} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_{k_m1} & \epsilon_{k_m2} & \epsilon_{k_m3} & \epsilon_{k_m4} & \cdots & 1 \end{pmatrix} \quad (5)$$

In the following  $m$  parts, we will use greedy algorithms to solve for the  $Z$  value on the Echelon of each part. Part 1 of the linear Equation 5 is shown in Equation 6.  $(\gamma_{k_0+1}, \gamma_{k_0+2}, \dots, \gamma_{k_1})$  is known and  $(\delta_{k_0+1}, \delta_{k_0+2}, \dots, \delta_{k_1})$  is unknown. It is important to ensure that  $\delta_i$  is as similar to  $\gamma_i$  as possible. At this moment  $z_1$  only has two possible values: 0 and 1. Therefore, the idea of a greedy algorithm is adopted here. If there are more 0's than 1's in the range from  $\gamma_{k_0+1}$  to  $\gamma_{k_1}$ , then  $z_1$  is set to 0. If there are more 1's than 0's, then  $z_1$  is set to 1. Therefore, we can directly obtain the value of  $x_1$  by solving it here, while ensuring that  $\sum_{i=k_0+1}^{k_1} u_i \leq (k_1 - k_0)/2$ .

$$\begin{cases} z_1 = \delta_{k_0+1} & \gamma_{k_0+1} \\ z_1 = \delta_{k_0+2} & \gamma_{k_0+2} \\ z_1 = \delta_{k_0+3} & \text{Compare to } \gamma_{k_0+3} \\ \vdots & \vdots \\ z_1 = \delta_{k_1} & \gamma_{k_1} \end{cases} \quad (6)$$

The value of  $z_2$  can be calculated through  $z_1$ . Part 2 of Equation 5 can be written as shown in Equation 7.  $(\gamma_{k_1+1}, \gamma_{k_1+2}, \dots, \gamma_{k_2})$  is known, and  $(\delta_{k_1+1}, \delta_{k_1+2}, \dots, \delta_{k_2})$  needs to satisfy the Equation 5 and be as similar to  $(\gamma_{k_1+1}, \gamma_{k_1+2}, \dots, \gamma_{k_2})$  as possible. The variables in Equation 7 are  $z_1$  and  $z_2$ , and  $z_1$  has been solved before through a greedy algorithm, so the unknown variable is only  $z_2$ . Since  $\epsilon_{i1}z_1$  are constants, we can move them from the left side of the equation to the right side, and these two equation systems are obviously equivalent. Then, we need to ensure that  $\epsilon_{(k_1+i)1}z_1 + \delta_{k_1+i}$  is as similar to  $\gamma_{k_1+i}$  as possible. It can be seen that another transformation can be carried out, which is equivalent to making  $\delta_{k_1+i}$  as similar to  $\epsilon_{(k_1+i)1}z_1 + \gamma_{k_1+i}$  as possible. In this way, we have separated the variables: the left side of the equation is the variable  $z_2$ , the right side of the equation is the variable  $\delta_{k_1+i}(\delta_{k_1+i} = z_2)$ , and the column of  $\epsilon_{(k_1+i)1}z_1 + \gamma_{k_1+i}$  are constants. At this point, we find that part 2 of Equation 5 has been transformed to be very similar to part 1. Therefore, if there are more 0's than 1's in the range from  $\epsilon_{(k_1+1)1}z_1 + \gamma_{k_1+1}$  to  $\epsilon_{(k_2)1}z_1 + \gamma_{k_2}$ , then  $z_2$  is set to 0. If there are more 1's than 0's, then  $z_2$  is set to 1. Therefore, the value of  $z_2$  can be solved here and  $\sum_{i=k_1+1}^{k_2} u_i \leq (k_2 - k_1)/2$  is ensured.

