A Lower Bound for Deterministic Asynchronous Rendez-Vous on the Line

Luis Barba*[1,2], Prosenjit Bose†[1], Jean-Lou De Carufel†‡[3], Stefan Langerman§[2], and Attila Pór¶[4]

[1]Carleton University, Ottawa, Canada
[2]Université Libre de Bruxelles, Brussels, Belgium
[3]University of Ottawa, Ottawa, Canada
[4]Western Kentucky University, Bowling Green, United States

February 9, 2016

Abstract

Two agents located at distance D from each other on an infinite line want to meet. This problem is known as the *rendez-vous problem*. In this paper, we study a version where the movements of the agents are not synchronized. To guarantee that the two agents can meet, we need to break symmetry. To do so, we assign two different *labels* (binary strings) to the agents. We denote the length of a binary string L by $|L|$. The goal is to design an algorithm which, given any label as an input, produces a sequence of moves, a *strategy*. Every pair of strategies produced by the algorithm must enable the two agents following them to meet in a finite amount of time. The *cost* of an algorithm is equal to the total distance the agents have to travel before they meet, in the worst case. Denote by $L_{\min}$ and $L_{\max}$ the two labels assigned to the agents, where $|L_{\min}| \leq |L_{\max}|$. When D is given to the agents, the best known algorithm has a cost of $\sim |L_{\min}|^2 D$. We write $f \sim g$ whenever $\lim_{x\to\infty} f(x)/g(x) = 1$. When $|L_{\min}| = |L_{\max}| = \ell$ is given to the agents, but D is not given to them, the best known algorithm has a cost of $\sim e^2\ell^2 D$, where $e \approx 2.71828$ is the Euler's number. When nothing is given to the agents, the best known algorithm has a cost of $O(D\log^2(D) + |L_{\max}|D\log(D) + |L_{\min}|^2 D + |L_{\min}||L_{\max}|\log(|L_{\min}|))$. We establish the first non-trivial lower bound on the cost of any deterministic algorithm that solves any of these three variants. We prove that asymptotically, any algorithm has a cost of at least $0.07302\,|L_{\min}|^2\,D$. Our lower bound argument relies on a new technique which uses the asymptotic formula for the number of strongly unimodal sequences.

*This research was funded by Vanier Scholarship.
†This research was funded by NSERC.
‡email: jdecaruf@uottawa.ca
§This research was funded by FNRS.
¶This research was funded by NSF.

1 Introduction

In distributed computing, one of the most basic coordination tasks is that of *gathering*: a group of agents located at arbitrary positions in a given environment want to meet at a single point. When exactly two agents are involved, this problem problem is usually called *rendez-vous*. Rendez-vous has numerous applications in Biology [2, Part IV], human interaction (a rescuer looking for a hiker lost in the woods), robotics [17], computer networks, etc.

Consider the following model for the rendez-vous problem (introduced by De Marco et al. [13]). The environment that the two agents are evolving on is an undirected graph. The agents have neither knowledge of the topology of the graph, its size nor the initial position of the other agent. The vertices are not labelled, but the agents can distinguish between the edges incident to a vertex. Indeed, at each vertex v, the edges incident to v are locally labelled 1, 2, 3,..., d, where d is the degree of v. No coherence between these local labellings is assumed. An agent currently located at v can only see the labels of the edges incident to v. An agent traversing an edge knows both the label of the port by which it leaves and the port it enters a node and the degrees of the nodes.

If we consider the case where the graph is a line, then we can think of the problem as two agents looking for each other on a line, where the distances they are allowed to walk are all integers. In this paper, we study the asynchronous version of this problem, where the movements of the agents are not synchronized. To guarantee that the two agents can meet, we need to break symmetry. Otherwise, they could be following each other forever without noticing. A common way of breaking symmetry deterministically is to assign two different *labels* (binary strings) to the agents (see [1, 11, 14] for instance). We denote the length of a binary string L (the number of bits) by $|L|$. The goal is to design an algorithm which, given any label as an input, produces a sequence of moves, a *strategy*. Every pair of strategies produced by the algorithm must enable the two agents following them to meet in a finite amount of time. The *cost* of an algorithm is equal to the total distance the agents have to travel before they meet, in the worst case. Let D be the distance between the agents at the beginning of the search. Denote by $|L_{\min}|$ and $|L_{\max}|$ the two labels assigned to the agents, where $|L_{\min}| \leq |L_{\max}|$. When D is given to the agents, the best known algorithm has a cost of $\sim |L_{\min}|^2 D$ (refer to [18]). We write $f \sim g$ whenever $\lim_{x\to\infty} f(x)/g(x) = 1$. When $|L_{\min}| = |L_{\max}| = \ell$ is given to the agents, but D is not given to them, the best known algorithm has a cost of $\sim e^2\ell^2 D$, where $e \approx 2.71828$ is the Euler's number (refer to [18]). When nothing is given to the agents, the best known algorithm has a cost of $O(D\log^2(D) + |L_{\max}|D\log(D) + |L_{\min}|^2 D + |L_{\min}||L_{\max}|\log(|L_{\min}|))$ (refer to [18]). Notice that all logarithms in this paper are in base e. We establish the first non-trivial lower bound on the cost of any deterministic algorithm that solves any of these three variants. Moreover, our lower bound applies to the more general setting where the agents are free to walk any real-number distance. Our lower bound argument relies on a new technique which uses the asymptotic formula for the number of strongly unimodal sequences.

Several variants of this rendez-vous problem have been studied: synchronous/asynchronous, randomized/deterministic, etc. In some other versions of the problem, each agent is allowed to leave markers on its path. Some authors also studied the impact of delaying the activation of the agents at the beginning of the search. We refer the reader to [2, 3, 14] for comprehensive surveys on the topic. Rendez-vous problems where the agents can move freely in a polygonal domain have also been studied [6, 9, 15]. Lower bounds were established in some of the frameworks we have mentioned so far: [4, 5, 7, 8, 10, 12]. For instance, in [7], Dessmark et al. establish lower bounds for deterministic asynchronous meeting on a cycle in different settings.

2 Preliminaries

In this section, we define formally the rendez-vous problem on the line we presented in the introduction. We start by explaining the power of the adversary in the worst case scenario. Before the two agents start searching for each other, an adversary assigns a *left/right-orientation* to each agent. The adversary is allowed to assign different orientations: what one agent considers to be "left" could be considered as "right" by the other. However, once the search has started, the left/right-orientations cannot be changed. Each agent sees the line as the real line and considers its starting position to be 0. Since the agents do not start at the same position, they have different origins. For each agent, the negative (respectively positive) side of the real line corresponds to its "left" (respectively "right") orientation. Nevertheless, what is considered to be at distance 1 by an agent is considered to be at distance 1 by both of them. We denote the distance between the two agents at the beginning of the search by D. During the search, each agent follows a *strategy*, where a strategy S consists in a sequence of numbers representing a sequence of positions on the real line.

Definition 1 (Strategy)**.** *A* strategy *S is a sequence of real numbers. If $S = (x_1, x_2, ..., x_n)$ is finite, we say that S is a* finite strategy*. If $S = (x_i)_{i\geq 1}$ is infinite, we say that S is an* infinite strategy*. The numbers x_i are called* steps*.*

A finite strategy $S = (x_1, x_2, ..., x_n)$ can be seen as an infinite strategy $(x_i)_{i\geq 1}$ such that $x_i = x_n$ for all $i > n$. When an agent follows S, the step x_i starts when the agent leaves x_{i-1} and stops when it reaches x_i. To simplify the presentation, we let $x_0 = 0$. The agent following S starts at x_0, then moves to x_1, then to x_2, etc. Each step x_i is thought of by the agent with respect to its starting position on the real line and with respect to its orientation. As the agents search for each other, the adversary controls their speeds independently. However, the adversary must comply to the following *liveness property*: an agent cannot be stopped forever unless it completed the last step of his strategy (in the case of a finite strategy). The absolute position of an agent following a strategy S with respect to time is called a *realization for S*.

Definition 2 (Realization)**.** *Let $S = (x_i)_{i\geq 0}$ be a strategy. A* realization *for S is a function $f_S : [0, \infty[\to \mathbb{R}$ which satisfies the following property. There exists a strictly increasing sequence of real numbers $(t_0, t_1, t_2, ...)$, called a* timeline *for f_S, such that $t_0 = 0$ and*

1. *One of the following two properties is true: (a) for all $i \geq 1$, $f_S(t_i) = f_S(0) + x_i$ or (b) for all $i \geq 1$, $f_S(t_i) = f_S(0) - x_i$.*
2. *For all $i \geq 1$, f_S is monotone on $[t_{i-1}, t_i]$ (we consider the constant function to be monotone).*

In Definition 2, the t_i's represent time. Item 1 represents the orientation of the agent following S. If its compass is accurate, then for all $i \geq 1$, $f_S(t_i) = f_S(0) + x_i$. Otherwise, for all $i \geq 1$, $f_S(t_i) = f_S(0) - x_i$. Moreover, Item 1 ensures that the liveness property is satisfied. Indeed, for all $i \geq 1$, there is a time t_i such that the agent following S reaches x_i (with respect to its own origin). The number $f_S(t_0) = f_S(0)$ is the absolute starting position of the agent following S. We do not necessarily have $f_S(0) = 0$ since the agent following S can be anywhere on the real line at the beginning of the search. As noted by Stachowiak [18, Section 1], assuming that the agents can go back and forth inside $[t_{i-1}, t_i]$ does not give them any additional capability to avoid each other. Therefore, we assume that f_S is monotone on $[t_{i-1}, t_i]$ (see Item 2).

