Data Gathering and Personalized Broadcasting in Radio Grids with Interferences ☆

Jean-Claude Bermond[a,b,*], Bi Li[b,a,c], Nicolas Nisse[b,a], Hervé Rivano[d], Min-Li Yu[e]

[a] *Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, Sophia Antipolis, France*
[b] *Inria, France*
[c] *Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China*
[d] *SWING, INRIA Rhônes Alpes and CITI Lab (INSA Lyon), Lyon, France*
[e] *University of the Fraser Valley, Dpt of Maths and Statistics, Abbotsford, BC, Canada*

Abstract

In the *gathering* problem, a particular node in a graph, the *base station*, aims at receiving messages from some nodes in the graph. At each step, a node can send one message to one of its neighbors (such an action is called a *call*). However, a node cannot send and receive a message during the same step. Moreover, the communication is subjet to interference constraints, more precisely, two calls interfere in a step, if one sender is at distance at most d_I from the other reciever. Given a graph with a base station and a set of nodes having some messages, the goal of the gathering problem is to compute a schedule of calls for the base station to receive all messages as fast as possible, i.e., minimizing the number of steps (called *makespan*). The gathering problem is equivalent to the *personalized broadcasting* problem where the base station has to send messages to some nodes in the graph, with same transmission constraints.

In this paper, we focus on the gathering and personalized broadcasting problem in grids. Moreover, we consider the non-buffering model: when a node receives a message at some step, it must transmit it during the next step. In this setting, though the problem of determining the complexity of computing the optimal makespan in a grid is still open, we present linear (in the number of messages) algorithms that compute schedules for gathering with $d_I \in \{0,1,2\}$. In particular, we present an algorithm that achieves the optimal makespan up to an additive constant 2 when $d_I = 0$. If no messages are "close" to the axes (the base station being the origin), our algorithms achieve the optimal makespan up to an additive constant 1 when $d_I = 0$, 4 when $d_I = 2$, and 3 when both $d_I = 1$ and the base station is in a corner. Note that, the approximation algorithms that we present also provide approximation up to a ratio 2 for the gathering with buffering. All our results are proved in terms of personalized broadcasting.

Keywords: gathering, personalized broadcasting, grid, interferences, radio networks

1. Introduction

1.1. Problem, model and assumptions

In this paper, we study a problem that was motivated by designing efficient strategies to provide internet access using wireless devices [8]. Typically, several houses in a village need access to a gateway (for example a satellite antenna) to transmit and receive data over the Internet. To reduce the cost of the transceivers, multi-hop wireless relay routing is used. We formulate this problem as gathering information in a Base Station (denoted by BS) of a wireless multi-hop network when interferences constraints are present. This problem is also known as data collection and is particularly important in sensor networks and access networks.

☆This work is partially supported by Project STREP FP7 EULER, ANR Verso project Ecoscells, ANR GRATEL and Region PACA.

*Corresponding author

Email addresses: jean-claude.bermond@inria.fr (Jean-Claude Bermond), bi.li@inria.fr (Bi Li), nicolas.nisse@inria.fr (Nicolas Nisse), herve.rivano@inria.fr (Hervé Rivano), joseph.yu@ufv.ca (Min-Li Yu)

Transmission model. We adopt the network model considered in [2, 4, 9, 11, 16]. The network is represented by a node-weighted symmetric digraph $G = (V, E)$, where V is the set of nodes and E is the set of arcs. More specifically, each node in V represents a *device* (sensor, station, ...) that can transmit and receive data. There is a special node $BS \in V$ called the *Base Station (BS)*, which is the final destination of all data possessed by the various nodes of the network. Each node may have any number of pieces of information, or *messages*, to transmit, including none. There is an arc from u to v if u can transmit a message to v. We suppose that the digraph is symmetric; so if u can transmit a message to v, then v can also transmit a message to u. Therefore G represents the graph of possible communications. Some authors use an undirected graph (replacing the two arcs (u, v) and (v, u) by an edge $\{u, v\}$). However *calls* (transmissions) are directed: a call (s, r) is defined as the transmission from the node s to node r, in which s is the *sender* and r is the *receiver* and s and r are adjacent in G. The distinction of sender and receiver will be important for our interference model.

Here we will consider grids as they model well both access networks and also random networks [14]. The network is assumed to be synchronous and the time is slotted into *steps*. During each step, a transmission (or a *call*) between two nodes can transport at most one message. That is, a step is a unit of time during which several calls can be done as long as they do not interfere with each other. We suppose that each device is equipped with an half duplex interface: a node cannot both receive and transmit during a step. This models the near-far effect of antennas: when one is transmitting, it's own power prevents any other signal to be properly received. Moreover, we assume that a node can transmit or receive at most one message per step.

Following [11, 12, 15, 16, 18] we assume that no buffering is done at intermediate nodes and each node forwards a message as soon as it receives it. One of the rationales behind this assumption is that it frees intermediate nodes from the need to maintain costly state information and message storage.

Interference model. We use a binary asymmetric model of interference based on the distance in the communication graph. Let $d(u, v)$ denote the distance, that is the length of a shortest directed path, from u to v in G and d_I be an nonnegative integer. We assume that when a node u transmits, all nodes v such that $d(u, v) \leq d_I$ are subject to the interference from u's transmission. We assume that all nodes of G have the same interference range d_I. Two calls (s, r) and (s', r') do not interfere if and only if $d(s, r') > d_I$ and $d(s', r) > d_I$. Otherwise calls interfere (or there is a collision). We will focus on the cases when $d_I \leq 2$. Note that we suppose in this paper $d_I \geq 0$. It implies that a node cannot receive and send simultaneously.

The binary interference model is a simplified version of the reality, where the Signal-to-Noise-and-Interferences Ratio (the ratio of the received power from the source of the transmission to the sum of the thermic noise and the received powers of all other simultaneously transmitting nodes) has to be above a given threshold for a transmission to be successful. However, the values of the completion times that we obtain will lead to lower bounds on the corresponding real life values. Stated differently, if the value of the completion time is fixed, then our results will lead to upper bounds on the maximum possible number of messages that can be transmitted in the network.

Gathering and Personalized broadcasting. Our goal is to design protocols that will efficiently, i.e., quickly, gather all messages to the base station BS subject to these interference constraints. More formally, let $G = (V, E)$ be a connected symmetric digraph, $BS \in V$ and and $d_I \geq 0$ be an integer. Each node in $V \setminus BS$ is assigned a set (possibly empty) of messages that must be sent to BS. A multi-hop schedule for a message consists of the path it must follow to reach BS together with the starting step (because no buffering is allowed, the starting step defines the whole schedule). The *gathering problem* consists in computing a multi-hop schedule for each message to arrive the BS under the constraint that during any step any two calls do not interfere within the interference range d_I. The completion time or *makespan* of the schedule is the number of steps used for all messages to reach BS. We are interested in computing the schedule with minimum makespan.

Actually, we will describe the gathering schedule by illustrating the schedule for the equivalent *personalized broadcasting problem* since this formulation allows us to use a simpler notation and simplify the proofs. In this problem, the base station BS has initially a set of *personalized* messages and they must be sent to their destinations, i.e., each message has a personalized destination in V, and possibly several messages may have the same destination. The problem is to find a multi-hop schedule for each message to reach its corresponding destination node under the same constraints as the gathering problem. The completion time or *makespan* of the schedule is the number of steps used for all messages to reach their destination and the problem aims at computing a schedule with minimum makespan. To see that these two problems are equivalent, from any personalized broadcasting schedule, we can always build a gathering schedule with the same makespan, and the other way around. Indeed, consider a personalized broadcasting schedule with makespan $\mathcal{T}$. Any call (s, r) occurring at step k corresponds to a call (r, s) scheduled at step $\mathcal{T} + 1 - k$ in the corresponding gathering schedule. Furthermore, as the digraph is symmetric, if two calls (s, r) and (s', r') do not interfere, then $d(s, r') > d_I$ and $d(s', r) > d_I$; so the reverse calls do not interfere. Hence, if there is an (optimal) personalized broadcasting schedule from BS, then there exists an (optimal) solution for gathering at BS with the same makespan. The reverse also holds. Therefore, in the sequel, we consider the personalized broadcasting problem.

1.2. Related Work

Gathering problems like the one that we study in this paper have received much recent attention. The papers most closely related to our results are [3, 4, 11, 12, 15, 16]. Paper [11] firstly introduced the data gathering problem in a model for sensor networks similar to the one adopted in this paper. It deals with $d_I = 0$ and gives optimal gathering schedules for trees. Optimal algorithms for star networks are given in [16] find the optimal schedule minimizing both the completion time and the average delivery time for all the messages. Under the same hypothesis, an optimal algorithm for general networks is presented in [12] in the case each node has exactly one message to deliver. In [4] (resp [3]) optimal gathering algorithms for tree networks in the same model considered in the present paper, are given when $d_I = 1$ (resp.,$d_I \geq 2$). In [3] it is also shown that the Gathering Problem is NP-complete if the process must be performed along the edges of a *routing tree* for $d_I \geq 2$ (otherwise the complexity is not determined). Furthermore, for $d_I \geq 1$ a simple $(1 + \frac{2}{d_I})$ factor approximation algorithm is given for general networks. In slightly different settings, in particular the assumption of directional antennas, the problem has been proved NP-hard in general networks [17]. The case of *open-grid* where BS stands at a corner and no messages have destinations in the first row or first column, called axis in the following, is considered in [15], where a 1.5-approximation algorithm is presented.

Other related results can be found in [1, 2, 6, 7, 10] (see [9] for a survey). In these articles data buffering is allowed at intermediate nodes, achieving a smaller makespan. In [2], a 4-approximation algorithm is given for any graph. In particular the case of grids is considered in [6], but with exactly one message per node. Another related model can be found in [13], where steady-state (continuous) flow demands between each pair of nodes have to be satisfied, in particular, the authors also study the gathering in radio grid networks.

1.3. Our results

In this paper, we propose algorithms to solve the personalized broadcasting problem (and so the equivalent gathering problem) in a grid with the model described above (synchronous, no buffering, one message transmission per step, with an interference parameter d_I). Initially all messages stand at the base station BS and each message has a particular destination node (possibly several messages may be sent to the same node). Our algorithms compute in linear time (in the number of messages) schedules with no calls interfering, with a makespan differing from the lower bound by a small additive constant. We first study the basic instance consisting of an open grid where no messages have destination on an axis, with a BS in the corner of the grid and with $d_I = 0$. This is exactly the same case as that considered in [15]. In Section 2 we give a simple lower bound LB. Then in Section 3 we design for this basic instance a linear time algorithm with a makespan at most $LB + 2$ steps, so obtaining a $+2$-approximation algorithm for the open grid, which greatly improves the multiplicative 1.5 approximation algorithm of [15] . Such an algorithm has already been given in the extended abstract [5]; but the one given here is simpler and we can refine it to obtain for the basic instance a $+1$-approximation algorithm. Then we prove in Section 4 that the $+2$-approximation algorithm works also for a general grid where messages can have destinations on the axis again with BS in the corner and $d_I = 0$. Then we consider in Section 5 the cases $d_I = 1$ and 2. We give lower bounds $LB_c(1)$ (when BS is in the corner) and $LB(2)$ and show how to use the $+1$-approximation algorithm given in Section 3 to design algorithms with a makespan at most $LB_c(1) + 3$ when $d_I = 1$ and BS is in the corner , and at most $LB(2) + 4$ when $d_I = 2$; however the coordinates of the destinations have in both cases to be at least 2. In Section 6, we extend our results to the case where BS is in a general position in the grid. In addition, we point out that our algorithms are 2-approximations if the buffering is allowed, which improves the result of [2] in the case of grids with $d_I \leq 2$. Finally, we conclude the paper in Section 7. The main results are summarized in Table 1.

Interference	Additional hypothesis	Performances	
		without buffering	with buffering
$d_I = 0$		+2-approximation	
	no messages on axes	+1-approximation	
$d_I = 1$	BS in a corner and no messages “close” to the axes (see Def. 2)	+3-approx.	$\times 1.5$-approx.
	no messages at distance ≤ 1 from an axis	$\times 1.5$-approximation	
$d_I = 2$	no messages at distance ≤ 1 from an axis	+4-approx.	$\times 2$-approx.

Table 1: Performances of the algorithms designed in this paper. Our algorithms deal with the gathering and personalized broadcasting problems in a grid with arbitrary base station (unless stated otherwise). In this table, $+c$-approximation means that our algorithm achieves an optimal makespan up to an additive constant c. Similarly, $\times c$-approximation means that our algorithm achieves an optimal makespan up to an multiplicative constant c.

2. Notations and Lower bound

In the following, we consider a grid $G = (V, E)$ with a particular node, the base station BS, also called *the source*. A node v is represented by its coordinates (x, y). The source BS has coordinates $(0, 0)$. We define the *axis* of the grid with respect to BS, as the set of nodes $\{(x, y) : x = 0\}$ or $\{(x, y) : y = 0\}$. The distance between two nodes u and v is the length of a shortest directed path in the grid and will be denoted by $d(u, v)$. In particular, $d(BS, v) = |x| + |y|$.

We consider a set of $M > 0$ messages that must be sent from the source BS to some destination nodes. Note that BS is not a destination node. Let $dest(m) \in V$ denote the destination of the message m. We use $d(m) > 0$ to denote the distance $d(BS, dest(m))$. We suppose that the messages are ordered by non-increasing distance from BS to their destination nodes, and we denote this ordered set $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ where $d(m_1) \geq d(m_2) \geq \cdots \geq d(m_M)$. The input of all our algorithms is the ordered sequence $\mathcal{M}$ of messages. For simplicity we suppose that the grid is infinite; however it suffices to consider a grid slightly greater than the one containing all the destinations of messages. Note that our work does not include the case of the paths, already considered in [1, 11, 15].

We will use the name of *open grid* to mean that no messages have destination on an axis that is when all messages have destination nodes in the set $\{(x, y) : x \neq 0 \text{ and } y \neq 0\}$.

Note that in our model the source can send at most one message per step. Given a set of messages that must be sent by the source, a *broadcasting scheme* consists in indicating for each message m the time at which the source sends the message m and the directed path followed by this message. More precisely a broadcasting scheme will be represented by an *ordered sequence* of messages $\mathcal{S} = (s_1, \cdots, s_k)$, where furthermore for each s_i we give the directed path P_i followed by s_i and the time t_i at which the source sends the message s_i. The sequence is ordered in such a way message s_{i+1} is sent after message s_i, that is we have $t_{i+1} > t_i$.

As we suppose there is no buffering, a message m sent at step t_m is received at step $t'_m = l_m + t_m - 1$, where l_m is the length of the directed path followed by the message m. In particular $t'_m \geq d(m) + t_m - 1$. The *completion time or makespan* of a broadcasting scheme is the step where all the messages have arrived at their destinations. Its value is $\max_{m \in \mathcal{M}} l_m + t_m - 1$. In the next proposition we give a lower bound of the makespan:

Proposition 1. *Given the set of messages $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ ordered by non-increasing distance from BS, the makespan of any broadcasting scheme is greater than or equal to $LB = \max_{i \leq M} d(m_i) + i - 1$.*

PROOF. Consider any personalized broadcasting scheme. For $i \leq M$, let t_i be the step where the last message in $(m_1, m_2, \cdots, m_i)$ is sent; therefore $t_i \geq i$. This last message denoted m is received at step $t'_i \geq d(m) + t_i - 1 \geq d(m_i) + t_i - 1 \geq d(m_i) + i - 1$. So the makespan is at least $LB = \max_{i \leq M} d(m_i) + i - 1$. $\square$

Note that this result is valid for any topology (not only grids) since it uses only the fact that the source sends at most one message per step. If there are no interference constraints, in particular if a node can send and receive messages simultaneously, then the bound is achieved by the greedy algorithm where at step i the source sends the message m_i of the ordered sequence $\mathcal{M}$ through a shortest directed path from BS to $dest(m_i)$, i.e. the makespan LB is attained by BS sending all the messages through the shortest paths to their destinations according to the non-increasing distance ordering.

If there are interferences and $d_I > 0$, we will design in Section 5 some better lower bounds. If $d_I = 0$, we will design in the next two sections linear time algorithms with a makespan at most $LB+2$ in the grid with the base station in the corner and a makespan at most $LB + 1$ when furthermore there is no message with a destination node on the axis (open-grid). In case $d_I = 0$ in open grid, our algorithms are simple in the sense that they use only very simple shortest directed paths and that BS never waits.

Example 1. Here, we exhibit examples for which the optimal makespan is strictly larger than LB. In particular, in the case of general grids, $LB + 2$ can be optimal. On the other hand, results of this paper show that the optimal makespan is at most $LB+1$ in the case of open-grids for $d_I = 0$ (Theorem 4) and at most $LB+2$ in general grids for $d_I = 0$ (Theorem 7). In case $d_I = 0$ and in open-grids, our algorithms use shortest paths and the BS sends a message at each step. We also give examples for which optimal makespan cannot be achieved in this setting.

Let us remark that there exist configurations for which no personalized broadcasting protocol can achieve better makespan than $LB + 1$. Figure 1(a) represents such a configuration. Indeed, in Figure 1(a), message m_i has a destination node v_i for $i = 1, 2, 3$ and $LB = 7$. However, to achieve the makespan $LB = 7$ for $d_I = 0$, BS must send the message m_1 to v_1 at step 1 (because v_1 is at distance 7 from BS) and must send message m_2 to v_2 at step 2 (because the message starts after the first step and must be sent to the destination node at distance 6) and these messages should be sent along shortest directed paths. To avoid interferences, the only possibility is that BS sends the first message to node $(0, 1)$, and the second one to the node $(1, 0)$. Intuitively, this is because otherwise the shortest paths followed by first two messages would intersect in such a way that interference cannot be avoided. A formal proof can be obtained from Fact 2 in Section 3.2. But then, if we want to achieve the makespan of 7, BS has to send the message m_3 via node $(0, 1)$ and it will reach v_3 at step 7; but the directed paths followed by m_2 and m_3 need to cross and at this crossing point m_3 arrives at a step where m_2 leaves and so the messages interfere. So BS has to wait one step and sends m_3 only at step 4. Then the makespan is $8 = LB + 1$.

