



HAL
open science

Gathering algorithms on paths under interference constraints

Jean-Claude Bermond, Ricardo Correa, Min-Li Yu

► **To cite this version:**

Jean-Claude Bermond, Ricardo Correa, Min-Li Yu. Gathering algorithms on paths under interference constraints. CTAC06, May 2006, Roma, Italy. pp.115-126. hal-00512286

HAL Id: hal-00512286

<https://hal.science/hal-00512286>

Submitted on 29 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gathering Algorithms on Paths under Interference Constraints

Jean-Claude Bermond^{*1}, Ricardo Corrêa^{**2}, and Minli Yu^{***3}

¹ MASCOTTE, joint project CNRS-INRIA-UNSA, 2004 Route des Lucioles, BP 93,
F-06902 Sophia-Antipolis, France

`bermond@sophia.inria.fr`

² Universidade Federal do Ceará, Departamento de Computação, Campus do Pici,
Bloco 910, 60455-760 Fortaleza, CE, Brazil

`correa@lia.ufc.br`

³ University College of the Fraser Valley, Department of Mathematics and Statistics,
Abbotsford, BC, Canada V2S 4N2

`joseph.yu@ucfv.ca`

Abstract. We study the problem of gathering information from the nodes of a multi-hop radio network into a pre-determined destination node under interference constraints which are modeled by an integer $d \geq 1$, so that any node within distance d of a sender cannot receive calls from any other sender. A set of calls which do not interfere with each other is referred to as a round. We give algorithms and lower bounds on the minimum number of rounds for this problem, when the network is a path and the destination node is either at one end or at the center of the path. The algorithms are shown to be optimal for any d in the first case, and for $1 \leq d \leq 4$, in the second case.

Key words: Gathering, interference, multi-hop radio network, path.

1 Introduction

1.1 Problem statement

The problem that we consider in this paper was motivated by a question asked by FRANCE TELECOM about “how to provide Internet to villages” (see [3]) and is related to the following scenario. Suppose we are given a set of communication devices (for instance, network interfaces that connect computers to the Internet) which are placed in houses in a village. They require access to a gateway (for instance, a satellite antenna) to send and receive data through a multi-hop wireless network. The nodes communicate exclusively by means of radio transmissions,

* Partially supported by the CRC CORSO with France Telecom, by the european FET project AEOLUS, by the cooperation with Brazil project REGAL and by the INRIA associated team RESEAUXCOM with S.F.U.

** Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, Brazil.

*** Partially supported by the INRIA associated team RESEAUXCOM.

referred to as *calls*. A call involves two nodes, the sender and the receiver, and is subject to the following constraints:

Reachability constraint: since every node has limited transmission power, the receiver must be close enough to the sender.

Interference constraint: unlike wired networks, a call can interfere with reception at certain nodes beyond the receiver. A node that is within interference distance of one call cannot be the receiver of another call.

Considering these two constraints, a message transmitted in a call can only be properly received if the receiver is reachable from the sender and there is no interference by another message being simultaneously transmitted. In this context, we study the following problem:

***t*-gathering problem:** suppose each node of the network has a piece of information. The *t*-gathering problem consists of collecting (gathering) all these pieces of information into a special node *t*, called the *gathering node*.

In this paper, we propose solutions to this problem for the particular case of a path. Before going into details about our results, let us introduce the mathematical formulation of the problem.

1.2 Model and assumptions

According to the model adopted in [1], the network described above is represented by an undirected graph $G = (V, E)$, where V is the set of nodes, each of them representing a communication device that is able to send and receive messages, and E is the set of edges, representing the possible communications. Let $d_G(s, r)$ indicate the distance in G , defined as the length of a shortest path between s and r . We model the reachability and the interference constraints by two positive integers $d_T \geq 1$ and $d_I \geq d_T$. A node $r \in V$ is reachable from $s \in V$ if and only if $d_G(r, s) \leq d_T$. An important case is $d_T = 1$, which means that a node is able to communicate only with its neighbors in the graph. The second parameter d_I models the interference constraint as follows: if s sends a message to r , then no node $w \in V$ such that $d_G(s, w) \leq d_I$ can receive another message.