Whether the agents meet each other or not depends on their relative orientations, on their relative positions at the beginning of the search and on their relative speeds. Pairs of realizations are called *space-time assignments*.

Definition 3 (Space-Time Assignment)**.** *Let S_1 and S_2 be two strategies. A* space-time assignment *for S_1 and S_2 is a pair (f_{S_1}, f_{S_2}) of realizations, where f_{S_1} and f_{S_2} are realizations for S_1 and S_2, respectively.*

We represent space-time assignments in *space-time diagrams* (refer to Appendix A for an example). In these diagrams, the horizontal axis represents time and the vertical axis represents position on the real line. On the vertical axis, "down" corresponds to "left" and "up" corresponds to "right". There are four general cases to consider for space-time assignments: either both agents have the same orientation or not, and either $f_{S_1}(0) < f_{S_2}(0)$ or $f_{S_1}(0) > f_{S_2}(0)$.

When the two functions f_{S_1} and f_{S_2} intersect, we say that the two agents *meet* by following S_1 and S_2 with respect to f_{S_1} and f_{S_2}, respectively. If, given any space-time assignment for S_1 and S_2, the two agents meet, we say that S_1 and S_2 are *D-compatible.*

Definition 4 (D-Compatible Strategies)**.** *Two strategies S_1 and S_2 are D-*compatible *if, given any space-time assignment (f_{S_1}, f_{S_2}) for S_1 and S_2 where $|f_{S_1}(0) - f_{S_2}(0)| = D$, there exists a time $t > 0$ such that $f_{S_1}(t) = f_{S_2}(t)$. A set $\mathcal{S}$ of strategies is said to be D-compatible if all strategies are pairwise D-compatible. Two strategies are D-*incompatible *if they are not D-compatible. In other words, there exists a space-time assignment (f_{S_1}, f_{S_2}) for S_1 and S_2 such that the two agents do not meet. When $D = 1$, we say* compatible *and* incompatible *instead of 1-compatible and 1-incompatible, respectively.*

Let S_1 and S_2 be two compatible strategies. Suppose that the two agents follow S_1 and S_2, respectively. We denote by $\gamma(S_1, S_2)$ the maximum distance the two agents travel altogether before they meet, in the worst case. That is, when considering all possible space-time assignments (f_{S_1}, f_{S_2}) such that $|f_{S_1}(0) - f_{S_2}(0)| = 1$.

Suppose now that S_1 and S_2 are incompatible. We denote by $\delta(S_1, S_2)$ the maximum of the minimum distances between the two agents as they are travelling, over all realizations. Formally, let $\mathcal{F}_{S_1}$ (respectively $\mathcal{F}_{S_2}$) be the set of realizations for S_1 (respectively for S_2). Then, $\delta(S_1, S_2) = \max \min_{t \geq 0} |f_{S_1}(t) - f_{S_2}(t)|$, where the maximum is taken over all pairs of realizations $(f_{S_1}, f_{S_2}) \in \mathcal{F}_{S_1} \times \mathcal{F}_{S_2}$ such that $|f_{S_1}(0) - f_{S_2}(0)| = 1$. Notice that if S_1 and S_2 are compatible, we have $\delta(S_1, S_2) = 0$.

The goal of the *Deterministic Asynchronous Rendez-Vous Problem on the Line* is to design a deterministic algorithm that produces a D-compatible set $\mathcal{S}$ of strategies. For short, in this paper, we write *rendez-vous problem on the line.* As we noted in the introduction, to guarantee that the problem can be solved, we need to break symmetry. Otherwise, with an appropriate choice of strategies and space-time assignment, the adversary could have the agents stay forever at distance D from each other (refer to Lemma 1). Therefore, we assign a unique *label* to each agent. We denote the labels of Agents 1 and 2 by L_1 and L_2, respectively. We denote the length (the number of bits) of a label L by $|L|$. An algorithm A is said to *solve the deterministic asynchronous rendez-vous problem on the line* if, given any two different labels as an input, it produces two D-compatible strategies. We have the following result.

Lemma 1. *Any strategy is D-incompatible with itself. Also, if two agents follow the same strategy, there exists a space-time assignment such that they stay at distance at least D from each other.*

Proof. Let S be a strategy and let f_S be any realization of S. Consider the realization $f'_S(t) = f_S(t) + D$. If Agent 1 follows S with respect to f_S and Agent 2 follows S with respect to f'_S, then they stay at distance D from each other. □

Lemma 1 implies that whatever are the labels of the two agents, an algorithm that solves the rendez-vous problem on the line provides the agents with different strategies. We know that there exist algorithms which solve the rendez-vous problem on the line (see [13, 18] for instance). The goal is to find an algorithm with minimum cost, where the cost is defined as follows.

Definition 5 (Cost of an Algorithm)**.** *Let A be an algorithm that solves the deterministic asynchronous rendez-vous problem on the line. Let $\mathcal{S}$ be the set of all strategies produced by A. The cost of A, noted $\phi(A)$, is defined by $\phi(A) = \max \gamma(S_1, S_2)$, where the maximum is taken over all pairs of strategies $S_1, S_2 \in \mathcal{S}$ such that $S_1 \neq S_2$.*

As we mentioned in the introduction, the rendez-vous problem on the line has been studied with respect to different sets of hypotheses. The best known algorithms are due to Stachowiak [18]. He concludes his paper by asking different questions about non-trivial lower bounds on any algorithms that solve these variants of the rendez-vous problem on the line. In this paper, we establish a lower bound on the cost of any algorithm that solves the rendez-vous problem on the line for the case where the agents know D. More precisely, we prove that, asymptotically, any algorithm has a cost of at least $\left(\sqrt{6}\log(2)/(2\pi)\right)^2 |L_{\min}|^2 D \approx 0.07302\, |L_{\min}|^2 D$. Our lower bound applies directly to the other two variants we described in the introduction where the agents do not know D. It also applies to the case where the agents are free to walk any real-number distance.

For the rest of the paper, we suppose that the agents know D. In this case, D plays the role of a scaling factor, therefore we suppose that $D = 1$. Also, notice that up to an appropriate choice of realization by the adversary, we can suppose without loss of generality that $x_1 > 0$ for all strategies $(x_1, x_2, x_3, ...)$. The paper is organized as follows. In Section 3, we explain how to deal with finite *integer strategies* $(x_1, x_2, , ..., x_n)$ (where $x_i \in \mathbb{Z}$ for all $1 \leq i \leq n$). This is a reasonable starting point given that the best known algorithm produces finite integer strategies only (refer to [18]). Moreover, we suppose that $|L_1| = |L_2| = \ell$. In Section 4, we explain how to deal with finite *real strategies* $(x_1, x_2, , ..., x_n)$ (where $x_i \in \mathbb{R}$ for all $1 \leq i \leq n$). We also suppose that $|L_1| = |L_2| = \ell$. In Section 5, we explain how to remove the finiteness hypothesis and the hypothesis $|L_1| = |L_2|$.

3 The Integer Case

Suppose that all strategies have the form $(x_1, x_2, ..., x_n)$, where $x_i \in \mathbb{Z}$ for all $1 \leq i \leq n$. Moreover, suppose that $|L_1| = |L_2| = \ell$. In this section, we develop tools to provide lower bounds on the cost of any algorithm that solves the rendez-vous problem on the line. Then, we prove an asymptotic lower bound of $\left(\sqrt{6}\log(2)/(2\pi)\right)^2 \ell^2 D \approx 0.07302\, \ell^2 D$.

The following lemma, together with Corollary 1, is a tool we need throughout the whole paper.

Lemma 2. *Let A be an optimal algorithm that solves the deterministic asynchronous rendez-vous problem on the line. Let $\mathcal{S}$ be the set of all strategies produced by A and suppose that all strategies in $\mathcal{S}$ are finite. We can suppose the following without loss of generality. For all strategies $S = (x_1, x_2, ..., x_n) \in \mathcal{S}$, there exists another strategy $S' \in \mathcal{S}$ together with a space-time assignment such that the agents meet once the agent following S is at x_n (and it completed step x_n).*

Proof. Let A be an optimal algorithm that solves the rendez-vous problem on the line. Denote by $\mathcal{S}$ the set of all strategies produced by A and let $S_1 = (x_1, x_2, ..., x_n) \in \mathcal{S}$ be any strategy. Without loss of generality, suppose that Agent 1 follows S_1. Moreover, suppose that, for all choices of strategies $S' \neq S_1$, and all space-time assignments for S_1 and S', the two agents never meet while Agent 1 is travelling from x_{n-1} to x_n and they never meet once Agent 1 is at x_n. Let A' be the algorithm that returns $S \in \mathcal{S}$ whenever A returns S, except when $S = S_1$, in which case A' returns $(x_1, x_2, ..., x_{n-1})$. Then A' is also an optimal algorithm that solves the rendez-vous problem on the line. Therefore, for all strategies $S = (x_1, x_2, ..., x_n) \in \mathcal{S}$, we can suppose the following without loss of generality. There exists a strategy $S' \in \mathcal{S}$ (followed by Agent 2) with $S' \neq S$ together with a space-time assignment such that the two agents meet while Agent 1 is travelling from x_{n-1} to x_n or once Agent 1 is at x_n.

