Partial is Full

Bernadette Charron-Bost[1], Matthias Függer[2], Jennifer L. Welch[3], and Josef Widder[3]
[1] LIX, Ecole polytechnique
[2] TU Wien
[3] Texas A&M University

April 22, 2011

Abstract

Link reversal is the basis of several well-known routing algorithms [10, 16, 12]. In these algorithms, logical directions are imposed on the communication links and a node that becomes a sink reverses some of its incident links to allow the (re)construction of paths to the destination. In the Full Reversal (FR) algorithm [10], a sink node reverses *all* its incident links. Other schemes have also been considered, in which a sink node reverses only some of its incident links; a notable example is the Partial Reversal (PR) algorithm [10]. Prior work [7] has introduced a generalization, called LR, of link-reversal routing, including FR and PR that uses binary labels on links and has given an exact analysis of the work complexity of this generalization.

In this paper, we show that every execution of LR on any link-labeled input graph corresponds, in a precise sense, to an execution of FR on a transformed graph. Thus, all the link reversal schemes captured by LR, in which a sink's incident edges are only partially reversed, can be reduced to FR, indicating that "partial is full." The correspondence preserves the work and time complexities. By referring to a companion paper on the time complexity of FR [6], we can, for the first time, obtain the *exact* time complexity for LR, and by specialization for PR. In fact, we provide a direct expression for the LR time complexity as function of simple characteristics of the link-labeled input graph, by identifying a key quantity called the *routing failure* of chains in the graph.

1 Introduction

Gafni and Bertsekas [10] proposed a family of distributed algorithms for routing to a destination node in a wireless network subject to communication link failures. In these networks, virtual directions are assigned to the links between nodes, and messages are sent over the links according to the virtual directions. If the directed graph induced by the virtual directions has no cycles and has the destination as the only sink, then such a scheme ensures that forwarded messages reach the destination. To accommodate link failures, it might be necessary to *reverse* the virtual directions of some of the links in order to reestablish the properties that will ensure proper routing of messages. The algorithms in [10] identified the formation of sinks as a detriment to achieving routing; whenever a node (other than the destination) in the graph becomes a sink, either because of the loss of an incident link or the reversal of the direction of an incident link, the node takes action so that it is no longer a sink.

In the first algorithm in [10], called *Full Reversal (FR)*, a sink node reverses *all* its incident links. However, other reversal schemes are also possible in which some, but not all, links incident on a sink are reversed. One particular such partial reversal scheme, called *Partial Reversal (PR)*, is proposed in [10]; roughly speaking, in PR a sink reverses only those incident links that have not been reversed since the last time the node was a sink. Implementations of these two algorithms assigned unbounded height values to the nodes, viewed a link as directed from the endpoint with higher height to the endpoint with lower height, and brought about link reversal by having a sink node increase its heights relative to its neighbors' heights according to some given rule. Various types of heights and of height-increasing rules result in a family of link reversal routing algorithms that include FR and PR. The link-reversal approach to routing is the basis of a couple of well-known routing protocols for mobile ad hoc networks [16, 12]. It has been adapted further to solve leader election [14, 9, 11], and the same general idea has also been exploited in solutions to other problems, including resource allocation [5, 2, 13], distributed queueing [18, 1], and mutual exclusion [17, 15, 19].

However, a complete analysis of the performance of link reversal routing algorithms has not yet been done, even for the original case FR and PR. Such work was pioneered by Busch *et al.* [3, 4]. In these papers, the authors considered the height-based implementations of FR and PR, and analyzed the *work* complexity measure, which is the total number of reversals done by all the nodes. An exact formula for the work complexity of every node in every graph was derived for FR, while for PR, asymptotically tight bounds were derived. The other natural complexity measure of link reversal algorithms is *time*, which is the number of iterations required until termination in "greedy" executions [2], where, in each iteration, all sinks take steps. Greedy executions are those with the highest possible parallelism. Clearly, global work complexity is the number of iterations in completely sequential executions, and so is at least equal to global time complexity. For both FR and PR, that implies a quadratic upper bound on global time complexity. Concerning time complexity, Busch *et al.* [3, 4] only obtained limited results: for each of FR and PR, they described a family of graphs on which the algorithm achieves quadratic global time complexity. However, these results were not conclusive, in that they did not demonstrate any significant differences between FR and PR.

Recently, Charron-Bost *et al.* [7] have taken a different approach to implementing link reversal, which does not use node heights. Instead, they considered an initial directed graph, assigned binary labels to the links of the graph, and used the labels on the incident links to decide which links to reverse and which labels to change. This generalized algorithm, called LR, can be specialized to different specific algorithms, including FR and PR, through different initial labelings of the links; the two initial labelings that are globally uniform correspond to FR and PR, and non-uniform initial labelings correspond to other partial reversal schemes. This approach allowed an exact analysis of the work complexity of each node in LR for link-labeled graphs. In particular, the exact work complexities for each node in FR and PR were obtained. A closer analysis of the work-complexity tradeoffs between FR and PR based on game theory appears in [8]. This analysis showed that PR is better than FR regarding work complexity in an interesting way: FR can be worse than the optimal link labeling by a factor that is linear in the number of nodes of the graph, whereas PR is never more than twice as bad as the optimal link labeling.

In this paper, we address the open problem of the exact time complexity of LR. In a companion paper [6], we used the fact that FR executions can be modeled as *linear* dynamical systems, in order to obtain an exact expression for the time complexity of an arbitrary node in an arbitrary graph, as a function of some simple properties of the input graph. The behavior of LR in general is not linear, and the method in [6] cannot directly work for LR. We overcome this difficulty by establishing a general reduction of LR to FR: we show that every execution of LR from any link-labeled input graph $G^\dagger$ corresponds to an execution of FR from a transformed graph $T(G^\dagger)$. The main point in the reduction is that the correspondence preserves the work and time complexities. Thus, the time complexity of LR (and in particular, of PR) from a graph $G^\dagger$ can be obtained by transforming $G^\dagger$ into $T(G^\dagger)$ and then applying the FR time complexity results from [6] to $T(G^\dagger)$.

In the FR case, the work complexity of node i, denoted w_i, is shown [3, 4, 7, 6] to be the minimum, over all chains from the destination to i, of the number of links in the chain that are directed away from the destination. Also, it is shown in [6] that the time complexity of node i is one greater than the maximum length of all chains that end at node i (no matter where they start) and have $w_i - 1$

links in the chain directed toward i. So it turns out that the key quantity for both work and time complexity of FR is the number r of links in a chain that are directed toward the last node in the chain. For LR, we introduce a new chain potential in link-labeled graphs, which we call the *routing failure*, that is a generalization of the quantity r for FR. We establish formulas for the time and work complexity of LR which are analogous to those for FR when replacing r with the routing failure of the chain. In this way, we provide a general expression for the time complexity of LR, and thus also of PR, as a function of characteristics of the input graph $G^\dagger$ only. Interestingly, we show that the routing failure of a chain c in a link-labeled graph $G^\dagger$ is actually a simple function of the values of r for the various chains that correspond to c in the transformed graph $T(G^\dagger)$, thus completing the reduction of LR to FR. Hence, all the link reversal schemes captured by LR, in which a sink's incident edges are only partially reversed, can be reduced to Full Reversal, indicating that "partial is full."

2 Preliminaries

We consider directed graphs with $N+1$ nodes, one of which is a specific node called the *destination* node, and a set E of directed links. For convenience, we refer to the destination as node 0 and we let $V = \{1, 2, \ldots, N\}$ be the remaining nodes.

The link (i, j) in E is said to be *incident* on both i and j, and to be *outgoing* from i and *incoming* to j. Let In_i and Out_i denote the set of incoming and outgoing neighbors of i, respectively. A node i is said to be a *sink* if and only if all its incident links are incoming to i. A *chain* is a sequence c of nodes $i_0, \ldots, i_k$, $k \geq 0$, such that for all m, $0 \leq m < k$, either (i_m, i_{m+1}) is in E or (i_{m+1}, i_m) is in E; the *length* of the chain c, which we denote by $\lambda(c)$, is defined to be k. We denote by c^{-1} the chain $i_k, \ldots, i_0$ that is the reverse of chain c. We denote by $\mathcal{C}(i, j, G)$ the set of all chains of finite length in G that start with node i and end with node j; we denote by $\mathcal{C}(\rightarrow j, G)$ the set of all chains of finite length in G that end with node j (no matter where they start). A *path* is a chain $i_0, \ldots, i_k$ such that for all m, $0 \leq m < k$, (i_m, i_{m+1}) is in E.

G is *connected* if for any two nodes i and j, there is a chain from i to j. A node i is *good in G* if there is a path from i to 0, otherwise i is *bad.* We say that G is *0-oriented* if every node is good in G.

A chain $c = i_0, \ldots, i_k$, where $k \geq 1$, is *closed* if $i_k = i_0$. A *cycle* is the subgraph of G induced by a simple closed chain that is a path; in a cycle, all the links point in the same direction. If G has no cycles, then it is *acyclic.*

A directed graph $G = \langle V \cup \{0\}, E\rangle$ is defined to be *routable* if it is connected, it has no self-loops, and for each (i, j) in E, (j, i) is not in E.[1]We only consider input graphs that are routable.

Given two routable graphs $G_1 = \langle V_1 \cup \{0\}, E_1\rangle$ and $G_2 = \langle V_2 \cup \{0\}, E_2\rangle$, G_2 is called a *reorientation* of G_1 if G_1 and G_2 have the same undirected support, in the sense that $V_1 = V_2$, and there is a bijection f from E_1 to E_2 such that $f((i, j))$ is either (i, j) or (j, i). The notion of being a reorientation is symmetric, in that G_1 is a reorientation of G_2 if and only G_2 is a reorientation of G_1.

Then we consider the *Routing Problem* [10]: given a routable graph (to 0) G, find a reorientation of G that is 0-oriented. If G is 0-oriented, then there is no sink in G other than 0. As shown in [10], the converse holds if G is acyclic:

Proposition 1. *Let G be a directed, connected and acyclic graph with a specific node 0. The following conditions are equivalent: (i) G is 0-oriented, (ii) node 0 is the sole sink of G.*

This leads to the link reversal strategy to solve the Routing problem: it consists in "fighting" sinks — that is, changing the direction of links incident to sinks — while maintaining acyclicity. To

[1]The routing algorithms under consideration assign dynamically changing virtual directions to the edges of an undirected graph that models bidirectional communication channels between nodes. Thus it does not make sense for both (i, j) and (j, i) to be in the corresponding directed graph.

implement this strategy in a distributed setting, we assume that a process is associated with each node of the graph, and for each node i, the process associated with i can both (a) determine the direction of all the links incident to i, and (b) change the direction of all incoming links incident to i. In this case, any process associated with a sink can locally detect that the graph is *not* destination-oriented, and then it reverses some of its incoming links in order to be no longer a sink. It remains to verify that this scheme maintains acyclicity and terminates. In the sequel, we shall work within the context of (a) and (b), and we shall identify a node i with the process associated with i.