Denote by $X_{s,r}$ a call where a node $s \in V$ sends message X to node $r \in V$. We assume that every call takes one unit of time (or one slot) to transmit one unit-length message. Two calls are said to be *compatible* if they do not interfere with each other. More precisely, two calls X_{s_1, r_1} and Y_{s_2, r_2} are compatible if $d_G(s_1, r_2) > d_I$ and $d_G(s_2, r_1) > d_I$. Observe that a consequence of the interference constraint is that $s_1 \neq r_2$ and $s_2 \neq r_1$, which implies that a node is not able to send and receive messages simultaneously. A *round* is a set of compatible calls, whereas an *algorithm* is a sequence of rounds.

In this paper, our aim is to find a *t*-gathering protocol using a minimum number of rounds in the specific case where G is a path. In fact, this stems from the assumption that the village consists of one main street. To our great surprise, the gathering problem is not so simple in this case, if one wants to obtain an exact optimal algorithm.

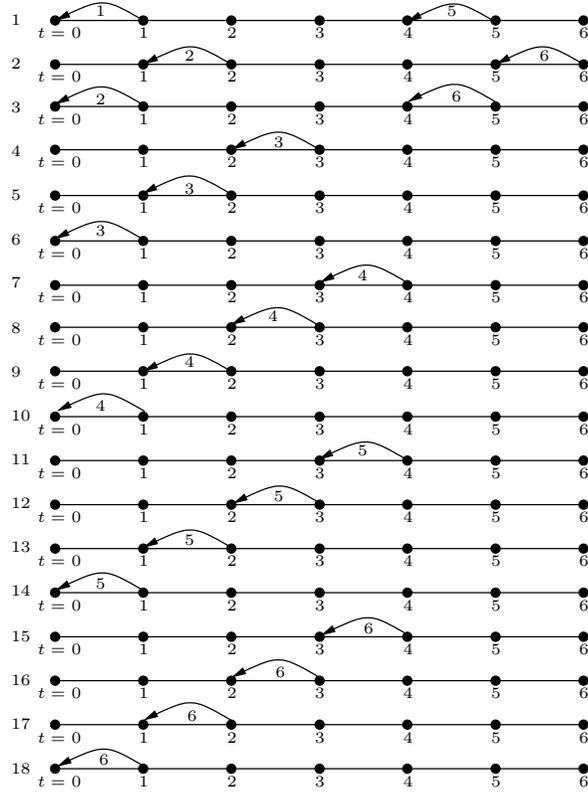


Fig. 1. Algorithm for a graph of 7 nodes and $d_I = 2$.

In the algorithm shown in Figure 1 (where $d_T = 1$ and $d_I = 2$), the call $1_{1,0}$ interferes with $4_{4,3}$ because $d_G(1,3) \leq 2 = d_I$. This is the reason why they do not appear in the same round. On the other hand, the calls $1_{1,0}$ and $5_{5,4}$ are compatible. All the rounds shown in the figure consist of a single call or two compatible calls. It will be shown later that the algorithm consisting of this sequence of 18 rounds is in fact optimal.

A final remark with respect to the model adopted in this paper is that another possibility would be to represent the radio devices as nodes in the plane, and to state the reachability and interference constraints according to the euclidean distances. However, since we only consider paths, the two models are equivalent.

1.3 Related work

The broadcasting and gossiping problems in radio networks with $d_T = d_I = 1$ are studied in [6, 8] and [4, 5, 7], respectively. Note that, in a broadcast, the same information has to be transmitted to all the other nodes and therefore flooding

techniques can be used. When a node needs to send different messages to the other nodes of the network, we have the *personalized broadcasting problem*, which is equivalent to the gathering problem as it suffices to reverse the calls in the solution of one problem to get a solution of the other one.

Some gathering problems have already been studied. For example, in [2] optimal solutions are provided for the two-dimensional square grid. In [1], general results are given (with the possibility of various sizes of messages in each node); in particular, an algorithm working on any graph with an approximation factor of at most 4 is presented. It is also shown that the problem of finding an optimal gathering algorithm (one that uses a minimum number of rounds) does not admit a Fully Polynomial Time Approximation Scheme if $d_I > d_T$, unless $P=NP$, and is NP-HARD if $d_I = d_T$. Another related model can be found in [9], where the authors study the case in which steady-state flow demands between each pair of nodes have to be satisfied.