Let $S_1 = (x_1, x_2, ..., x_n) \in \mathcal{S}$ be a strategy followed by Agent 1. Consider all strategies $S' \in \mathcal{S}$ for which there exist a space-time assignment such that if Agent 2 follows S', the agents meet while Agent 1 is travelling from x_{n-1} to x_n or once Agent 1 is at x_n. Let x'_n be the furthest distance from x_{n-1} where such a meeting can happen (when considering all strategies and all space-time assignments). Therefore, if $x_n \geq x_{n-1}$, then $x'_n \in [x_{n-1}, x_n]$. And if $x_n \leq x_{n-1}$, then $x'_n \in [x_n, x_{n-1}]$. Let A' be the algorithm that returns $S \in \mathcal{S}$ whenever A returns S, except when $S = S_1$, in which case A' returns $(x_1, x_2, ..., x'_n)$. Then A' is also an optimal algorithm that solves the rendez-vous problem on the line. Therefore, for all strategies $S = (x_1, x_2, ..., x_n) \in \mathcal{S}$, we can suppose the following without loss of generality. There exists a strategy $S' \in \mathcal{S}$ (followed by Agent 2) with $S' \neq S$ together with a space-time assignment such that the agents meet once Agent 1 is at x_n (and it completed step x_n). □

Following the notation of Lemma 2, let $\mathcal{S}_\ell \subseteq \mathcal{S}$ be the set of strategies produced by A from all possible input labels of size ℓ. In the proof of Lemma 2, suppose that we modify only the strategies in $\mathcal{S}_\ell$. Moreover, for each modified strategy $(x_1, x_2, ..., x_n)$, replace x_n by $x'_n = \lceil x_n \rceil$ if $x_n \geq 0$ or by $x'_n = \lfloor x_n \rfloor$ if $x_n < 0$. Then, we get the following corollary from Definition 5.

Corollary 1. *Suppose that only finite strategies are allowed and that $|L_1| = |L_2| = \ell$. There exists an optimal algorithm A that solves the deterministic asynchronous rendez-vous problem on the line such that for all strategies $(x_1, x_2, ..., x_n)$ produced by A from an input label of size ℓ, $x_n \in \mathbb{Z}$ and*

$$\phi(A) + 1 \geq \sum_{i=1}^{n} |x_i - x_{i-1}| . \tag{1}$$

Without loss of generality, we can also suppose that the strategies have no *redundancies*.

Definition 6 (Redundant Strategy)**.** *A strategy $S = (x_1, x_2, ..., x_n)$ ($n \geq 2$) is said to be* redundant *whenever there exists an $0 \leq i \leq n-2$ such that $x_i \leq x_{i+1} \leq x_{i+2}$ or $x_i \geq x_{i+1} \geq x_{i+2}$. In such a case, we say that S* has a redundancy at i. *Otherwise, S is said to be* non-redundant.

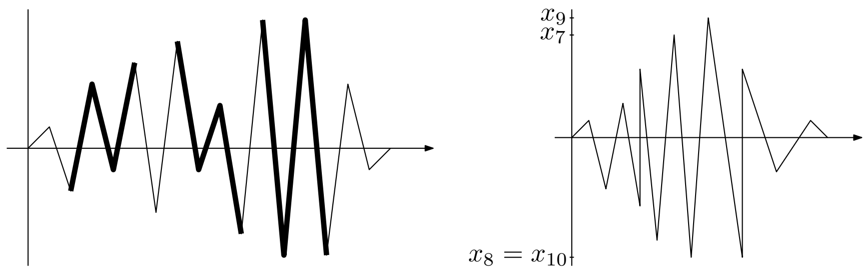
Suppose that S has a redundancy at i (where $0 \leq i \leq n-2$). We say that a strategy S' is obtained from S by removing its redundancy at i *whenever $S' = (x_2, ..., x_n)$ (if $i = 0$), $S' = (x_1, x_2, ..., x_i, x_{i+2}, ..., x_n)$ (if $0 < i < n-2$) or $S' = (x_1, x_2, ..., x_{n-2}, x_n)$ (if $i = n-2$).*

The following lemma, which proof is provided in Appendix B.2, states, without surprise, that we can remove redundancies for free.

Lemma 3. *Let $S_1 = (x_1, x_2, ..., x_n)$ and S_2 be two finite strategies. Suppose that S_1 has a redundancy at i (where $0 \leq i \leq n-2$). Let S'_1 be the strategy obtained from S_1 by removing its redundancy at i. Then S_1 and S_2 are compatible if and only if S'_1 and S_2 are compatible.*

Moreover, if S_1 and S_2 are compatible, then $\gamma(S_1, S_2) = \gamma(S'_1, S_2)$. And if S_1 and S_2 are incompatible, then $\delta(S_1, S_2) = \delta(S'_1, S_2)$.

Let $S = (x_1, x_2, ..., x_n)$ be a non-redundant strategy. For all $0 \leq i \leq n-2$, $x_i \neq x_{i+1}$ otherwise, there would be a redundancy i. Moreover, if $x_i < x_{i+1}$, then $x_{i+1} > x_{i+2}$ whereas if $x_i > x_{i+1}$, then $x_{i+1} < x_{i+2}$. Therefore, if $x_{i+1} - x_i > 0$, then $x_{i+2} - x_{i+1} < 0$, whereas if $x_{i+1} - x_i < 0$, then $x_{i+2} - x_{i+1} > 0$. In other words, the sequence $(x_{i+1} - x_i)_{0 \leq i \leq n-1}$ alternates between positive and negative values. Recall that at the end of Section 2, we supposed without loss of generality that $x_1 > 0$. Consequently, S can be represented as a sequence of positive integers $(\Delta_1, \Delta_2, ..., \Delta_n)$ such that for all $1 \leq i \leq n$, $x_i = x_{i-1} + (-1)^{i+1}\Delta_i$. Then, from Corollary 1, there exists an optimal



(a) (b)

Figure 1: (a) Example of a strategy with 3 lightnings (in bold). (b) Example of a unimodal strategy $(x_1, x_2, ..., x_{13}, 0)$. Following Definition 8, we can let $m = 9$ and $m' = 8$, or $m = 7$ and $m' = 8$.

algorithm A that solves the deterministic asynchronous rendez-vous problem on the line such that for all strategies $S = (x_1, x_2, ..., x_n)$ produced by A, we have

$$\phi(A) + 1 \geq \sum_{i=1}^{n} |x_i - x_{i-1}| = \sum_{i=1}^{n} (-1)^{i+1}(x_i - x_{i-1}) = \sum_{i=1}^{n} \Delta_i. \quad (2)$$

Let A be such an optimal algorithm. To obtain a lower bound on $\phi(A)$, we find an upper bound $\eta(\phi(A))$ on the number of strategies that satisfy (2). Since the 2^ℓ strategies produced by A (recall that $|L_1| = |L_2| = \ell$) satisfy (2), we can write $\eta(\phi(A)) \geq 2^\ell$ and then solve for $\phi(A)$. Counting the number of strategies satisfying (2) corresponds to counting the number of sequences of strictly positive numbers that sum up to $\phi(A)$. We can argue that there are $2^{\phi(A)+1} - 1$ such sequences. However, this leads to $\phi(A) \geq \ell - 1$ which is not all that strong. We show that we can make additional assumptions on the strategies (see Lemmas 4 and 5, and Corollary 2), enabling us to prove an asymptotic lower bound of $0.07302\,\ell^2 D$.

Suppose that an agent follows a strategy S that has a redundancy at i. By Lemma 3, removing the redundancy does not decrease the total distance travelled by the agent. However, it simplifies the presentation of the proofs. Some parts of some strategies —the *lightnings*— can be removed to decrease the total distance travelled by an agent.

Definition 7 (Lightning)**.** *Let $S = (x_1, x_2, ..., x_n)$ ($n \geq 4$) be a strategy. A* lightning *is a subsequence $\mathcal{L}_i = (x_i, x_{i+1}, x_{i+2}, x_{i+3})$ of length 4 (where $0 \leq i \leq n-3$) such that $x_i \leq x_{i+3}$ and $x_{i+1}, x_{i+2} \in [x_i, x_{i+3}]$, or $x_{i+3} \leq x_i$ and $x_{i+1}, x_{i+2} \in [x_{i+3}, x_i]$ (refer to Figure 1(a)).*

Let $S = (x_1, x_2, ..., x_n)$ ($n \geq 4$) be a strategy with a lightning $\mathcal{L}_i$ (where $0 \leq i \leq n-3$). We say that a strategy S' is obtained from S by removing $\mathcal{L}_i$ *whenever $S' = (x_3, ..., x_n)$ (if $i = 0$), $S' = (x_1, x_2, ..., x_i, x_{i+3}, ..., x_n)$ (if $0 < i < n-3$) or $S' = (x_1, x_2, ..., x_{n-3}, x_n)$ (if $i = n-3$).*

Notice that a strategy without lightning can contain a redundancy and a strategy without redundancy can contain a lightning. Stachowiak [18] observed the following property without proof due to space constraints. We provide a proof in Appendix B.3 for the sake of completeness.

Lemma 4 (Stachowiak [18])**.** *Let $S_1 = (x_1, x_2, ..., x_n)$ and S_2 be two finite strategies. Suppose that S_1 has a lightning $\mathcal{L}_i$ (for some $0 \leq i \leq n-3$). Let S_1' be the strategy obtained from S_1 by removing $\mathcal{L}_i$. Then S_1 and S_2 are compatible if and only if S_1' and S_2 are compatible.*

Moreover, if S_1 and S_2 are compatible, then $\gamma(S_1', S_2) \leq \gamma(S_1, S_2)$. And if S_1 and S_2 are incompatible, then $\delta(S_1, S_2) \leq \delta(S_1', S_2)$.