In addition, there are also examples in which BS has to wait for some steps after sending one message in order to reach the lower bound LB for $d_I = 0$. Figure 1(b) represents such an example. To achieve the lower bound 7, BS has to send messages using shortest directed paths firstly to v_1 via $(3, 0)$ and then consecutively sends messages to v_2 via $(0, 4)$ and v_3 via $(2, 0)$. If BS sends message m_4 at step 4, then m_4 will interfere with m_3. But, to avoid this interference, BS can send message m_4 at step 5 and will reach v_4 at step 7.

There are also examples in which no schedule using only shortest directed paths achieves the optimal makespan[1]. For instance, consider the grid with four messages to be sent to $(0, 4), (0, 3), (0, 2)$ and $(0, 1)$ (all on the first column) and let $d_I = 0$ (a more elaborate example with an open-grid is given in Example 6(a)). Clearly, sending all messages through shortest directed paths implies that BS sends messages every two steps. Therefore, it requires 7 steps. On the other hand, the following scheme has makespan 6: send the message to $(0, 4)$ through the unique shortest directed path at step 1; send the message to $(0, 3)$ at step 2 via nodes $(1, 0), (1, 1), (1, 2)(1, 3)$; send the message to $(0, 2)$ through the shortest directed path at step 3 and, finally, send the message to $(0, 1)$ at step 4 via nodes $(1, 0), (1, 1)$. Note that the optimal makespan is in this example $LB + 2$.

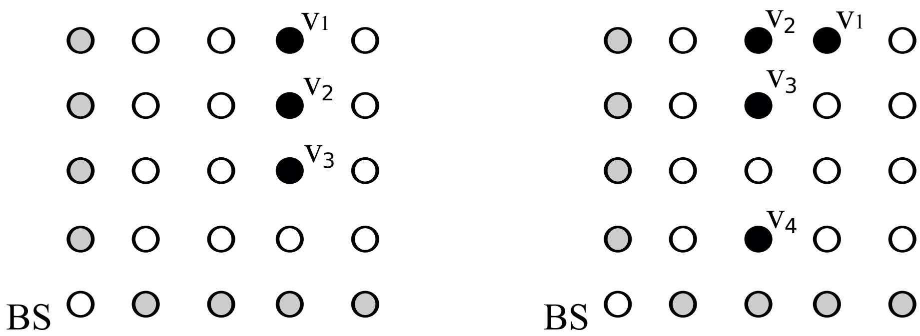
3. Basic instance: $d_I = 0$, open-grid, and BS in the corner

In this section we study simple configurations called *basic instances*. A basic instance is a configuration where $d_I = 0$, messages are sent in the open grid (no destinations on the axis) and BS is in the corner (a node with degree 2 in the grid). We will see that we can find personalized broadcasting algorithms using a basic scheme, where each message is sent via a simple shortest directed path (with one horizontal and one vertical segment) and where the source sends a message at each step (it never waits) and achieving a makespan of at most $LB + 1$.

3.1. Basic schemes

A message is said to be sent *horizontally* to its destination $v = (x, y)$ $(x > 0, y > 0)$, if it goes first horizontally then vertically, that is if it follows the shortest directed path from BS to v passing through $(x, 0)$. Correspondingly,

[1]The authors would like to thanks Prof. Frédéric Guinand who raised this question.



(a) Configuration when the trivial lower bound cannot be achieved.

(b) BS has to wait for one step to achieve the trivial lower bound.

Figure 1: Two particular configurations

the message is sent *vertically* to its destination $v = (x, y)$, if it goes first vertically then horizontally, that is if it follows the shortest directed path from BS to v passing through $(0, y)$. We will use the notation **a message is sent in direction** D, where $D = H$ (for horizontally) (resp. $D = V$ (for vertically)) if it is sent horizontally (resp. vertically). Also, $\bar{D}$ **will denote the direction different from** D that is $\bar{D} = V$ (resp. $\bar{D} = H$) if $D = H$ (resp. $D = V$).

Definition 1. [basic scheme] *A basic scheme is a broadcasting scheme where BS sends a message at each step alternating horizontal and vertical sendings. Therefore it is represented by an ordered sequence $\mathcal{S} = (s_1, s_2, \ldots, s_M)$ of the M messages with the properties: message s_i is sent at step i and furthermore, if s_i is sent in direction D, then s_{i+1} is sent in direction $\bar{D}$.*

Notation: Note that, by definition of horizontal and vertical sendings, the basic scheme defined below uses shortest paths. Moreover, as soon as we fix $\mathcal{S}$ and the sending direction D of the first or last message, the directed paths used in the scheme are uniquely determined. Hence, the scheme is characterized by the sequence $\mathcal{S}$ and the direction D. We will use when needed, the notation $(\mathcal{S}, first = D)$ to indicate a basic scheme where the first message is sent in direction D, and the notation $(\mathcal{S}, last = D)$ when the last message is sent in direction D.

3.2. Interference of messages

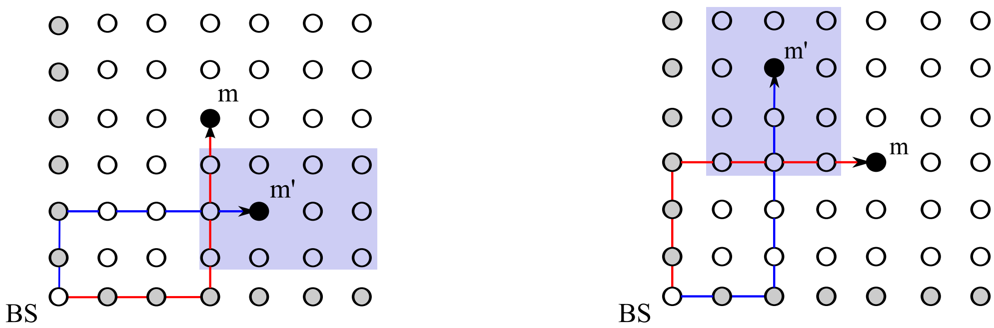
Our aim is to design an *admissible* basic scheme in which the messages are broadcasted without any collisions. The following simple fact shows that we only need to take care of consecutive sendings. In the following, we say that two messages are *consecutive* if the source sends them consecutively (one at step t and the other at step $t + 1$)

Fact 1. *When $d_I = 0$, in any broadcast scheme using only shortest paths (in particular in a basic scheme), only consecutive messages may interfere.*

PROOF. By definition, a basic scheme uses only shortest paths. Let the message m be sent at step t and the message m' at step $t' \geq t + 2$. Let $t' + h$ ($h \geq 0$) be a step such that the two messages have not reached their destinations. As we use shortest directed paths the message m is sent on an arc (u, v) with $d(v, BS) = d(u, BS) + 1 = t' + h - t + 1$, while message m' is sent on an arc (u', v') with $d(v', BS) = d(u', BS) + 1 = h + 1$. Therefore, $d(u, v') \geq t' - t - 1 \geq 1 > 0 = d_I$ and $d(u', v) \geq t' - t + 1 \geq 3 > d_I$. □

We now characterize the situations when two consecutive messages interfere in a basic scheme. For that we use the following notation:

Notation: In the case $d_I = 0$, if BS sends in direction $D \in \{V, H\}$ the message m at step t and sends the message m' in the other direction $\bar{D}$, at step $t' = t + 1$, we will write $(m, m') \in D\bar{D}$ if they do not interfere and $(m, m') \notin D\bar{D}$ if they interfere.



(a) $(m, m') \notin HV$ (b) $(m, m') \notin VH$

Figure 2: Cases of interferences

Fact 2. *Let m and m' be two consecutive messages in a basic scheme. Then, $(m, m') \notin D\bar{D}$ if and only if the paths followed by the messages in the basic scheme intersect in a vertex which is not the destination of m.*

PROOF. Suppose the directed paths intersect in a node v that is not the destination of m. The message m sent at step t has not reached its destination and so leaves the node v at step $t + d(v, BS)$; but the message m' sent at step $t + 1$ arrives at node v at step $t + d(v, BS)$ and therefore the two messages interfere.

Conversely if the two directed paths used for m and m' do not cross then the messages do not interfere. If the paths intersect only in the destination $dest(m)$ of m, then m' arrives in $dest(m)$ one step after m has stopped in $dest(m)$ and so the two messages do not interfere. □

Remark 1. Note that Fact 2 does not hold if we do not impose basic schemes (i.e., this is not true if any shortest paths are considered). Moreover, we emphasize that the two paths may intersect, but the corresponding messages do not necessarily interfere.

In some proofs throughout the paper, we will need to use the coordinates of the messages. Therefore, the following equivalent statement of Fact 2 will be of interest. Let $dest(m) = (x, y)$ and $dest(m') = (x', y')$. Then

- $(m, m') \notin HV$ if and only if $\{x' \geq x$ and $y' < y\}$;
- $(m, m') \notin VH$ if and only if $\{x' < x$ and $y' \geq y\}$;

Figure 2 shows when there are interferences and also illustrates Fact 2 for $D = H$ (resp. V) in case (a) (resp. (b)).

3.3. Basic lemmata

We now prove some simple but useful lemmata.

Lemma 1. *If $(m, m') \notin D\bar{D}$, then $(m, m') \in \bar{D}D$ and $(m', m) \in D\bar{D}$.*

PROOF. By Fact 2, if $(m, m') \notin D\bar{D}$, then the two directed paths followed by m and m' in the basic scheme (in directions D and $\bar{D}$ respectively) intersect in a node different from $dest(m)$. Then, the two directed paths followed by m and m' in the basic scheme (in directions $\bar{D}$ and D respectively) do not intersect. Hence, by Fact 2, $(m, m') \in \bar{D}D$. Similarly, the two directed paths followed by m' and m in the basic scheme (in directions D and $\bar{D}$ respectively) do not intersect. Hence, by Fact 2, $(m', m) \in D\bar{D}$. □

Note that this lemma is enough to prove the multiplicative $\frac{3}{2}$ approximation obtained in [15]. Indeed the source can send at least two messages every three steps, in the order of $\mathcal{M}$. More precisely, BS sends any pair of messages m_{2i-1} and m_{2i} consecutively by sending the first one horizontally and the second one vertically if $(m_{2i-1}, m_{2i}) \in HV$, otherwise sending the first one vertically and the second one horizontally if $(m_{2i-1}, m_{2i}) \notin HV$ (since this implies that $(m_{2i-1}, m_{2i}) \in VH$). Then the source does not send anything during the third step. So we can send $2q$ messages in $3q$ steps. Such a scheme has makespan at most $\frac{3}{2}LB$.

Note that in general, $(m, m') \in D\bar{D}$ does not imply $(m', m) \in \bar{D}D$, namely when the directed paths intersect only in the destination of m which is not the destination of m'.

Lemma 2. *If $(m, m') \in D\bar{D}$ and $(m', m'') \notin \bar{D}D$, then $(m, m'') \in D\bar{D}$.*

PROOF. By Fact 2, $(m, m') \in D\bar{D}$ implies that the paths followed by m and m' (in directions D and $\bar{D}$ respectively) in the basic scheme may intersect only in $dest(m)$. Moreover, $(m', m'') \notin \bar{D}D$ implies that the paths followed by m' and m'' (in directions $\bar{D}$ and D respectively) intersect in a node which is not $dest(m')$. Simple check shows that the paths followed by m and m'' (in directions D and $\bar{D}$ respectively) may intersect only in $dest(m)$. Therefore, by Fact 2, $(m, m'') \in D\bar{D}$. □

Lemma 3. *If $(m, m') \notin D\bar{D}$ and $(m, m'') \notin \bar{D}D$, then $(m', m'') \in D\bar{D}$.*

PROOF. By Lemma 1 $(m, m') \notin D\bar{D}$ implies $(m', m) \in D\bar{D}$. Then we can apply the preceding Lemma 2 with m', m, m'' in this order to get the result. The second claim is obtained similarly. □

3.4. Makespan can be approximated up to additive constant 2

Recall that $\mathcal{M} = (m_1, \ldots, m_M)$ is the set of messages ordered by non-increasing distance from BS. Throughout this paper, $S \odot S'$ denotes the sequence obtained by the concatenation of two sequences S and S'.

In [5], we use a basic scheme to design an algorithm for broadcasting the messages in the basic instance with a makespan at most $LB + 2$. We give here a different algorithm with similar properties, but easier to prove and which presents two improvements: it can be adapted to the case where the destinations of the messages may be on the axes (i.e. for general grid) (see Section 4) and it can be refined to give in the basic instance a makespan at most $LB + 1$. We denote the algorithm by $TwoApprox[d_I = 0, last = D](\mathcal{M})$; for an input set of messages $\mathcal{M}$ ordered by non-increasing distances from BS, and a direction $D \in \{H, V\}$, it gives as output an ordered sequence $\mathcal{S}$ of the messages such that the basic scheme $(\mathcal{S}, last = D)$ has makespan at most $LB+2$. Recall that D is the direction of the last sent message in $\mathcal{S}$ in Definition 1.

The algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ is given in Figure 3. It uses a basic scheme, where the non-increasing order is kept, if there are no interferences; otherwise we change the order a little bit. To do that, we apply dynamic programming. We examine the messages in their order and at a given step we add to the current ordered sequence the two next unconsidered messages. We show that we can avoid interferences, only by reordering these two messages and the last one in the current sequence.

Remark 2. Notice that, there are instances (see examples below) for which Algorithm $TwoApprox$ computes an optimal makespan only for one direction. Hence, it may sometimes be interesting to apply the algorithm for each direction and take the better resulting schedule.

Because of the behavior of a basic scheme, the direction of the final message and of the first one are simply linked via the parity of the number of messages. Hence, we can also derive an algorithm $TwoApprox[d_I = 0, first = D](\mathcal{M})$ that has the first direction D of the message as an input.

Example 2. Here, we give examples that illustrate the execution of Algorithm $TwoApprox$. Moreover, we describe instances for which it is not optimal.

Consider the example of Figure 4(a). The destinations of the messages m_i ($1 \leq i \leq 6$) are $v_1 = (7, 3)$, $v_2 = (7, 1)$, $v_3 = (3, 3)$, $v_4 = (2, 4)$, $v_5 = (1, 5)$ and $v_6 = (2, 2)$. Here $LB = 10$. Let us apply the Algorithm $TwoApprox[d_I = 0, last = V](\mathcal{M})$. First we apply the algorithm for m_1, m_2. As $(m_1, m_2) \notin HV$, we are at line 4 and $\mathcal{S} = (m_2, m_1)$. Then we consider m_3, m_4. The value of p (line 6) is m_1 and as $(m_1, m_3) \notin VH$ and $(m_1, m_4) \in HV$, we get (line 9, case 3) $\mathcal{S} = (m_2, m_3, m_1, m_4)$. We now apply the algorithm with m_5, m_6. The value of p (line 6) is m_4 and as $(m_4, m_5) \notin VH$ and $(m_4, m_6) \notin HV$, we get (line 10, case 4) $\mathcal{S} = (m_2, m_3, m_1, m_4, m_6, m_5)$. The makespan of the algorithm is $LB + 2 = 12 = d(m_1) + 2$ achieved for $s_3 = m_1$.

But, if we apply to this example the Algorithm $TwoApprox[d_I = 0, last = H](\mathcal{M})$, we get a makespan of 10. Indeed $(m_1, m_2) \in VH$ and we get (line 3) $\mathcal{S} = (m_1, m_2)$. Then as $p = m_2$, $(m_2, m_3) \in HV$ and $(m_3, m_4) \notin VH$, we get (line 8, case 2) $\mathcal{S} = (m_1, m_2, m_4, m_3)$. Finally, with $p = m_3$, $(m_3, m_5) \in HV$ and $(m_5, m_6) \in VH$ we get (line 7, case 1) the final sequence $\mathcal{S} = (m_1, m_2, m_4, m_3, m_5, m_6)$ with makespan $10 = LB$.

Consider the example of Figure 4(b). The destinations of the messages m'_i ($1 \leq i \leq 6$) are v'_i, which are placed in symmetric positions with respect to the diagonal as v_i in Figure 4(a). So $v'_1 = (3, 7)$, $v'_2 = (1, 7)$,..., $v'_6 = (2, 2)$.