1.4 Our results

The results of this paper are presented in the remaining sections as follows. We assume $d_T = 1$ and denote d_I simply by d . In Section 2, we deal with the case where the gathering node is at one end of the path. This case is simple and we describe an optimal algorithm. In Section 3, we consider the case where the gathering node is at the center of the path with $2p + 1$ nodes. We first give a lower bound (this bound is also valid for the flow model of [9]). Then, we design an algorithm which meets the lower bound for $p \leq p_1 = d + 1 + \frac{k(k+1)}{2}$. In the next subsection, we show how to strengthen the preceding lower bound. In fact, we show that, for $p \geq d + 2$, any algorithm for the path with the gathering node at the center needs $2\lfloor(d-1)/2\rfloor + 1$ more rounds than that for the path of length p with the gathering node at one end. Our algorithm meets this strengthened lower bound for $d = 1, 2, 3, 4$ (which correspond to the practical cases). We close the paper with some concluding remarks in Section 4.

2 Paths with the gathering node at one end

Let Π_p be the path of length p (consisting of p edges and $p + 1$ nodes). The nodes are denoted $0, 1, 2, \dots, p$, and the edges are of the type $(i, i - 1)$. Assume that the gathering node is $t = 0$. To simplify the notation, we denote the call $X_{i, i-1}$ by X_i and the minimum number of rounds by $g_d(p)$. The recursive scheduler depicted in Algorithm 1 is used to prove the result below (see Figure 1 for an example with $p = 6$ and $d = 2$).

Theorem 1. *For the path Π_p and $d \geq 1$,*

$$g_d(p) = \begin{cases} p(p+1)/2, & \text{if } p \leq d+1 \\ (d+2)(2p-d-1)/2, & \text{otherwise} \end{cases}$$

Algorithm 1 Gathering scheduler on Π_p

```
1: if  $p > 0$  then
2:   Call recursively the gathering scheduler on  $\Pi_{p-1}$ 
3:   for  $j \leftarrow p, \dots, d+3$  do
4:     Let  $x = p - (d+2)$  and  $i = j - (d+2)$ 
5:     Schedule  $P_j$  in the same round as  $X_i$ 
6:   for  $j \leftarrow \min\{p, d+2\}, \dots, 1$  do
7:     Schedule  $P_j$  in a new round
```

Proof. The upper bound is given by Algorithm 1. Suppose that all calls involving messages smaller than P are scheduled in existing rounds as indicated in line 2. The calls involving the message P leaving a node $j \geq d+3$ are scheduled as indicated in lines 3-5. New rounds are then created for the remaining calls. Hence, proceeding by recurrence, we find that

$$g_d(p) \leq \sum_{i=1}^p \min\{i, d+2\},$$

which gives the upper bound of the theorem.

To show the lower bound, note that the information X of a node x must be transmitted via the calls X_j , $1 \leq j \leq x$. Furthermore, the interference constraint implies that at most one call X_j , for $1 \leq j \leq d+2$, can occur in a round. So, to send X , for $1 \leq x \leq d+1$, from node x to the gathering node, we need at least x rounds, all containing a call in the interval $[0, d+2]$. It follows that the rounds used for two distinct nodes x and x' , $1 \leq x, x' \leq d+1$, are disjoint. Therefore, if $p \leq d+1$, then at least $1 + 2 + \dots + p = p(p+1)/2$ rounds are required.

Now, consider $x \geq d+2$. To bring X to the gathering node, all the $d+2$ calls X_j , $1 \leq j \leq d+2$, for X must occur at different rounds. Moreover, these rounds must be different from those used for Y_j , $X \neq Y$ and $1 \leq j \leq d+1$. Consequently, at least $(d+2)[p - (d+1)]$ calls are required for X , $d+2 \leq x \leq p$, thus we have the lower bound for the case $p \geq d+2$. \square

3 Paths with the gathering node at the center

3.1 Preliminaries

Let us denote by $\Pi_{-p}\Pi_p$ the path of length $2p$ with the $2p+1$ nodes $-p, -(p-1), \dots, -1, 0, 1, 2, \dots, p$, and with edges $(-i, -(i-1))$ and $(i, i-1)$. Assume that the gathering node is $t = 0$. We write $d = 2k+1$ or $d = 2k+2$, depending whether d is odd or even, respectively, and denote the minimum number of rounds by $g_d(p, p)$. Clearly, $g_d(p, p) \geq g_d(p)$ since $\Pi_{-p}\Pi_p$ is composed by two symmetric paths of length p . However, in order to attain any tight lower bound, it often requires the calls on one side of the paths to be paired with calls on the other side. When p is small, all the calls are incompatible and every algorithm is optimal.