Let $S = (x_1, x_2, ..., x_n)$ be a non-redundant strategy without lightning. Recall that at the end of Section 2, we supposed without loss of generality that $x_1 > 0$. Denote by n' (respectively by n'') the largest odd (respectively even) integer such that $n' \leq n$ (respectively $n'' \leq n$). We can prove that the sequence $(x_1, x_3, ..., x_{n'})$ has a unique maximum and the sequence $(x_0, x_2, x_4, ..., x_{n''})$ has a unique minimum. Moreover, these maxima are closely related. We gather all these properties in Definition 8 and Lemma 5.

Definition 8 (Monotonic and Unimodal Strategies, Stachowiak [18]). *Let $S = (x_1, x_2, ..., x_n)$ be a strategy. Without loss of generality, suppose that $x_1 > 0$. Denote by n' (respectively by n'') the largest odd (respectively even) integer such that $n' \leq n$ (respectively $n'' \leq n$). We say that S is* monotonic *if the sequence $(x_1, x_3, ..., x_{n'})$ is strictly increasing and the sequence $(x_0, x_2, x_4, ..., x_{n''})$ is strictly decreasing.*

We say that S is unimodal *(refer to Figure 1(b)) if there are an odd integer $1 \leq m < n'$ and an even integer $2 \leq m' < n''$ such that $|m - m'| = 1$, $x_m \neq x_{m+2}$ or $x_{m'} \neq x_{m'+2}$, $(x_1, x_3, ..., x_m)$ is strictly increasing, $(x_{m+2}, x_{m+4}, ..., x_{n'})$ is strictly decreasing, $(x_0, x_2, x_4, ..., x_{m'})$ is strictly decreasing and $(x_{m'+2}, x_{m'+4}, ..., x_{n''})$ is strictly increasing.*

If we had $x_m = x_{m+2}$ and $x_{m'} = x_{m'+2}$, or if $|m - m'| > 1$, it would create a redundancy or a lightning. However, our goal is for Definition 8 to characterize non-redundant strategies without lightning. We have the following property.

Lemma 5. *If a non-redundant strategy S does not contain any lightning, then it is monotonic or unimodal.*

Proof. We refer to the proof of Theorem 2 in [18]. Stachowiak proved Lemma 5, but for strategies $(x_1, x_2, x_3, ...)$ such that $x_i > 0$ for all odd integers i and $x_i < 0$ for all even integers i. However, the same proof applies in our case. □

From Lemmas 3, 4 and 5, we get the following corollary.

Corollary 2. *There exists an optimal algorithm A —which solves the deterministic asynchronous rendez-vous problem on the line— such that all strategies produced by A are monotonic or unimodal.*

By Lemmas 3 and 4, we know that we can (and we should) focus on non-redundant strategies without lightning, which are monotonic or unimodal by Lemma 5. Let $S = (x_1, x_2, ..., x_n)$ be a monotonic or unimodal strategy. Recall that any strategy can be represented as a finite sequence of integers $(\Delta_1, \Delta_2, ..., \Delta_n)$ (refer to the discussion following Lemma 3). Since S is monotonic or unimodal, the sequence $(\Delta_1, \Delta_2, ..., \Delta_n)$ is monotonic or unimodal. Therefore, to lower bound $\phi(A)$, it is sufficient to count the number of *strongly unimodal sequences* satisfying (2).

Definition 9 (Strongly Unimodal Sequence of Weight m). *A strongly unimodal sequence of weight m is a sequence of integers $(x_1, x_2, ..., x_n)$ such that $\sum_{i=1}^{n} x_i = m$ and $0 < x_1 < ... < x_{j-1} < x_j > x_{j+1} > ... > x_n > 0$ for some $1 \leq j \leq n$. We denote by $u^*(m)$ the number of strongly unimodal sequences of weight m.*

There is an asymptotic formula for the number of strongly unimodal sequences.

Theorem 1 (Rhoades [16]). $\log(u^*(m)) \sim \pi\sqrt{24m - 1}/6$.

The following corollary is an easy exercise which we solve in Appendix B.4.

Corollary 3. *Let $u^*_\Sigma(m)$ be the number of strongly unimodal sequences of weight at most m. In other words, $u^*_\Sigma(m) = \sum_{i=1}^{m} u^*(i)$. We have $\log(u^*_\Sigma(m)) \sim \pi\sqrt{24m - 1}/6$.*

We are ready to prove the following theorem.

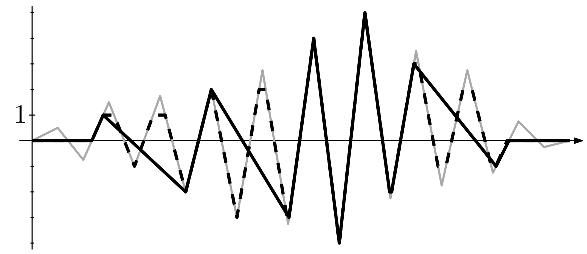


Figure 2: Example of a strategy (in gray) together with its signature (in dashed black) and its simplified signature (in black). Recall that $D = 1$.

Theorem 2. *Suppose that only strategies of the form $(x_1, x_2, ..., x_n)$ are allowed, where $x_i \in \mathbb{Z}$ for all $1 \le i \le n$. Moreover, suppose that $|L_1| = |L_2| = \ell$. Let A be any algorithm that solves the deterministic asynchronous rendez-vous problem on the line. Asymptotically, we have $\phi(A) \ge \left(\sqrt{6}\log(2)/(2\pi)\right)^2 \ell^2 D \approx 0.07302\,\ell^2 D$.*

Proof. Without loss of generality, we can suppose that A is an optimal algorithm that satisfies Corollary 2. Let $\mathcal{C} = \phi(A)$ and $S = (x_1, x_2, ..., x_n)$ be any strategy produced by A, where $x_i = x_{i-1} + (-1)^{i+1}\Delta_i$ for all $1 \le i \le n$. By Corollary 2, S is non-redundant and it is monotonic or unimodal. By (2),

$$\mathcal{C} + 1 \ge \sum_{i=1}^{n} \Delta_i. \tag{3}$$

How many integer sequences satisfy (3)? The numbers in (3) are all positive and they add up to at most $\mathcal{C}+1$. Moreover, as we explained, each sequence $(\Delta_1, \Delta_2, ..., \Delta_n)$ which satisfies (3) is strongly unimodal (refer to Definition 9). Therefore, there cannot be more than $u^*_\Sigma(\mathcal{C}+1)$ strategies produced by A. Since $|L_1| = |L_2| = \ell$, there are exactly 2^ℓ strategies produced by A. Therefore, $u^*_\Sigma(\mathcal{C}+1) \ge 2^\ell$, from which we get $\log\left(u^*_\Sigma(\mathcal{C}+1)\right)/\log(2) \ge \ell$. Thus, asymptotically, we have $\pi\sqrt{24(\mathcal{C}+1)-1}/\left(6\log(2)\right) \ge \ell$ by Corollary 3, from which $\mathcal{C} \ge \left(\sqrt{6}\log(2)/(2\pi)\right)^2 \ell^2 - 23/24 \sim \left(\sqrt{6}\log(2)/(2\pi)\right)^2 \ell^2$. □

4 A Lower Bound for the Real Case

In this section, we consider *real strategies* $(x_1, x_2, ..., x_n)$, where $x_i \in \mathbb{R}$ for all $1 \le i \le n$. We maintain the assumption $|L_1| = |L_2| = \ell$, which will be removed in Section 5. The main result of this section is that the lower bound of Theorem 2 stands for finite real strategies. The definitions of redundancy (refer to Definition 6) and lightning (refer to Definition 7) apply to real strategies. Moreover, in the proofs of Lemmas 3, 4 and 5, and Corollary 2, we did not use the fact that the strategies were made of integers. Therefore, when needed, we can suppose that the strategies we consider are monotonic or unimodal. We define the *signature* of a strategy, which intuitively is the integer part of a real strategy.

Definition 10 (Signature of a Strategy)**.** *Let $S = (x_1, x_2, ..., x_n)$ be a real strategy. Let $\sigma(S) = (x'_1, x'_2, ..., x'_n)$ be the strategy such that for all $1 \le i \le n$, if $x_i \ge 0$, then $x'_i = \lfloor x_i \rfloor$, and if $x_i \le 0$, then $x'_i = \lceil x_i \rceil$. The strategy $\sigma(S)$ is called the* signature of S *(refer to Figure 2).*

The following lemma explains why we do not need to consider real strategies.

Lemma 6. *Let S_1 and S_2 be two real strategies with signatures $\sigma(S_1)$ and $\sigma(S_2)$, respectively. If S_1 and S_2 are compatible, then $\sigma(S_1) \neq \sigma(S_2)$.*

Proof. We prove the following equivalent statement: if $\sigma(S_1) = \sigma(S_2)$, then S_1 and S_2 are incompatible. Let $S_1 = (x_1, x_2, ..., x_n)$ and $S_2 = (x'_1, x'_2, ..., x'_n)$, and suppose that Agents 1 and 2 both follow $\sigma(S_1) = \sigma(S_2)$. Let $f_{\sigma(S_1)}$ be any realization for $\sigma(S_1)$ and take $f_{\sigma(S_1)} + 1$ as a realization for $\sigma(S_2)$. Therefore, the two agents stay at distance 1 from each other (refer to Lemma 1) and all steps are synchronized. We prove that $x_i \neq x'_i$ for all $0 \leq i \leq n$. We consider two cases: either (1) the two agents are on the same side of the real axis or (2) not.