Input: $\mathcal{M} = (m_1, \cdots, m_M)$, the set of messages ordered by non-increasing distances from BS and the direction $D \in \{H, V\}$ of the last message.
Output: $\mathcal{S} = (s_1, \cdots, s_M)$ an ordered sequence of the M messages satisfying (i) and (ii) (See in Theorem 1)
begin
1 **Case** $M = 1$**: return** $\mathcal{S} = (m_1)$
2 **Case** $M = 2$**:**
3 **if** $(m_1, m_2) \in \bar{D}D$ **return** $\mathcal{S} = (m_1, m_2)$
4 **else return** $\mathcal{S} = (m_2, m_1)$
5 **Case** $M > 2$**:**
6 let $\mathcal{O} \odot p = TwoApprox[d_I = 0, last = D](m_1, \cdots, m_{M-2})$
(p is the last message in the obtained sequence)
7 **Case 1: if** $(p, m_{M-1}) \in D\bar{D}$ and $(m_{M-1}, m_M) \in \bar{D}D$ **return** $\mathcal{O} \odot (p, m_{M-1}, m_M)$
8 **Case 2: if**$(p, m_{M-1}) \in D\bar{D}$ and $(m_{M-1}, m_M) \notin \bar{D}D$ **return** $\mathcal{O} \odot (p, m_M, m_{M-1})$
9 **Case 3: if**$(p, m_{M-1}) \notin D\bar{D}$ and $(p, m_M) \in \bar{D}D$ **return** $\mathcal{O} \odot (m_{M-1}, p, m_M)$
10 **Case 4: if**$(p, m_{M-1}) \notin D\bar{D}$ and $(p, m_M) \notin \bar{D}D$ **return** $\mathcal{O} \odot (p, m_M, m_{M-1})$

end

Figure 3: Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$

So we can apply the algorithm by exchanging the x and y, V and H. By the Algorithm $TwoApprox[d_I = 0, last = V](\mathcal{M})$, we get $\mathcal{S} = (m'_1, m'_2, m'_4, m'_3, m'_5, m'_6)$ with makespan 10; by the Algorithm $TwoApprox[d_I = 0, last = H](\mathcal{M})$, we get $\mathcal{S} = (m'_2, m'_3, m'_1, m'_4, m'_6, m'_5)$ with makespan 12.

However there are sequences $\mathcal{M}$ such that both Algorithms $TwoApprox[d_I = 0, last = V](\mathcal{M})$ and $TwoApprox[d_I = 0, last = H](\mathcal{M})$ give a makespan $LB + 2$. Consider the example of Figure 4(c) with $\mathcal{M} = (m_1, \ldots, m_6, m'_1, \ldots, m'_6)$. The destinations of $m_1, \ldots, m_6$ are obtained from the destination nodes in Figure 4(a) by translating them along a vector $(3, 3)$, i.e. we move $v_i = (x_i, y_i)$ to $(x_i + 3, y_i + 3)$. So $LB = 16$ and Algorithm $TwoApprox[d_I = 0, last = V](m_1, \ldots, m_6)$ gives the sequence $\mathcal{S}_V = (m_2, m_3, m_1, m_4, m_6, m_5)$ with makespan 18 and Algorithm $TwoApprox[d_I = 0, last = H](m_1, \ldots, m_6)$ gives the sequence $\mathcal{S}_H = (m_1, m_2, m_4, m_3, m_5, m_6)$ with makespan 16. Note that the destinations of $m'_1, \ldots, m'_6$ are in the same configuration as those of Figure 4(b). Now, if we run the Algorithm $TwoApprox[d_I = 0, last = V](\mathcal{M})$ on the sequence $\mathcal{M} = (m_1, \ldots, m_6, m'_1, \ldots, m'_6)$, we get as $(m_5, m'_1) \in VH$ and $(m'_1, m'_2) \in HV$, the sequence $\mathcal{S}_V \odot \mathcal{S}'_V = (m_2, m_3, m_1, m_4, m_6, m_5, m'_1, m'_2, m'_4, m'_3, m'_5, m'_6)$ with makespan 18 achieved for $s_3 = m_1$. If we run Algorithm $TwoApprox[d_I = 0, last = H](\mathcal{M})$ on the sequence $\mathcal{M} = (m_1, \ldots, m_{12})$, we get as $(m_6, m'_1) \in HV$ and $(m'_1, m'_2) \notin VH$ the sequence $\mathcal{S}_H \odot \mathcal{S}'_H = (m_1, m_2, m_4, m_3, m_5, m_6, m'_2, m'_3, m'_1, m'_4, m'_6, m'_5)$ with makespan 18 achieved for $s_9 = m'_1$.

However we can find a sequence with a makespan 16 achieving the lower bound with a basic scheme namely $\mathcal{S}^* = (m_1, m_5, m_2, m_4, m_3, m'_1, m_6, m'_2, m'_5, m'_3, m'_4, m'_6)$ with the first message sent horizontally.

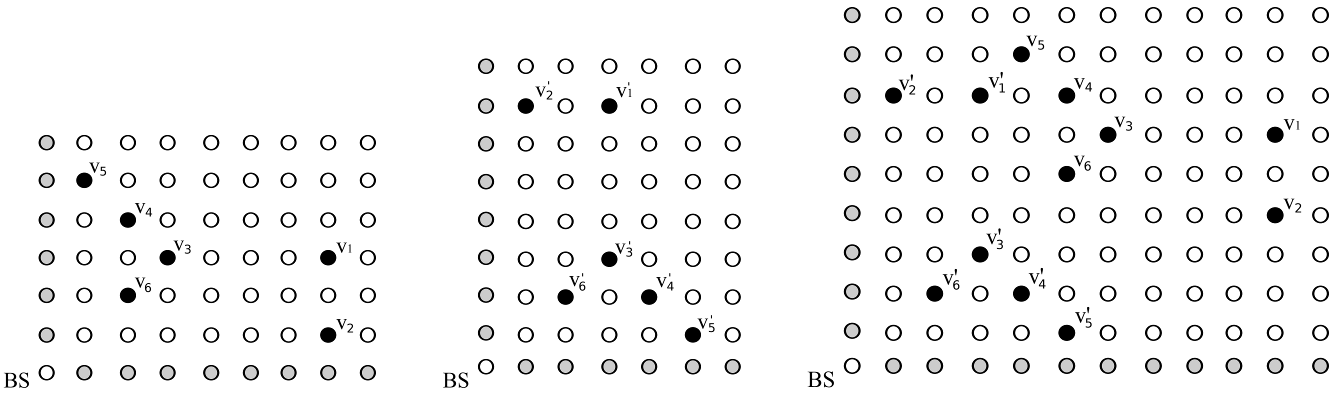
Theorem 1. *Given a basic instance and the set of messages ordered by non-increasing distances from BS,* $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ *and a direction* $D \in \{H, V\}$*, Algorithm* $TwoApprox[d_I = 0, last = D](\mathcal{M})$ *computes in linear-time an ordering* $\mathcal{S} = (s_1, \cdots, s_M)$ *of the messages satisfying the following properties:*

(i) the basic scheme$(\mathcal{S}, last = D)$ *broadcasts the messages without collisions;*

(ii) $s_1 \in \{m_1, m_2\}$, $s_2 \in \{m_1, m_2, m_3\}$ *and* $s_i \in \{m_{i-2}, m_{i-1}, m_i, m_{i+1}, m_{i+2}\}$ *for any* $3 \leq i \leq M - 2$ *and* $s_{M-1} \in \{m_{M-3}, m_{M-2}, m_{M-1}, m_M\}$, $s_M \in \{m_{M-1}, m_M\}$.

PROOF. The proof is by induction on M. If $M = 1$, we send m_1 in direction D (line 1). So the theorem is true. If $M = 2$, either $(m_1, m_2) \in \bar{D}D$ and $\mathcal{S} = (m_1, m_2)$ satisfies all properties or $(m_1, m_2) \notin \bar{D}D$ and by Lemma 1 $(m_2, m_1) \in \bar{D}D$ and $\mathcal{S} = (m_2, m_1)$ satisfies all properties .

If $M > 2$, let $\mathcal{O} \odot p = TwoApprox[d_I = 0, last = D](m_1, \cdots, m_{M-2})$ be the sequence computed by the algorithm for $(m_1, m_2, \cdots, m_{M-2})$. By the induction hypothesis, we may assume that $\mathcal{O} \odot p$ satisfies properties



(a) (b) (c)

Figure 4: Examples for Algorithms $TwoApprox[d_I = 0, last = D](\mathcal{M})$ and $OneApprox[d_I = 0, last = V](\mathcal{M})$

(i) and (ii). In particular p is sent in direction D and $p \in \{m_{M-3}, m_{M-2}\}$. Now we will prove that the sequence $S = \{s_1, \ldots, s_M\}$ satisfies properties (i) and (ii). Property (ii) is satisfied in all cases: for $s_i, 1 \leq i \leq M-3$, as it is verified by induction in $\mathcal{O}$; for s_{M-2}, as either $s_{M-2} = p \in \{m_{M-3}, m_{M-2}\}$ or $s_{M-2} = m_{M-1}$; for s_{M-1}, as either $s_{M-1} = p \in \{m_{M-3}, m_{M-2}\}$ or $s_{M-1} = m_{M-1}$ or $s_{M-1} = m_M$ and finally for s_M, as $s_M \in \{m_{M-1}, m_M\}$. For property (i) we consider the four cases of the algorithm (lines 7-10). Obviously, the last message is sent in direction D in all cases. In the following we prove that there are no interferences in any case. For cases 1, 2 and 4, $\mathcal{O} \odot p$ is by induction a scheme that results in no collision.

In case 1, by hypothesis, $(p, m_{M-1}) \in D\bar{D}$ and $(m_{M-1}, m_M) \in \bar{D}D$.

In case 2, since $(p, m_{M-1}) \in D\bar{D}$ and $(m_{M-1}, m_M) \notin \bar{D}D$, Lemma 2 with p, m_{M-1}, m_M in this order implies that $(p, m_M) \in D\bar{D}$. Furthermore, by Lemma 1, $(m_{M-1}, m_M) \notin \bar{D}D$ implies $(m_M, m_{M-1}) \in \bar{D}D$.

For case 4, by Lemma 1 $(p, m_M) \notin \bar{D}D$ implies $(p, m_M) \in D\bar{D}$. Furthermore Lemma 3, applied with p, m_M, m_{M-1} in this order and direction $\bar{D}$, implies $(m_M, m_{M-1}) \in \bar{D}D$

For case 3, $(p, m_M) \in \bar{D}D$; furthermore by Lemma 1, $(p, m_{M-1}) \notin D\bar{D}$ implies $(m_{M-1}, p) \in D\bar{D}$. It remains to verify that if q is the last message of $\mathcal{O}$, $(q, m_{M-1}) \in \bar{D}D$. As $\mathcal{O} \odot p$ is an admissible scheme we have $(q, p) \in \bar{D}D$ and since also $(p, m_{M-1}) \notin D\bar{D}$, by Lemma 2 applied with q, p, m_{M-1} in this order and direction $\bar{D}$, we get $(q, m_{M-1}) \in \bar{D}D$. □

As corollary we get by property (ii) and definition of LB that the basic scheme $(\mathcal{S}, last = D)$ achieves a makespan at most $LB + 2$. We emphasize this result as a Theorem and note that in view of Example 2 it is the best possible for the algorithm.

Theorem 2. *In the basic instance, the basic scheme $(\mathcal{S}, last = D)$ obtained by the Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ achieves a makespan at most $LB + 2$.*

PROOF. It is sufficient to consider the arrival time of each message. Because Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ uses a basic scheme, each message follows a shortest path. By Property (ii) of Theorem 1, the message s_1 arrives at its destination at step $d(s_1) \leq d(m_1) \leq LB$ and the message s_2 arrives at step $d(s_2) + 1 \leq d(m_1) + 1 \leq LB + 1$; for any $2 < i \leq M$, the message s_i arrives at its destination at step $d(s_i) + i - 1 \leq d(m_{i-2}) + i - 1 = d(m_{i-2}) + (i-2) - 1 + 2 \leq LB + 2$. □

3.5. Makespan can be approximated up to additive constant 1

In this subsection, we show how to improve Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ in the basic instance (open grid with BS in the corner) to achieve makespan at most $LB + 1$. For that we will distinguish two cases according to the value of last term s_M which can be either m_M or m_{M-1}. In the later case, $s_M = m_{M-1}$ we will also maintain another ordered admissible sequence $\mathcal{S}'$ of the $M - 1$ messages $(m_1, \cdots, m_{M-1})$ which can be extended in the induction step when $\mathcal{S}$ cannot be extended. Both sequences $\mathcal{S}$ and $\mathcal{S}'$should satisfy some technical properties (see Theorem 3).

We denote the algorithm as $OneApprox[d_I = 0, last = D](\mathcal{M})$. For an ordered input sequence $\mathcal{M}$ of messages and the direction $D \in \{H, V\}$, it gives as output an ordered sequence $\mathcal{S}$ of the messages such that the basic scheme $(\mathcal{S}, last = D)$ has makespan at most $LB+1$. Algorithm $OneApprox[d_I = 0, last = D](\mathcal{M})$ is depicted in Figure 5. As we explain in Remark 2, we can also obtain algorithms with the first message sent in direction D.

Input: $\mathcal{M} = (m_1, \cdots, m_M)$, the set of messages ordered by non-increasing distances from BS and the direction $D \in \{V, H\}$ of the last message.
Output: $\mathcal{S} = (s_1, \cdots, s_M)$ an ordered sequence of $\mathcal{M}$ satisfying properties (a) and (b) and, only when $s_M = m_{M-1}$, an ordering $\mathcal{S}' = (s'_1, \cdots, s'_{M-1})$ of the messages $(m_1, \cdots, m_{M-1})$ satisfying properties (a'), (b') and (c') (See in Theorem 3). When $\mathcal{S}'$ is not specified below, it means $\mathcal{S}' = \emptyset$.
begin
1 **Case** $M = 1$**: return** $\mathcal{S} = (m_1)$
2 **Case** $M = 2$**:**
3 **if** $(m_1, m_2) \in \bar{D}D$ **return** $\mathcal{S} = (m_1, m_2)$
4 **else return** $\mathcal{S} = (m_2, m_1)$ and $\mathcal{S}' = (m_1)$
5 **Case** $M > 2$**:**
6 let $\mathcal{O} \odot p = OneApprox[d_I = 0, last = D](m_1, \cdots, m_{M-2})$ and when $p = m_{M-3}$, let $\mathcal{O}'$ be the ordering of $\{m_1, \cdots, m_{M-3}\}$ satisfying (a')(b')(c').
7 **Case 1: if** $(p, m_{M-1}) \in D\bar{D}$ and $(m_{M-1}, m_M) \in \bar{D}D$ **return** $\mathcal{S} = \mathcal{O} \odot (p, m_{M-1}, m_M)$
8 **Case 2: if** $(p, m_{M-1}) \in D\bar{D}$ and $(m_{M-1}, m_M) \notin \bar{D}D$ **return** $\mathcal{S} = \mathcal{O} \odot (p, m_M, m_{M-1})$ and $\mathcal{S}' = \mathcal{O} \odot (p, m_{M-1})$
9 **Case 3: if** $(p, m_{M-1}) \notin D\bar{D}$ and $(m_{M-2}, m_M) \in \bar{D}D$
10 **Case 3.1: if** $p = m_{M-2}$ **return** $\mathcal{S} = \mathcal{O} \odot (m_{M-1}, m_{M-2}, m_M)$
11 **Case 3.2: if** $p = m_{M-3}$ **return** $\mathcal{S} = \mathcal{O}' \odot (m_{M-1}, m_{M-2}, m_M)$
12 **Case 4: if** $(p, m_{M-1}) \notin D\bar{D}$ and $(m_{M-2}, m_M) \notin \bar{D}D$
13 **Case 4.1: if** $p = m_{M-2}$ **return** $\mathcal{S} = \mathcal{O} \odot (m_{M-2}, m_M, m_{M-1})$ and $\mathcal{S}' = \mathcal{O} \odot (m_{M-1}, m_{M-2})$
14 **Case 4.2: if** $p = m_{M-3}$ **return** $\mathcal{S} = \mathcal{O} \odot (m_{M-3}, m_M, m_{M-1})$ and $\mathcal{S}' = \mathcal{O}' \odot (m_{M-1}, m_{M-2})$
end

Figure 5: Algorithm $OneApprox[d_I = 0, last = D](\mathcal{M})$

Example 3. Here, we give examples that illustrate the execution of Algorithm $OneApprox$. Moreover, we describe instances for which it is not optimal.

Consider again the Example of Figure 4(a) (see Example 2). Let us apply the Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M})$. First we apply the algorithm for m_1, m_2; $(m_1, m_2) \notin HV$, we are at line 4 and $\mathcal{S} = (m_2, m_1)$ and $\mathcal{S}' = (m_1)$. Then we consider m_3, m_4; the value of p (line 6) is m_1; as $(m_1, m_3) \notin VH$ and $(m_2, m_4) \in HV$, we are in case 3.2 line 11 ($p = m_{M-3}$). So we get, as $\mathcal{O}' = (m_1)$, $S = (m_1, m_3, m_2, m_4)$. We now apply the algorithm with m_5, m_6; the value of p (line 6) is m_4; as $(m_4, m_5) \notin VH$ and $(m_4, m_6) \notin HV$, we are in case 4.1 line 13. So we get $S = (m_1, m_3, m_2, m_4, m_6, m_5)$. The makespan of the algorithm is $LB + 1 = 11 = d(m_5) + 5$ achieved for $s_6 = m_5$.

But, if we apply to this example the Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M})$, we get a makespan of 10. Indeed $(m_1, m_2) \in HV$ and so the algorithm applied to (m_1, m_2) gives $S = (m_1, m_2)$. Then as $p = m_2$, $(m_2, m_3) \in HV$ and $(m_3, m_4) \notin VH$, we are in case 2 line 8. So we get $S = (m_1, m_2, m_4, m_3)$ and $\mathcal{S}' = (m_1, m_2, m_3)$. Finally, with $p = m_3$, $(m_3, m_5) \in HV$ and $(m_5, m_6) \in VH$ we get (line 7 case 1) the final sequence $S = (m_1, m_2, m_4, m_3, m_5, m_6)$ with makespan $10 = LB$.