Proposition 1. *If $p \leq k + 1$, then $g_d(p, p) = 2g_d(p) = p(p + 1)$.*

In the sequel, we consider $p > k + 1$, in which case an optimal algorithm requires some compatible calls to be appropriately paired. Special attention needs to be devoted to the *critical calls*, that is the calls in the *critical interval* $[-(d + 2), d + 2]$ of nodes. A round is called an *obstruction* if it contains only one critical call. Like in the previous section, write X_i and $-X_i$ for the calls $X_{i, i-1}$ and $-X_{-i, -(i-1)}$, respectively.

In the critical interval, two calls X_i and Y_j interfere, and so do two calls $-X_i$ and $-Y_j$. Moreover, two calls $-X_i$ and Y_j interfere if and only if $i + j \leq d + 1$ because the distance between $-i$ and $j - 1$ is $i + j - 1$. For example, a call $-X_1$ can be paired only with calls Y_{d+1} or Y_{d+2} . Consequently, every round contains at most two critical calls and, in addition, a round contains two critical calls $-X_i$ and Y_j only if $i + j \geq d + 2$.

Let

$$A^+ = \bigcup_{i=1}^{k+1} \{X_i \mid i \leq x \leq p\} \text{ and } A^- = \{-X_i \mid X_i \in A^+\}. \quad (1)$$

Observe that these two sets are such that a call in A^+ cannot be paired with any call in A^- . The remaining critical calls define the sets

$$B^+ = \bigcup_{i=d-k+1}^{p'} \{X_i \mid i \leq x \leq p\} \text{ and } B^- = \{-X_i \mid X_i \in B^+\}, \quad (2)$$

where $p' = \min\{p, d + 2\}$. When d is odd, these sets partition the set of possible calls. But when d is even, there are also all the calls $-X_{k+2}$ and X_{k+2} . Observe that two critical calls can be paired only if one of them belongs to neither A^+ nor A^- .

3.2 A lower bound when $p \geq k + 2$

Let us turn our attention to a lower bound which will turn to be optimal when p is not too large.

Theorem 2. $g_d(p, p) \geq p(k + 2) + \lfloor d/2 \rfloor (p - k - 1)$.

Proof. To obtain the lower bound, we count the maximum number M of pairs $\{-X_i, Y_j\}$ which can be formed and we get $g_d(p, p) \geq 2g_d(p) - M$. It can be checked from (1) and (2) that $|A^-| \geq |B^+|$ and $|A^+| \geq |B^-|$. Since the calls of A^- (resp. A^+) can only be paired with calls in B^+ (resp. B^-), the maximum number of pairs involving calls in A^+ , A^- , B^+ and B^- occurs when all calls in B^+ and B^- are paired with A^- and A^+ , respectively. In addition, if d is even, we can also pair $-X_{k+2}$ with X_{k+2} , for $k + 2 \leq x \leq p$. Thus, $M = |B^+| + |B^-| + (d - 2k - 1)(p - k - 1)$.

First consider the case $p \leq d + 1$. Then $g_d(p) = \frac{p(p+1)}{2}$ by Theorem 1. If d is odd, then $|B^+| = |B^-| = \sum_{k+2 \leq i \leq p} p - i + 1 = \frac{(p-k-1)(p-k)}{2}$ and so

$2g_d(p) - M = p(p+1) - (p-k-1)(p-k) = p(k+2) + k(p-k-1)$. Otherwise, d is even, and $|B^+| = |B^-| = \sum_{k+3 \leq i \leq p} p-i+1 = \frac{(p-k-2)(p-k-1)}{2}$. This leads to $M = (p-k-1)^2$ and $2g_d(p) - M = p(p+1) - (p-k-1)^2 = p(k+2) + (k+1)(p-k-1)$.