2.1 The LR Algorithm

Following the above strategy, Charron-Bost *et al.* [7] proposed a general algorithm for the routing problem, called *LR* for *Link Reversal*, in which binary labels are assigned to the links; when a node takes a step, it uses a local rule to update the directions and labels on the incident links. The LR algorithm unifies the Full Reversal and Partial Reversal algorithms given by Gafni and Bertsekas [10] just by varying the input binary labeling of LR, while the local rule is always the same.

In more detail, we consider a graph $G = \langle V \cup \{0\}, E \rangle$ and a function μ from E to $\{\mathbb{0}, \mathbb{1}\}$, which assigns the label $\mathbb{0}$ or $\mathbb{1}$ to each link of G. We denote the resulting link-labeled graph as $G^\dagger = \langle G, \mu \rangle$, and G is called the *support* of $G^\dagger$. The dagger superscript will be used throughout to indicate such a link-labeled graph. If the support of $G^\dagger$ is routable, then $G^\dagger$ is said to be a *link-labeled routable graph*. All the definitions given in the above subsection still hold for link-labeled graphs.

Each node i other than 0 can apply the following (mutually exclusive) rules if it is a sink:

R1: If at least one link incident on i is labeled $\mathbb{0}$, then all the links incident on node i that are labeled with $\mathbb{0}$ are reversed, the other incident links are not reversed, and the labels on all the incident links are flipped.

R2: If all the links incident on node i are labeled $\mathbb{1}$, then all the links incident on i are reversed, but none of their labels is changed.

When node i applies R1 or R2, then i is said to *take a step*.

Given a link-labeled routable graph $G^\dagger$, any nonempty set S of sinks in $G^\dagger$ is said to be *applicable to* $G^\dagger$, as each node in S may "simultaneously" take a step. Since two neighboring nodes cannot both be a sink, the resulting graph depends only on S and $G^\dagger$, and is denoted by $S.G^\dagger$. In case $S = \{i\}$, we write $i.G$ for short, instead of $\{i\}.G$. By induction, we easily generalize the notion of applicability to $G^\dagger$ for a sequence $\mathbf{S}$ of nonempty sets of nodes, and we denote the resulting graph by $\mathbf{S}.G^\dagger$.

An *execution* of the LR algorithm from a link-labeled routable graph $G_0^\dagger$ is a sequence $G_0^\dagger$, S_1, ... $G_{t-1}^\dagger$, S_t, ... of alternating link-labeled routable graphs and sets of nodes satisfying:
1. For each $t \geq 1$, S_t is applicable to $G_{t-1}^\dagger$.
2. For each $t \geq 1$, $G_t^\dagger$ equals $S_t.G_{t-1}^\dagger$.
3. If the sequence is finite, then it ends with a graph that contains no sinks other than 0.

The transition from $G_{t-1}^\dagger$ to $G_t^\dagger$ is called *iteration* t, and node i in S_t is said to *take a step at iteration* t.

Since each S_t can be any nonempty subset of the sink nodes in $G_{t-1}^\dagger$ other than 0, there are multiple possible LR executions starting from the same initial graph: the flexibility for the sets S_t captures asynchronous behaviors of the nodes. We may thus model a range of situations, with one extreme being the maximally concurrent situation in which all sinks take a step at each iteration, and the other extreme being a single node taking a step in each iteration. An LR execution $G_0^\dagger$, S_1, ..., $G_{t-1}^\dagger$, S_t, ... that exhibits the maximal amount of parallelism is called *greedy*, i.e., each S_t consists of all the sinks in $G_{t-1}^\dagger$. Since the algorithm is deterministic, there is exactly one greedy LR execution for a given initial link-labeled routable graph $G_0^\dagger$.

In an LR execution in which the links are all initially labeled with $\mathbb{1}$, links remain labeled with $\mathbb{1}$ and a sink can execute exclusively R2, i.e., it makes a *full* reversal of its incident links. Otherwise, some links are initially labeled with $\mathbb{0}$, and certain nodes may apply R1, and then make a *partial* reversal as they reverse a strict subset of their incident links.

We now review the basic properties of the LR algorithm shown in [7].

Theorem 1. *Let $G^\dagger$ be any link-labeled routable graph, and G its directed support.*
(a) Any LR execution from $G^\dagger$ is finite.
(b) Each node takes the same number of steps in every LR execution from $G^\dagger$. Moreover, the final graph depends on $G^\dagger$ only.
(c) If all links are labeled with $\mathbb{1}$ in $G^\dagger$, then the LR executions from $G^\dagger$ are the executions from G of the Full Reversal algorithm.
(d) If all links are labeled with $\mathbb{0}$ in $G^\dagger$, then the executions from $G^\dagger$ are the executions from G of the Partial Reversal algorithm.

Theorem 1.c and 1.d assert that the Full and Partial Reversal algorithms from [10] discussed in the introduction correspond to two specific link labeling initializations for the LR algorithm, namely the globally uniform labelings with $\mathbb{1}$ and with $\mathbb{0}$. This is why the LR executions with such initial labelings are called *FR executions* and *PR executions*, respectively.

2.2 Complexity measures

Given any LR execution $G_0^\dagger$, S_1, $G_1^\dagger$, S_2, ..., we know from Theorem 1.a that the execution is finite, and thus ends with $G_k^\dagger$ for some k. The *work complexity of node i* in the execution, denoted w_i, is the number of steps taken by i; formally, $w_i = |\{1 \leq t \leq k : i \in S_t\}|$. Since the execution is finite, so is each w_i. Moreover, by Theorem 1.b, w_i depends only on the initial link-labeled routable graph and not on the order in which nodes take steps. We define the *work complexity* to be $w = \sum_{i=0}^{N} w_i$.

The *time complexity* is measured by counting the number of iterations in greedy executions: assuming that between any two consecutive iterations in a greedy LR execution $G_0^\dagger$, S_1, ... S_k, $G_k^\dagger$ one time unit elapses, the *time complexity of node $i \in V \cup \{0\}$*, denoted by θ_i, is the last iteration when i takes a step. Formally, $\theta_i = 0$ if $w_i = 0$, and $\theta_i = \max\{t : i \in S_t\}$ otherwise. We define the *time complexity* to be $\theta = \max\{\theta_i : i \in V \cup \{0\}\}$.

2.3 The FR executions

Theorem 1.a shows that starting with any link-labeled routable graph, whether acyclic or not, the LR algorithm converges, i.e., reaches a graph with no sink other than 0. By Proposition 1, the LR algorithm is a solution to the Routing Problem when the final graphs are acyclic. Moreover, one easily observes that acyclicity of the graph is maintained in FR executions. Consequently, LR solves the Routing Problem for the globally uniform labeling with $\mathbb{1}$ when the initial routable graph is acyclic.

In a companion paper [6], we have analyzed work and time complexity for FR. The formulas we obtained are stated in terms of an important characteristic of the input graph G: for any chain c in G, let $r(c)$ be the number of links in c that are directed the "right" way, i.e., the number of consecutive nodes i_k, i_{k+1} in c such that (i_k, i_{k+1}) is a link of G. We have shown that:

Theorem 2. *In any execution of Full Reversal from any acyclic routable graph G, for any node i, the work complexity of node i is $w_i = \min\{r(c) \colon c \in \mathcal{C}(0, i, G)\}$.*

Theorem 3. *If w_i is the work of some node i in V in executions of Full Reversal from any routable graph G, then the time complexity θ_i of node i in the greedy execution is 0 if $w_i = 0$, and otherwise $\theta_i = \max\{\lambda(c) \colon c \in \mathcal{C}(\rightarrow i, G) \wedge r(c) = w_i - 1\} + 1$.*

Theorem 2 has been previously proved by Busch *et al.* [3, 4], using a node layering specific to the Full Reversal algorithm. In [7], Charron-Bost *et al.* gave a general formula for the work complexity of LR, which also provides the expression in Theorem 2 when specializing it to FR executions. The authors introduced for link-labeled graphs, a chain potential Φ that regularly decreases in LR executions, and a property called (AC), which is shown to guarantee the acyclicity of graphs. As we will show, Φ and the (AC) property are actually the translations of the potential r and the acyclicity property in our reduction of LR to FR.

3 Reduction to FR executions

In this section, we explain how to transform any routable link-labeled graph $G^\dagger$ into a routable (unlabeled) graph H such that any LR execution from $G^\dagger$ is equivalent to an FR execution from H. The proof of the execution equivalence uses diagram chasing arguments.

3.1 The graph transformation

By inspection of the LR algorithm and its two rules R1 and R2, we observe that each node in V which initially has at least one incoming link labeled with $\mathbb{0}$, and at least one link which either is outgoing, or is incoming and labeled with $\mathbb{1}$ always execute R1, and thus reverses only a *proper subset* of its incoming links when it is a sink. More precisely, the incident links of such a node can be partitioned into two sets, where the links in one set are reversed at odd steps and the others at even steps. These nodes are called *double nodes.* On the contrary, every other node in V either initially is a sink with all incoming links labeled with $\mathbb{0}$, or initially has all its incoming links labeled by $\mathbb{1}$, and always reverses the direction of *all* its incident links. Such a node is called a *single node.* For convenience, the destination node 0 is supposed to be a single node. We are thus led to partition the set of nodes $V \cup \{0\}$ into the two disjoint subsets of nodes $\mathcal{S}(G^\dagger)$ and $\mathcal{D}(G^\dagger)$, which are the set of single nodes and the set of double nodes in $G^\dagger$, respectively. The above sketched argument shows the following invariance property of this bi-partitioning (a complete proof is given in [7]):

Theorem 4 ([7])**.** *For any LR execution $G_0^\dagger$, $S_1 \ldots G_k^\dagger$, and for any t, $0 \le t \le k$, $\mathcal{D}(G_t^\dagger) = \mathcal{D}(G_0^\dagger)$ and $\mathcal{S}(G_t^\dagger) = \mathcal{S}(G_0^\dagger)$.*

In the FR case, we easily check that $\mathcal{D}$ is empty. The exclusive use of R2 in this case leads to a regular interleaving given in [6]:

Proposition 2. *Consider any routable graph G, any FR execution from G, and any node i in G.*
(a) Between two consecutive steps by a node i other than 0, each neighbor of i takes exactly one step.
(b) If i takes at least one step, then before the first step by i, each node $j \in In_i$ takes no step and each node $k \in Out_i$ takes exactly one step.

The point in general LR executions is to cope with the double nodes: between two steps of a double node i, not all i's neighbors take steps. From the viewpoint of one of i's neighbors j that is a single node, i takes two steps between two steps of j.