After obtaining the value of  $z_2$ , the value of  $\epsilon_{i1}z_1 + \epsilon_{i2}z_2$  can be calculated, and the value of  $z_3$  can be calculated again. Following this pattern, the values of  $Z = (z_1, z_2, \dots, z_m)$  can be obtained. Then  $\epsilon Z + \gamma = U$ , we obtain  $U$ . The complete algorithm is shown in Algorithm 1.

$$\begin{cases} \epsilon_{(k_1+1)1}z_1 + z_2 = \delta_{k_1+1} \\ \epsilon_{(k_1+2)1}z_1 + z_2 = \delta_{k_1+2} \\ \epsilon_{(k_1+3)1}z_1 + z_2 = \delta_{k_1+3} \\ \vdots \\ \epsilon_{(k_2)1}z_1 + z_2 = \delta_{k_2} \end{cases} \quad \text{Compare to} \quad \begin{cases} \gamma_{k_1+1} \\ \gamma_{k_1+1} \\ \gamma_{k_1+1} \\ \vdots \\ \gamma_{k_2} \end{cases}$$

$$\Downarrow$$

$$\begin{cases} z_2 = \delta_{k_1+1} + \epsilon_{(k_1+1)1}z_1 \\ z_2 = \delta_{k_1+2} + \epsilon_{(k_1+2)1}z_1 \\ z_2 = \delta_{k_1+3} + \epsilon_{(k_1+3)1}z_1 \\ \vdots \\ z_2 = \delta_{k_2} + \epsilon_{(k_2)1}z_1 \end{cases} \quad \text{Compare to} \quad \begin{cases} \gamma_{k_1+1} \\ \gamma_{k_1+1} \\ \gamma_{k_1+1} \\ \vdots \\ \gamma_{k_2} \end{cases} \quad (7)$$

$$\Downarrow$$

$$\begin{cases} z_2 = \delta_{k_1+1} \\ z_2 = \delta_{k_1+2} \\ z_2 = \delta_{k_1+3} \\ \vdots \\ z_2 = \delta_{k_2} \end{cases} \quad \text{Compare to} \quad \begin{cases} \gamma_{k_1+1} + \epsilon_{(k_1+1)1}z_1 \\ \gamma_{k_1+2} + \epsilon_{(k_1+2)1}z_1 \\ \gamma_{k_1+3} + \epsilon_{(k_1+3)1}z_1 \\ \vdots \\ \gamma_{k_2} + \epsilon_{(k_2)1}z_1 \end{cases}$$

### 3. Algorithm performance evaluation

In this section, we present the complexity of Algorithm 1 and analyze its approximation guarantees.

**Proposition 3.1.** *Algorithm 1 has a complexity of  $O(n^3)$ , and if the fundamental solution set  $\eta$  for the equation  $AU = B$  has been obtained and is in column echelon form, then the complexity will reduce to  $O(mn)$ .*

*Proof.* In Algorithm 1, step 1 involves solving a system of linear equations, which has a complexity of  $O(n^3)$ . Step 8 involves transforming the matrix  $\epsilon$  into column echelon form, which has a complexity of  $O(m^2n)$  where  $m \leq n$ . Step 2 to 7 is  $O(1)$ . Step 9 involves calculating the location of pivots in the column echelon matrix  $\epsilon$ , which has a complexity of  $O(mn)$ . Steps 10 to 24 consist of a nested loop with three layers. However, each element in the matrix  $\epsilon$  is only accessed once, resulting in a complexity of  $O(mn)$ .  $\square$

**Proposition 3.2.** *If a given instance  $\mathcal{I}$  of the min generalized all-ones problem has a solution, the value  $\text{sol}$  of the solution obtained by Algorithm 1 satisfies  $\text{sol} \leq r$ , where  $r$  is the rank of the matrix  $A$ .*

*Proof.* In Equation 6 and 7, if  $\delta_i = \gamma_i$ , then the resulting  $u_i$  will be 0. In each part, we always make more  $u_i$  equal to 0, so each part has at least one  $u_i$  that takes on the value of 0. Furthermore, the rank of  $\eta$  is  $m = n - r$  because  $\eta$  is the fundamental solution set of the system  $AU = B$ . Therefore, at least  $m$  values of  $u_i$  are 0, so  $\sum U \leq n - m = r$ .  $\square$

**Proposition 3.3.** *If a given instance  $\mathcal{I}$  of the min generalized all-ones problem has a solution, the value  $\text{sol}$  of the solution obtained by Algorithm 1 satisfies  $\text{sol} \leq (n + \text{opt})/2$ , where  $\text{opt}$  is the value of an optimal solution of  $\mathcal{I}$ .*