For the case $p \geq d+2$, we have to use the value of $g_d(p)$ given in Theorem 1 and observe that $g_d(p)$ increases by $d+2$ as p increases by 1. To compute M , we also observe that now $p' = d+2$. So, when p increases by 1, $|B^+|$ increases by $k+2$ and M by $d+3$. Therefore, $2g_d(p) - M$ increases by $d+1 = k + \lfloor d/2 \rfloor$, ending the proof. \square

3.3 An optimal algorithm

In this subsection, we present an algorithm whose number of rounds meets the lower bound described in the previous subsection. This algorithm corresponds to the sequence of rounds obtained with Algorithm 2. In the next subsection, we will show that this algorithm also gives optimal solution for larger values of p and $1 \leq d \leq 4$.

Algorithm 2 Gathering scheduler for $\Pi_{-p}\Pi_p$

```

1: if  $p > 0$  then
2:   Call recursively the gathering scheduler for  $\Pi_{-(p-1)}\Pi_{p-1}$ 
3:   for  $j \leftarrow p, \dots, d+3$  do
4:     Let  $x = p - (d+2)$  and  $i = j - (d+2)$ 
5:     Schedule  $P_j$  in the same round as  $X_i$ 
6:     Schedule  $-P_j$  in the same round as  $-X_i$ 
7:   for  $j \leftarrow \min\{p, d+2\}, \dots, k+2$  do
8:     if There is obstruction compatible with  $P_j$  then
9:       Schedule  $P_j$  in the smallest round that is compatible with  $P_j$ 
10:    else
11:      Schedule  $P_j$  in a new round
12:    Schedule  $-P_j$  in the smallest round that is compatible with  $-P_j$ 
13:  for  $j \leftarrow \min\{p, k+1\}, \dots, 1$  do
14:    Schedule  $P_j$  in a new round
15:    Schedule  $-P_j$  in a new round

```

Algorithm 2 schedules the calls in a sequence of pairs of symmetric rounds in such a way that, if a pair of compatible critical calls $\{X_i, -Y_j\}$, with $x \neq y$, is scheduled in a certain round, then the round immediately after consists of the symmetric counterpart $\{-X_i, Y_j\}$. Similarly, if a round consists of a single positive call X_i , the next round consists of the single negative call $-X_i$. The algorithm for $d = 3$ and $d = 4$ are illustrated in Table 1 and 2, respectively.

The rounds in Algorithm 2 are scheduled recursively in the sense that the rounds involving the calls P_j and $-P_j$, for all $j \in \{1, 2, \dots, p\}$, are scheduled after all the calls associated with the path consisting of $p-1$ positive and negative

Round	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$	$p = 10$	$p = 11$	
1	1 ₁			-4 ₄		6 ₆			-9 ₉		11 ₁₁	
3		2 ₂	-3 ₃				7 ₇	-8 ₈				
5		2 ₁			-5 ₅		7 ₆			-10 ₁₀		
7			3 ₂	-4 ₃				8 ₇	-9 ₈			
9			3 ₁		-5 ₄			8 ₆		-10 ₉		
11				4 ₂	-5 ₃			9 ₇	-10 ₈			
13				4 ₁		-6 ₅		9 ₆			-11 ₁₀	
15					5 ₂	-6 ₄				10 ₇	-11 ₉	
17					5 ₁		-7 ₅			10 ₆		
19						{-6 ₃ , 6 ₃ }						
20						6 ₂	-7 ₄					
22						6 ₁		-8 ₅				11 ₆
24						{-7 ₃ , 7 ₃ }						
25							7 ₂	-8 ₄				
27							7 ₁		-9 ₅			

Table 1. Pairs and obstructions in the rounds derived from Algorithm 2 for $d = 3$. For every round shown in the table but those between horizontal lines, the algorithm also includes its symmetric counterpart.

nodes are scheduled in line 2. This is done without modifying the order of rounds, but only by including the new calls in existing rounds, when possible, or creating new pairs and obstructions. In addition, the new calls are scheduled greedily in lines 9 and 12. More precisely, for the new calls outside the critical interval, they are included in existing rounds. Then, the critical calls are handled. First, the call $P_{\min\{p, d+2\}}$ is paired with the first available and compatible obstruction $-X_i$. Next, the symmetric counterpart is created by pairing $-P_{\min\{p, d+2\}}$ with X_i . The call $P_{\min\{p, d+2\}-1}$ is then paired with the next available and compatible positive obstruction and so on, until $-P_{k+2}$ is paired with either the first available and compatible obstruction (which will turn to be $(P-1)_{d-k}$) or with $-P_{k+2}$.