However there are sequences $\mathcal{M}$ such that both Algorithms $OneApprox[d_I = 0, last = V](\mathcal{M})$ and $OneApprox[d_I = 0, last = H](\mathcal{M})$ give a makespan $LB + 1$. Consider the example of Figure 4(c). Like in Example 2, $LB = 16$; furthermore, for the messages $m_1, \ldots, m_6$ Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M})$ gives the sequence $(m_1, m_3, m_2, m_4, m_6, m_5)$ denoted by S_V with makespan 17 and Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M})$ gives the sequence $(m_1, m_2, m_4, m_3, m_5, m_6)$ denoted by $S_H$with makespan 16. For the messages $m'_1, \ldots, m'_6$, we get (similarly as in Example 2) by applying Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M})$ the sequence $S'_V = (m'_1, m'_2, m'_4, m'_3, m'_5, m'_6)$ with makespan 10 and by applying the Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M})$ the sequence $S'_H = (m'_1, m'_3, m'_2, m'_4, m'_6, m'_5)$ with makespan 11 achieved for $s'_6 = m'_5$. Now if we run the Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M})$ on the global sequence $\mathcal{M} = (m_1, \ldots, m_6, m'_1, \ldots, m'_6)$, we get as $(m_5, m'_1) \in VH$ and $(m'_1, m'_2) \in HV$, the sequence $S_V \odot S'_V = (m_1, m_3, m_2, m_4, m_6, m_5, m'_1, m'_2, m'_4, m'_3, m'_5, m'_6)$ with makespan 17 achieved for $s_6 = m_5$. If we run Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M})$ on the global sequence $\mathcal{M} = (m_1, \ldots, m_6, m'_1, \ldots, m'_6)$, we get as $(m_6, m'_1) \in HV$ and $(m'_1, m'_2) \notin VH$, the sequence $S_H \odot S'_H = (m_1, m_2, m_4, m_3, m_5, m_6, m'_1, m'_3, m'_2, m'_4, m'_6, m'_5)$ with makespan 17 achieved for $s_{12} = m'_5$.

However, we know that the sequence S^* (defined in Example 2) achieves a makespan 16.

Theorem 3. *Given a basic instance and the set of messages ordered by non-increasing distances from BS, $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ and a direction $D \in \{H, V\}$, Algorithm OneApprox$[d_I = 0, last = D](\mathcal{M})$ computes in linear-time an ordering $\mathcal{S} = (s_1, \cdots, s_M)$ of the messages satisfying the following properties:*

(a) the basic scheme $(\mathcal{S}, last = D)$ broadcasts the messages without collisions;

(b) $s_1 \in \{m_1, m_2\}$ and $s_i \in \{m_{i-1}, m_i, m_{i+1}\}$ for any $1 < i \leq M - 1$, and $s_M \in \{m_{M-1}, m_M\}$.

When $s_M = m_{M-1}$, it also computes an ordering $\mathcal{S}' = (s'_1, \cdots, s'_{M-1})$ of the messages $(m_1, \cdots, m_{M-1})$ satisfying properties (a')-(c').

(a') the scheme$(\mathcal{S}', last = \bar{D})$ broadcasts the messages without collisions;

(b') $s'_1 \in \{m_1, m_2\}$, and $s'_i \in \{m_{i-1}, m_i, m_{i+1}\}$ for any $1 < i \leq M - 2$, and $s'_{M-1} \in \{m_{M-2}, m_{M-1}\}$.

(c') $(s'_{M-1}, m_M) \notin \bar{D}D$ and if $s'_{M-1} = m_{M-2}$, $(m_{M-2}, m_{M-1}) \notin D\bar{D}$

PROOF. The proof is by induction. If $M = 1$, the result is correct as we send $m_1$in direction D (line 1). If $M = 2$, either $(m_1, m_2) \in \bar{D}D$ and $\mathcal{S} = (m_1, m_2)$ satisfies properties (a) and (b) or $(m_1, m_2) \notin \bar{D}D$ and by Lemma 1 $(m_2, m_1) \in \bar{D}D$ and $\mathcal{S} = (m_2, m_1)$ satisfies properties (a) and (b) and $\mathcal{S}' = (m_1)$ satisfies all properties (a'), (b') and (c').

Now, let $M > 2$ and let $\mathcal{O} \odot p = OneApprox[d_I = 0, last = D](m_1, \cdots, m_{M-2})$ be the sequence computed by the algorithm for $(m_1, m_2, \cdots, m_{M-2})$. By the induction hypothesis, we may assume that $\mathcal{O} \odot p$ satisfies properties (a) and (b). In particular p is sent in direction D and $p \in \{m_{M-3}, m_{M-2}\}$. We have also that, if $p = m_{M-3}$, $\mathcal{O}'$ satisfies properties (a'), (b') and (c').

Property (b) is also satisfied for $s_i, 1 \leq i \leq M - 3$ as it is verified by induction either in $\mathcal{O}$ or in case 3.2 in $\mathcal{O}'$. Furthermore, either $s_{M-2} = p \in \{m_{M-3}, m_{M-2}\}$ or $s_{M-2} = m_{M-1}$ in case 3. Similarly, $s_{M-1} \in \{m_{M-2}, m_{M-1}, m_M\}$ and $s_M \in \{m_{M-1}, m_M\}$. Hence, Property (b) is satisfied. Property (b') is also satisfied for $s'_i, 1 \leq i \leq M - 3$, as it is verified by induction in $\mathcal{O}$ or for case 4.2 in $\mathcal{O}'$. Furthermore $s'_{M-2} \in \{m_{M-3}, m_{M-2}, m_{M-1}\}$ and $s'_{M-1} \in \{m_{M-2}, m_{M-1}\}$. Hence, Property (b') is satisfied.

Now let us prove that $\mathcal{S}$ satisfies property (a) and $\mathcal{S}'$ properties (a') and (c') in the six cases of the algorithm (lines 7-14). Obviously the last message in $\mathcal{S}$ (resp. $\mathcal{S}'$) is sent in direction D (resp. $\bar{D}$).

In cases 1, 2, 3.1, 4.1 the hypothesis and sequence $\mathcal{S}$ are exactly the same as that given by Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$. Therefore, by the proof of Theorem 1, $\mathcal{S}$ satisfies property (a) and so the proof is complete for cases 1 and 3.1 as there are no sequences $\mathcal{S}'$.

In case 2, $\mathcal{S}'$ satisfies (a') as by hypothesis (line 8) $(p, m_{M-1}) \in D\bar{D}$. Property (c') is also satisfied as $s'_{M-1} = m_{M-1}$ and by hypothesis (line 8) $(m_{M-1}, m_M) \notin \bar{D}D$.

In case 4.1 ($p = m_{M-2}$), let q be the last element of $\mathcal{O}$; $(q, m_{M-2}) \in \bar{D}D$ as $\mathcal{O} \odot p$ is admissible. By hypothesis (line 12), $(m_{M-2}, m_{M-1}) \notin D\bar{D}$ and then by Lemma 2 applied with q, m_{M-2}, m_{M-1} in this order, we get $(q, m_{M-1}) \in \bar{D}D$; furthermore, by Lemma 1 ,$(m_{M-2}, m_{M-1}) \notin D\bar{D}$ implies $(m_{M-1}, m_{M-2}) \in D\bar{D}$. So, $\mathcal{S}'$ satisfies Property (a'). Finally $s'_{M-1} = m_{M-2}$ and by hypothesis (line 12) $(m_{M-2}, m_M) \notin \bar{D}D$ and $(m_{M-2}, m_{M-1}) \notin D\bar{D}$ and therefore $\mathcal{S}'$ satisfies property (c').

The following claims will be useful to conclude the proof in cases 3.2 and 4.2. In these cases $p = m_{M-3}$ and let p' be the last element of $\mathcal{O}'$. By induction on $\mathcal{O}'$, and by property (b'), $p' \in \{m_{M-4}, m_{M-3}\}$.

Claim 1. : In cases 3.2 and 4.2, $(m_{M-2}, m_{M-1}) \notin D\bar{D}$

PROOF. To write a convincing proof, we use coordinates and the expression of Fact 2 in terms of coordinates (see Remark 1). We use $dest(m_{M-i}) = (x_{M-i}, y_{M-i})$. Let us suppose $D = V$ (the claim can be proved for $D = H$ by exchanging H and V and exchanging x and y).

By hypothesis (lines 9 and 12) $(m_{M-3}, m_{M-1}) \notin VH$.

- If $p' = m_{M-3}$, by induction hypothesis (c') applied to $\mathcal{O}'$, we have $(p', m_{M-2}) \notin HV$. Then $(m_{M-3}, m_{M-1}) \notin VH$ and $(m_{M-3}, m_{M-2}) \notin HV$ imply by Fact 2: $\{x_{M-1} < x_{M-3}$ and $y_{M-1} \geq y_{M-3}\}$ and $\{x_{M-2} \geq x_{M-3}$ and $y_{M-2} < y_{M-3}\}$.

  So we have $x_{M-1} < x_{M-3} \leq x_{M-2}$ implying $x_{M-1} < x_{M-2}$ and $y_{M-1} \geq y_{M-3} > y_{M-2}$ implying $y_{M-1} > y_{M-2}$. These conditions imply by Fact 2 that $(m_{M-2}, m_{M-1}) \notin VH$.

- If $p' = m_{M-4}$, by induction hypothesis (c') applied to $\mathcal{O}'$, we have $(p', m_{M-2}) \notin HV$ and $(m_{M-4}, m_{M-3}) \notin VH$. So $(m_{M-3}, m_{M-1}) \notin VH$, $(m_{M-4}, m_{M-2}) \notin HV$ and $(m_{M-4}, m_{M-3}) \notin VH$ imply respectively by Fact 2: $\{x_{M-1} < x_{M-3}$ and $y_{M-1} \geq y_{M-3}\}$; $\{x_{M-2} \geq x_{M-4}$ and $y_{M-2} < y_{M-4}\}$ and $\{x_{M-3} < x_{M-4}$ and $y_{M-3} \geq y_{M-4}\}$.

  So we have $x_{M-1} < x_{M-3} < x_{M-4} \leq x_{M-2}$ implying $x_{M-1} < x_{M-2}$ and $y_{M-1} \geq y_{M-3} \geq y_{M-4} > y_{M-2}$ implying $y_{M-1} > y_{M-2}$. These conditions imply by Fact 2 that $(m_{M-2}, m_{M-1}) \notin VH$.

Claim 2. : In cases 3.2 and 4.2, $(p', m_{M-1}) \in \bar{D}D$.

PROOF. If $p' = m_{M-3}$ by hypothesis lines 9 and 12 $(m_{M-3}, m_{M-1}) \notin D\bar{D}$ and by Lemma 1 $(m_{M-3}, m_{M-1}) \in \bar{D}D$. If $p' = m_{M-4}$, by induction hypothesis (c') applied to $\mathcal{O}'$, $(m_{M-4}, m_{M-3}) \notin D\bar{D}$ and so by Lemma 1 $(m_{M-4}, m_{M-3}) \in \bar{D}D$; furthermore by hypothesis $(m_{M-3}, m_{M-1}) \notin D\bar{D}$ and so by Lemma 2 applied with $m_{M-4}, m_{M-3}, m_{M-1}$ in this order, we get $(m_{M-4}, m_{M-1}) \in \bar{D}D$.

In case 3.2, by hypothesis (line 9) $(m_{M-2}, m_M) \in \bar{D}D$; by the claim 1 $(m_{M-2}, m_{M-1}) \notin D\bar{D}$ and so by Lemma 1 $(m_{M-1}, m_{M-2}) \in D\bar{D}$; and by claim 2, $(p', m_{M-1}) \in \bar{D}D$. So the theorem is proved in case 3.2.

Finally it remains to deal with the case 4.2. Let us first prove that $\mathcal{S}$ satisfies (a). By hypothesis line 12 $(m_{M-2}, m_M) \notin \bar{D}D$ and by the claim $(m_{M-2}, m_{M-1}) \notin D\bar{D}$ and so by Lemma 3 applied with m_{M-2}, m_M, m_{M-1} in this order we get $(m_M, m_{M-1}) \in \bar{D}D$. We claim that $(m_{M-3}, m_{M-2}) \in D\bar{D}$; indeed, if $p' = m_{M-3}$, by induction hypothesis (c') applied to $\mathcal{O}'$, we have $(m_{M-3}, m_{M-2}) \notin \bar{D}D$ and so $(m_{M-3}, m_{M-2}) \in D\bar{D}$. If $p' = m_{M-4}$, by induction hypothesis (c') applied to $\mathcal{O}'$, we have $(m_{M-4}, m_{M-2}) \notin \bar{D}D$ and $(m_{M-4}, m_{M-3}) \notin D\bar{D}$ and so by Lemma 3 applied with $m_{M-4}, m_{M-3}, m_{M-2}$ in this order we get $(m_{M-3}, m_{M-2}) \in D\bar{D}$. Now the property $(m_{M-3}, m_{M-2}) \in D\bar{D}$ combined with the hypothesis line 12 $(m_{M-2}, m_M) \notin \bar{D}D$ gives by Lemma 2 applied with m_{M-3}, m_{M-2}, m_M in this order $(m_{M-3}, m_M) \in D\bar{D}$.

Finally, by claim 1, $(m_{M-2}, m_{M-1}) \notin D\bar{D}$ and so by Lemma 1 $(m_{M-1}, m_{M-2}) \in D\bar{D}$. By claim 2, $(p', m_{M-1}) \in \bar{D}D$ and so $\mathcal{S}'$ satisfies Property (a'). $\mathcal{S}'$ satisfies also Property (c') as $(m_{M-2}, m_M) \notin \bar{D}D$ by hypothesis and $(m_{M-2}, m_{M-1}) \notin D\bar{D}$ by claim 1. □

As corollary we get by property (b) and definition of LB that the basic scheme $(\mathcal{S}, last = D)$ achieves a makespan at most $LB + 1$. We emphasize this result as a Theorem and note that in view of Example 3 it is the best possible for the algorithm. The proof is similar to that Theorem 2.

Theorem 4. *In the basic instance, the basic scheme* $(\mathcal{S}, last = D)$ *obtained by the Algorithm* $OneApprox[d_I = 0, last = D](\mathcal{M})$ *achieves a makespan at most* $LB + 1$.

As we have seen in Example 3, Algorithms $OneApprox[d_I = 0, last = V](\mathcal{M})$ and $OneApprox[d_I = 0, last = H](\mathcal{M})$ are not always optimal since there are instances for which the optimal makespan equals LB while our algorithms only achieves $LB+1$. However there are other cases where Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M})$ or Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M})$ can be used to obtain an optimal makespan LB. The next theorem might appear as specific, but it includes the case where each node in a finite grid receives exactly one message (case considered in many papers in the literature, such as in [6] for the grid when buffering is allowed).

Theorem 5. *Let $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ be an ordered sequence of messages (i.e., by decreasing distance), if the bound $LB = \max_{i \le M} d(m_i) + i - 1$ is reached for an unique value of i, then we can design an algorithm with optimal makespan $= LB$.*

PROOF. Let k be the value for which LB is achieved that is $d(m_k) + k - 1 = LB$ and $d(m_i) + i - 1 < LB$ for $i \neq k$. We divide $\mathcal{M} = (m_1, \cdots, m_M)$ into two ordered subsequences $\mathcal{M}_k = (m_1, \ldots, m_k)$ and $\mathcal{M}'_k = (m_{k+1}, \ldots, m_M)$. So $|\mathcal{M}_k| = k$ and $|\mathcal{M}'_k| = M - k$. Let $\mathcal{S}_V$ (resp., $\mathcal{S}_H$) be the sequence obtained by applying Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M}_k)$ (resp., Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M}_k)$) to the sequence $\mathcal{M}_k$. The makespan is equal to LB; indeed if the sequence is $(s_1, \ldots, s_k)$, then the makespan is $\max_{i \le k} d(s_i) + i - 1$. But we have $s_i \in \{m_{i-1}, m_i, m_{i+1}\}$ for any $i \le k - 1$, and so $d(s_i) + i - 1 \le d(m_{i-1}) + (i - 1) \le LB$ (as $d(m_{i-1}) + (i-1) - 1 < LB$); we also have $s_k \in \{m_{k-1}, m_k\}$ and so either $d(s_k) + (k-1) = d(m_{k-1}) + (k-1) \le LB$ or $d(s_k) + (k - 1) = d(m_k) + k - 1 = LB$.

Suppose $k > 1$, then the destination of m_{k-1} is at the same distance of that of m_k; indeed if $d(m_{k-1}) > d(m_k)$, then $d(m_{k-1}) + k - 2 \ge d(m_k) + k - 1 = LB$ and LB will also be achieved for $k - 1$ contradicting the hypothesis. Consider the set D_k of all the messages with destinations at the same distance as that of m_k (so if $k > 1$ $|D_k| \ge 2$) and let m_u (resp.,m_ℓ) be the uppermost message (resp., lowest message) of D_k, that is the message in D_k with destination the node with the highest y (resp., the lowest y); (in case there are many such messages with this property, i.e. they have the same destination node, we choose one of them).