**Algorithm 1:** Approximation Algorithm of Minimum All-Ones Problem

---

**Data:** An adjacency matrix  $A_{n \times n}$  and a initial state  $B_{1 \times n}$   
**Result:** Answer  $U$

```

1  $(\eta, \gamma, m) = \text{solveEquations}(A, B)$ ;
2 if  $m == 0$  and  $\gamma$  is null then
3   | return null;
4 end
5 if  $m == 0$  and  $\gamma$  is not null then
6   | return  $\gamma$ ;
7 end
8  $(P, \epsilon, Q) = \text{matrixEchelon}(\eta, \gamma)$ ;
9  $K = \text{calculatePart}(\epsilon)$ ;
10 for  $i$  from 1 to  $m$  do
11   |  $cnt, tmp = 0$ ;
12   | for  $j$  from  $K[i - 1] + 1$  to  $K[i]$  do
13     | for  $p$  from 1 to  $i - 1$  do
14       | |  $tmp = tmp \oplus (\epsilon[j][p] * X[p])$ 
15     | end
16     |  $cnt = cnt + (tmp \oplus \gamma[j])$ ;
17   | end
18   | if  $cnt \leq K[i] - K[i - 1] / 2$  then
19     | |  $X[i] = 0$ ;
20   | end
21   | else
22     | |  $X[i] = 1$ ;
23   | end
24 end
25  $U = P * (\epsilon * X + \gamma)$ ;
26 return  $U$ ;

```

---

*Proof.* In the Subsection 2.2, we partitioned the matrix  $\eta$  into  $m + 1$  parts and proved that for the 1 to  $m$  parts,  $\sum_{k_i+1}^{k_{i+1}} u \leq (k_{i+1} - k_i) / 2$ . Only the 0th part remains to be discussed. The 0th part is quite special in that it contains no variables, only differing in the value of  $\gamma$ . Let the number of 0's in  $\gamma$  in the 0th part be  $g_0$  and the number of 1's be  $g_1$ .  $g_0$  indicates that the switch at that point must not be pressed; otherwise, the conditions for the all-ones problem cannot be satisfied. Similarly,  $g_1$  indicates that the switch must be pressed. Now we have:

$$\text{sol} \leq g_1 + (n - g_1 - g_0) / 2 = (n + g_1 - g_0) / 2 \quad (8)$$

Then add the parameter opt. We can easily prove that  $\text{sol} \geq \text{opt} \geq g_1$ , because the switches for these points must be pressed in any case. So we have

$$g_1 \leq \text{opt} \leq \text{sol} \leq (n + g_1 - g_0) / 2 \quad (9)$$

Next, we will bound sol by replacing  $g_1$  with opt and  $g_0$  with 0, resulting in the following expression:

$$\text{sol} \leq (n + \text{opt}) / 2 \quad (10)$$

□

#### 4. Conclusion and future work

This article presents an approximation algorithm for the min generalized all-ones problem on arbitrary graphs, making it possible to process the problem in batches. The algorithm has a complexity of  $O(n^3)$ . If the equation

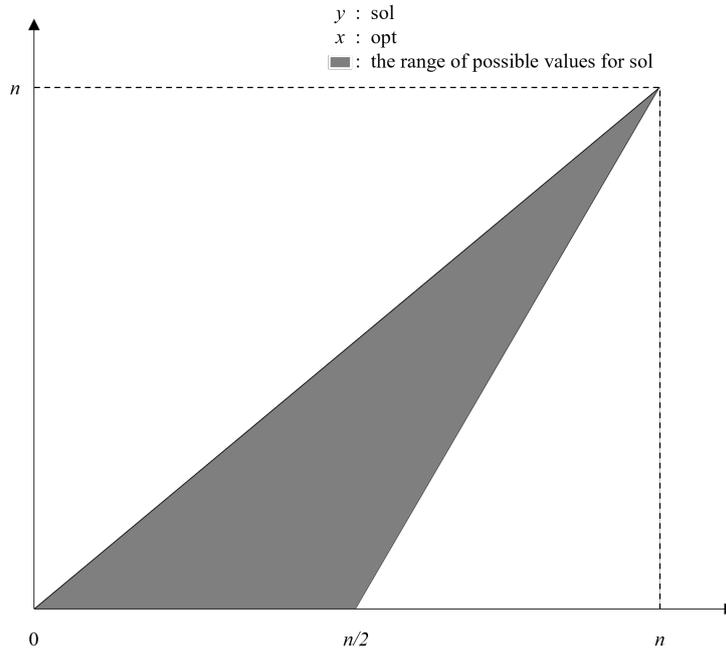


Figure 1. The range of possible values for sol

system  $AU = B$  has been solved and the solution is in column echelon form, the complexity will be reduced to  $O(n(n-r))$ , which is the lowest complexity for general graphs. The upper bound of the solution value sol obtained by this algorithm satisfies the inequality  $\text{sol} \leq (n + \text{opt})/2$  and  $\text{sol} \leq r$ . This ensures that the obtained solution, as shown in Figure 1, is always the optimal half.