1. Suppose that the two agents are on the same side of the real axis. Without loss of generality, they are both on the positive side of the real axis. Therefore, Agent 1 is at $\lfloor x_i \rfloor$, Agent 2 is at $\lfloor x'_i \rfloor$ and $|\lfloor x_i \rfloor - \lfloor x'_i \rfloor| = 1$. Let $x_i = m_i + \varepsilon_i$ and $x'_i = m'_i + \varepsilon'_i$, where $m_i = \lfloor x_i \rfloor$ and $m'_i = \lfloor x'_i \rfloor$. If $x_i > x'_i$, we have $1 = |\lfloor x_i \rfloor - \lfloor x'_i \rfloor| = \lfloor x_i \rfloor - \lfloor x'_i \rfloor = m_i - m'_i$. Hence, since $0 \leq \varepsilon_i < 1$ and $0 \leq \varepsilon'_i < 1$, we have $x_i - x'_i = (m_i + \varepsilon_i) - (m'_i + \varepsilon'_i) = (m_i - m'_i) - (\varepsilon'_i - \varepsilon_i) = 1 - (\varepsilon'_i - \varepsilon_i) > 0$. Therefore, $x_i > x'_i$. If $x_i \leq x'_i$, a symmetric argument applies.

2. Suppose that the two agents are not on the same side of the real axis. Without loss of generality, Agent 1 is on the positive side and Agent 2 on the negative side of the real axis. Therefore, Agent 1 is at $\lfloor x_i \rfloor$, Agent 2 is at $\lceil x'_i \rceil$ and $\lfloor x_i \rfloor - \lceil x'_i \rceil = 1$. We have $x_i - x'_i \geq \lfloor x_i \rfloor - \lceil x'_i \rceil = 1$. Therefore, $x_i \geq x'_i + 1 > x'_i$.

We now define a space-time assignment (f_{S_1}, f_{S_2}) for S_1 and S_2, showing that S_1 and S_2 are incompatible. Let $f_{S_1}(i) = x_i$ and $f_{S_2}(i) = x'_i$ for all $0 \leq i \leq n$. Therefore, $f_{S_1}(i) = x_i \neq x'_i = f_{S_2}(i)$ for all $0 \leq i \leq n$. Without loss of generality, $x_i > x'_i$ and $x_{i+1} > x'_{i+1}$ for all $0 \leq i \leq n$. On the interval $[i, i+1]$ let f_{S_1} be the linear function from (i, x_i) to $(i+1, x_{i+1})$ and let f_{S_2} be the linear function from (i, x'_i) to $(i+1, x'_{i+1})$. In a space-time diagram, since $x_i > x'_i$ and $x_{i+1} > x'_{i+1}$, this corresponds to two line segments that do not intersect. Therefore, the two agents do not meet. Hence, S_1 and S_2 are incompatible. □

Even though a strategy S is non-redundant and does not contain any lightning, taking the signature of S can introduce redundancies or lightnings.

Definition 11 (Simplified Signature of a Strategy)**.** *Let $S = (x_1, x_2, ..., x_n)$ be a real strategy. Let $\overline{\sigma}(S)$ be the strategy obtained from $\sigma(S)$ by removing all redundancies and all lightnings. The strategy $\overline{\sigma}(S)$ is called the* simplified signature *of S (refer to Figure 2).*

By applying Lemmas 1 3, 4 and 6, we get the following lemma.

Lemma 7. *Let S_1 and S_2 be two real unimodal strategies with simplified signatures $\overline{\sigma}(S_1)$ and $\overline{\sigma}(S_2)$, respectively. If S_1 and S_2 are compatible, then $\overline{\sigma}(S_1) \neq \overline{\sigma}(S_2)$.*

By applying Lemma 7 and Corollary 2, we get the following corollary.

Corollary 4. *There is an optimal algorithm A —which solves the deterministic asynchronous rendez-vous problem on the line— such that all strategies produced by A are integer strategies, and they are monotonic or unimodal.*

We can now prove the following lower bound for real strategies.

Theorem 3. *Suppose that only strategies of the form $(x_1, x_2, ..., x_n)$ are allowed, where $x_i \in \mathbb{R}$ for all $1 \leq i \leq n$. Moreover, suppose that $|L_1| = |L_2| = \ell$. Let A be any algorithm that solves the deterministic asynchronous rendez-vous problem on the line. Asymptotically, we have $\phi(A) \geq \left(\sqrt{6}\log(2)/(2\pi)\right)^2 \ell^2 D \approx 0.07302\, \ell^2 D$.*

Proof. By Corollary 4, we can focus on integer strategies that are monotonic or unimodal. Therefore, the proof of Theorem 2 applies directly. □

5 A General Lower Bound

In this section, we remove the last two assumptions. We explain why our lower bound (refer to Theorem 3) applies to infinite strategies and to the case where $|L_1| \neq |L_2|$.

The proofs of Lemmas 3, 4, 5 and 6 do not use the fact that the strategy is finite. Therefore, without loss of generality, we can consider infinite integer strategies that are monotonic or unimodal. By Definition 8, integer strategies that are unimodal are finite. Thus, let us consider infinite integer strategies that are monotonic. We know that there exist algorithms that produce only finite strategies (refer to [18]). Therefore, it is not worth walking farther than a certain distance D^*. Moreover, in the worst case, an agent have to reach D^* to meet the other agent. Suppose that there is an algorithm which produces infinite strategies. Then the steps defining these strategies must be bounded by D^*. Therefore, these strategies are finite, which is a contradiction.

Suppose that $|L_1| \neq |L_2|$ and let $L_{\min}$ and $L_{\max}$ be the smallest and the longest labels, respectively. Let us revisit Lemma 2 and Corollary 1. In the proof of Lemma 2, we consider an optimal algorithm A which produces a set $\mathcal{S}$ of strategies. We shorten, as much as possible, all strategies in $\mathcal{S}$. At the end, for each strategy $S \in \mathcal{S}$, there exists another strategy $S' \in \mathcal{S}$ which forces the agent following S to complete its last step. Let $\mathcal{S}_{\min} \subseteq \mathcal{S}$ (respectively $\mathcal{S}_{\max} \subseteq \mathcal{S}$) be the set of strategies produced by A from all possible input labels of size $|L_{\min}|$ (respectively of size $|L_{\max}|$). We modify the proof of Lemma 2 in the following way. We shorten all strategies $S \in \mathcal{S}_{\min}$ such that there is a strategy $S' \in \mathcal{S}_{\max}$ which forces the agent following S to complete its last step. Moreover, for each modified strategy $(x_1, x_2, ..., x_n) \in \mathcal{S}_{\min}$, we replace x_n by $x'_n = \lceil x_n \rceil$ if $x_n \geq 0$ or by $x'_n = \lfloor x_n \rfloor$ if $x_n < 0$. Then, we get the following corollary from Definition 5.

Corollary 5. *Suppose that only finite strategies are allowed and that $|L_1| \neq |L_2|$. There exists an optimal algorithm A that solves the deterministic asynchronous rendez-vous problem on the line such that for all strategies $(x_1, x_2, ..., x_n)$ produced by A from an input label of size $|L_{\min}|$, $x_n \in \mathbb{Z}$ and*

$$\phi(A) + 1 \geq \sum_{i=1}^{n} |x_i - x_{i-1}| \,. \tag{4}$$

The number of strategies satisfying (4) is no longer equal to 2^ℓ (refer to the proof of Theorem 2). It is lower bounded by $2^{|L_{\min}|} + 1$. Indeed, there is at least one strategy produced by A from an input label of length $|L_{\max}|$ that satisfies (4). And all strategies produced by A from an input label of length $|L_{\min}|$ satisfy (4). The rest of the proofs of Theorems 2 and 3 are identical.

Since we removed all hypotheses, we can now state the final theorem.

Theorem 4. *Let A be any algorithm that solves the deterministic asynchronous rendez-vous problem on the line. Asymptotically, we have $\phi(A) \geq \left(\sqrt{6}\log(2)/(2\pi)\right)^2 \ell^2 D \approx 0.07302\, \ell^2 D$.*

Acknowledgement
This work was initiated at the *Second Workshop on Geometry and Graphs*, held at the Bellairs Research Institute, March 9-14, 2014. We are grateful to the other workshop participants for providing a stimulating research environment.

References

[1] S. Alpern. Rendezvous search on labeled networks. *Naval Research Logistics (NRL)*, 49(3):256–274, 2002.

[2] S. Alpern, R. Fokkink, L. Gasieniec, R. Lindelauf, and V. S. Subrahmanian. *Search Theory: A Game Theoretic Perspective*. Springer New York, 2013.

[3] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous*. International Series in Operations Research & Management Science. Kluwer Academic Publishers, 2003.

[4] D. Baba, T. Izumi, F. Ooshita, H. Kakugawa, and T. Masuzawa. Space-optimal rendezvous of mobile agents in asynchronous trees. In *Structural Information and Communication Complexity, 17th International Colloquium, SIROCCO 2010, Sirince, Turkey, June 7-11, 2010. Proceedings*, pages 86–100, 2010.

[5] E. Bampas, J. Czyzowicz, L. Gasieniec, D. Ilcinkas, and A. Labourel. Almost optimal asynchronous rendezvous in infinite multidimensional grids. In *Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings*, pages 297–311, 2010.

[6] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. In *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, pages 1181–1196, 2003.

[7] A. Dessmark, P. Fraigniaud, D. R. Kowalski, and A. Pelc. Deterministic rendezvous in graphs. *Algorithmica*, 46(1):69–96, 2006.