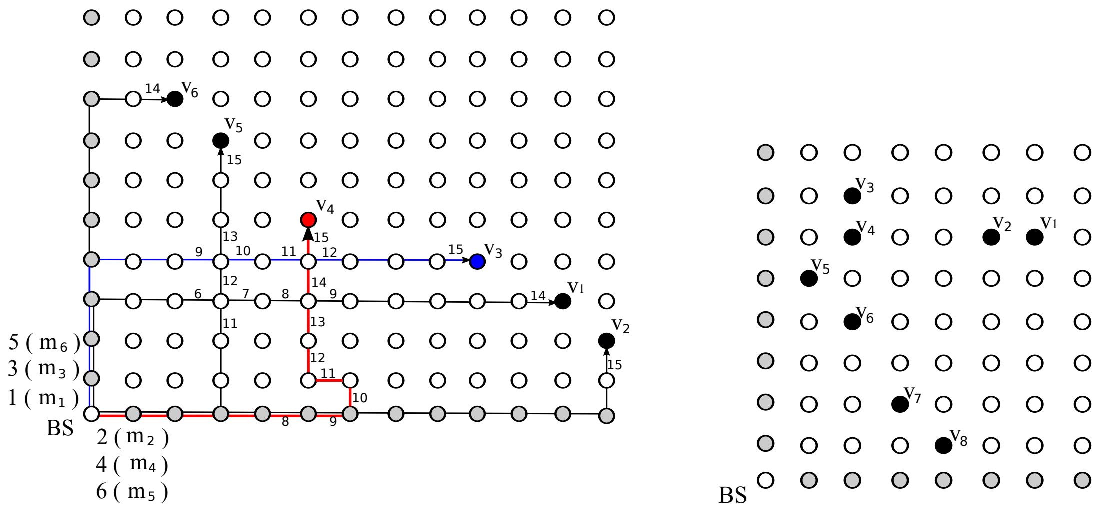
We claim that there exists a basic scheme for $\mathcal{M}_k$, such that if the last message is sent vertically (resp., horizontally) it is m_u (resp. m_ℓ). Indeed, suppose we want the last message sent vertically to be m_u it suffices to order the messages in $\mathcal{M}_k$ such that the last one $m_k = m_u$; then if we apply Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M}_k)$ we get a sequence where $s_k \in \{m_{k-1}, m_k\}$. Either $s_k = m_k = m_u$ and we are done or $s_k = m_{k-1}$ and $s_{k-1} = m_u$; but in that case $(s_{k-1}, s_k) \in HV$ implies, by Fact 2, that $x_{k-1} < x_u$ or $y_{k-1} \ge y_u$, where (x_u, y_u) and (x_{k-1}, y_{k-1}) are the destinations of m_u and m_{k-1}. But $m_u, m_{k-1} \in D_k$ and m_u being the uppermost vertex, $y_{k-1} \le y_u$ and $x_{k-1} \ge x_u$. Therefore, s_{k-1} and s_k have the same destination. So, we can interchange them. Similarly using Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M}_k)$ we can obtain an HV-scheme denoted $\mathcal{S}_H$ with the last message sent horizontally being m_ℓ.

If k=1, $\mathcal{M}_k$ is reduced to one message m_1 and the claims are satisfied with $m_u = m_\ell = m_1$ and $\mathcal{S}_V = \mathcal{S}_H = m_1$.

Now, we consider the sequence $\mathcal{M}'_k$; the lower bound is $LB' = \max_{k < i \le M} d(m_i) + i - k - 1 < LB - k$ as LB is not achieved for any $i \neq k$. Let $\mathcal{S}'_H$ be the sequence obtained by applying Algorithm $OneApprox[d_I = 0, first = H](\mathcal{M}'_k)$ with the first element of $\mathcal{S}'_H$ sent horizontally and let s'_h be this first element. (We obtain this algorithm from Algorithm $OneApprox[d_I = 0, last = V](\mathcal{M}'_k)$ if $|M'_k| = M - k$ is even or Algorithm $OneApprox[d_I = 0, last = H](\mathcal{M}'_k)$ if $|M'_k|$ is odd). Similarly, let $\mathcal{S}'_V$ be the sequence obtained by applying Algorithm $OneApprox[d_I = 0, first = V](\mathcal{M}'_k)$ with the first element of $\mathcal{S}'_V$ sent vertically and let s'_v be this first element. In all the cases the makespan is at most $LB' + 1 \le LB - k$.

Finally, we consider the concatenation of the sequences $\mathcal{S}_V \odot \mathcal{S}'_H$ and $\mathcal{S}_H \odot \mathcal{S}'_V$. We claim that one of these two sequences has no interferences. If the claim is true, then the theorem is proved as the makespan will be LB for the first k messages and $LB' + 1 + k \le LB$ for the last $M - k$ messages. In what follows, let as usual (x_u, y_u), (x_l, y_l), (x'_h, y'_h) and (x'_v, y'_v) denote respectively the destinations of messages m_u, m_l, s'_h and s'_v. Now, suppose the claim is not true, that is $(m_u, s'_h) \notin VH$ and $(m_\ell, s'_v) \notin HV$. That implies by Fact 2 that $x'_h < x_u$ and $y'_h \ge y_u$ and $x'_v \ge x_\ell$ and $y'_v < y_\ell$. But we choose the destination of m_u (resp.,m_ℓ) to be the uppermost one (resp., the lowest one) in D_k. So, $x_u \le x_l$ and $y_u \ge y_l$. Therefore $x'_h < x'_v$ and $y'_h > y'_v$ which imply first that $s'_h \neq s'_v$ and by Fact 2 that $(s'_v, s'_h) \notin VH$ and $(s'_h, s'_v) \notin HV$.

Note that, by the property of Algorithm $OneApprox[d_I = 0, last = D](\mathcal{M})$, $s'_h \in \{m_{k+1}, m_{k+2}\}$ and $s'_v \in \{m_{k+1}, m_{k+2}\}$; thus, as they are different, one of s'_h, s'_v is m_{k+1} and the other m_{k+2}. Suppose that $s'_h = m_{k+1}$ and $s'_v = m_{k+2}$; then in the sequence $\mathcal{S}'_V$ the first message is $s'_v = m_{k+2}$ and from property (c) in Theorem 3, the second message is necessarily $m_{k+1} = s'_h$, but that implies $(s'_v, s'_h) \in VH$ a contradiction. The case $s'_h = m_{k+2}$ and $s'_v = m_{k+1}$ implies similarly in the sequence $\mathcal{S}'_H$ that $(s'_h, s'_v) \in HV$, a contradiction. So the claim and the theorem are proved. $\square$



(a) Example for optimal schedule with not shortest directed path.

(b) Example for optimal schedule sending the m_6 at step 3.

Figure 6: Examples for optimal schedules are difficult to obtain.

Example 4. As mentioned above, Algorithm $OneApprox$ is not always optimal. The design of a polynomial-time optimal algorithm seems challenging because of some reasons that we discuss now. First, the first example below shows that there are open-grid instances for which any broadcast scheme using shortest paths in not optimal (a general grid with such property was already given in Example 1). In this example described in Figure 6(a), we have 6 messages m_i ($1 \leq i \leq 6$) with destinations at distance d for m_1 and m_2, $d-1$ for m_3 and $d-4$ for m_4, m_5, m_6. Here $LB = d+1$, achieved for m_2, m_3 and m_6. In the Figure 6(a), $d = 14$, $v_1 = (11,3)$, $v_2 = (12,2)$, $v_3 = (9,4)$, $v_4 = (5,5)$, $v_5 = (3,7)$ and $v_6 = (2,8)$ and $LB = 15$. If we apply $OneApprox[d_I = 0, last = V](\mathcal{M})$ we get the sequence $(m_1, m_3, m_2, m_5, m_4, m_6)$ with a makespan 16 attained for $s_3 = m_2$. If we apply $OneApprox[d_I = 0, last = H](\mathcal{M})$ we get the sequence $(m_1, m_2, m_4, m_3, m_6, m_5)$ also with a makespan 16 attained for $s_4 = m_3$. Consider any algorithm where the messages are sent via shortest directed paths. If the makespan is LB then m_1 and m_2 should be sent in the first two steps and to avoid interferences the source should send m_1 via $(0,1)$ and m_2 via $(1,0)$. m_3 should be sent at step 3. If m_2 was sent at step 1 and so m_1 at step 2, then m_3 should be sent at step 3 via $(1,0)$ and will interfere with m_1. Therefore, the only possibility is to send m_1 at step 1 via $(0,1)$, m_2 at step 2 via $(1,0)$ and m_3 at step 3 via $(0,1)$. But then at step 4, we cannot send any of m_4, m_5, m_6 without interference. So the source does no sending at step 4, but the last sent message will be sent at step 7 and the makespan will be $d+2 = LB+1$. However there exists a tricky schedule with makespan LB, but not with shortest directed paths routing. We sent m_1 vertically, m_2 horizontally, m_3 vertically but m_4 with a detour to introduce a delay of 2. More precisely, if $v_4 = (x_4, y_4)$, we send m_4 horizontally till $(x_4+1, 0)$, then to $(x_4+1, 1)$ and $(x_4, 1)$ (the detour) and then vertically till (x_4, y_4). Finally we send m_6 vertically at step 5 and m_5 horizontally at step 6. m_4 has been delayed by two but the message arrives at time LB and there is no interference between the messages.

Secondly, even if we restrict ourselves to use shortest paths, the computation of an optimal schedule seems difficult. Indeed, the second example below illustrates the fact that optimal schedule may be very different compared to the non-increasing distance schedule. The example is decribed in Figure 6(b). We have 8 messages m_i ($1 \leq i \leq 8$) with destinations at $v_1 = (6,6)$, $v_2 = (5,6)$, $v_3 = (2,7)$, $v_4 = (2,6)$, $v_5 = (1,5)$, $v_6 = (2,4)$, $v_7 = (3,2)$ and $v_8 = (4,1)$. Here $LB = 12$, achieved for m_1, m_2 and m_8. We will prove that there is a unique sequence of messages reaching the bound LB which is the ordered sequence $(m_1, m_2, m_6, m_3, m_4, m_5, m_8, m_7)$ with the first message sent horizontally. Indeed to reach the makespan LB, m_1 and m_2 have to be sent first and second because their distances are 12 and 11 and in order they do not interfere m_1 has to be sent horizontally and m_2 vertically.The next message to be sent cannot be m_3 nor m_4 as they will interfere with m_2. If the third message sent is m_i for some $i \in 5, 7, 8$, then the fourth and fifth messages have to be m_3 vertically then m_4 horizontally since their distances are 9 and 8. Now only message m_5 can be sent vertically at step 6, otherwise there will an interference with m_4. Then message m_6 has to be sent horizontally at step 7 since its distance is 6. But then the last message m_7 or m_8 (the one not sent at the third step) can not be sent vertically as it will interfere with m_6. So the only possibility consists in sending m_6 at the third step and then the ordered sequence is forced.

In the last example, some specific message (m_6) has to be chosen to be sent early (while being close to BS compared with other messages) to achieve the optimal solution. Deciding of such "critical" message seems to be not easy. Hence it shows that the complexity of determining the value of the minimum makespan might be a difficult problem (even when considering only shortest path schedules).

4. Case $d_I = 0$; general grid, and BS in the corner

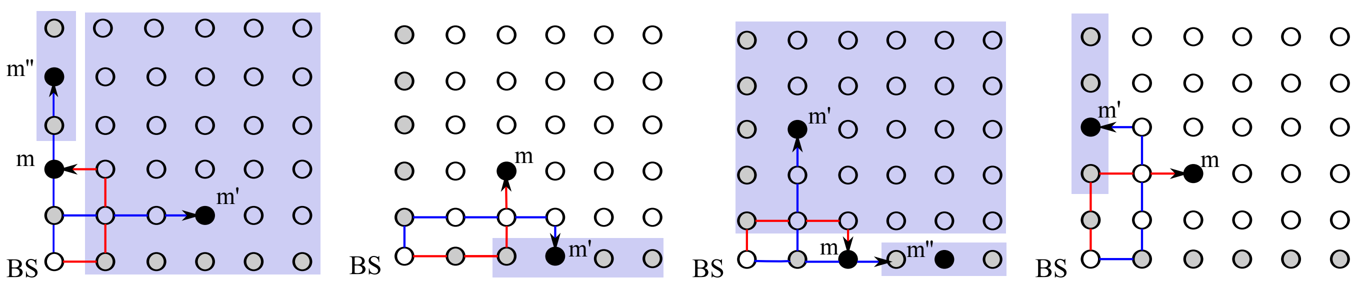
We will see in this section that, by generalizing the notion of basic scheme, Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ also achieves a makespan at most $LB + 2$ in the case of a general grid, that is when the destinations of the messages can be on one or both axes and with BS in the corner. First we have to generalize the notions of horizontal sendings for a destination node on Y-axis and vertical sendings for a destination node on the X-axis. However the proofs of the basic lemmata are more complicated as Lemma 2 is not fully valid in this case. Furthermore, we cannot present the conditions only in simple terms like in Fact 2 and so to be precise we need to use coordinates.

The following definitions are illustrated on Figure 7. We will say that a message is sent "horizontally to reach the Y axis", denoted by H_Y*-sending*, if the destination of m is on the Y axis, i.e., $dest(m) = (0, y)$, and the message is sent first horizontally from BS to (1,0) then it follows the vertical directed path from $(1, 0)$ till $(1, y)$ and finally the horizontal arc $((1, y), (0, y))$. For instance, an H_Y-sending of message m is illustrated in Figure 7(a) and of message m' in Figure 7(d).

Similarly a message is sent "vertically to reach the X axis", denoted by V_X*-sending*, if the destination of m is on the X axis, i.e., $dest(m) = (x, 0)$, and the message is sent first vertically from BS to (0,1) then it follows the horizontal directed path from $(0, 1)$ till $(x, 1)$ and finally the vertical arc $((x, 1), (x, 0))$. For instance, an V_X-sending of message m' is illustrated in Figure 7(b) and of message m in Figure 7(c).

Notations. Definition 1 of basic scheme in Section 3.1 is generalized by allowing H_Y (resp., V_X)-sendings as horizontal (resp., vertical) sendings. For emphasis, we call it *modified basic scheme*. We will also generalize the notation HV (resp., VH) by including H_Y (resp., V_X)-sendings.

Note that we cannot have an H_Y-sending followed by a V_X-sending (or a V_X-sending followed by an H_Y-sending) as there will be interference in $(1, 1)$.



(a) $(m, m') \notin HV$ Case 3.1; $(m, m'') \notin HV$ Case 3.2
(b) $(m, m') \notin HV$ Case 3.3
(c) $(m, m') \notin VH$ Case 4.1; $(m, m'') \notin VH$ Case 4.2
(d) $(m, m') \notin VH$ Case 4.3

Figure 7: Cases of interferences with destinations on the axis.

Fact 3. *Let $dest(m) = (x, y), dest(m') = (x', y')$ and suppose at least one of $dest(m)$ and $dest(m')$ is on an axis. Then*

- $(m, m') \notin HV$ *if and only if we are in one of the following cases, see in Fig. 7(a) and 7(b)*

3.1: $x = 0$ *and* $x' > 0$

3.2: $x = 0$, $x' = 0$ *and* $y' > y$

3.3: $x > 0$, $y' = 0$, $x \leq x'$ *and* $y \geq 2$

or equivalently

- $(m, m') \in HV$ *if and only if we are in one of the following cases*

3.4: $y = 0$

3.5: $x = 0$, $x' = 0$ *and* $y' \leq y$

3.6: $x > 0, y > 0$, $x' = 0$

3.7: $x > 0, y > 0$, $y' = 0$, *and either* $y = 1$ *or* $x' < x$

PROOF. First suppose dest(m) is on one of the axis. If, $y = 0$ there is no interference (3.4). If $x = 0$ and $y' > y$ message m arrives at its destination $(0, y)$ at step $y+2$, but message m' leaves $(0, y)$ at step $y+2$ and so they interfere (3.2 and 3.1 with $y' > y$). If $x = 0$ and $y' \leq y$, either $x' = 0$ and the directed paths followed by the messages do not cross (3.5), or $x' > 0$, but then message m leaves $(1, y')$ at step $y' + 2$, while message m' arrives at $(1, y')$ at step $y' + 2$ and so they interfere (3.1 with $y' \leq y$).

Suppose now that dest(m) is not on one of the axis, that is $x > 0$ and $y > 0$. If $x' = 0$, the directed paths followed by the messages do not cross (3.6). If $y' = 0$, then either $x' < x$ and the messages do not interfere (3.7) or $x' \geq x$, and the directed paths cross at $(x, 1)$ and there either $y = 1$ and the messages do not interfere (3.7) or $y \geq 2$, but then message m leaves $(x, 1)$ at step $x + 2$, while message m' arrives at $(x, 1)$ at step $x + 2$ and so they interfere (3.3). $\square$

Fact 4. *Let* $dest(m) = (x, y), dest(m') = (x', y')$ *and suppose at least one of* $dest(m)$ *and* $dest(m')$ *is on an axis. Then*

- $(m, m') \notin VH$ *if and only if we are in one of the following cases, see in Fig. 7(c) and 7(d)*

4.1: $y = 0$ *and* $y' > 0$

4.2: $y = 0$, $y' = 0$ *and* $x' > x$

4.3: $y > 0$, $x' = 0$, $y \leq y'$ *and* $x \geq 2$

or equivalently

- $(m, m') \in VH$ *if and only if we are in one of the following cases*

4.4: $x = 0$

4.5: $y = 0$, $y' = 0$ *and* $x' \leq x$

4.6: $x > 0, y > 0$, $y' = 0$

4.7: $x > 0, y > 0$, $x' = 0$, *and either* $x = 1$ *or* $y' < y$

Lemma 4. *If* $(m, m') \notin D\bar{D}$*, then* $(m, m') \in \bar{D}D$ *and* $(m', m) \in D\bar{D}$*.*

PROOF. We prove that if $(m, m') \notin HV$ (case $D = H$), then $(m, m') \in VH$ in the following. Other results are proved similarly. If none of the destinations of m and m' are on the axis, the result holds by Lemma 1. If at least one destination is on an axis, suppose that $(m, m') \notin HV$. If conditions of Fact 3.1 or 3.2 are satisfied, then $x = 0$ but then by Fact 4.4 $(m, m') \in VH$. If condition of Fact 3.3 is satisfied , so $x > 0$, $y' = 0$ and $y \geq 2$ which implies by Fact 4.6 that $(m, m') \in VH$. $\square$

However Lemma 2 is no more valid in its full generality.