Algorithm 2 leads to the following upper bound for $g_d(p, p)$.

Theorem 3. $g_d(p, p) \leq p(k+2) + \lfloor d/2 \rfloor (p-k-1) + \max\{0, p-p_1\}$, for $p \geq k+1$, where $p_1 = d + 1 + \frac{k(k+1)}{2}$.

Sketch of the proof. To prove the theorem, we count the number $r_d(p)$ of rounds scheduled with Algorithm 2. First let us consider the odd case $d = 2k + 1$. The proof is by induction on p . We indicate only half of the rounds (the other being obtained by symmetry) which consist either of an obstruction or of a pair of critical calls exclusively. The calls outside the critical interval are easily handled in lines 5 and 6.

If $p = k + 1$, all rounds are scheduled in lines 13-15. So, $r_d(k + 1) = (k + 1)(k + 2)$. Next, we consider $p \geq k + 2$ and give the sequence A_d^p of obstructions left after line 2 which have to be paired with the sequence

$$\langle -P_\ell, -P_{\ell-1}, \dots, -P_{k+2} \rangle, \ell = \min\{p, d + 2\}.$$

Round	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$
1	1_1				-5_5		7_7		
3		2_2		-4_4				8_8	
5		2_1				-6_6		8_7	
7			$\{-3_3, 3_3\}$						$\{-9_9, 9_9\}$
8			3_2		-5_4				9_8
10			3_1			-6_5			9_7
12				$\{-4_3, 4_3\}$					
13				4_2		-6_4			
15				4_1			-7_6		
17					$\{-5_3, 5_3\}$				
18					5_2		-7_5		
20					5_1			-8_6	
22						$\{-6_3, 6_3\}$			
23						6_2	-7_4		
25						6_1		-8_5	
27							$\{-7_3, 7_3\}$		
28							7_2	-8_4	
30							7_1		-9_6
32								$\{-8_3, 8_3\}$	
33								8_2	-9_5
35								8_1	
37									$\{-9_4, 9_4\}$
38									$\{-9_3, 9_3\}$
39									9_2
41									9_1

Table 2. Similar to Table 1, but for $d = 4$.

The first element of A_d^p is paired with $-P_\ell$, the second with $-P_{\ell-1}$ and so on. When p is large enough (as can be seen in Table 1) the last element of A_d^p is not an obstruction at step $p - 1$ but, in fact, P_{k+2} which is paired with $-P_{k+2}$.

First assume that $p \leq 2k + 2 = d + 1$. It turns out that

$$r_d(p) = r_d(p - 1) + 2(p - |A_d^p|). \quad (3)$$

Let $p = k + 1 + i$, where $1 \leq i \leq k + 1$. Then,

$$A_d^{k+1+i} = \langle (k - i + 2)_{k-i+2}, (k - i + 4)_{k-i+3}, \dots, (k + i)_{k+1} \rangle,$$

which gives $|A_d^p| = p - k - 1$ and $r_d(p) = r_d(p - 1) + 2(k + 1)$.

When $p \geq d + 2$, write $s = k(k - 1)/2$ and let \oplus denote a concatenation of sequences. In the two cases considered in the sequel, we replace $2(p - |A_d^p|)$ by $2(d + 2 - |A_d^p|)$ in (3) and obtain $|A_d^p| = k + 2$, as follows:

1. $2k + 3 \leq p \leq 2k + 2 + s$. Let $p = 2k + 2 + i$, where $1 \leq i \leq s$. The definition of A_d^p in this case is recursive:

$$A_d^{2k+2+i} = \begin{cases} \langle 2_1, 3_1, 4_2 \rangle, & \text{if } p = 5 \text{ and } d = 3 \\ \langle (P-2)_1, (P-1)_2, P_3 \rangle, & \text{if } p \geq 6 \text{ and } d = 3 \\ A_{d-2}^{2k+i} \oplus \langle (2k+1+i)_{k+1} \rangle, & \text{otherwise} \end{cases}$$

The recurrence for the number of elements in A_d^{2k+2+i} is

$$|A_d^p| = \begin{cases} 3, & \text{if } p \geq 5 \text{ and } d = 3 \\ |A_{d-2}^{p-2}| + 1, & \text{otherwise} \end{cases}$$

whose solution gives the desired result.