[8] P. Flocchini, E. Kranakis, D. Krizanc, N. Santoro, and C. Sawchuk. Multiple mobile agent rendezvous in a ring. In *LATIN 2004: Theoretical Informatics, 6th Latin American Symposium, Buenos Aires, Argentina, April 5-8, 2004, Proceedings*, pages 599–608, 2004.

[9] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous oblivious robots with limited visibility. In *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings*, pages 247–258, 2001.

[10] P. Fraigniaud and A. Pelc. Delays induce an exponential memory gap for rendezvous in trees. *ACM Transactions on Algorithms*, 9(2):17, 2013.

[11] Q. Han, D. Du, J. C. Vera, and L. F. Zuluaga. Improved bounds for the symmetric rendezvous value on the line. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 69–78, 2007.

[12] E. Kranakis, N. Santoro, C. Sawchuk, and D. Krizanc. Mobile agent rendezvous in a ring. In *23rd International Conference on Distributed Computing Systems (ICDCS 2003), 19-22 May 2003, Providence, RI, USA*, pages 592–599, 2003.

[13] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, and U. Vaccaro. Asynchronous deterministic rendezvous in graphs. *Theoretical Computer Science*, 355(3):315–326, 2006.

[14] A. Pelc. Deterministic rendezvous in networks: A comprehensive survey. *Networks*, 59(3):331–347, 2012.

[15] G. Prencipe. Impossibility of gathering by a set of autonomous mobile robots. *Theor. Comput. Sci.*, 384(2-3):222–231, 2007.

[16] R. C. Rhoades. Asymptotics for the number of strongly unimodal sequences. *International Mathematics Research Notices*, 2014(3):700–719, 2014.

[17] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. Intelligent robotics and autonomous agents. MIT Press, 2011.

[18] G. Stachowiak. Asynchronous deterministic rendezvous on the line. In *SOFSEM*, pages 497–508, 2009.

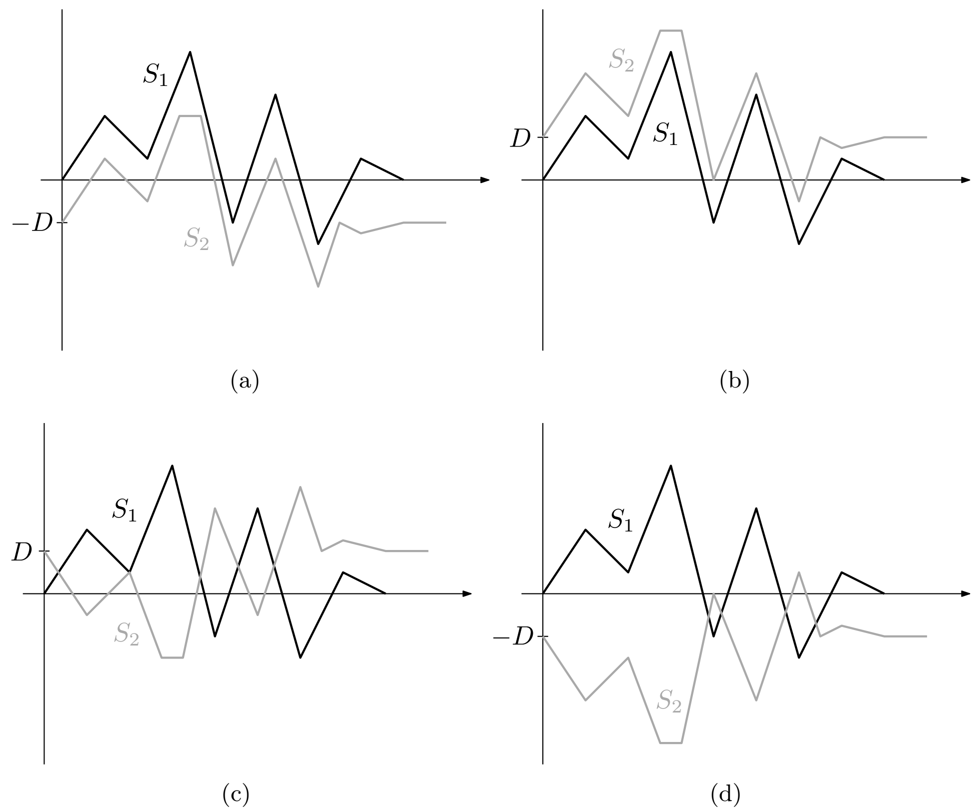


Figure 3: Example of space-time diagram.

A Example of a Space-Time Diagram

In this appendix, we provide an example of space-time diagram (refer to Figure 3). Two agents follow $S_1 = (x_1, x_2, ..., x_7, 0)$ (depicted in black) and $S_2 = (x'_1, x'_2, ..., x'_8, 0)$ (depicted in dark gray), respectively. To help the discussion, suppose that x_1 and x'_1 are both positive. In Figures 3(a) and 3(b), Agents 1 and 2 have the same orientation. In Figures 3(c) and 3(d), Agents 1 and 2 have different orientations. In Figures 3(a) and 3(d), $f_{S_1}(0) = 0 > -D = f_{S_2}(0)$. In Figures 3(b) and 3(c), $f_{S_1}(0) = 0 < D = f_{S_2}(0)$. The same speed is assigned to agent 1 in all figures. The same speed is assigned to agent 2 in all figures. Notice that in Figures 3(a) and 3(b), since the two curves do not intersect, the two agents do not meet each other with these space-time assignments. In Figures 3(c) and 3(d), the two agents do meet each other with these space-time assignments. Therefore, S_1 and S_2 are D-incompatible since there exists a space-time assignment such that two agents following S_1 and S_2, respectively, do not meet.

B Missing Proofs

In this appendix, we provide the missing proofs from Section 3. That is, we prove Lemmas 3 and 4 (refer to Sections B.2 and B.3, respectively), and Corollary 3 (refer to Section B.4). We first introduce *standard space-time assignments* in Section B.1. We do not need this concept for the proofs we present in this appendix. However, they lead to interesting observations and they mildly simplify the presentation of the proof of Lemma 4.

B.1 Standard Space-Time Assignments

Suppose that exactly one agent is moving at a time. Moreover, suppose that when an agent moves, it moves at speed -1 or 1. Then, the time elapsed (since the beginning of the search) corresponds to the total distance travelled by the two agents. Without loss of generality, we can always restrict our attention to such space-time assignments (refer to Propositions 1 and 2). A space-time assignment which satisfies these properties is said to be *standard*.

Definition 12 (Standard Space-Time Assignment)**.** *Let S_1 and S_2 be two strategies, and $\Xi = (f_{S_1}, f_{S_2})$ be a space-time assignment for S_1 and S_2. We say that Ξ is* standard *if there exists a sequence or real numbers $(t_0, t_1, t_2...)$ such that*

- $0 = t_0 < t_1 < t_2 < \cdots$
- *For all $i \geq 1$, f_{S_1} is linear on $[t_{i-1}, t_i]$ with slope -1, 0 or 1.*
- *For all $i \geq 1$, f_{S_2} is linear on $[t_{i-1}, t_i]$ with slope -1, 0 or 1.*
- *For all $i \geq 1$, exactly one of f_{S_1} and f_{S_2} has slope 0 on $[t_{i-1}, t_i]$.*

The sequence $(t_0, t_1, t_2...)$ is called a timeline for Ξ.

We have the following proposition.

Proposition 1. *Let S_1 and S_2 be two compatible strategies. There exists a standard space-time assignment (f_{S_1}, f_{S_2}) for S_1 and S_2 such that $f_{S_1}(t) \neq f_{S_2}(t)$ for every $t < \gamma(S_1, S_2)$ and $f_{S_1}(\gamma(S_1, S_2)) = f_{S_2}(\gamma(S_1, S_2))$.*

To prove Proposition 1, we need the following technical lemma. Intuitively, Lemma 8 states that each non-standard but monotone section of a space-time assignment can be made standard, provided that we stretch time.

Lemma 8. *Let $\delta > 0$ be a real number. Let S_1 and S_2 be two strategies, and $\Xi = (f_{S_1}, f_{S_2})$ be a space-time assignment for S_1 and S_2. Suppose that there exists a time interval (that can consist in a single point) $[s, s']$ such that $|f_{S_1}(t) - f_{S_2}(t)| \geq \delta$ for every $t \in [s, s']$, and both f_{S_1} and f_{S_2} are monotone on $[s, s']$.*

Then there exists a space-time assignment $\Xi' = (f'_{S_1}, f'_{S_2})$ and a time interval $[\tau, \tau']$ such that $\tau = s$ and

- *For all $t \in [0, \tau]$, $f'_{S_1}(t) = f_{S_1}(t)$ and $f'_{S_2}(t) = f_{S_2}(t)$*
- *For all $t \in [\tau, \tau']$, $|f'_{S_1}(t) - f'_{S_2}(t)| \geq \delta$.*
- *For all $t \in [\tau', \infty[$, $f'_{S_1}(t) = f_{S_1}(t - (\tau' - s'))$ and $f'_{S_2}(t) = f_{S_2}(t - (\tau' - s'))$.*
- *During the time interval $[\tau, \tau']$, Ξ' is standard.*

As a direct consequence of Lemma 8, the total distance travelled by the two agents during the time interval $[\tau, \tau']$ (with respect to Ξ') is equal to the total distance travelled by the two agents during the time interval $[s, s']$ (with respect to Ξ). To prove Lemma 8, we take out the part of Ξ corresponding to the time interval $[s, s']$ and we modify it in such a way that

- it becomes standard on the time interval $[\tau, \tau'] = [s, \tau']$ and
- $|f'_{S_1}(t) - f'_{S_2}(t)| \geq \delta$ for all $t \in [\tau, \tau']$.