Lemma 5. *Let* $dest(m) = (x, y)$*,* $dest(m') = (x', y')$ *and* $dest(m'') = (x'', y'')$*.*
If $(m, m') \in D\bar{D}$ *and* $(m', m'') \notin \bar{D}D$*, then* $(m, m'') \in D\bar{D}$ *except if:*

- *Case $D = H$: $y' = 0$ (V_X-sending is used for m'), and $y \geq max(2, y'' + 1)$, and $0 < x' < x \leq x''$.*
- *Case $D = V$: $x' = 0$ (H_Y-sending is used for m'), and $x \geq max(2, x'' + 1)$, and $0 < y' < y \leq y''$.*

PROOF. Let us prove the case $D = H$. If none of the destinations of m, m', m'' are on an axis the result holds by Lemma 2. If $y = 0$, then $(m, m'') \in HV$ by Fact 3.4. By Fact 4, $(m', m'') \notin VH$ implies $x' > 0$. If $x = 0$, then by Fact 3.5, $(m, m') \in HV$ implies $x' = 0$ a contradiction with the preceding assertion. Therefore $x > 0$ and dest(m) is not on an axis. If $x'' = 0$, then by Fact 3.6 $(m, m'') \in HV$. If $y' > 0$, then $(m', m'') \notin VH$ implies $x'' = 0$ by Fact 4.3, where we already know that by Fact 3.6 $(m, m'') \in HV$. So $y' = 0$, $x > 0$, $y > 0$ and by Fact 3.7 $(m, m') \in HV$ implies that either $y = 1$ or $x' < x$.

If $y'' = 0$, by Fact 3.3, $(m, m'') \notin HV$ if and only if $y \geq 2$ and $x \leq x''$. If $y'' > 0$, none of the destinations of m and m'' are on the axis and so by Fact 2, $(m, m'') \notin HV$, if and only if $x'' \geq x$ and $y'' < y$. So again $y \geq 2$ and $x \leq x''$. In summary $(m, m'') \notin HV$, if and only if $y \geq 2$ and when $y'' > 0$, $y > y''$ and $0 < x' < x \leq x''$

The case $D = V$ is obtained similarly. □

We give the following useful corollary for the proof of the next theorem.

Corollary 1. *If $d(m') \geq d(m'')$ then: If $(m, m') \in D\bar{D}$ and $(m', m'') \notin \bar{D}D$, then $(m, m'') \in D\bar{D}$.*

We now show that:

Lemma 6. *Lemma 3 is still valid in general grid.*

PROOF. We prove it for $D = H$. The case $D = V$ can be obtained similarly.

If none of the destinations of m, m', m'' are on an axis the result holds by Lemma 3. Suppose first dest(m'') is on an axis; by Fact 4 $(m, m'') \notin VH$ implies $x > 0$. If furthermore dest(m) or dest(m') are on an axis, by Fact 3.3 $(m, m') \notin HV$ implies $y' = 0$ and so by Fact 3.4 $(m', m'') \in HV$. Otherwise if none of dest(m) and dest(m') are on an axis, $y > 0$ and by Fact 4.3 $(m, m'') \notin VH$ implies $x'' = 0$, and with $x' > 0$ and $y' > 0$ Fact 3.6 implies $(m', m'') \in HV$.

If dest(m'') is not on an axis, then one of dest(m) and dest(m') is on an axis and $(m, m') \notin HV$ implies $y > 0$. We cannot have $x = 0$ otherwise it contradicts $(m, m'') \notin VH$. If $x > 0$, then by Fact 3.3 $(m, m') \notin HV$ implies $y' = 0$, but then Fact 3.4 implies $(m', m'') \in HV$. □

Theorem 6. *Let $d_I = 0$, and BS be in the corner of the general grid. Given the set of messages ordered by non-increasing distances from BS, $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ and a direction D, Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ computes in linear-time an ordering $\mathcal{S}$ of the messages satisfying following properties*

(i) the modified basic scheme($\mathcal{S}, last = D$) broadcasts the messages without collisions;

(ii) $s_1 \in \{m_1, m_2, m_3\}$, $s_2 \in \{m_1, m_2, m_3, m_4\}$ and $s_i \in \{m_{i-2}, m_{i-1}, m_i, m_{i+1}, m_{i+2}\}$ for any $2 < i \leq M - 2$, and $s_{M-1} \in \{m_{M-3}, m_{M-2}, m_{M-1}, m_M\}$ and $s_M \in \{m_{M-1}, m_M\}$;

(iii) for any $i \leq M$, if s_i is an H_Y (resp., V_X) sending with destination on column 0 (resp., on line 0), then either $s_i \in \{m_i, m_{i+1}, m_{i+2}\}$ if $i < M - 1$, or $s_i \in \{m_i, m_{i+1}\}$ if $i = M - 1$, or $s_i = m_i$ if $i = M$.

PROOF. We prove the theorem for $D = V$. The case $D = H$ can be proved similarly. The proof is by induction on M and follows the proof of Theorem 1. We have to verify the new property (iii) and property (i) when one of p, q, m_{M-1}, m_M has its destination on one of the axis. Recall that q is the last message in $\mathcal{O}$. We will denote $dest(p) = (x_p, y_p)$,and as usual $dest(m_{M-1}) = (x_{M-1}, y_{M-1})$ and $dest(m_M) = (x_M, y_M)$.

For property (i) the proof of Theorem 1 works if, when using Lemma 2, we are in a case where it is still valid, that is when Lemma 5 is valid. We use Lemma 2 to prove case 2 of the Algorithm $TwoApprox[d_I = 0, last = V](\mathcal{M})$ with p, m_{M-1}, m_M in this order. The order on the messages implies $d(m_{M-1}) \geq d(m_M)$ and so by Corollary 1, Lemma 5 is valid. We also use Lemma 2 to prove the case 3 of the algorithm with q, p, m_{M-1} in this order. The order on the messages implies $d(p) \geq d(m_{M-1})$ and so by Corollary 1, Lemma 5 is valid. Note that to prove case 4 of the algorithm we use Lemma 3 which is still valid (Lemma 6).

It remains to verify property (iii). In case 2 of the algorithm, we have to show that $s_M = m_{M-1}$ is not using V_X-sending because we use induction for $(m_1, \ldots, m_{M-2})$. So it is sufficient to prove $y_{M-1} > 0$. Indeed, by Fact 3, $(m_{M-1}, m_M) \notin HV$ implies $y_{M-1} > 0$.

In case 3 of the algorithm, to verify property (iii) we have to show that $s_{M-1} = p$ is not using H_Y-sending because we use induction for $(m_1, \ldots, m_{M-2})$. So it is sufficient to prove $x_p > 0$. Indeed, by Fact 4, $(p, m_{M-1}) \notin VH$ implies $x_p > 0$.

In case 4 of the algorithm, to verify property (iii) we have to show that $s_M = m_{M-1}$ is not using V_X-sending. Suppose it is not the case i.e. $y_{M-1} = 0$; as $(p, m_{M-1}) \notin VH$, we have by Fact 4.2 $y_p = 0$ and $x_{M-1} > x_p$. But then $d(p) < d(m_{M-1})$ contradicts the order of the messages. □

As corollary we get by properties (ii) and (iii) and the definition of LB, that the modified basic scheme$(\mathcal{S}, last = D)$ achieves a makespan at most $LB + 2$. We emphasize this result as a Theorem and note that in view of Example 2 or the example given at the end of Section 2 it is the best possible.

Theorem 7. *In the general grid with BS in the corner and $d_I = 0$, the modified basic scheme $(\mathcal{S}, last = D)$ obtained by the Algorithm $TwoApprox[d_I = 0, last = D](\mathcal{M})$ achieves a makespan at most $LB + 2$.*

5. d_I-Open Grid when $d_I \in \{1, 2\}$

In this section, we use the Algorithm $OneApprox[d_I = 0, last = D](\mathcal{M})$ and the detour similar with the one in Example 4 to solve the personalized broadcasting problem for $d_I \in \{1, 2\}$ in d_I-open grids, defined as follows:

Definition 2. *A grid with BS$(0,0)$ in the corner is called 1-open grid if at least one of the following conditions is satisfied: (1) All messages have destination nodes in the set $\{(x, y) : x \geq 2$ and $y \geq 1\}$; (2) All messages have destination nodes in the set $\{(x, y) : x \geq 1$ and $y \geq 2\}$.*

The 1-open grid differs from the open grid only by excluding destinations of messages either on the line $x = 1$ (condition (1)) or on the column $y = 1$ (condition (2)). For $d_I \geq 2$ the definition is simpler.

Definition 3. *For $d_I \geq 2$, a grid with BS$(0,0)$ in the corner is called d_I-open grid if all messages have destination nodes in the set $\{(x, y) : x \geq d_I$ and $y \geq d_I\}$.*

5.1. Lower bounds

In this subsection, we give the lower bounds of the makespan for $d_I \in \{1, 2\}$ in d_I-open grids:

Proposition 2. *Let G be a grid with BS in the corner, $d_I = 1$ and the set of messages, $\mathcal{M} = (m_1, m_2, \cdots, m_M)$, ordered by non-increasing distances from BS, with all the destinations at distance at least 3 ($d(m_M) \geq 3$), then the makespan of any broadcasting scheme is greater than or equal to $LB_c(1) = max_{i \leq M} d(m_i) + \lceil 3i/2 \rceil - 2$.*

PROOF. First we claim that if the source sends two messages in two concecutive steps t and $t + 1$, then it cannot send at step $t + 2$. Indeed, suppose that the source sends a message m at step t on one axis; then at step $t + 1$ it must send the message m' on the other axis. But then at step $t + 2$, both the two neighbors of the source are at distance at most 1 from the sender of messages m or m'. So if the source sends m'' at step $t + 2$, m'' will interfere with m or m'.

Let t_i be the step where the last message in $(m_1, m_2, \cdots, m_i)$ is sent; therefore $t_i \geq \lceil 3i/2 \rceil - 1$. This last message denoted m is received at step $t'_i \geq d(m) + t_i - 1 \geq d(m_i) + t_i - 1 \geq d(m_i) + \lceil 3i/2 \rceil - 2$ and for every $i \leq M$, $LB_c(1) \geq d(m_i) + \lceil 3i/2 \rceil - 2$. □

Remark 3. (A): Obviously, this bound is valid for 1-open grid according to Definition 2.

(B): This bound is valid for $d_I = 1$ only when the source has a degree 2 (case BS in the corner of the grid). If BS is in a general position in the grid we have no better bound than LB.

(C): One can check that the bound is still valid if at most one message has a destination at distance 1 or 2. But if two or more messages have such destinations ($d(m_{M-1}) \leq 2$), then the bound is no more valid. As an example, let dest$(m_i) = v_i$, with $v_1 = (1, 2)$, $v_2 = (2, 1)$, $v_3 = (1, 2)$ and $v_4 = v_5 = (1, 1)$, then $d(m_1) = d(m_2) = d(m_3) = 3$ and $d(m_4) = d(m_5) = 2$ and $LB_c(1) = d(m_5) + 6 = 8$. However we can achieve a makespan of 7 by sending m_4 horizontally at step 1, then m_1 vertically at step 2 and m_2 horizontally at step 3, then the source sends m_3 vertically at step 5 and m_5 horizontally at step 6. m_3 and m_5 reach their destinations at step 7.

(D): Finally let us also remark that there exist configurations for which no gathering protocol can achieve better makespan than $LB_c(1) + 1$. Let dest$(m_1) = v_1 = (x, y)$, with $x + y = d$, dest$(m_2) = v_2 = (x, y - 1)$ and dest$(m_3) = v_3 = (x - 1, y - 2)$. To achieve a makespan of $LB_c(1) = d$, m_1 should be sent at step 1 via a shortest directed path; m_2 should be sent at step 2 via a shortest directed path; and m_3 should be sent at step 4 via a shortest directed path. But, at step d, the sender of m_2 (either $(x, y-2)$ or $(x-1, y-1)$) is at distance 1 from $v_3 = dest(m_3)$ and so m_2 and m_3 interfere.

For $d_I \geq 2$, we have the following lower bound.

Proposition 3. *Let $d_I \geq 2$ and suppose we are in d_I-open grid. Let $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ be the set of messages ordered by non-increasing distances from BS, then the makespan of any broadcasting scheme is greater than or equal to $LB(d_I) = max_{i \leq M} d(m_i) + (i - 1)d_I$.*

PROOF. Indeed if a source sends a message at some step the next message has to be sent at least d_I steps after. □

Remark 4. For $d_I = 2$, there exist configurations for which no gathering protocol can achieve a better makespan than $LB(2) + 2$. Let dest$(m_1) = v_1 = (x, y)$, with $x + y = d$ and dest$(m_2) = v_2 = (x - 1, y - 1)$. Note that $LB(2) = d$. Let s_1, s_2 be the sequence obtained by some algorithm ; to avoid interferences s_1 being sent at step 1 , s_2 should be sent at step ≥ 3. If $s_2 = m_1$, the makespan is at least $d + 2$; Furthermore, if m_1 is not sent via a shortest directed path again the makespan is at least $d + 2$. So $s_1 = m_1$ is sent at step 1 via a shortest directed path. At step d the sender of m_1 (either $(x, y - 1)$ or $(x - 1, y)$ is at distance 1 from v_2. Therefore, if m_2 is sent at step 3 (resp., 4) it arrives at v_2 (resp.,at a neighbor of v_2) at step d and so m_2 interferes with m_1. Thus, m_2 can be sent in the best case at step 5 and arrives at step $d + 2$. In all the cases, the makespan of any algorithm is $LB(2) + 2$.

5.2. Routing with ϵ-detours

To design the algorithms for $d_I \in \{1, 2\}$, we will use the sequence $\mathcal{S}$ obtained by Algorithm $OneApprox[d_I = 0, first = D](\mathcal{M})$. First, as seen in the proof of lower bounds, the source will no more send a message at each step. Second, we need to send the messages via directed paths more complicated than horizontal or vertical sendings; however we will see that we can use relatively simple directed paths with at most 2 turns and simple detours. Let us define precisely such sendings.

Definition 4. *We say that a message to be sent to node (x, y) is sent vertically with an ϵ-detour, if it follows the directed path from BS$(0, 0)$ to $(0, y + \epsilon)$, then from $(0, y + \epsilon)$ to $(x, y + \epsilon)$ and finally from $(x, y + \epsilon)$ to (x, y). Similarly a message to be sent to node (x, y) is sent horizontally with an ϵ-detour, if it follows the directed path from BS$(0, 0)$ to $(x + \epsilon, 0)$, then from $(x + \epsilon, 0)$ to $(x + \epsilon, y)$ and finally from $(x + \epsilon, y)$ to (x, y).*

Note that $\epsilon = 0$ corresponds to a message sent horizontally (or vertically) as defined earlier (in that case we will also say that the message is sent without detour). Note also that in the previous section we use directed paths with 1-detour but only to reach vertices on the axes which are now excluded, since we are in open grid. A message sent at step t with an ϵ-detour reaches its destination at step $t + d(m) + 2\epsilon - 1$. We also note that the detours introduced here are slightly different from the one used in Example 4. They are simpler in the sense that they are doing only two turns and for the case $\epsilon = 1$ (1-detour) going backward only at the last step.

We will design algorithms using the sequence obtained by Algorithm $OneApprox[d_I = 0, first = D](\mathcal{M})$ but we will have to send some of the messages with a 1-detour. We will first give some lemmata which characterize when two messages m and m' interfere when $d_I = 1$, but not interfere in the basic scheme that is when $d_I = 0$, according to the detours of their sendings. For that the following fact which gives precisely the arcs used by the messages will be useful.

Fact 5. • *If $dest(m) = (x, y)$ and m is sent horizontally at step t with an ϵ-detour ($\epsilon = 0$ or 1) then it uses at step $t + h$ the following arc*

case 1: $((h, 0), (h + 1, 0))$ *for* $0 \leq h < x + \epsilon$

case 2: $((x + \epsilon, h - (x + \epsilon)), (x + \epsilon, h + 1 - (x + \epsilon))$ *for* $x + \epsilon \leq h < x + y + \epsilon$

case 3: *if* $\epsilon = 1$ $((x + 1, y), (x, y))$ *for* $h = x + y + 1$

• *If $dest(m') = (x', y')$ and m' is sent vertically with an ϵ'-detour ($\epsilon' = 0$ or 1) at step t', then it uses at step $t' + h'$ the following arc*

case 1': $((0, h'), (0, h' + 1))$ *for* $0 \leq h' < y' + \epsilon'$

case 2': $((h' - (y' + \epsilon'), y' + \epsilon'), (h' + 1 - (y' + \epsilon'), y' + \epsilon'))$ *for* $y' + \epsilon' \leq h' < x' + y' + \epsilon'$

case 3': *if* $\epsilon' = 1$ $((x', y' + 1), (x', y'))$ *for* $h' = x' + y' + 1$

Lemma 7. *Let G be an open grid. Let $dest(m) = (x, y)$ and m be sent at step t horizontally without detour, i.e. $\epsilon = 0$. Let $dest(m') = (x', y')$ and m' be sent vertically with an ϵ'-detour ($\epsilon' = 0$ or 1) at step $t' = t + 1$. Let furthermore $\{x' < x$ or $y' \geq y\}$ (i.e. $(m, m') \in HV$ in the basic scheme). Then for $d_I = 1$, m and m' do not interfere.*

PROOF. To prove that the two messages do not interfere, we will prove that at any step for any pair of messages sent but not arrived at destination, the distance between the sender of one and the receiver of the other is ≥ 2. Consider a step $t + h = t' + h'$ where $h' = h - 1$ and $1 \leq h < min\{x + y, x' + y' + 1 + 2\epsilon'\}$. By Fact 5 we have to consider 6 cases. We label them as case i-j' if we are in case i for m and in case j' for m', $i = 1, 2$ and $1 \leq j \leq 3$:

case 1-1': $1 \leq h < x$ and $0 \leq h - 1 < y' + \epsilon'$. Then, the distance between a sender and a receiver is at least $2h \geq 2$.

case 1-2': $1 \leq h < x$ and $y' + \epsilon' \leq h - 1 < x' + y' + \epsilon'$. Then, the distance between a sender and a receiver is at least $2(y' + \epsilon') \geq 2$, as $y' \geq 1$.

case 1-3': $1 \leq h < x$ and $\epsilon' = 1$ $h - 1 = x' + y' + 1$. Then, the distance between a sender and a receiver is at least $h - x' + y' = 2y' + 2 \geq 4$.

case 2-1': $x \leq h < x + y$ and $0 \leq h - 1 < y' + \epsilon$. Then, the distance between a sender and a receiver is at least $|x| + |x - 2| \geq 2$.

case 2-2': $x \leq h < x + y$ and $y' + \epsilon' \leq h - 1 < x' + y' + \epsilon'$. Recall that $(m, m') \in HV$; so, by Fact 2, $x' < x$ or $y' \geq y$. If $x' < x$, as $h \leq x' + (y' + \epsilon')$, we get $h \leq x + (y' + \epsilon') - 1$. If $y' \geq y$, $h < x + y$ implies $h \leq x + y' - 1$. But, the distance between a sender and a receiver is at least $2(x + (y' + \epsilon') - h) \geq 2$ in both cases.

case 2-3': $x \leq h < x + y$ and $\epsilon' = 1$ $h - 1 = x' + y' + 1$. Then, the distance between a sender and a receiver is at least $|x' - x| + |x' - x + 2| \geq 2$. □

The next lemma will be used partly for proving the correctness of algorithm for $d_I = 1$ (since the last case in the lemma will not happen in the algorithm) and fully for the algorithm for $d_I = 2$.