2. $2k + 3 + s \leq p \leq 3k + 2 + s = p_1$. Let $p = 2k + 2 + s + i$, where $1 \leq i \leq k$. Then, the cardinality of A_d^p follows directly from

$$A_d^{2k+2+i} = \langle (s+k+i)_1, (s+k+1+i)_2, \dots, (s+2k)_{k-i+1} \rangle \oplus \langle (s+2k+1)_{k-i+1}, (s+2k+2)_{k-i+2}, \dots, (s+2k+i+1)_{k+1} \rangle.$$

Finally, assume $p \geq 3k + 3 + s$, in which case we write $p = 3k + 2 + s + i$, where $i \geq 1$. In this case, the last element of A_d^p is P_{k+2} , which is paired with $-P_{k+2}$. Then, $r_d(p) = r_d(p-1) + 2(d+2 - |A_d^p|) + 1$ and

$$A_d^p = \langle (s+2k+i+1)_1, (s+2k+i+2)_2, \dots, (s+3k+i+2)_{k+2} \rangle.$$

Putting the pieces together, we get the recurrence

$$r_d(p) \leq \begin{cases} (k+1)(k+2), & \text{if } p = k+1 \\ r_d(p-1) + d + 1, & \text{if } k+1 < p \leq p_1 \\ r_d(p-1) + d + 2, & \text{if } p > p_1. \end{cases} \quad (4)$$

When d is even ($d = 2k + 2$, illustrated in Table 2), we obtain the result from the case $d - 1$ odd. First, observe that, if $p < 3k + 3 + s$, then $A_{2k+2}^p = A_{2k+1}^{p-1} \oplus \langle P_{k+2} \rangle$. Otherwise, A_d^p includes P_{k+2} and pairs kept from A_{2k+1}^{p-1} and A_{2k+1}^p depending on the call X_{k+2} , where $x = 3k + 3 + s$. A call Y_j is kept from A_{2k+1}^{p-1} if $y < x$ or ($y = x$ and $j > k + 2$), and from A_{2k+1}^p otherwise. The recurrence is then the same as above. The solution of (4) concludes the proof. \square

Combining Theorem 2 and Theorem 3, we get

Theorem 4. $g_d(p, p) = p(k+2) + \lfloor d/2 \rfloor (p - k - 1)$, for $k+1 \leq p \leq p_1$, where $p_1 = d + 1 + \frac{k(k+1)}{2}$.

3.4 A lower bound for $p \geq d + 2$

Both Algorithm 1 and Algorithm 2 have a common property: X_i (resp. $-X_i$) appears in a round occurring before that of X_j (resp. $-X_j$) if $i > j$, and X_i (resp. $-X_i$) appears after Y_i (resp. $-Y_i$) if $x > y$. Indeed, one can easily modify any algorithm in order to satisfy such a property. For this reason, and without loss of generality, we suppose that the obstructions are maximal in the following sense.

Assumption 1 If X_i (resp. $-X_i$) is an obstruction, then the following conditions hold:

1. either $x = p$ or $(X + 1)_i$ (resp. $-(X + 1)_i$) is an obstruction; and
2. either $i = 1$ or $-X_{i-1}$ (resp. $-X_{i-1}$) is an obstruction.

Moreover, the property mentioned above naturally defines a partial order \preceq , illustrated in Figure 2, in which $X_i \preceq Y_j$ if $x \leq y$ and $i \geq j$ and $-X_i \preceq -Y_j$ if $x \leq y$ and $i \geq j$. We may use the notation $X_i \prec Y_j$ when $x \neq y$ or $i \neq j$.