Then, we glue the part of Ξ corresponding to the time interval $[0, s]$ (respectively $[s', \infty[$) to the left (respectively to the right) of the time interval $[\tau, \tau'] = [s, \tau']$.

Proof. For the proof, we consider the constant function to be increasing. Without loss of generality, let

$$\delta = \min_{t\in[s,s']} |f_{S_1}(t) - f_{S_2}(t)|$$

and recall that $\tau = s$. For all $t \in [0, \tau]$, let $f'_{S_1}(t) = f_{S_1}(t)$ and $f'_{S_2}(t) = f_{S_2}(t)$. To define τ' and (f'_{S_1}, f'_{S_2}) on $[\tau, \tau']$, we consider four cases where f_{S_1} is either increasing or decreasing on $[s, s']$ and f_{S_2} is either increasing or decreasing on $[s, s']$.

1. Suppose that both f_{S_1} and f_{S_2} are increasing. We consider two subcases: either (a) $f_{S_1}(s) > f_{S_2}(s)$ or (b) $f_{S_1}(s) < f_{S_2}(s)$.

   (a) If $f_{S_1}(s) > f_{S_2}(s)$, let $\tau^* = \tau + f_{S_1}(s') - f_{S_1}(s)$ and $\tau' = \tau^* + f_{S_2}(s') - f_{S_2}(s)$. For all $t \in [\tau, \tau^*]$, let $f'_{S_1}(t) = f_{S_1}(s) + (t - \tau)$ and $f'_{S_2}(t) = f_{S_2}(s)$ For all $t \in [\tau^*, \tau']$, let $f'_{S_1}(t) = f_{S_1}(s')$ and $f'_{S_2}(t) = f_{S_2}(s) + (t - \tau^*)$. By construction, (f'_{S_1}, f'_{S_2}) is standard on $[\tau, \tau']$. Moreover, for all $t \in [\tau, \tau^*]$,

   $$f'_{S_1}(t) - f'_{S_2}(t) = f_{S_1}(s) + (t - \tau) - f_{S_2}(s) \geq f_{S_1}(s) - f_{S_2}(s) \geq \delta.$$

   And for all $t \in [\tau^*, \tau']$,

   $$\begin{aligned} f'_{S_1}(t) - f'_{S_2}(t) &= f_{S_1}(s') - (f_{S_2}(s) + (t - \tau^*)) \\ &\geq f_{S_1}(s') - f_{S_2}(s) - (\tau' - \tau^*) \\ &= f_{S_1}(s') - f_{S_2}(s) - \big(f_{S_2}(s') - f_{S_2}(s)\big) \\ &= f_{S_1}(s') - f_{S_2}(s') \\ &\geq \delta. \end{aligned}$$

   Finally,

   $$f'_{S_1}(s) = f'_{S_1}(\tau) = f_{S_1}(s),$$
   $$f'_{S_2}(s) = f'_{S_2}(\tau) = f_{S_2}(s),$$
   $$f'_{S_1}(\tau') = f_{S_1}(s')$$

   and

   $$f'_{S_2}(\tau') = f_{S_2}(s) + (\tau' - \tau^*) = f_{S_2}(s').$$

   (b) If $f_{S_1}(s) < f_{S_2}(s)$, let $\tau^* = \tau + f_{S_2}(s') - f_{S_2}(s)$ and $\tau' = \tau^* + f_{S_1}(s') - f_{S_1}(s)$. For all $t \in [\tau, \tau^*]$, let $f'_{S_1}(t) = f_{S_1}(s)$ and $f'_{S_2}(t) = f_{S_2}(s) + (t - \tau)$ For all $t \in [\tau^*, \tau']$, let $f'_{S_1}(t) = f_{S_1}(s) + (t - \tau^*)$ and $f'_{S_2}(t) = f_{S_2}(s')$. By construction, (f'_{S_1}, f'_{S_2}) is standard on $[\tau, \tau']$. An analysis similar to the one for the case where $f_{S_1}(s) > f_{S_2}(s)$ shows that $f'_{S_2}(t) - f'_{S_1}(t) \geq \delta$ for all $t \in [\tau, \tau']$. It also shows that $f'_{S_1}(s) = f'_{S_1}(\tau) = f_{S_1}(s)$, $f'_{S_2}(s) = f'_{S_2}(\tau) = f_{S_2}(s)$, $f'_{S_1}(\tau') = f_{S_1}(s')$ and $f'_{S_2}(\tau') = f_{S_2}(s')$.

2. Suppose that both f_{S_1} and f_{S_2} are decreasing. This case is similar to the case where both f_{S_1} and f_{S_2} are increasing.

3. Suppose that f_{S_1} is increasing and f_{S_2} is decreasing. We can apply any of the two constructions for the case where both f_{S_1} and f_{S_2} are increasing.

4. Suppose that f_{S_1} is decreasing and f_{S_2} is increasing. This case is similar to the case where f_{S_1} is increasing and f_{S_2} is decreasing.

For all $t \in [\tau', \infty[$, let $f'_{S_1}(t) = f_{S_1}(t - \tau' + s')$ and $f'_{S_2}(t) = f_{S_2}(t - \tau' + s')$. □

Proof. [*Proposition 1*] Suppose that the two agents follow S_1 and S_2, respectively. By the definition of $\gamma(\cdot,\cdot)$, there exists a space-time assignment $\Xi = (f_{S_1}, f_{S_2})$ such that the total distance travelled by the agents before they meet is $\gamma(S_1, S_2)$. If Ξ is standard, then we are done. Therefore, suppose that Ξ is not standard.

Let $T = (t_0, t_1, t_2, ...)$ and $T' = (t'_0, t'_1, t'_2, ...)$ be timelines for f_{S_1} and f_{S_2}, respectively. Suppose that the two agents meet when Agent 1 is executing step x_n and Agent 2 is executing step $x'_{n'}$. Recall that by Definition 2, Agent 1 (respectively Agent 2) is executing step x_n (respectively step $x'_{n'}$) during the time interval $[t_{n-1}, t_n]$ (respectively $[t'_{n'-1}, t'_{n'}]$). Therefore, the meeting occurs during the time interval $[t_{n-1}, t_n] \cap [t'_{n'-1}, t'_{n'}] \neq \emptyset$. Let $(s_0, s_1, ..., s_k)$ be the sorted list obtained by merging $(t_0, t_1, ..., t_n)$ and $(t'_0, t'_1, ..., t'_{n'})$. Since $[t_{n-1}, t_n] \cap [t'_{n'-1}, t'_{n'}] \neq \emptyset$, the meeting occurs during the time interval[1] $[s_{k-2}, s_{k-1}]$.

For all $1 \leq i \leq k-2$, let $[\tau_{i-1}, \tau_i]$ be the time interval obtained by applying Lemma 8 on $[s_{i-1}, s_i]$. This defines a space-time assignment Ξ' for all $t \in [\tau_0, \tau_{k-2}] = [0, \tau_{k-2}]$. By Lemma 8, the total distance travelled by the two agents during the time interval $[\tau_{i-1}, \tau_i]$ (with respect to Ξ') is equal to $\tau_i - \tau_{i-1}$, which is equal to the total distance travelled by the two agents during the time interval $[s_{i-1}, s_i]$ (with respect to Ξ). Therefore, the total distance travelled by the two agents during the time interval $[\tau_0, \tau_i]$ (with respect to Ξ') is equal to $\sum_{j=1}^{i}(\tau_j - \tau_{j-1}) = \tau_i - \tau_0 = \tau_i$. Notice that $\tau_i < \gamma(S_1, S_2)$, otherwise by Lemma 8, the two agents could follow Ξ and meet after having travelled less than $\gamma(S_1, S_2)$ altogether.

We now focus on the time interval $[s_{k-2}, s_{k-1}]$ (with respect to Ξ) during which the meeting occurs. Let $s^* \in [s_{k-2}, s_{k-1}]$ be the time when the meeting occurs. For any $\varepsilon > 0$, we can apply Lemma 8 on $[s_{k-2}, s^* - \varepsilon]$. Consequently, by a continuity argument, we can define τ_{k-1} and Ξ' on $[\tau_{k-2}, \tau_{k-1}]$, where $f'_{S_1}(t) \neq f'_{S_2}(t)$ for every $t \in [\tau_{k-2}, \tau_{k-1}[$ and $f'_{S_1}(\tau_{k-1}) = f'_{S_2}(\tau_{k-1})$. Moreover, by Lemma 8, $\tau_{k-1} = \gamma(S_1, S_2)$.

By construction, Ξ' is standard, $f'_{S_1}(t) \neq f'_{S_2}(t)$ for every $t < \gamma(S_1, S_2)$ and $f'_{S_1}(\gamma(S_1, S_2)) = f'_{S_2}(\gamma(S_1, S_2))$. □

The following proposition, is an incompatible version of Proposition 1.

Proposition 2. *Let S_1 and S_2 be two incompatible strategies. There exists a standard space-time assignment (f_{S_1}, f_{S_2}) for S_1 and S_2 such that $|f_{S_1}(t) - f_{S_2}(t)| \geq \delta(S_1, S_2)$ for every $t \geq 0$.*

Proof. By the definition of $\delta(\cdot,\cdot)$, there exists a space-time assignment $\Xi = (f_{S_1}, f_{S_2})$ such that $|f_{S_1}(t) - f_{S_2}(t)| \geq \delta(S_1, S_2)$ for every $t \geq 0$. If Ξ is standard, then we are done. Therefore, suppose that Ξ is not standard.