Hence Proposition 2 does not hold anymore in the general LR case. However, the LR interleaving is always regular. To see that, we distinguish between odd and even steps of each double node i, and hence introduce two types of steps by i which alternate. We consider only one type of steps for single nodes. We then observe for any two neighbors i and j that *between two steps of a certain type by node i there is exactly one step of each type by node j.* That yields an alternating pattern of typed steps of neighbors, similar to the alternating pattern of the FR interleaving, and an initial offset of typed steps is similar to the FR initial offset.

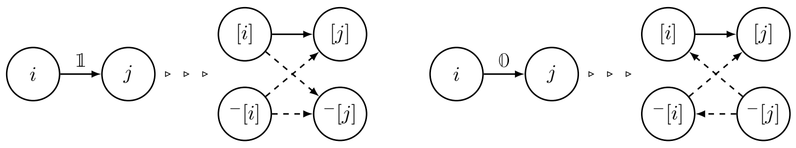

Figure 1: Transformation of link (i, j) in $G^\dagger$ to a collection of links in $T(G^\dagger)$.

This leads to the idea of simulating an LR execution from a routable link-labeled graph $G^\dagger$ with an FR execution from a directed graph H which is obtained from $G^\dagger$ by splitting each double node into two distinct nodes, and by connecting the resulting nodes with directed links so that the FR interleaving given by Proposition 2 corresponds to the regular interleaving in the LR executions from $G^\dagger$ described just above.

Formally, we consider the graph transformation T that maps any link-labeled graph $G^\dagger$ to a directed graph $H = T(G^\dagger)$. In order not to confuse nodes from $G^\dagger$ with the nodes from $T(G^\dagger)$, we keep the convention that nodes in $G^\dagger$ are from $\{0, \dots, N\}$ and introduce the convention that nodes in $T(G^\dagger)$ are strings of the form $[i]$ or $^-[i]$, where i is in $\mathbb{N}$. For a set $S \subseteq \mathbb{N}$, we define $[S] = \{[i] \colon i \in S\}$.

Definition 1. *Given a routable link-labeled graph $G^\dagger = \langle V \cup \{0\}, E, \mu\rangle$, we define $T(G^\dagger) = \langle V^T \cup \{[0]\}, E^T\rangle$ to be the routable graph with destination node $[0]$, $V^T = \{[i] \colon i \in V\} \cup \{^-[i] \colon i \in \mathcal{D}(G^\dagger)\}$, and E^T defined as follows: for any link $e = (i, j) \in E$ labeled with $\mu = \mu(e)$,*

1. *$([i], [j])$ is in E^T;*
2. *if $i \in \mathcal{D}(G^\dagger)$, then $(^-[i], [j])$ is in E^T;*
3. *if $j \in \mathcal{D}(G^\dagger)$, then*

   if $\mu = \mathbb{0}$, then $(^-[j], [i])$ is in E^T, else $([i], ^-[j])$ is in E^T;
4. *if both $i \in \mathcal{D}(G^\dagger)$ and $j \in \mathcal{D}(G^\dagger)$, then*

   if $\mu = \mathbb{0}$, then $(^-[j], ^-[i])$ is in E^T, else $(^-[i], ^-[j])$ is in E^T.

Figure 1 illustrates the transformation of a link (i, j) in $G^\dagger$. Links drawn with dashed arrows only exist if the incident nodes $^-[i]$ or $^-[j]$ exist, that is, if i or j is a double node. Through the transformation T, each labeled link e in E is mapped into a set of links in E^T which we denote by $T(e)$. Further, if e is a link in $G^\dagger$ and i is a sink in $G^\dagger$, then we denote by $i.e$ the corresponding link in $i.G^\dagger$. Similarly, we denote by $[i].T(e)$ the set of links in $[i].T(G^\dagger)$ corresponding to link $e \in G^\dagger$.

From the very definition of $T(G^\dagger)$, we easily establish that:

Proposition 3. *A node i in $G^\dagger$ is a sink in $G^\dagger$ if and only if $[i]$ is a sink in $T(G^\dagger)$. Moreover, if $j \in \mathcal{D}(G^\dagger)$, then the node $^-[j]$ is not a sink in $T(G^\dagger)$.*

Hence, if i is a sink in $G^\dagger$, then $\{i\}$ is applicable to $G^\dagger$, and $\{[i]\}$ is applicable to $T(G^\dagger)$.

3.2 Dynamics in LR executions

In this section, we show that the dynamics in any LR execution from $G^\dagger$ are in some sense equivalent to the dynamics in an FR execution from $T(G^\dagger)$. For that, we use graph isomorphisms: given two directed graphs $G = \langle V, E\rangle$ and $G' = \langle V', E'\rangle$, and a bijection α, from V to V', we denote by $G \simeq_\alpha G'$ that α is an *isomorphism* from G to G'. This notation is not symmetric in general as $G \simeq_\alpha G'$ if and only if $G' \simeq_{\alpha^{-1}} G$. Recall that if $G \simeq_\alpha G'$ and $G' \simeq_\beta G''$, then $G \simeq_{\beta\circ\alpha} G''$. First we easily check the following proposition on isomorphic routable graphs:

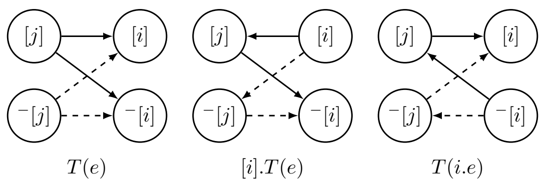

Figure 2: Case 2 (a): $i \in \mathcal{D}(G^\dagger)$ and $\mu(e) = \mathbb{1}$.

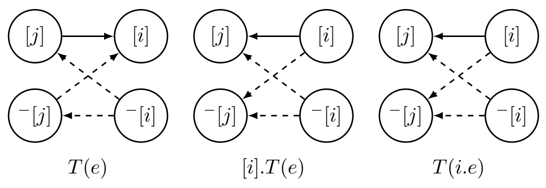

Figure 3: Case 2 (b): $i \in \mathcal{S}(G^\dagger)$ or $\mu(e) = 0$.

Proposition 4. *For any two graphs G and G' routable to destination u and v, respectively, such that $G \simeq_\alpha G'$ and $v = \alpha(u)$, it holds that if set S is applicable to G, then $\alpha(S)$ is applicable to G' and $S.G \simeq_\alpha \alpha(S).G'$.*

For comparing FR and LR dynamics, we first prove that for any set of sinks S in $G^\dagger$, the graphs $[S].T(G^\dagger)$ and $T(S.G^\dagger)$ are isomorphic. For that, we introduce some additional notation. Given any routable graph $G^\dagger$ and any node i in $G^\dagger$, we define the permutation of the set of nodes in $T(G^\dagger)$, denoted $\tau_{G^\dagger,i}$, as the identity function if i is a single node, and as just permuting $[i]$ and $^-[i]$ if i is a double node. Obviously, $\tau_{G^\dagger,i}$ is bijective. Moreover, for any graph $G^\dagger$, any two functions $\tau_{G^\dagger,i}$ and $\tau_{G^\dagger,j}$ commute, and we may thus unambiguously define $\tau_{G^\dagger,S}$ for a set S of nodes in $G^\dagger$ by the composition of the functions $\tau_{G^\dagger,i}$, i in S, irrespective of the order. From this, and since for each i in S, $(\tau_{G^\dagger,i})^{-1} = \tau_{G^\dagger,i}$, we deduce that $(\tau_{G^\dagger,S})^{-1} = \tau_{G^\dagger,S}$. Note that $\tau_{G^\dagger,i}$ depends on $G^\dagger$ only in the partitioning of nodes into single and double nodes. By Theorem 4, this bi-partitioning is constant during any LR execution. Hence $\tau_{S.G^\dagger,i} = \tau_{G^\dagger,i}$. We begin with the case S is a singleton $\{i\}$:

Lemma 1. *If i is a sink in $G^\dagger$ and $\tau = \tau_{G^\dagger,i}$, then $[i]$ is applicable to $T(G^\dagger)$ and $[i].T(G^\dagger) \simeq_\tau T(i.G^\dagger)$.*

Proof. Clearly, the directed graphs $[i].T(G^\dagger)$ and $T(i.G^\dagger)$ have the same set of nodes. So it remains to show that for any pair (u,v) of nodes in $T(G^\dagger)$, (u,v) is a link of $[i].T(G^\dagger)$ if and only if $(\tau(u),\tau(v))$ is a link of $T(i.G^\dagger)$. In other words, for any pair (u,v) of nodes in $T(G^\dagger)$ and any link e of $G^\dagger$, (u,v) is in $[i].T(e)$ if and only if $(\tau(u),\tau(v))$ is in $T(i.e)$. For that, we consider two cases:

1. Link e is not incident to i. Then, $i.e = e$ and $T(i.e) = T(e)$. Moreover, no link in $T(e)$ is incident to $[i]$, and so $[i].T(e) = T(e)$. Obviously, (u,v) is in $T(e)$ if and only if $(\tau(u),\tau(v))$ is in $T(e)$.
2. Link e is incident to i, i.e., $e = (j,i)$ for some node j in $G^\dagger$. There are two cases to consider.
   - $i \in \mathcal{D}(G^\dagger)$ and e is labeled with $\mathbb{1}$ in $G^\dagger$. In this case, $i.e = (j,i)$ and $i.e$ is labeled with $\mathbb{0}$ in $i.G^\dagger$. We easily check that $T(i.e)$ and $[i].T(e)$ are as in Figure 2, where the links drawn with dashed arrows exist only when $j \in \mathcal{D}$.
   - $i \in \mathcal{S}(G^\dagger)$ or e is labeled with $\mathbb{0}$ in $G^\dagger$. Then, $i.e = (i,j)$ and $i.e$ is labeled with $\mathbb{1}$ in $i.G^\dagger$. The sets of links $T(i.e)$ and $[i].T(e)$ are as in Figure 3, where the links drawn with dashed arrows exist only when their extremities are both in $\mathcal{D}$.