Lemma 8. *Let G be an open-grid. Let $dest(m) = (x, y)$ with $x \geq 2$ and m be sent horizontally at step t with an ϵ-detour ($\epsilon = 0$ or 1). Let $dest(m') = (x', y')$ and m' be sent vertically with an ϵ'-detour ($\epsilon' = 0$ or 1) at step $t' = t + 2$. Let furthermore $\{x' < x$ or $y' \geq y\}$ (i.e. $(m, m') \in HV$ in the basic scheme). Then, for $d_I = 1$ or 2, m and m' interfere if and only if*

case 00. $\epsilon = 0, \epsilon' = 0$*:* $x' = x - 1$ *and* $y' \leq y - 1$

case 01. $\epsilon = 0, \epsilon' = 1$*:* $x' = x - 1$ *and* $y' \leq y - 2$

case 10. $\epsilon = 1, \epsilon' = 0$*:* $x' \geq x$ *and* $y' = y$

case 11. $\epsilon = 1, \epsilon' = 1$: $x' = x - 1$ *and* $y' = y - 1$

PROOF. Consider a step $t + h = t' + h'$ so $h' = h - 2$. By Fact 5 we have to consider 9 cases according the 3 possibilities for an arc used by m and the 3 possibilities for an arc used by m'. We label them as case i-j' if we are in case i for m and in case j' for m', $1 \leq i, j \leq 3$. We will prove that in all the cases, the distance of the sender and receiver of these two messages is either at most 1 or at least 3. So the interference happens in the same condition for $d_I = 1$ and $d_I = 2$.

case 1-1': Then, the distance between a sender and a receiver is at least $2h - 1 \geq 3$ as $h' = h - 2 \geq 0$.

case 1-2': Then, the distance between a sender and a receiver is at least $2(y' + \epsilon') + 1 \geq 3$ as $y' \geq 1$.

case 1-3': $h = h' + 2 = x' + y' + 3$. The distance between a sender and a receiver is at least $h - x' + y' = 2y' + 3 \geq 5$, as $y' \geq 1$.

case 2-1': Then, the distance between a sender and a receiver is either $|x + \epsilon| + |x + \epsilon - 3| \geq 3$ or $2(x + \epsilon) - 1 \geq 3$ as $x \geq 2$.

case 2-2': Then, the distance between a sender and a receiver is at least $2(x + \epsilon + y' + \epsilon' - h) + 1$. If $y' \geq y - \alpha$, then $h \leq x + y + \epsilon - 1 \leq x + y' + \alpha + \epsilon - 1$ implies $x + \epsilon + y' + \epsilon' - h \geq \epsilon' + 1 - \alpha$ and the distance is at least $2\epsilon' + 3 - 2\alpha$. If $\alpha \leq 0$ ($y' \geq y$) then the distance is ≥ 3. Furthermore if $\epsilon' = 1$ and $\alpha = 1$, the distance is also ≥ 3.

Otherwise, $y' < y$ and by the hypothesis $x' < x$. Let $x' = x - 1 - \beta$ with $\beta \geq 0$; $h' + 2 = h \leq x' + y' + \epsilon' + 1 = x - \beta + y' + \epsilon'$ implies $x + \epsilon + y' + \epsilon' - h \geq \epsilon + \beta$ and the distance is at least $2\epsilon + 1 + 2\beta$. If $\beta \geq 1$ or $\epsilon = 1$, then the distance is ≥ 3. Otherwise when $\beta = 0$ (i.e. $x' = x - 1$) and $\epsilon = 0$, we have a distance 1, achieved for $h = x' + y' + \epsilon' + 1$. More precisely when $\epsilon' = 0$, it is achieved with $x' = x - 1$ and $y' \leq y - 1$, which corresponds to case 00. When $\epsilon' = 1$, we have already seen that the distance is 3, for $y' = y - 1$ (case $\alpha = 1$); otherwise the distance is 1 with $x' = x - 1$ and $y' \leq y - 2$ (case 01).

case 2-3': In this case $\epsilon' = 1$ and $h = x' + y' + 3 \leq x + y + \epsilon - 1$. The distance between a sender and a receiver is $|x + \epsilon - x'| + |x + \epsilon + y' - h|$. If $y' \geq y - 1$, $h = x' + y' + 3 \leq x + y + \epsilon - 1 \leq x + y' + \epsilon$ implies $x' \leq x + \epsilon - 3$ and so $|x + \epsilon - x'| \geq 3$. Otherwise, by hypothesis, $x' < x$; if $x' \leq x - 3$, then $|x + \epsilon - x'| \geq 3$. In the remaining case $x' = x - 1$ or $x' = x - 2$. If $x' = x - 1$, then $|x + \epsilon - x'| = 1 + \epsilon$ and $h = x' + y' + 3 = x + y' + 2$, which implies $|x + \epsilon + y' - h| = 2 - \epsilon$. So the distance is 3; If $x' = x - 2$, then $|x + \epsilon - x'| = 2 + \epsilon$ and $h = x' + y' + 3 = x + y' + 1$, which implies $|x + \epsilon + y' - h| = 1 - \epsilon$. So the distance is 3.

case 3-1': Then, the distance between a sender and a receiver is at least $2x - 1 \geq 3$ as $x \geq 2$.

case 3-2': In that case $\epsilon = 1$ and $h = x + y + 1$ and $h \leq x' + y' + \epsilon' + 1$. The distance between a sender and a receiver is $|x + y' + \epsilon' + 2 - h| + |y' + \epsilon' - y|$. If $x' \leq x - 1$, $h = x + y + 1 \leq x' + y' + \epsilon' + 1 \leq x + y' + \epsilon'$ implies $|x + y' + \epsilon' + 2 - h| + y' + \epsilon' - y \geq 2 + 1 = 3$. Otherwise $x' \geq x$, and, by hypothesis, $y' \geq y$; Let $y' = y + \gamma$ with $\gamma \geq 0$. So $x + y' + \epsilon' = x + y + \gamma + \epsilon' = h - 1 + \gamma + \epsilon'$ implies $|x + y' + \epsilon' + 2 - h| + y' + \epsilon' - y \geq 2\epsilon' + 2\gamma + 1$. If $\epsilon' = 1$ or $\gamma \geq 1$, then the distance is at least 3; otherwise the distance is 1 and so we have interference if $\epsilon = 1, \epsilon' = 0, x' \geq x$ and $y' = y$ (case 10).

case 3-3': Then , $\epsilon = 1, \epsilon' = 1$ and $h = x + y + 1 = x' + y' + 3$. The distance between a sender and a receiver is either $|x + 1 - x'| + |y' - y|$ or $|x - x'| + |y' + 1 - y|$. If $y' \geq y$, then $h = x + y + 1 = x' + y' + 3$ implies $x \geq x' + 2$ and the distance is 3. If $y' \leq y - 1$, then by hypothesis $x' \leq x - 1$ and $x + y + 1 = x' + y' + 3$ implies $y' = y - 1$ and $x' = x - 1$. Then the distance is 1 we have interference. In summary, we have interference if $\epsilon = 1, \epsilon' = 1, x' = x - 1$ and $y' = y - 1$ (case 11). □

By exchanging horizontally and vertically , x and y and x' and y' in Lemma 7 and Lemma 8 we get the following two lemmata:

Lemma 9. *Let G be open grid. Let $dest(m) = (x, y)$ and m be sent vertically (without detour) at step t. Let $dest(m') = (x', y')$ and m' be sent horizontally with an ϵ'-detour ($\epsilon' = 0$ or 1) at step $t' = t + 1$. Let furthermore $\{x' \geq x$ or $y' < y\}$ (i.e. $(m, m') \in VH$ in the basic scheme). Then, for $d_I = 1$, m and m' do not interfere.*

Lemma 10. *let G be an open grid. Let $dest(m) = (x, y)$ with $y \geq 2$ and m be sent vertically at step t with an ϵ-detour ($\epsilon = 0$ or 1). Let $dest(m') = (x', y')$ and m' be sent horizontally with an ϵ'-detour ($\epsilon' = 0$ or 1) at step $t' = t + 2$. Let furthermore $\{x' \geq x$ or $y' < y\}$ (i.e. $(m, m') \in VH$ in the basic scheme). Then for $d_I = 1$ or 2, m and m' interfere if and only if*

case 00. $\epsilon = 0, \epsilon' = 0$: $x' \leq x - 1$ *and* $y' = y - 1$

case 01. $\epsilon = 0, \epsilon' = 1$: $x' \leq x - 2$ *and* $y' = y - 1$

case 10. $\epsilon = 1, \epsilon' = 0$: $x' = x$ *and* $y' \geq y$

case 11. $\epsilon = 1, \epsilon' = 1$: $x' = x - 1$ *and* $y' = y - 1$

5.3. General-scheme $d_I = 1$.

We will have to define general-scheme by indicating not only the ordered sequence of messages $\mathcal{S} = (s_1, \cdots, s_M)$ sent by the source, but also by specifying for each s_i the time t_i at which the message s_i is sent and the directed path followed by the message s_i, in fact the direction D_i and the ϵ_i-detour used for sending it. More precisely,

Definition 5. *A general-scheme is defined as a sequence of M quadruples $(s_i, t_i, D_i, \epsilon_i)$, where the i-th message sent by the source is s_i. This message is sent at step t_i in direction D_i with an ϵ_i-detour.*

Note that we will send the messages alternatively horizontally and vertically in our algorithm. Therefore, we have only to specify the direction of the first (or last) message. We will see in the next theorem that the sequence $\mathcal{S}$ obtained by the algorithm $OneApprox[d_I = 0, first = D](\mathcal{M})$ in Section 3 almost works when $d_I = 1$. More precisely, we propose a scheme that sends the messages in the same order as in $\mathcal{S}$. However, BS waits one step every three steps; i.e., the source sends two messages of the sequence $\mathcal{S}$ during two consecutive steps and then stops sending for one step. Furthermore, a message must sometimes be sent with a detour to avoid interference. That is, the messages are sent without detours like in $\mathcal{S}$, except that, if the first message is sent in direction D, an even message s_{2k+2} is sent in direction $\bar{D}$ with a 1-detour if and only if without detour it would interfere with s_{2k+3}.

Theorem 8. *Let $d_I = 1$, and let BS be in a corner of a 1-open grid. Let $\mathcal{M} = (m_1, \ldots, m_M)$ be the set of messages ordered by non-increasing distances from BS and suppose that the destination $v = (x, y)$ of any message satisfies $\{x \geq 1, y \geq 2\}$ (condition (2) of 1-open grid). Let us define:*

- *$\mathcal{S} = (s_1, \ldots, s_M)$ is the ordered sequence obtained by the Algorithm $OneApprox[d_I = 0, first = H](\mathcal{M})$*
- *for any $i = 2k + 1$, $0 \leq k \leq \lfloor (M - 1)/2 \rfloor$, let $t_i = 3k + 1$ $D_i = H$ and $\epsilon_i = 0$,*
- *for any $i = 2k + 2$, $0 \leq k < \lfloor M/2 \rfloor$, let $t_i = 3k + 2$, $D_i = V$ and $\epsilon_{2k+2} = 0$ if s_{2k+2} does not interferes with s_{2k+3} for $d_I = 1$, otherwise $\epsilon_{2k+2} = 1$.*

Then the general-scheme defined by the sequence $(s_i, t_i, D_i, \epsilon_i)_{i \leq M}$ broadcasts the messages without collisions for $d_I = 1$ and the first message is sent in direction H.

PROOF. To prove the theorem, we need to prove that any two messages do not interfere at any step in the general scheme with parameters $(s_i, t_i, D_i, \epsilon_i)$. A message s_i cannot interfere with a message s_{i+j} for $j \geq 2$ sent at least 3 steps after; indeed the senders of such two messages will be at distance at least 3 (at each step, including the last step when the messages do a 1-detour, the distance of a sender to the base station increases by one). So we have only to care about s_i and s_{i+1}.

First consider the message s_{2k+1}. Let $s_{2k+1} = m$, with $dest(m) = (x, y)$ and $s_{2k+2} = m'$, with $dest(m') = (x', y')$. Message m is sent horizontally at step $t = 3k + 1$ without detour and m' is sent vertically at step $t' = t + 1 =$

$3k+2$ with an ϵ'-detour for $\epsilon' = \epsilon_{2k+2}$. Furthermore, by Theorem 3, we have $(m, m') \in HV$. We conclude by Lemma 7 that s_{2k+1} and s_{2k+2} do not interfere.

Now let us prove that s_{2k+2} does not interfere with s_{2k+3}. Let $s_{2k+2} = m$ with $dest(m) = (x, y)$ and $s_{2k+3} = m'$ with $dest(m') = (x', y')$. Message m is sent vertically with an ϵ-detour, $\epsilon = \epsilon_{2k+2}$ at step $t = 3k+2$ and m' is sent horizontally at step $t' = t+2 = 3k+4$. Furthermore by Theorem 3 $(m, m') \in VH$ and so $\{x' \geq x$ or $y' < y\}$ by Fact 2. So we can apply Lemma 10. If $\{x' \leq x-1$ and $y' = y-1\}$, we are in the case 00 of Lemma 10 and so if m and m' were sent without detour they would interfere. Then by the algorithm we have to choose $\epsilon_{2k+2} = 1$, but now we are in the case 10 of Lemma 10 which implies no interference. (Case 11 never happens in the Theorem.) Otherwise we have $\{x' > x-1$ or $y' \neq y-1\}$; also we have $\epsilon = 0$ according to the Theorem. By case 00 of Lemma 10, they do not interfere. The proof works because interferences in case 00 and 10 of Lemma 10 cannot appear simultaneously.
□

Remark 5. Note that we cannot relax the hypothesis that the messages satisfy $y \geq 2$. Indeed if $y = 1$, we might have to do a 1-detour for $m = s_{2k+2}$ when $x' \geq x$ as at any step $t+h$ $(2 \leq h \leq x)$ the sender of m is at distance 1 from the receiver of $m' = s_{2k+3}$ (case 2-1' in the proof). So we have to send m vertically with a 1-detour; but at step $t+x+2$ the sender of m' $(x', 0)$ is at distance 1 from the receiver of m $(x', 1)$ (case 3-1' in the proof). A simple example is given with 3 messages m_1, m_2, m_3 whose destinations are respectively $(5,1), (4,1), (3,1)$. Then $OneApprox[d_I = 0, first = H](\mathcal{M})$gives the sequence (m_1, m_3, m_2), where $m_3 = s_2$ is sent vertically at step 2 and $m_2 = s_3$ is sent horizontally at step 3. Now, for $d_I = 1$, m_2 is sent at step 4. If m_3 is sent without detour, it interferes with m_2 at step 4 and 5; otherwise if m_3 is sent with a 1-detour it interferes with m_2 at step 7.

By exchanging x and y, H and V, we also get that when the destination $v = (x, y)$ of any message satisfies $\{x \geq 2, y \geq 1\}$ (condition (1) of 1-open grid) we can adapt our algorithm to compute a general-scheme that broadcasts the messages without collisions for $d_I = 1$ and where the first message is sent in direction V. Furthermore, if we are in a 2-open grid we can have a general-scheme where the direction of the first message is arbitrary.

Theorem 9. *In the 1-open grid with BS in the corner and $d_I = 1$, there exists a general-scheme achieving a makespan at most $LB_c(1) + 3$.*

PROOF. Applying the Algorithm $OneApprox[d_I = 0, first = D](\mathcal{M})$, we get an ordered sequence $\mathcal{S}$ which satisfies the Property (b) of Theorem 3: $m_i \in \{s_{i-1}, s_i, s_{i+1}\}$. Consider parameters as in Theorem 8 in case of condition (2) of 1-open grid (the proof is similar for condition(1)). Recall that a message m sent at step t with an ϵ-detour reaches its destination at step $d(m) + 2\epsilon + t - 1$. Then s_{2k+1} reaches its destination (the worst case being $s_{2k+1} = m_{2k}$ sent without detour at step $3k+1$) at step at most $d(m_{2k}) + 3k + 1 - 1 = d(m_{2k}) + \lceil \frac{3(2k)}{2} \rceil - 2 + 2$. Similarly s_{2k+2} reaches its destination (the worst cases being $s_{2k+2} = m_{2k+1}$ sent with a 1-detour at step $3k+2$) at step at most $d(m_{2k+1}) + 2 + 3k + 2 - 1 = d(m_{2k+1}) + 3k + 3 = d(m_{2k+1}) + \lceil \frac{3(2k+1)}{2} \rceil - 2 + 3$. So the makespan is at most $max_{i \leq M} d(m_i) + \lceil 3i/2 \rceil + 1 = LB_c(1) + 3$. □

5.4. General-scheme $d_I = 2$.

In this section, we present a linear-time (in the number of messages) algorithm that computes a general-scheme (Definition 5) broadcasting the messages without collisions for $d_I = 2$ in a 2-open grid, and achieving a makespan up to 4 from the optimal.