Let $T = (t_0, t_1, t_2, ...)$ and $T' = (t'_0, t'_1, t'_2, ...)$ be timelines for f_{S_1} and f_{S_2}, respectively. Let $(s_0, s_1, s_2 ...)$ be the sorted list obtained by merging T and T'.

For all $i \geq 1$, let $[\tau_{i-1}, \tau_i]$ be the time interval obtained by applying Lemma 8 on $[s_{i-1}, s_i]$. This defines a space-time assignment $\Xi' = (f'_{S_1}, f'_{S_2})$ for all $t \in [\tau_0, \infty[= [0, \infty[$.

By Lemma 8, $|f'_{S_1}(t) - f'_{S_2}(t)| \geq \delta(S_1, S_2)$ for every $t \geq 0$. Moreover, by construction, Ξ' is standard. □

[1] This takes into account the fact that some numbers may be duplicate in $(s_0, s_1, ..., s_k)$.

B.2 Proof of Lemma 3

In this section, we prove Lemma 3.

Proof. Let f_{S_1} be any realization for S_1 and $T = (t_0, t_1, ..., t_n)$ be a timeline for f_{S_1}. Then f_{S_1} is also a realization for S_1'. Indeed, it suffices to take $(t_0, t_1, ..., t_i, t_{i+2}, ..., t_n)$ as a timeline for f_{S_1}.

Conversely, let $f_{S_1'}$ be any realization for S_1' and $T' = ({t'}_0, {t'}_1, ..., {t'}_{n-1})$ be a timeline for $f_{S_1'}$. There three cases to consider: (1) S_1 has a redundancy at 0, (2) S_1 has a redundancy at $0 < i < n-2$ or (3) S_1 has a redundancy at $n-2$.

1. Suppose that S_1 has a redundancy at 0. Since $f_{S_1'}$ is monotone on $[{t'}_0, {t'}_1] = [0, {t'}_1]$, the function $f_{S_1'}$ has an inverse on $[0, {t'}_1]$. Let $f_{S_1'}^{-1}(x_1)$ be the number such that $f_{S_1'}\left(f_{S_1'}^{-1}(x_1)\right) = x_1$. Then, $f_{S_1'}$ is also a realization for S_1. Indeed, it suffices to take $\left({t'}_0, f_{S_1'}^{-1}(x_1), {t'}_1, ..., {t'}_{n-1}\right)$ as a timeline for $f_{S_1'}$.

2. Suppose that S_1 has a redundancy at $0 < i < n-2$. This case is similar to the one where S_1 has a redundancy at 0.

3. Suppose that S_1 has a redundancy at $n-2$. This case is similar to the one where S_1 has a redundancy at 0.

Therefore, S_1 and S_2 are compatible if and only if S_1' and S_2 are compatible. Moreover, if S_1 and S_2 are compatible, then $\gamma(S_1, S_2) = \gamma(S_1', S_2)$. And if S_1 and S_2 are incompatible, then $\delta(S_1, S_2) = \delta(S_1', S_2)$. □

B.3 Proof of Lemma 4

In this section, we prove Lemma 4.

Proof. We prove the following equivalent statement: S_1 and S_2 are incompatible if and only if S_1' and S_2 are incompatible.

[$\Longrightarrow$] Suppose that Agents 1 and 2 follow S_1 and S_2, respectively. Let $\Xi = (f_{S_1}, f_{S_2})$ be a space-time assignment for S_1 and S_2 such that Agents 1 and 2 miss each other. Let $(t_0, t_1, t_2, ...)$ be a timeline for f_{S_1}. Without loss of generality, suppose that $f_{S_1}(t_0) = f_{S_1}(0) = 0$.

We construct a space-time assignment $\Xi' = (f_{S_1'}, f'_{S_2})$ such that Agents 1 and 2 miss each other by following S_1' and S_2, respectively. For all $t \in [0, t_i]$, we define $f_{S_1'}(t) = f_{S_1}(t)$ and $f'_{S_2}(t) = f_{S_2}(t)$. In particular, $f_{S_1'}(t_i) = f_{S_1}(t_i) = x_i$ and $f'_{S_2}(t_i) = f_{S_2}(t_i)$. For any $t' > t_i$, if we define $f_{S_1'}(t) = f_{S_1}(t - (t' - t_{i+3}))$ and $f'_{S_2}(t) = f_{S_2}(t - (t' - t_{i+3}))$, then $f_{S_1'}$ and f'_{S_2} do not intersect on the time interval $[t', \infty[$. Indeed, $f_{S_1'}$ and f'_{S_2} are simply a translation by $t' - t_{i+3}$ of f_{S_1} and f_{S_2}, respectively. Moreover, $f_{S_1'}(t') = f_{S_1}(t_{i+3}) = x_{i+3}$ and $f'_{S_2}(t') = f_{S_2}(t_{i+3})$.

Therefore, it suffices to find a time $t' > t_i$ for which we can define $f_{S_1'}$ and f'_{S_2} on the time interval $[t_i, t']$, such that $f_{S_1'}(t) \neq f'_{S_2}(t)$ for all $t \in [t_i, t']$.

Let

$$\delta = \min_{t \in [t_i, t_{i+3}]} |f_{S_1}(t) - f_{S_2}(t)|.$$

We can apply Proposition 2 to f_{S_1} and f_{S_2} restricted to the time interval $[t_i, t_{i+3}]$. We get a time $t^* > t_i$ together with a space-time assignment $\left(\overline{f}_{S_1}, \overline{f}_{S_2}\right)$ that is standard on the time interval $[t_i, t^*]$. Moreover, $\overline{f}_{S_2}(t_i) = f_{S_2}(t_i) = f'_{S_2}(t_i)$, $\overline{f}_{S_2}(t^*) = f_{S_2}(t_{i+3}) = f'_{S_2}(t')$ and

$$\min_{t \in [t_i, t^*]} \left|\overline{f}_{S_1}(t) - \overline{f}_{S_2}(t)\right| = \delta.$$

We construct t', $f_{S_1'}$ and f'_{S_2} from t^*, $\overline{f}_{S_1}$ and $\overline{f}_{S_2}$. We let $t' = t^* + (x_{i+3} - x_i)$. To define $f_{S_1'}$ and f'_{S_2}, we consider two cases: either (1) $x_i \leq x_{i+3}$ or (2) $x_{i+3} < x_i$.

1. Suppose that $x_i \leq x_{i+3}$. We consider two cases: either (a) Agent 2 stays to the left of Agent 1 or (b) Agent 2 stays to the right of Agent 1.

   (a) For all $t \in [t_i, t_i + (x_{i+3} - x_i)]$, let

   $$\begin{aligned} f_{S_1'}(t) &= t - t_i + x_i, \\ f'_{S_2}(t) &= \overline{f}_{S_2}(t_i). \end{aligned}$$

   For all $t \in [t_i + (x_{i+3} - x_i), t^* + (x_{i+3} - x_i)]$, let

   $$\begin{aligned} f_{S_1'}(t) &= x_{i+3}, \\ f'_{S_2}(t) &= \overline{f}_{S_2}(t - (x_{i+3} - x_i)). \end{aligned}$$

   (b) For all $t \in [t_i, t^*]$, let

   $$\begin{aligned} f_{S_1'}(t) &= x_i, \\ f'_{S_2}(t) &= \overline{f}_{S_2}(t). \end{aligned}$$

   For all $t \in [t^*, t^* + (x_{i+3} - x_i)]$, let

   $$\begin{aligned} f_{S_1'}(t) &= t - t^* + x_i, \\ f'_{S_2}(t) &= \overline{f}_{S_2}(t^*). \end{aligned}$$

2. Suppose that $x_{i+3} \leq x_i$. The proof is similar to the previous case.

[$\Longleftarrow$] The proof for the converse statement is similar.

The proof for the second statement of the lemma is a direct consequence of the definition of Ξ'. □

B.4 Proof of Corollary 3

In this section, we prove Corollary 3.

Proof. For all $m \geq 1$,

$$\begin{aligned} u^*(m) \leq u^*_\Sigma(m) &\leq m\, u^*(m) && \text{by the definition of } u^*_\Sigma(\cdot), \\ \log\left(u^*(m)\right) \leq \log\left(u^*_\Sigma(m)\right) &\leq \log(m) + \log\left(u^*(m)\right) \\ \frac{\log\left(u^*(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} \leq \frac{\log\left(u^*_\Sigma(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} &\leq \frac{\log(m) + \log\left(u^*(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}}. \end{aligned}$$

Therefore,

$$\begin{aligned} \lim_{m\to\infty} \frac{\log\left(u^*(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} \leq \lim_{m\to\infty} \frac{\log\left(u^*_\Sigma(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} &\leq \lim_{m\to\infty} \frac{\log(m) + \log\left(u^*(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} \\ 1 \leq \lim_{m\to\infty} \frac{\log\left(u^*_\Sigma(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} &\leq 1 && \text{by Theorem 1.} \end{aligned}$$

Thus,

$$\lim_{m\to\infty} \frac{\log\left(u^*_\Sigma(m)\right)}{\frac{\pi}{6}\sqrt{24m-1}} = 1,$$

from which

$$\log\left(u^*_\Sigma(m)\right) \sim \pi\sqrt{24m-1}/6. \quad \square$$