In both cases, we easily check that (u,v) is in $[i].T(e)$ if and only if $(\tau(u),\tau(v))$ is in $T(i.e)$. ☐

Lemma 2. *If S is a nonempty set of sinks in $G^\dagger$ and $\tau = \tau_{G^\dagger,S}$, then $[S]$ is applicable to $T(G^\dagger)$ and $[S].T(G^\dagger) \simeq_\tau T(S.G^\dagger)$.*

To combine iterations of Lemma 2, let $\mathbf{S} = (S_t)_{1\le t\le k}$ be a sequence of k, $k \ge 1$, sets of nodes applicable to some $G^\dagger$. Let $\mathbf{\Sigma} = (\Sigma_t)_{1\le t\le k}$ be the sequence of k sets of nodes in $T(G^\dagger)$ defined by:

$$\begin{aligned} \Sigma_1 &= [S_1] \\ \Sigma_t &= \left(\tau_{G^\dagger,S_1} \circ \cdots \circ \tau_{G^\dagger,S_{t-1}}\right)([S_t]), \quad 2 \le t \le k. \end{aligned} \tag{1}$$

$$\begin{tikzcd}
G_0^\dagger \arrow[r, "S_1 \text{ applicable}"] \arrow[ddd, dotted, "T"] & S_1.G_0^\dagger \arrow[r, "S_2 \text{ applicable}"] \arrow[dd, dotted, "T"] & S_2.(S_1.G_0^\dagger) \arrow[d, dotted, "T"] \\
 & & T(S_2.S_1.G_0^\dagger) \\
 & T(S_1.G_0^\dagger) \arrow[r, "{[S_2]} \text{ applicable}"] & {[S_2]}.T(S_1.G_0^\dagger) \arrow[u, squiggly, "\tau_{G_0^\dagger,S_2}"'] \\
T(G_0^\dagger) \arrow[r, "\Sigma_1={[S_1]} \text{ applicable}"] & \Sigma_1.T(G_0^\dagger) \arrow[r, "\Sigma_2=\tau_{G_0^\dagger,S_1}({[S_2]}) \text{ applicable}"] \arrow[u, squiggly, "\tau_{G_0^\dagger,S_1}"'] & \Sigma_2.\Sigma_1.T(G_0^\dagger) \arrow[u, squiggly, "\tau_{G_0^\dagger,S_1}"']
\end{tikzcd}$$

Figure 4: Relation of the first two iterations in an LR execution and its corresponding FR execution.

Lemma 3. *If $G^\dagger$ is a link-labeled routable graph, $\mathbf{S}$ is a sequence of length k applicable to $G^\dagger$, and $\tau = \tau_{G^\dagger,S_1} \circ \cdots \circ \tau_{G^\dagger,S_k}$, then $\mathbf{\Sigma}$ is applicable to $T(G^\dagger)$, and $\mathbf{\Sigma}.T(G^\dagger) \simeq_\tau T(\mathbf{S}.G^\dagger)$.*

The next theorem shows the relation between the LR execution and its corresponding FR execution. Figure 4 depicts this relation for the first two iterations, and it shows how graph isomorphisms combine in the commutative diagrams involved in this correspondence.

Theorem 5. *If $G_0^\dagger, S_1, \ldots, G_k^\dagger$ is an LR execution from some initial link-labeled graph $G_0^\dagger$ routable to 0, then there is an FR execution $H_0, \Sigma_1, \ldots, H_k$ from the graph $H_0 = T(G_0^\dagger)$ routable to $[0]$, where each H_t is isomorphic to $T(G_t^\dagger)$, and where for each node i in G_0, the steps by $[i]$ and $^-[i]$ alternate. Moreover, $H_0, \Sigma_1, \ldots, H_k$ is a greedy execution if and only if $G_0^\dagger, S_1, \ldots, G_k^\dagger$ is a greedy execution.*

Proof. Let a *partial execution* refer to any finite prefix of an execution ending with a graph. Lemma 3 shows that the sequence $\mathbf{\Sigma}$ defined by (1) is applicable to $H_0 = T(G_0^\dagger)$, and for each t, $1 \le t \le k$, $\Sigma_t. \ldots .\Sigma_1.H_0 \simeq_{\tau_t} T(S_t. \ldots .S_1.G_0^\dagger)$ where $\tau_t = \tau_{G_0^\dagger,S_t} \circ \cdots \circ \tau_{G_0^\dagger,S_1}$. Thus, there are k reorientations of H_0, denoted $H_1, \ldots, H_k$, such that $H_0, \Sigma_1, \ldots, H_k$ is a partial FR execution.

Since $H_t \simeq_{\tau_t} T(G_t^\dagger)$, u is a sink of H_t iff $\tau_t(u)$ is a sink of $T(G_t^\dagger)$. By Proposition 3 and the definition of $\tau_{G_0^\dagger,i}$, for each node i in $G_0^\dagger$, the steps of $[i]$ and $^-[i]$ alternate in the partial execution $H_0, \Sigma_1, \ldots, H_k$, and they correspond to the odd and even steps of node i in $G_0^\dagger, S_1, \ldots, G_k^\dagger$. Moreover, since the single node 0 is the sole sink of $G_k^\dagger$, the directed graph H_k has also a sole sink which is its destination node $[0]$. In other words, $H_0, \Sigma_1, \ldots, H_k$ is a (complete) execution.

Finally, the one-to-one correspondence between the sinks in H_t and in $T(G_t^\dagger)$, and the definition of the sets Σ_t by (1) show that $H_0, \Sigma_1, \ldots, H_k$ is greedy iff $G_0^\dagger, S_1, \ldots, G_k^\dagger$ is greedy. $\square$

4 The routing failure

From Theorems 2, 3, and 5 we can in principle derive the work and time complexity for LR executions from any routable link-labeled graph. However, we would like to define a generalization of the chain potential r for link-labeled graphs which provides simple expressions of the work and time complexity in terms of the initial link-labeled graph. To do that, we introduce the *routing failure* of a chain c, which coincides with $r(c)$ in the FR case, and quantifies the failure for the last node of c to route information to the destination node along c.

First, we study how any chain in $G^\dagger$ is transformed by the mapping T in Definition 1. For that, we introduce some notation. Let $c = i_0, \ldots, i_n$ be any chain in $G^\dagger$, and let first$(c) = i_0$ and last$(c) = i_n$ denote the first and the last node in c, respectively. In the directed graph $T(G^\dagger)$, we consider the sequences of nodes $u_0, \ldots u_n$ in $T(G^\dagger)$ such that for all k, $0 \leq k \leq n$, $u_k \in \{[i_k], {}^-[i_k]\}$. Clearly, such sequences are chains in $T(G^\dagger)$. If i_n is a double node, we denote by $T^+(c)$ and $T^-(c)$ the sets of such chains ending in $[i_n]$ and ${}^-[i_n]$, respectively. Further $T(c) = T^+(c) \cup T^-(c)$. Otherwise i_n is a single node, $T^+(c)$, $T^-(c)$, and $T(c)$ are all equal and denote the set of such chains ending in $[i_n]$. Clearly any chain in $T(c)$ has the same length as c.

The expression of the FR work complexity in Theorem 2 naturally leads us to introduce the potential φ for chains in $G^\dagger$

$$\varphi(c) = \min\{r(\gamma) \colon \gamma \in T(c)\},$$

and two refinements defined by $\varphi^+(c) = \min\{r(\gamma) \colon \gamma \in T^+(c)\}$ and $\varphi^-(c) = \min\{r(\gamma) \colon \gamma \in T^-(c)\}$. Obviously, $\varphi(c) = \min\{\varphi^+(c), \varphi^-(c)\}$.

Besides, let us recall some notation introduced in [7]. A node i in $G^\dagger$ is a *$\mathbb{0}$-sink in c*, if it occurs in c between two consecutive links in c which are incoming to i and labeled with $\mathbb{0}$. We denote by $s^{\mathbb{0}}(c)$ the number of $\mathbb{0}$-sinks in c. A link which is labeled with $\mathbb{1}$ and directed toward the end of the chain is called a $\mathbb{1}$-right link, and the number of such links in c is denoted by $r^{\mathbb{1}}(c)$. Hence in the FR case $r^{\mathbb{1}}(c) = r(c)$. Similarly we define $\mathbb{0}$-right links. The *residue* $Res(c)$ is defined to be 1 if c's last link is a $\mathbb{0}$-right link, and otherwise $Res(c) = 0$. In particular, for a chain c of length 0, $Res(c) = 0$.

Proposition 5. *For any chain c in $G^\dagger$, $\varphi(c) = \varphi^-(c)$. In more detail, $\varphi^+(c) = \varphi^-(c)$ if $\mathrm{last}(c) \in \mathcal{S}$, and $\varphi^+(c) = \varphi^-(c) + Res(c)$ if $\mathrm{last}(c) \in \mathcal{D}$.*

Interestingly, the chain potentials φ, φ^+, and φ^- admit simple expressions just in terms of the link-labeled graph $G^\dagger$. To show that, we define the chain potential $\varrho(c)$ as:

$$\varrho(c) = r^{\mathbb{1}}(c) + s^{\mathbb{0}}(c) + Res(c).$$

Proposition 6. *For any chain c in $G^\dagger$, $\varrho(c) = \varphi^+(c)$.*

We now define the *routing failure* $\Phi(c)$ by,

$$\Phi(c) = \begin{cases} \varrho(c) & \text{if } \mathrm{last}(c) \in \mathcal{S}, \\ 2\varrho(c) - Res(c) & \text{if } \mathrm{last}(c) \in \mathcal{D}, \end{cases}$$

and relate it to the FR potential r in $T(G^\dagger)$, using Propositions 5 and 6.

Theorem 6. *For any chain c in $G^\dagger$,*

$$\Phi(c) = \begin{cases} \min\{r(\gamma) \colon \gamma \in T(c)\} & \text{if } \mathrm{last}(c) \in \mathcal{S}, \\ \min\{r(\gamma) \colon \gamma \in T^+(c)\} + \min\{r(\gamma) \colon \gamma \in T^-(c)\} & \text{if } \mathrm{last}(c) \in \mathcal{D}. \end{cases}$$

5 Application

In this section we first give a sufficient condition of the input graph for LR to solve the Routing problem, and then establish for the first time the exact time complexity.

5.1 The acyclicity issue

As explained in Section 2, in the FR case, one easily observes that if the initial graph is acyclic, it remains acyclic during the whole FR execution. On the contrary, in the general LR case, acyclicity of the initial graph is not sufficient to guarantee acyclicity during the LR execution.

By Theorem 5, we know that each LR execution from a link-labeled routable graph $G^\dagger$ is equivalent to an FR execution from $T(G^\dagger)$; in particular, the two final graphs are isomorphic. Hence, acyclicity of $T(G^\dagger)$ guarantees acyclicity of the final graph in the LR executions from $G^\dagger$. That leads us to seach a characterization of the link-labeled graphs that are transformed into acyclic graphs by T.

Theorem 7. *Let $G^\dagger$ be a link-labeled routable graph. Then the following are equivalent: (i) the graph $T(G^\dagger)$ is acyclic, and (ii) for any closed chain c in $G^\dagger$, $\varrho(c) > 0$.*

One can further show that (ii) is actually equivalent to a simple condition, called (AC), being satisfied on all circuits (i.e. subgraphs induced by simple closed chains): a circuit satisfies (AC) if it contains links labeled with $\mathbb{1}$ in opposite directions, or if it contains a node that is a sink relative to the circuit where both incoming links are labeled with $\mathbb{0}$. From the definition of (AC), it follows immediately that condition (ii) implies that (AC) holds in all circuits. Proving the converse direction is more subtle. This equivalence combined with Theorem 7 reveals (AC) to be the counterpart of acyclicity for FR. The benefit of (AC) over (ii) is that it allows us to check just simple closed chains instead of all closed chains.