As in the case $d_I = 1$, BS will send the messages in the same order as in $\mathcal{S}$. However, BS sends one message only every two steps (which is necessary when $d_I = 2$). The difficulty here is to decide the detour that must be followed by each message, in order to avoid interference. Next algorithm, described in Figure 8, is dedicated to compute the sequence $(\epsilon_i)_{i \leq M}$ of the detours.

Theorem 10. *Let $d_I = 2$, and let BS be in a corner of a 2-open grid. Let $\mathcal{M} = (m_1, \ldots, m_M)$ be the set of messages ordered by non-increasing distances from BS. Let us define $(s_i, t_i, D_i, \epsilon_i)_{i \leq M}$ such that*

- *$\mathcal{S} = (s_1, \ldots, s_M)$ is the ordered sequence obtained by the Algorithm $OneApprox[d_I = 0, first = D](\mathcal{M})$*
- *for any $i \leq M$, $t_i = 2i - 1$ and $D_i = D$ if i is odd and $D_i = \bar{D}$ otherwise.*

Input: $\mathcal{M} = (m_1, \cdots, m_M)$ the set of messages ordered by non-increasing distances from BS, in a 2-open grid, and the direction D of the first message.
Output: $\varepsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_M)$ where $\epsilon_i \in \{0, 1\}$
begin
    Let $(s_1, \cdots, s_M) = OneApprox[d_I = 0, first = D](\mathcal{M})$
    Let $t_i = 2i - 1$, and $D_i = D$ if i is odd and $D_i = \bar{D}$ otherwise, for any $1 \leq i \leq M$
    Let start with $\epsilon_i = 1$ for $1 \leq i \leq M$.
    for $i = M - 1$ to 1
        if s_i interferes with s_{i+1} in the general-scheme defined by $(s_i, t_i, D_i, \epsilon_i)_{i \leq M}$ when $d_I = 2$ **then**
            (we emphasis that we consider interferences with the *current* values of the $(\epsilon_i)_{i \leq M}$)
            $\epsilon_i \leftarrow 0$
    return $\varepsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_M)$
end

Figure 8: Algorithm $Epsilon(\mathcal{M}, first = D)$

- $\varepsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_M)$ *is the sequence obtained by Algorithm* $Epsilon(\mathcal{M}, first = D)$

Then the general-scheme defined by the sequence $(s_i, t_i, D_i, \epsilon_i)_{i \leq M}$ *broadcasts the messages without collisions for* $d_I = 2$ *and the first message is sent in direction* D.

PROOF. We need to prove that any two messages do not interfere at any step. A message s_i cannot interfere with a message s_{i+j}, for $j \geq 2$, sent at least 4 steps after. Indeed, at any step, the senders of two such messages are at distance at least 4. This is because, at each step including the last step when the messages do a 1-detour the distance of a sender to the base station increases by one. So we have only to show that s_i does not interfere with s_{i+1} for any $1 \leq i < M$. For this purpose, we need the following claim that we will prove thanks to Lemma 8 and 10.

Claim 3. For $d_I = 2$, if s_i sent with an $\epsilon_i = 1$-detour interferes with s_{i+1} , then if we send s_i without detour, s_i does not interfere with s_{i+1}.

Indeed suppose s_i is sent in direction D. As the sequence $\mathcal{S}$ is obtained by Algorithm $OneApprox[d_I = 0, first = D](\mathcal{M})$, $(s_i, s_{i+1}) \in D\bar{D}$. So we are in cases 10 if $\epsilon_{i+1} = 0$ or in case 11 if $\epsilon_{i+1} = 1$ of Lemma 8 ($D = H$) or Lemma 10 ($D = V$). First suppose that we are in case 10, then we are not in the case 00; therefore if we send s_i without detour, s_i does not interfere with s_{i+1}. Now assume that we are in case 11, then we are not in the case 01; therefore if we send s_i without detour, s_i does not interfere with s_{i+1}.

Now the algorithm $Epsilon(\mathcal{M}, first = D)$ was designed in such a way it gives either $\epsilon_i = 1$ in which case s_i does not interfere with s_{i+1} or it gives $\epsilon_i = 0$ because s_i sent with a 1 detour was interfering with s_{i+1}, but then, by the claim 3, s_i sent without detour does not interfere with s_{i+1}. □

Theorem 11. *In the* 2*-open grid with BS in the corner and* $d_I = 2$*, the general-scheme defined in Theorem 10 achieves a makespan at most* $LB(2) + 4$.

PROOF. By definition of the scheme, the messages are sent in the same order as computed by $OneApprox[d_I = 0, first = D](\mathcal{M})$. Therefore, by Property (b) of Theorem 3, $s_i \in \{m_{i-1}, m_i, m_{i+1}\}$. So the message s_i arrives at its destination at step $d(s_i) + 2\epsilon_i + t_i - 1 \leq d(m_{i-1}) + 2 + 2i - 1 - 1 = d(m_{i-1}) + 2(i - 1 - 1) + 4$. Then the result follows from the definition of $LB(2)$. □

6. Personalized Broadcasting in Grid with Arbitrary Base Station

In this section, we show how to use the algorithms proposed above to broadcast (or equivalently to gather) a set of personalized messages $\mathcal{M}$, in a grid with a base station placed in an arbitrary node. More precisely, BS will still have coordinates $(0, 0)$, but the coordinates of the other nodes are in $\mathbb{Z}$. A grid with arbitrary base station is said to be an *open-grid* if no destination nodes are on the axes. More generally, a grid with arbitrary base station is said to be an 2*-open-grid* if no destination nodes are at distance at most 1 from any axis.

We divide the grid into four *quadrants* $Q_q, 1 \leq q \leq 4$, where $Q_1 = \{(x, y)$ such that $x \geq 0, y \geq 0\}$, $Q_2 = \{(x, y)$ such that $x \leq 0, y \geq 0\}$, $Q_3 = \{(x, y)$ such that $x \leq 0, y \leq 0\}$, and $Q_4 = \{(x, y)$ such that $x \geq 0, y \leq 0\}$. Note that, BS belongs to all quadrants, and any other node on an axis belongs to two different quadrants.

Each quadrant can be considered itself as a grid with the BS in the corner. Therefore, we can extend all the definitions of the preceding sections, in particular the basic scheme and general-scheme by considering a move in Q_1 (resp., Q_2, Q_3, Q_4) as horizontal, if it is on the positive x-axis (reps. positive y-axis, negative x-axis, negative y-axis) and a vertical move as one on the other half-axis of the quadrant. Then, if we have a sequence of consecutive messages, still ordered by non-increasing distance to BS, and all in the same quadrant we can apply any of the preceding algorithms. Otherwise, we can extend the algorithms by splitting the sequence of messages into maximal subsequences, where all the messages are in the same quadrant and applying any of the algorithms to this subsequence. We have just to be careful that there is no interference between the last message of a subsequence and the first one of the next subsequence; fortunately we will take advantage of the fact that we can choose the direction of the first message of any subsequence.

Theorem 12. *Given a grid with any arbitrary base station BS, and $\mathcal{M} = (m_1, m_2, \cdots, m_M)$ the set of messages ordered by non-increasing distances from BS, then there are linear-time algorithms which broadcast the messages without interferences, with makespan:*

- *at most $LB + 2$ if $d_I = 0$;*
- *at most $LB + 1$ if $d_I = 0$ in an open-grid;*
- *at most $LB_c(1) + 3$ if $d_I = 1$ in a 2-open-grid;*
- *at most $LB(2) + 4$ if $d_I = 2$ in a 2-open-grid;*

PROOF. We partition the ordered set of messages into maximal subsequences, of messages in the same quadrant. That is $\mathcal{M} = \mathcal{M}_1 \odot \mathcal{M}_2 \ldots \mathcal{M}_j \ldots \odot \mathcal{M}_t$, where all the messages in $\mathcal{M}_j$ belong to the same quadrant and the messages of $\mathcal{M}_j$ and $\mathcal{M}_{j+1}$ belong to different quadrants. Then, depending on the cases of the theorem, we apply Algorithms $TwoApprox[d_I = 0, first = D](\mathcal{M})$, $OneApprox[d_I = 0, first = D](\mathcal{M})$, or the algorithms defined in Theorems 8 or 10 to each $\mathcal{M}_j$, in order to obtain a sequence $\mathcal{S}_j$. Now we define the value of D in the algorithms by induction. The direction of the first message of $\mathcal{S}_1$ is arbitrary. Then the direction of the first message of $\mathcal{S}_{j+1}$ has to be chosen on an half-axis different from that of the last message of $\mathcal{S}_j$, which is always possible as two quadrants have at most one half axis in common. For example, suppose the messages of $\mathcal{M}_j$ belong to Q_1 and the last message of $\mathcal{S}_j$ is sent vertically (i.e. on the positive y-axis) and that the messages of $\mathcal{M}_{j+1}$ belong to Q_2, then the first message of $\mathcal{S}_{j+1}$ cannot be sent on the the positive y-axis (that is horizontally in Q_2), but should be sent to avoid interferences on the negative x-axis (that is vertically in Q_2). Otherwise if the last message of $\mathcal{S}_j$ is sent horizontally (i.e. on the positive x-axis), we can sent the first message of $\mathcal{S}_{j+1}$ as we want (as the positive x-axis does not belong to Q_2); similarly if the messages of $\mathcal{M}_{j+1}$ belong to Q_3 we can send the first message of $\mathcal{S}_{j+1}$ as we want (as there are no half axes in common between Q_1 and Q_3). Finally, in the case $d_I = 2$, we have to wait one step between the sending of the last message of $\mathcal{S}_j$ and the first message of $\mathcal{S}_{j+1}$. With these restrictions, we have no interferences between two consecutive messages inside the same $\mathcal{S}_j$ by the correctness of the various algorithms; furthermore we choose the direction of the first message of $\mathcal{S}_{j+1}$ and we add in the case $d_I = 2$ a waiting step in order to avoid interferences between the last message of $\mathcal{S}_j$ and the first message of $\mathcal{S}_{j+1}$. Unconsecutive messages are sent far apart to avoid interferences; indeed the distance between two senders is $> d_I + 1$. Finally the values of the makespan follow from that of the respective algorithms. $\square$

Note that the values of LB (resp., $LB(2)$) are lower bounds for the case of an arbitrary position of BS. Therefore, we get the following corollary

Corollary 2. *There are linear-time (in the number of messages) algorithms that solve the gathering and the personalized broadcasting problems in any grid, achieving an optimal makespan up to an additive constant c where:*

- *$c = 2$ when $d_I = 0$;*

- $c = 1$ *in open-grid when* $d_I = 0$*;*
- $c = 3$ *in* 1*-open-grid when* $d_I = 1$ *and* BS *is a corner;*
- $c = 4$ *in* 2*-open-grid when* $d_I = 2$*.*

However, for $d_I = 1$, LB_c is not a lower bound when BS is not in the corner; the best lower bound we know is LB. In fact this bound can be achieved in some cases. For example suppose that, in the ordered sequence $\mathcal{M}$, the message m_{4j+q} belong to the quadrant Q_q, then we send the messages m_{4j+q} horizontally in Q_q that is on the positive x-axis for $q = 1$, on the positive y-axis for $q = 2$, on the negative x-axis for $q = 3$, and on the negative y-axis for $q = 4$. There is no interferences and the makespan is exactly LB. On the opposite, we conjecture that, when all the messages are in the same quadrant, we can obtain a makespan differing of $LB_c(1)$ by a small constant; so in that case our algorithm will give a good approximation.

Remark 6. Note that when buffering is allowed at the intermediate nodes, LB is still a lower bound for the makespan of any personalized broadcasting or gathering scheme. All our algorithms get makespans at most $\frac{3}{2}LB+3$ for $d_I = 1$, since $LB_c(1) \leq \frac{3}{2}LB$ and $2LB + 4$ for $d_I = 2$, since $LB(2) \leq 2LB$. So we have almost $\frac{3}{2}$ and 2-approximation algorithms for $d_I = 1$ and $d_I = 2$ in 2-open grid respectively when buffering is allowed. For the special grid networks, this improves the result in [2], which gives a 4-approximation algorithm.

7. Conclusion and Further Works

In this article we give several algorithms for the personalized broadcasting and so the gathering problem in grids with arbitrary base station. For $d_I = 0$ and $d_I = 2$, our algorithms have makespans very close to the optimum, in fact, differing from the lower bound by some small additive constants. For $d_I = 1$, we have also efficient algorithms, but only when the base station is in a corner. The general case seems to be difficult to solve and depending on the destinations of the messages. It will be nice to have additive approximations for $d_I \geq 3$; we try to generalize the ideas developed before by using ϵ detours with $\epsilon \geq 2$; doing so, we can avoid interferences between consecutive messages, but not between messages s_i and s_{i+2}. Another challenging problem consists in determining the complexity of finding an optimal schedule and routing of messages for achieving the gathering in the minimum completion time or characterizing when the lower bound is achieved. Example 4 shows it might not be an easy problem. Determining if there is a polynomial algorithm to compute the makespan in the restricted case where messages should be sent via shortest directed paths seems also to be a challenging problem (See Example 4). Last but not least, a natural extension will be to consider the gathering problem for other network topologies.

Acknowledgments: We thank all referees for their very helpful comments

References

[1] J-C. Bermond, R. Correa, and M.-L. Yu. Optimal gathering protocols on paths under interference constraints. *Discrete Mathematics*, 309(18):5574–5587, September 2009.

[2] J-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Pérennes. Hardness and approximation of gathering in static radio networks. *Parallel Processing Letters*, 16(2):165–183, 2006.

[3] J-C. Bermond, L. Gargano, S. Pérennes, A.A. Rescigno, and U. Vaccaro. Optimal time data gathering in wireless networks with omni-directional antennas. In *SIROCCO 2011*, volume 6796 of *Lecture Notes in Computer Science*, pages 306–317, Gdansk, Poland, June 2011. Springer-Verlag.

[4] J-C. Bermond, L. Gargano, and A. A. Rescigno. Gathering with minimum completion time in sensor tree networks. *Journal of interconnection networks*, 11(1-2):1–33, 2010.

[5] J. C. Bermond, N. Nisse, P. Reyes, and H. Rivano. Minimum delay data gathering in radio networks. In *Proceedings of the 8th international conference on Ad Hoc Networks and Wireless (AdHoc-Now)*, volume 5793 of *Lecture Notes in Computer Science*, pages 69–82. Springer, 2009.

[6] J-C. Bermond and J. Peters. Optimal gathering in radio grids with interference. *Theoretical Computer Science*, 457:10–26, October 2012.

[7] J-C. Bermond and M-L. Yu. Optimal gathering algorithms in multi-hop radio tree networks with interferences. *Ad Hoc and Sensor Wireless Networks*, 9(1-2):109–128, 2010.

[8] P. Bertin, J-F. Bresse, and B. Le Sage. Accès haut débit en zone rurale: une solution ”ad hoc”. *France Telecom R&D*, 22:16–18, 2005.

[9] V. Bonifaci, R. Klasing, P. Korteweg, A. Marchetti-Spaccamela, and L. Stougie. Data gathering in wireless networks. In A.Koster and X. Munoz, editors, *Graphs and Algorithms in Communication Networks*, pages 357–377. Springer Monograph, 2010.
[10] V. Bonifaci, P. Korteweg, A. Marchetti Spaccamela, and L. Stougie. An approximation algorithm for the wireless gathering problem. *Operations Research Letters*, 36(5):605–608, 2008.
[11] C. Florens, M. Franceschetti, and R. McEliece. Lower bounds on data collection time in sensory networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1110–1120, 2004.
[12] L. Gargano and A. Rescigno. Optimally fast data gathering in sensor networks. *Discrete Applied Mathematics*, 157:1858–1872, 2009.
[13] C. Gomes, S. Perennes, P. Reyes, and H. Rivano. Bandwidth allocation in radio grid networks. In *Algotel'08*, Saint Malo, May 2008.
[14] R. Klasing, Z. Lotker, A. Navarra, and S. Pérennes. From balls and bins to points and vertices. *Algorithmic Operations Research (AlgOR)*, 4(2):133–143, 2009.
[15] Y. Revah and M. Segal. Improved algorithms for data-gathering time in sensor networks ii: Ring, tree and grid topologies. In *Networking and Services, 2007. ICNS. Third International Conference on*, page 46, 2007.
[16] Y. Revah and M. Segal. Improved bounds for data-gathering time in sensor networks. *Computer Communications*, 31(17):4026–4034, 2008.
[17] Y. Revah, M. Segal, and L. Yedidsion. Real-time data gathering in sensor networks. *Discrete Applied Mathematics*, 158(5):543–550, 2010.
[18] X. Zhu, B. Tang, and H. Gupta. Delay efficient data gathering in sensor network. In X. Jia, J. Wu, and Y. He, editors, *Proc. of MSN'05*, volume 3794 of *Lecture Notes in Computer Science*, pages 380–389. Springer-Verlag, 2005.