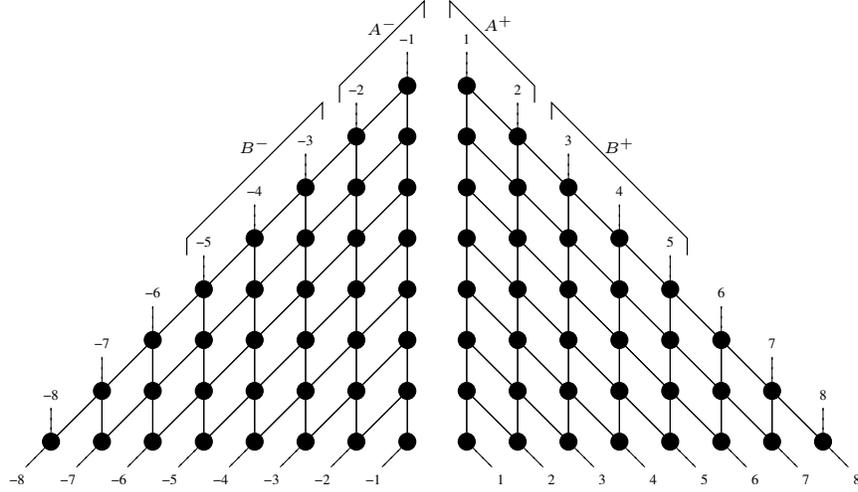


Fig. 2. Partial order \preceq on the calls of an algorithm for $p = 8$ and $d = 3$.

In the rest of this subsection, we present a lower bound for $p \geq d + 2$. This lower bound is based on the minimum number of obstructions that are induced by \preceq . The proofs are omitted due to space limitations.

Lemma 1 (Non-Crossing Lemma). *An algorithm cannot have two different pairs $\{-X_i, W_j\}$ and $\{-Y_k, Z_\ell\}$ with either $-Y_k \prec -X_i$ and $W_j \prec Z_\ell$, or $-X_i \prec -Y_k$ and $Z_\ell \prec W_j$.*

The previous lemma is applied in the results that follow.

Lemma 2. P_1 and $-P_1$ are obstructions.

An immediate consequence is the optimality of Algorithm 2 for $d = 1, 2$. In addition, we have

Lemma 3. *If $p \geq d + 2$, then every algorithm has at least $2k + 1$ positive and $2k + 1$ negative obstructions.*

This leads to our final result

Theorem 5. *If $p \geq d + 2$, then $g_d(p, p) = g_d(p) + 1$, for $d = 1, 2$ and $g_d(p, p) = g_d(p) + 3$, for $d = 3, 4$.*

4 Concluding remarks

We presented algorithms for gathering information from nodes of a path with $2p+1$ nodes to its center or one end node, satisfying reachability and interference constraints. Optimal solutions are given for the first case, and for the second case when the interference distance d is at most 4. We conjecture that the algorithm for the second case is also optimal for larger values of d , leading to $g_d(p, p) = g_d(p) + \frac{(k+1)(k+2)}{2}$, for $p \geq p_1$.

The results in this paper can be extended for more general cases; for instance, when the gathering node is placed anywhere in the path. However, the choice of the center of the path is the one that minimizes the number of rounds.

Acknowledgements

The two last authors thank the MASCOTTE project where some of the research was done during their visits. Also, the authors would like to thank the anonymous referees for the useful suggestions.

References

1. J.-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Hardness and approximation of gathering in static radio networks. In *FAWN06*, Pisa, Italy, March 2006.
2. J.-C. Bermond and J. Peters. Efficient gathering in radio ids with interference. In *AlgoTel'05*, pages 103–106, Presqu'île de Giens, May 2005.
3. P. Bertin, J.-F. Bresse, and B. Le Sage. Accs haut dbit en zone rurale: une solution "ad hoc". *France Telecom R&D*, 22:16–18, 2005.
4. M. Christersson, L. Gasieniec, and A. Lingas. Gossiping with bounded size messages in ad-hoc radio networks. In *Proceedings of ICALP'02*, volume 2380 of *LNCS*, pages 377–389. Springer-Verlag, 2002.
5. M. Chrobak, L. Gasieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. *Journal of Algorithms*, 43(2):177–189, 2002.
6. M. L. Elkin and G. Kortsarz. Logarithmic inapproximability of the radio broadcast problem. *Journal of Algorithms*, 52(1):8–25, 2004.
7. I. Gaber and Y. Mansour. Centralized broadcast in multihop radio networks. *Journal of Algorithms*, 46(1):1–20, 2003.
8. L. Gasieniec and I. Potapov. Gossiping with unit messages in known radio networks. In *Proceedings of the IFIP 17th World Computer Congress*, pages 193–205. Kluwer, B.V., 2002.
9. R. Klasing, N. Morales, and S. Pérennes. On the complexity of bandwidth allocation in radio networks with steady traffic demands. Technical report, INRIA Research Report RR-5432 and I3S Research Report I3S/RR-2 004-40-FR, 2004.