5.2 Exact complexity of the LR algorithm

We now develop for the first time an exact expression for the time complexity of LR (and thus PR). To this end, we require an exact expression for the work complexity of each node beforehand. We note that from Theorems 2 and 5, we derive a new proof that the LR algorithm terminates, i.e., each node i takes a finite number w_i of steps. More precisely, either i is a single node in $G^\dagger$, there is one corresponding node $[i]$ in the routable graph $T(G^\dagger)$, and $w_i = w_{[i]}$, or else i is a double node, $[i]$ and $^-[i]$ are the two corresponding nodes in $T(G^\dagger)$, and $w_i = w_{[i]} + w_{-[i]}$ where $w_{[i]}$ and $w_{-[i]}$ denote the work by $[i]$ and $^-[i]$ in any FR execution from the routable graph $T(G^\dagger)$.

In the case each circuit in the initial link-labeled graph satisfies (AC), we show that w_i is equal to the minimum routing failure of the chains from 0 to i. This work complexity result has already been established in [7], but the proof here is interesting on its own as it illustrates how to use our reduction to translate a result previously established for FR, namely Theorem 2, into a general result for LR.

Theorem 8. *Let $G^\dagger$ be a link-labeled routable graph where all circuits satisfy the (AC) property. In any LR execution from $G^\dagger$, the number of steps taken by any node i in $V \cup \{0\}$ is equal to*

$$w_i = \min\{\Phi(c)\colon c \in \mathcal{C}(0, i, G^\dagger)\}.$$

We use the same technique to generalize Theorem 3 to LR greedy executions. In this way, we establish a new result, namely the exact time complexity of any LR execution. We only sketch the correctness arguments here and give the full proof in the appendix.

Theorem 9. *Let $G^\dagger$ be a link-labeled routable graph. If w_i is the work of some node i in $G^\dagger$, then the termination time θ_i in the greedy LR execution is 0 if $w_i = 0$, and otherwise*

$$\theta_i = \max\{\lambda(c)\colon c \in \mathcal{C}(\rightarrow i, G^\dagger) \wedge \Phi(c) = w_i - 1\} + 1.$$

Proof sketch. We consider executions from $G^\dagger$ and $T(G^\dagger)$ and use w_x and θ_x for the work and the termination time of node x in the respective executions as no confusion may arise. One observes that from Theorem 5 follows that for single nodes $w_i = w_{[i]}$ and $\theta_i = \theta_{[i]}$, and for double nodes $w_i = w_{[i]} + w_{-[i]}$ and $\theta_i = \max\{\theta_{[i]}, \theta_{-[i]}\}$.

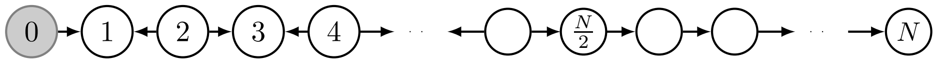

Figure 5: Chain family with quadratic time complexity of PR.

Using this, we may employ Theorem 3 to relate the time complexity of a node i from $G^\dagger$ to the length of chains in $T(G^\dagger)$. For instance, for a single node we get

$$\theta_i = \max\{\lambda(\gamma)\colon \gamma \in \mathcal{C}(\to [i], T(G^\dagger) \wedge r(\gamma) = w_i - 1\} + 1.$$

For each chain γ in $T(G^\dagger)$, there exists a chain c in $G^\dagger$ such that $\gamma \in T(c)$ and γ and c have the same length. The potential r of all chains in $T(c)$ may then be related to the routing failure of c by Theorem 6. This finally yields an expression for the exact time complexity of node i in terms of the length and the routing failure of chains in $G^\dagger$, as required. $\square$

Theorem 9 can be easily specialized to PR: by Theorem 1.d, PR executions are those where all links are initially labeled with $\mathbb{0}$. We observe that (i) a node other than 0 is a single node if and only if it is a source or a sink, and (ii) the chain potential $\varrho(c)$ of a chain c reduces to $s(c) + \mathit{Res}(c)$, where $s(c)$ is the number of sinks relative to c.

Corollary 1. *If w_i is the work of node i in V in executions of Partial Reversal from the initial routable graph G, then the termination time θ_i in the greedy execution is* 0 *if $w_i = 0$, and otherwise*

$$\theta_i = \max\left\{\lambda(c)\colon c \in \mathcal{C}(\to i, G) \wedge w_i - 1 = \left\{\begin{array}{ll} s(c) + \mathit{Res}(c) & \textit{if } i \textit{ is a source or sink} \\ 2s(c) + \mathit{Res}(c) & \textit{otherwise} \end{array}\right\}\right\} + 1.$$

This sheds new light on the comparison of FR and PR. In [6], we have shown that in trees the time complexity of FR is linear in the number of nodes (while only a quadratic upper bound was previously known). Using our transformation, we observe that given an initial labeled tree $G^\dagger$, the transformed graph $T(G^\dagger)$ is in general not a tree, which indicates that the time complexity of PR on trees may be nonlinear. Indeed, based on Corollary 1, one can design a family of chain graphs in which the time complexity of PR is quadratic in the number of nodes (cf. Figure 5). Hence, when considering time instead of work, we arrive at the conclusion opposite to [8] that PR is not better than FR. In other words, FR allows more concurrency than PR, thus possibly compensating nodes for work overload.

6 Conclusions

In a companion paper [6], we showed that the dynamic behavior of Full Reversal (FR) from a directed graph G can be captured by a linear dynamical min-plus system of order equal to the number of nodes in G. It can be easily shown that unlike FR, the dynamic behavior of Partial Reversal (PR) cannot in general be described by such a system of the same order. This paper therefore tackles the complexity of LR by a different approach, namely by reduction to FR. In this way we reveal that FR is in fact the paradigm in the link reversal algorithmic scheme.

The transformation approach laid out in this paper proved to be very efficient from a technical viewpoint. First, we were able to establish a condition on the link-labeled graph that exactly corresponds to acyclicity for FR; a condition that is central to the link reversal approach. This legitimates a posteriori a magic invariant from [7] in a constructive way. In addition, we were able to develop the exact work, and — for the first time — time complexity in terms of the initial labeled graph by using properties of the transformation.

Finally, the reduction also works in applications of link reversal where there is no destination node. For instance in [2] resource allocation was studied for FR, and various properties of correctness and concurrency were established. Our results allow extension of these results to LR.

References

[1] Hagit Attiya, Vincent Gramoli, and Alessia Milani. A provably starvation-free distributed directory protocol. In *12th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 405–419, 2010.

[2] Valmir C. Barbosa and Eli Gafni. Concurrency in heavily loaded neighborhood-constrained systems. *ACM Trans. Program. Lang. Syst.*, 11(4):562–584, 1989.

[3] Costas Busch, Srikanth Surapaneni, and Srikanta Tirthapura. Analysis of link reversal routing algorithms for mobile ad hoc networks. In *Proceedings of the 15th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 210–219, 2003.

[4] Costas Busch and Srikanta Tirthapura. Analysis of link reversal routing algorithms. *SIAM Journal on Computing*, 35(2):305–326, 2005.

[5] K. Mani Chandy and Jayadev Misra. The drinking philosopher's problem. *ACM Transactions on Programming Languages and Systems*, 6(4):632–646, 1984.

[6] Bernadette Charron-Bost, Matthias Függer, Jennifer L. Welch, and Josef Widder. Full reversal routing as a linear dynamical system. To appear in SIROCCO 2011.

[7] Bernadette Charron-Bost, Antoine Gaillard, Jennifer L. Welch, and Josef Widder. Routing without ordering. In *Proceedings of the 21st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 145–153, 2009.

[8] Bernadette Charron-Bost, Jennifer L. Welch, and Josef Widder. Link reversal: How to play better to work less. In *Proceedings of the 5th International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors'09)*, volume 5304 of *LNCS*, pages 88–101, 2009.

[9] Abdelouahid Derhab and Nadjib Badache. A self-stabilizing leader election algorithm in highly dynamic ad hoc mobile networks. *IEEE Trans. Parallel Distrib. Syst.*, 19(7):926–939, 2008.

[10] Eli Gafni and Dimitri P. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. *IEEE Transactions on Communications*, 29(1):11–18, January 1981.

[11] Rebecca Ingram, Patrick Shields, Jennifer E. Walter, and Jennifer L. Welch. An asynchronous leader election algorithm for dynamic networks. In *Proceedings of the IEEE International Parallel & Distributed Processing Symposium*, pages 1–12, 2009.

[12] Young-Bae Ko and N. H. Vaidya. Geotora: a protocol for geocasting in mobile ad hoc networks. In *Proceedings of the 2000 International Conference on Network Protocols*, ICNP '00, pages 240–250, 2000.

[13] Yossi Malka, Shlomo Moran, and Shmuel Zaks. A lower bound on the period length of a distributed scheduler. *Algorithmica*, 10(5):383–398, 1993.

[14] Navneet Malpani, Jennifer L. Welch, and Nitin Vaidya. Leader election algorithms for mobile ad hoc networks. In *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, 2000.

[15] Mohamed Naimi, Michel Trehel, and André Arnold. A log(n) distributed mutual exclusion algorithm based on path reversal. *Journal on Parallel and Distributed Computing*, 34(1):1–13, 1996.

[16] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *16th Conference on Computer Communications (Infocom)*, pages 1405–1413, April 1997.

[17] Kerry Raymond. A tree-based algorithm for distributed mutual exclusion. *ACM Transactions on Computer Systems*, 7(1):61–77, 1989.

[18] Srikanta Tirthapura and Maurice Herlihy. Self-stabilizing distributed queuing. *IEEE Transactions on Parallel and Distributed Systems*, 17(7):646–655, 2006.

[19] Jennifer E. Walter, Jennifer L. Welch, and Nitin H. Vaidya. A mutual exclusion algorithm for ad hoc mobile networks. *Wireless Networks*, 7(6):585–600, 2001.

A Appendix

A.1 Proofs of Section 3

An important property of the LR algorithm is, that it is *delay-insensitive* in the sense that the directed graph generated by the algorithm depends on the routable input graph only, and not on the order in which nodes take steps. This property can be easily proved with the following property that we require to prove Lemma 2.

Proposition 7. *If S be any subset of the sinks in $G^\dagger$, and S_1 and S_2 are non-empty subsets of S that partition S, then S_1 is applicable to $S_2.G^\dagger$, S_2 is applicable to $S_1.G^\dagger$, and*

$$S.G^\dagger = S_1.(S_2.G^\dagger) = S_2.(S_1.G^\dagger).$$

Lemma 2. *If S is a nonempty set of sinks in $G^\dagger$ and $\tau = \tau_{G^\dagger,S}$, then $[S]$ is applicable to $T(G^\dagger)$ and $[S].T(G^\dagger) \simeq_\tau T(S.G^\dagger)$.*

Proof. The proof is by induction on the cardinality of S.