In future work, there still remain two questions to be solved. One of them is whether there is a polynomial-time algorithm for the min generalized all-ones problem which always finds a solution of size at most  $c \cdot \text{opt}$  for some constant  $c$ ? The other one is whether we can get such an algorithm for the minimum all-colors problem?

## References

- [1] Barua, R., Ramakrishnan, S., 1996.  $\sigma$ -game,  $\sigma^+$ -game and two-dimensional additive cellular automata. *Theoretical Computer Science* 154, 349–366.
- [2] Broersma, H., Li, X., 2007. On the complexity of dominating set problems related to the minimum all-ones problem. *Theoretical Computer Science* 385, 60–70.
- [3] Caro, Y., 1996. Simple proofs to three parity theorems. *Ars Comb.* 42.
- [4] Chen, W.Y., Li, X., Wang, C., Zhang, X., 2004. The minimum all-ones problem for trees. *SIAM Journal on Computing* 33, 379–392.
- [5] Cowen, R., Hechler, S., Kennedy, J., Ryba, A., 1999. Inversion and neighborhood inversion in graphs. *Graph Theory Notes of New York* 37, 37–41.
- [6] Dodis, Y., Winkler, P., 2001. Universal configurations in light-flipping games, in: Kosaraju, S.R. (Ed.), *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms*, January 7-9, 2001, Washington, DC, USA, ACM/SIAM. pp. 926–927.
- [7] Eriksson, H., Eriksson, K., Sjöstrand, J., 2001. Note on the lamp lighting problem. *Advances in Applied Mathematics* 27, 357–366.
- [8] Fleischer, R., Yu, J., 2013. A survey of the game lights out! *Space-Efficient Data Structures, Streams, and Algorithms: Papers in Honor of J. Ian Munro on the Occasion of His 66th Birthday*, 176–198.
- [9] Galvin, F., 1989. Solution to problem 88-8. *Math. Intelligencer* 11, 31–32.
- [10] Goldwasser, J., Klostermeyer, W., Trapp, G., 1997. Characterizing switch-setting problems. *Linear and Multilinear Algebra* 43, 121–136.
- [11] Li, X., Wang, C., Zhang, X., 2008. The general  $\sigma$  all-ones problem for trees. *Discrete Applied Mathematics* 156, 1790–1801.
- [12] Li, X., Zhang, X., 2005. New versions of the all-ones problem. *arXiv preprint math/0512011*.
- [13] Lossers, O., 1993. Solution to problem 10197 [1992, 162]-an all-ones problem. *American Mathematical Monthly* 100, 806–807.
- [14] Lu, Y., Li, Y., 2007. The minimum all-ones problem for graphs with small treewidth, in: *Combinatorial Optimization and Applications: First International Conference, COCOA 2007, Xi'an, China, August 14-16, 2007. Proceedings 1*, Springer. pp. 335–342.
- [15] Manuel, P., Rajasingh, I., Rajan, B., Prabha, R., 2012. The all-ones problem for binomial trees, butterfly and benes networks. *International Journal of Mathematics and soft computing* 2, 1–6.

- [16] Sutner, K., 1988a. Additive automata on graphs. *Complex Systems* 2, 649–661.
- [17] Sutner, K., 1988b. Problem 88-8. *The Mathematical Intelligencer* 10.
- [18] Sutner, K., 1989. Linear cellular automata and the Garden-of-Eden. *The Mathematical Intelligencer* 11, 49–53.
- [19] Sutner, K., 1990. The  $\sigma$ -game and cellular automata. *The American Mathematical Monthly* 97, 24–34.
- [20] Zhang, X., Wang, C., 2019. Solutions to all-colors problem on graph cellular automata. *Complexity* 2019. Article ID 3164692.