Base: $|S| = 1$. Since S is a singleton, the lemma follows from Lemma 1.

Inductive step: Assume that S has cardinality $k \geq 2$, and assume that the lemma holds for any set of sinks in $G^\dagger$ of cardinality $k-1$. Choose an arbitrary i from S and let $S' = S \setminus \{i\}$. By Proposition 3, $[S]$ is a set of sinks in $T(G^\dagger)$, and by Proposition 7, we obtain

$$[S].T(G^\dagger) = [i].([S'].T(G^\dagger)).$$

By induction hypothesis, $[S'].T(G^\dagger) \simeq_{\tau_{G^\dagger,S'}} T(S'.G^\dagger)$. Since $[i]$ is not in $[S']$, $[i]$ is fixed point of $\tau_{G^\dagger,S'}$, and by Proposition 4,

$$[i].([S'].T(G^\dagger)) \simeq_{\tau_{G^\dagger,S'}} [i].(T(S'.G^\dagger)).$$

From Lemma 1, we derive

$$[i].(T(S'.G^\dagger)) \simeq_{\tau_{S'.G^\dagger,i}} T(i.(S'.G^\dagger)).$$

By Proposition 7,

$$T(i.(S'.G^\dagger)) = T(S.G^\dagger).$$

Since $\tau_{S'.G^\dagger,i} = \tau_{G^\dagger,i}$ and $\tau_{G^\dagger,S} = \tau_{G^\dagger,i} \circ \tau_{G^\dagger,S'}$, the lemma follows. ☐

Lemma 3. *If $G^\dagger$ is a link-labeled routable graph, $\mathbf{S}$ is a sequence of length k applicable to $G^\dagger$, and $\tau = \tau_{G^\dagger,S_1} \circ \cdots \circ \tau_{G^\dagger,S_k}$, then $\mathbf{\Sigma}$ is applicable to $T(G^\dagger)$, and $\mathbf{\Sigma}.T(G^\dagger) \simeq_\tau T(\mathbf{S}.G^\dagger)$.*

Proof. First note that $\tau^{-1} = \tau$, and therefore $\mathbf{\Sigma}.T(G^\dagger) \simeq_\tau T(\mathbf{S}.G^\dagger)$ if and only if $T(\mathbf{S}.G^\dagger) \simeq_\tau \mathbf{\Sigma}.T(G^\dagger)$. The proof of the lemma is by induction on the length k of $\mathbf{S}$.

Base: $k = 1$. Since $\mathbf{S}$ comprises a single set, the lemma follows from Lemma 2.

Inductive step: Assume as induction hypothesis that the lemma holds for any applicable sequence with length up to $k-1 \geq 1$. We abbreviate

$$\tau' = \tau_{G^\dagger,S_1} \circ \cdots \circ \tau_{G^\dagger,S_{k-1}} \quad \text{and} \quad \mathbf{S}' = (S_t)_{1 \leq t \leq k-1},$$

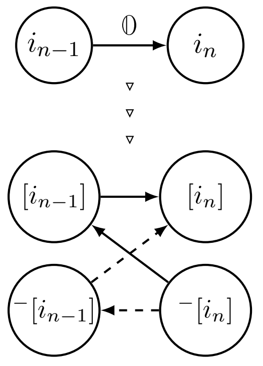

Figure 6: Case 2a

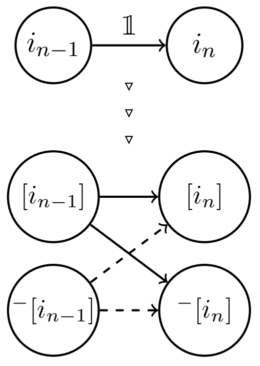

Figure 7: Case 2b

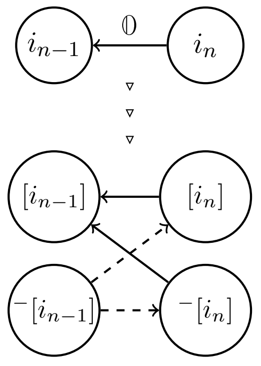

Figure 8: Case 2b

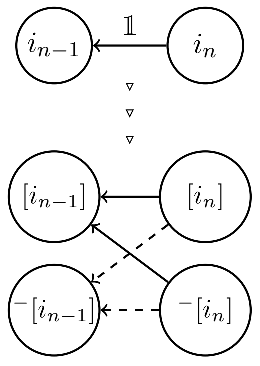

Figure 9: Case 2b

and $\mathbf{\Sigma}'$ is the sequence of length $k-1$ defined from $\mathbf{S}'$ by (1). Hence, $\Sigma_k = \tau'([S_k])$. Moreover, $\mathbf{\Sigma}'$ is the prefix of $\mathbf{\Sigma}$ of length $k-1$, i.e., for any index t, $1 \leq t \leq k-1$, we have $\Sigma'_t = \Sigma_t$.

Since $\mathbf{S}$ is applicable to $G^\dagger$, S_k is applicable to $\mathbf{S}'.G^\dagger$. From Lemma 2 then follows that $[S_k]$ is applicable to $T(\mathbf{S}'.G^\dagger)$ and

$$[S_k].T(\mathbf{S}'.G^\dagger) \simeq_{\tau_{G^\dagger,S_k}} T(\mathbf{S}.G^\dagger)$$

since $\tau_{\mathbf{S}'.G^\dagger,S_k} = \tau_{G^\dagger,S_k}$ and $S_k.\mathbf{S}'.G^\dagger = \mathbf{S}.G^\dagger$. With $(\tau_{G^\dagger,S_k})^{-1} = \tau_{G^\dagger,S_k}$ we further obtain

$$T(\mathbf{S}.G^\dagger) \simeq_{\tau_{G^\dagger,S_k}} [S_k].T(\mathbf{S}'.G^\dagger). \tag{2}$$

Moreover by the induction hypothesis, $\mathbf{\Sigma}'$ is applicable to $T(G^\dagger)$, and

$$T(\mathbf{S}'.G^\dagger) \simeq_{\tau'} \mathbf{\Sigma}'.T(G^\dagger). \tag{3}$$

By Proposition 4, we deduce from (3) that $\tau'([S_k]) = \Sigma_k$ is applicable to $\mathbf{\Sigma}'.T(G^\dagger)$, and

$$[S_k].T(\mathbf{S}'.G^\dagger) \simeq_{\tau'} \Sigma_k.\mathbf{\Sigma}'.T(G^\dagger). \tag{4}$$

Since $\Sigma_k.\mathbf{\Sigma}'.T(G^\dagger) = \mathbf{\Sigma}.T(G^\dagger)$, (4) shows $\mathbf{\Sigma}$ is applicable to $T(G^\dagger)$, and

$$[S_k].T(\mathbf{S}'.G^\dagger) \simeq_{\tau'} \mathbf{\Sigma}.T(G^\dagger). \tag{5}$$

Composing (2) and (5), we deduce that $\mathbf{\Sigma}.T(G^\dagger) \simeq_\tau T(\mathbf{S}.G^\dagger)$, as $\tau = \tau' \circ \tau_{G^\dagger,S_k}$. ☐

A.2 Proofs of Section 4

Proposition 5. *For any chain c in $G^\dagger$, $\varphi(c) = \varphi^-(c)$. In more detail, $\varphi^+(c) = \varphi^-(c)$ if* $\mathrm{last}(c) \in \mathcal{S}$, *and $\varphi^+(c) = \varphi^-(c) + Res(c)$ if* $\mathrm{last}(c) \in \mathcal{D}$.

Proof. The proof is by induction on the length n of the chain c.
Base: $n = 0$. For any chain c of length 0, it holds that $\varphi^+(c) = \varphi^-(c) = 0$, and the base case follows.
Inductive step: We consider some chain $c = i_0, i_1, \dots i_n$ in $G^\dagger$, and its prefix $c' = i_0, \dots i_{n-1}$. We assume that the lemma holds for c'. There are two cases to consider.

1. Node i_n is a single node. By definition of φ^+ and φ^-, we immediately obtain $\varphi^+(c) = \varphi^-(c)$.

2. Node i_n is a double node. Then we proceed by exhaustive examination of the four possible cases for the last labeled-link of c. Let e denote this link, namely $e = (i_{n-1}, i_n)$ or $e = (i_n, i_{n-1})$.

(a) $e = (i_{n-1}, i_n)$, and e is labeled with $\mathbb{0}$ (Figure 6). By definition of φ^+,

$$\varphi^+(c) = \min\left\{\varphi^+(c') + 1, \varphi^-(c') + 1\right\}$$

regardless of i_{n-1} being a single or a double node. From the inductive hypothesis then follows that

$$\varphi^+(c) = \varphi^+(c') + 1.$$

Similarly, $\varphi^-(c) = \min\{\varphi^+(c'), \varphi^-(c')\}$ regardless of i_{n-1} being a single or a double node, and so

$$\varphi^-(c) = \varphi^+(c').$$

Therefore, $\varphi^+(c) = \varphi^-(c) + 1$, and as $Res(c) = 1$, the proposition follows for c.

(b) Otherwise, $e = (i_{n-1}, i_n)$ is labeled with $\mathbb{1}$, or $e = (i_n, i_{n-1})$. That corresponds to the three cases in Figures 7, 8, and 9, respectively. We observe that $[i_n]$ and $^-[i_n]$ play identical roles in $T(e)$, and so $\varphi^-(c) = \varphi^+(c)$, regardless of i_{n-1} being a single or a double node. Moreover, in the three cases, $Res(c) = 0$, and therefore the proposition holds for c in each of these three cases.

The proposition holds in all cases. □

Proposition 6. *For any chain c in $G^\dagger$, $\varrho(c) = \varphi^+(c)$.*

Before giving the proof, we introduce two refinements of φ^+, denoted by φ^+_+ and φ^+_-:

$$\varphi^+_+(c) = \begin{cases} \varphi^+(c) & \text{if } \mathrm{first}(c) \in \mathcal{S}, \\ \min\{r(\gamma) \colon \gamma \in T^+(c) \wedge \mathrm{first}(\gamma) = [\mathrm{first}(c)]\} & \text{if } \mathrm{first}(c) \in \mathcal{D}, \end{cases}$$

and

$$\varphi^+_-(c) = \begin{cases} \varphi^+(c) & \text{if } \mathrm{first}(c) \in \mathcal{S}, \\ \min\{r(\gamma) \colon \gamma \in T^+(c) \wedge \mathrm{first}(\gamma) = {}^-[\mathrm{first}(c)]\} & \text{if } \mathrm{first}(c) \in \mathcal{D}. \end{cases}$$

Obviously, for any chain c in $G^\dagger$ it holds that

$$\varphi^+(c) = \min\left\{\varphi^+_+(c), \varphi^+_-(c)\right\}.$$

A proof similar to the one of Proposition 5 shows the following relation between the chain potentials φ^+_- and φ^+_+.

Proposition 8. *For any chain c in $G^\dagger$, $\varphi^+(c) = \varphi^+_+(c)$. In more detail, if c^{-1} is the reverse chain of c, then*

$$\varphi^+_+(c) = \begin{cases} \varphi^+_-(c) & \textit{if } \mathrm{first}(c) \in \mathcal{S}, \\ \varphi^+_-(c) - Res(c^{-1}) & \textit{if } \mathrm{first}(c) \in \mathcal{D}. \end{cases}$$

In the particular FR case, every node is a single node, and the routing failure of a chain c is equal to the number of its right links $r(c)$. We proceed by showing Proposition 6, that is, that the two chain potentials φ^+ and ϱ actually coincide:

Proof of Proposition 6. If the chain c has length 0, then it holds that $\varrho(c) = 0$ and $\varphi^+(c) = 0$, and the proposition follows in this case. Otherwise, that is, if the length is greater than 0 we proceed by induction on the length n of the chain c.

Base case: $n = 1$. There are two cases to consider.

1. (i_0, i_1) is a link in $G^\dagger$. If its label is $\mathbb{0}$, then $s^{\mathbb{0}}(c) = r^{\mathbb{1}}(c) = 0$, and $Res(c) = 1$. Otherwise, its label is $\mathbb{1}$, $s^{\mathbb{0}}(c) = Res(c) = 0$, and $r^{\mathbb{1}}(c) = 1$. In both cases, we get $\varrho(c) = 1$. Further, $\varphi^+(c) = 1$.
2. (i_1, i_0) is a link in $G^\dagger$. We easily check that $\varrho(c) = 0$ and $\varphi^+(c) = 0$.

The base case $n = 1$ then follows.

Inductive step: We consider some chain $c = i_0, i_1, \dots i_n$ in $G^\dagger$ with length $n \geq 2$, and its suffix $c' = i_1, \dots i_n$. We assume that $\varrho(c') = \varphi^+(c')$. By Proposition 8, we know that

$$\varphi^+(c') = \varphi^+_+(c'), \quad \text{and} \quad \varphi^+(c) = \varphi^+_+(c).$$

Then we proceed by exhaustive examination with the following four cases.

1. (i_0, i_1) is a $\mathbb{0}$-link in $G^\dagger$, and i_1 is a single node (Figure 6 with $n = 1$). We easily check that $\varphi^+_+(c) = \varphi^+_+(c') + 1$. Consequently,
$$\varphi^+(c) = \varphi^+(c') + 1.$$
Since i_1 is a single node and has at least one incoming link labeled with $\mathbb{0}$, i_1 is actually a sink, and all its incident links are labeled with $\mathbb{0}$. By definition of ϱ,
$$\varrho(c) = \varrho(c') + 1.$$
By inductive hypothesis, it follows that $\varrho(c) = \varphi^+(c)$.

2. (i_0, i_1) is a $\mathbb{0}$-link in $G^\dagger$, and i_1 is a double node (Figure 6 with $n = 1$). By definition of φ^+_+, we have
$$\varphi^+_+(c) = \min\left\{\varphi^+_+(c') + 1, \varphi^+_-(c')\right\}.$$
There are two subcases to consider.
   (a) (i_2, i_1) is a $\mathbb{0}$-link in $G^\dagger$. Since $Res((c')^{-1}) = 1$, Proposition 8 implies that $\varphi^+_-(c') = \varphi^+_+(c') + 1$. Consequently,
   $$\varphi^+(c) = \varphi^+(c') + 1.$$
   Moreover by definition of ϱ,
   $$\varrho(c) = \varrho(c') + 1.$$
   By inductive hypothesis, it follows that $\varrho(c) = \varphi^+(c)$.
   (b) (i_2, i_1) is not a $\mathbb{0}$-link in $G^\dagger$. In this case, $Res((c')^{-1}) = 0$, and by Proposition 8, $\varphi^+_-(c') = \varphi^+(c')$. Consequently,
   $$\varphi^+(c) = \varphi^+(c').$$
   By definition of ϱ,
   $$\varrho(c) = \varrho(c').$$
   By inductive hypothesis, it follows that $\varrho(c) = \varphi^+(c)$.

3. (i_0, i_1) is a $\mathbb{1}$-link in $G^\dagger$ (Figure 7 with $n = 1$). By definition of φ^+_+, we have $\varphi^+_+(c) = \min\left\{\varphi^+_+(c'), \varphi^+_-(c')\right\} + 1$, regardless of i_1 being a single or a double node. Therefore, Proposition 8 implies that $\varphi^+_+(c) = \varphi^+_+(c') + 1$, and so
$$\varphi^+(c) = \varphi^+(c') + 1.$$
By definition of ϱ,
$$\varrho(c) = \varrho(c') + 1.$$
By inductive hypothesis, it follows that $\varrho(c) = \varphi^+(c)$.

4. (i_1, i_0) is a link in $G^\dagger$ (Figures 8 and 9 with $n = 1$). By definition of φ^+_+, we verify that $\varphi^+_+(c) = \min\left\{\varphi^+_+(c'), \varphi^+_-(c')\right\}$, regardless of i_1 being a single or a double node. Proposition 8 implies that $\varphi^+_+(c) = \varphi^+_+(c')$, and so
$$\varphi^+(c) = \varphi^+(c').$$
By definition of ϱ,
$$\varrho(c) = \varrho(c').$$
By inductive hypothesis, it follows that $\varrho(c) = \varphi^+(c)$.

□

Theorem 6. *For any chain c in $G^\dagger$,*
$$\Phi(c) = \begin{cases} \min\{r(\gamma) \colon \gamma \in T(c)\} & \textit{if } \mathrm{last}(c) \in \mathcal{S}, \\ \min\{r(\gamma) \colon \gamma \in T^+(c)\} + \min\{r(\gamma) \colon \gamma \in T^-(c)\} & \textit{if } \mathrm{last}(c) \in \mathcal{D}. \end{cases}$$

Proof. If the last node in c is a single node, then by definition of Φ we have $\Phi(c) = \varrho(c)$. The theorem immediately follows from Propositions 5 and 6.

Otherwise the last node in c is a double node, and $\Phi(c) = 2\varrho(c) - \mathit{Res}(c)$. From Propositions 5 and 6 follows that $\varrho(c) = \varphi^+(c)$, and $\varrho(c) = \varphi^-(c) + \mathit{Res}(c)$. Hence, $\Phi(c) = \varphi^+(c) + \varphi^-(c)$, as needed. □

A.3 Proofs of Section 5

Theorem 7. *Let $G^\dagger$ be a link-labeled routable graph. Then the following are equivalent: (i) the graph $T(G^\dagger)$ is acyclic, and (ii) for any (closed) chain c in $cl(G^\dagger)$, $\varrho(c) > 0$.*

Proof. From (i) follows (ii): assume that $T(G^\dagger)$ is acyclic. Consider an arbitrary closed chain $c = i_0, i_1, \ldots, i_k$ in $G^\dagger$. By the acyclicity of $T(G^\dagger)$, each chain γ in $T(c)$ with $\mathrm{first}(\gamma) = \mathrm{last}(\gamma) = [i_0]$ must have at least one right link in it. Thus $\varphi^+_+(c) > 0$ holds. Since further by Proposition 8 and Proposition 6 it holds that $\varrho(c) = \varphi^+(c) = \varphi^+_+(c)$, we deduce that $\varrho(c) > 0$. The implication follows.

From (ii) follows (i): assume the potential ϱ of each closed chain is greater than 0. Further assume by means of contradiction that $T(G^\dagger)$ is cyclic, and thus there exists a closed chain $\gamma = u_0, u_1, \ldots, u_n$ in $T(G^\dagger)$ with $r(\gamma) = 0$. By construction of $T(G^\dagger)$, there exists a closed chain $c = i_0, i_1, \ldots, i_n$ in $G^\dagger$ with γ in $T(c)$.

We next construct a chain $d = j_0, j_1, \ldots, j_n$ that is a permutation of chain c which has Res equal to 0: choose an index k, $0 \le k \le n$, such that for each ℓ, $0 \le \ell \le n$, it holds that $j_\ell = i_{\ell+k \bmod (n+1)}$ and $\mathit{Res}(d) = 0$. Such a k, and therefore d, must exist: if not, c would be a sequence of nodes with only $\mathbb{0}$-right links, a contradiction to the hypothesis that $\varrho(c) > 0$ and $\varrho(c^{-1}) > 0$.

Since d was generated by rotating c, chain d is a closed chain in $G^\dagger$. It follows from the hypothesis that $\varrho(d) > 0$.

Further, there is a closed chain δ in $T(d)$ which can be generated by rotating γ. By construction, $r(\delta) = r(\gamma)$. By the hypothesis, $r(\gamma) = 0$, and thus it follows that $r(\delta) = 0$. By definition of φ, it follows from $r(\delta) = 0$ that $\varphi(d) = 0$. Since $\mathit{Res}(d) = 0$, we derive from Proposition 5 that $\varphi^+(d) = \varphi^-(d) = \varphi(d) = 0$. From Proposition 6, we conclude that the closed chain d satisfies $\varrho(d) = 0$, a contradiction. The implication follows. □

Theorem 8. *Let $G^\dagger$ be a link-labeled routable graph where all circuits satisfy the (AC) property. In any LR execution from $G^\dagger$, the number of steps taken by any node i in $V \cup \{0\}$ is equal to*

$$w_i = \min\{\Phi(c) \colon c \in \mathcal{C}(0, i, G^\dagger)\}.$$

Proof. Let $H = T(G^\dagger)$ be the directed graph defined from $G^\dagger$. Since all circuits in $G^\dagger$ satisfy (AC), by Theorem 7, H is acyclic, and so the expression of the work complexity given by Theorem 2 holds in any FR execution from H. We consider any LR execution from $G^\dagger$ and the equivalent FR execution from H given by Theorem 5. For both executions, we use the notation w_x for the work by node x as no confusion may arise. Let i be any node of $G^\dagger$. There are two cases to consider.

1. Node i is a single node in $G^\dagger$. Hence, $w_i = w_{[i]}$. By Theorem 2, we obtain

$$w_{[i]} = \min\{r(\gamma) \colon \gamma \in \mathcal{C}([0], [i], H)\}.$$

By definition of the graph H,

$$\{\gamma \colon \gamma \in \mathcal{C}([0], [i], H)\} = \bigcup_{c \in \mathcal{C}(0,i,G^\dagger)} T(c).$$

Using Theorem 6, we easily check that $\min\{\Phi(c) \colon c \in \mathcal{C}(0, i, G^\dagger)\} = \min\{r(\gamma) \colon \gamma \in \mathcal{C}([0], [i], H)\}$, and the formulas for w_i immediately follows in the case i is a single node.

2. Node i is a double node in $G^\dagger$. Then, $w_i = w_{[i]} + w_{-[i]}$. By Theorem 2, we have

$$\begin{aligned} w_{[i]} &= \min\{r(\gamma) \colon \gamma \in \mathcal{C}([0], [i], H)\} \\ w_{-[i]} &= \min\{r(\gamma) \colon \gamma \in \mathcal{C}([0], {}^{-}[i], H)\}. \end{aligned}$$

Thanks to the decomposition

$$\{\gamma \colon \gamma \in \mathcal{C}([0], [i], H)\} = \bigcup_{c \in \mathcal{C}(0,i,G^\dagger)} T^+(c)$$

and Theorem 6, we easily check that

$$w_{[i]} = \min\{\varphi^+(c) \colon c \in \mathcal{C}(0, i, G^\dagger)\}. \tag{6}$$

Similarly, we show that

$$w_{-[i]} = \min\{\varphi^-(c) \colon c \in \mathcal{C}(0, i, G^\dagger)\}. \tag{7}$$

It follows that $w_i \leq \min\{\Phi(c) \colon c \in \mathcal{C}(0, i, G^\dagger)\}$.

From Proposition 5, we derive that $w_{-[i]} \leq w_{[i]} \leq w_{-[i]} + 1$, as the residue of any chain is equal to 0 or 1. Thereby, $w_{[i]} = w_{-[i]}$ or $w_{[i]} = w_{-[i]} + 1$. We now consider the following two subcases:

- $w_{[i]} = w_{-[i]}$. Hence, $w_i = 2w_{[i]}$. By (6), there is a chain c_0 in $\mathcal{C}(0, i, G^\dagger)$ such that $w_i = 2\varphi^+(c_0)$. Moreover by Proposition 6 and the definition of $\Phi(c_0)$, $2\varphi^+(c_0) \geq \Phi(c_0)$, as the residue of c_0 is equal to 0 or 1. It follows that $w_i \geq \Phi(c_0)$.

- $w_{[i]} = w_{-[i]} + 1$. Hence, $w_i = 2w_{-[i]} + 1$. By (7), there is a chain c_1 in $\mathcal{C}(0, i, G^\dagger)$ such that $w_i = 2\varphi^-(c_1) + 1$. Moreover by Propositions 5 and 6 and the definition of $\Phi(c_1)$, $2\varphi^-(c_1) + 1 \geq \Phi(c_1)$, as the residue of c_1 is equal to 0 or 1. It follows that $w_i \geq \Phi(c_1)$.

In both cases, we can conclude that $w_i \geq \min\{\Phi(c)\colon c \in \mathcal{C}(0, i, G^\dagger)\}$, and the theorem follows in the case i is a double node. $\square$

Theorem 9. *Let $G^\dagger$ be a link-labeled routable graph. If w_i is the work of some node i in V from the initial link-labeled routable graph $G^\dagger$, then the termination time θ_i in the greedy LR execution is* 0 *if $w_i = 0$, and otherwise*

$$\theta_i = \max\{\lambda(c)\colon c \in \mathcal{C}(\to i, G^\dagger) \wedge \Phi(c) = w_i - 1\} + 1.$$

For that, we first prove that the chain potential ϱ is not bounded, and satisfies some property of intermediate values.

Lemma 4. *For any chain c in $G^\dagger$, and any integer k, $k \geq \varrho(c)$, there exists a chain c' in $G^\dagger$ such that c is a suffix of c', and $\varrho(c') = k$.*

Proof. By definition of ϱ in $G^\dagger$, we first observe that it suffices to prove the lemma when c is a chain in $G^\dagger$. The proof is by induction on k.

Base: $k = \varrho(c)$. We choose $c' = c$, and so the base case immediately follows.

Inductive step: We assume that the lemma holds for $k \geq \varrho(c)$, and so there is a chain $\tilde{c}$ such that c is a suffix of $\tilde{c} = i_0, i_1, \dots i_n$, and $\varrho(\tilde{c}) = k$. There are two cases to consider.

1. $\lambda(\tilde{c}) = 0$. Then $k = 0$, and $\tilde{c} = c = i_0$. There are two subcases to consider.
   (a) Node i_0 is a source. Let i denote any of i_0's neighbors, and let $c' = i_0, i, i_0$. Then $\varrho(c') = 1$.
   (b) Otherwise i_0 is not a source, and so i_0 has at least one incoming neighbor. Let i denote any of i_0's incoming neighbors, and let $c' = i, i_0$. Then $\varrho(c') = 1$.

   The lemma thus follows in the case $\lambda(\tilde{c}) = 0$.

2. $\lambda(\tilde{c}) \geq 1$. Let e denote the first labeled link of $\tilde{c}$, namely $e = (i_0, i_1)$ or $e = (i_1, i_0)$.
   (a) $e = (i_1, i_0)$. Then let $c' = i_1; \tilde{c}$. We easily check that $\varrho(c') = \varrho(\tilde{c}) + 1$ regardless e being labeled with $\mathbb{0}$ or with $\mathbb{1}$.
   (b) $e = (i_0, i_1)$. By considering $\tilde{c}' = i_1; \tilde{c}$, we go to the previous case since $\varrho(\tilde{c}') = \varrho(\tilde{c})$.

$\square$

We can now prove the theorem:

Proof of Theorem 9. Let $H = T(G^\dagger)$ be the directed graph defined from $G^\dagger$ in Definition 1. We consider the LR greedy execution from $G^\dagger$ and the equivalent FR execution from H given by Theorem 5 which is also a greedy execution. For both executions, we use the notation w_x and θ_x for the work and the termination time of node x as no confusion may arise.

Let i be any node in $G^\dagger$. If $w_i = 0$, then i takes no step, and so $\theta_i = 0$. Otherwise we have to consider two cases.

1. Node i is a single node in $G^\dagger$. From Theorem 5, we derive that

$$w_i = w_{[i]} \text{ and } \theta_i = \theta_{[i]}.$$

We may apply Theorem 3 and obtain

$$\theta_i = \max\{\lambda(\gamma)\colon \gamma \in \mathcal{C}(\rightarrow [i], H) \wedge r(\gamma) = w_i - 1\} + 1.$$

Let γ_0 be a chain in $\mathcal{C}(\rightarrow [i], H)$ such that

$$r(\gamma_0) = w_i - 1 \text{ and } \lambda(\gamma_0) = \theta_i - 1.$$

By definition of the graph transformation T, the chain γ_0 belongs to some set $T(c_0)$, where c_0 is a chain in $\mathcal{C}(\rightarrow i, G^\dagger)$. Since $\gamma_0 \in T(c_0)$, $\lambda(c_0) = \lambda(\gamma_0)$, and $\varphi(c_0) \leq r(\gamma_0)$. By Theorem 6, we deduce that $\Phi(c_0) \leq r(\gamma_0)$. Since c_0 ends with a single node, $\Phi(c_0) = \varrho(c_0)$. By Lemma 4, there exists a chain c'_0 in $G^\dagger$ such that c_0 is a suffix of c'_0, and $\varrho(c'_0) = r(\gamma_0)$. It follows that c'_0 is in $\mathcal{C}(\rightarrow i, G^\dagger)$, $\lambda(c'_0) \geq \lambda(\gamma_0)$, and $\Phi(c'_0) = w_i - 1$. Thereby, the set $\{c \in \mathcal{C}(\rightarrow i, G^\dagger)\colon \Phi(c) = w_i - 1\}$ is non empty, and

$$\theta_i \leq \max\{\lambda(c)\colon c \in \mathcal{C}(\rightarrow i, G^\dagger) \wedge \Phi(c) = w_i - 1\} + 1.$$

Conversely, since $\{c \in \mathcal{C}(\rightarrow i, G^\dagger)\colon \Phi(c) = w_i - 1\}$ is non empty, there exits a chain c_1 in $\mathcal{C}(\rightarrow i, G^\dagger)$ such that $\Phi(c_1) = w_i - 1$, and

$$\lambda(c_1) = \max\{\lambda(c)\colon c \in \mathcal{C}(\rightarrow i, G^\dagger) \wedge \Phi(c) = w_i - 1\}.$$

By Theorem 6, there is a chain γ_1 in $T(c_1)$ such that $r(\gamma_1) = \Phi(c_1)$. By definition of the graph transformation T, the chain γ_1 is in $\mathcal{C}(\rightarrow [i], H)$, and $\lambda(\gamma_1) = \lambda(c_1)$. It follows that

$$\lambda(c_1) \leq \max\{\lambda(\gamma)\colon \gamma \in \mathcal{C}(\rightarrow [i], H) \wedge r(\gamma) = w_i - 1\},$$

and so

$$\max\{\lambda(c)\colon c \in \mathcal{C}(\rightarrow i, G^\dagger) \wedge \Phi(c) = w_i - 1\} \leq \theta_i - 1.$$

2. Node i is a double node. From Theorem 5, we derive that

$$w_i = w_{[i]} + w_{-[i]} \text{ and } \theta_i = \max\{\theta_{[i]}, \theta_{-[i]}\}.$$

In more details, either w_i is odd, and for some integer k,

$$w_i = 2k + 1,\ w_{[i]} = k + 1,\ w_{-[i]} = k,\ \theta_i = \theta_{[i]}, \text{ and } \theta_{[i]} > \theta_{-[i]},$$

or w_i is even, and for some integer k,

$$w_i = 2k,\ w_{[i]} = w_{-[i]} = k,\ \theta_i = \theta_{-[i]}, \text{ and } \theta_{-[i]} > \theta_{[i]}.$$

The two above cases are similar, and so we only give the proof when w_i is odd.

Suppose that $w_i = 2k + 1$ for some integer k. We may apply Theorem 3 and obtain

$$\theta_i = \max\{\lambda(\gamma)\colon \gamma \in \mathcal{C}(\rightarrow [i], H) \wedge r(\gamma) = k\} + 1.$$

Let γ_0 be a chain in $\mathcal{C}(\rightarrow [i], H)$ such that

$$r(\gamma_0) = k \text{ and } \lambda(\gamma_0) = \theta_i - 1.$$

By definition of the graph transformation T, the chain γ_0 belongs to some set $T^+(c_0)$, where c_0 is a chain in $\mathcal{C}(\rightarrow i, G^\dagger)$. Since $\gamma_0 \in T^+(c_0)$, $\lambda(c_0) = \lambda(\gamma_0)$, and $\varphi^+(c_0) \leq r(\gamma_0)$. By Proposition 6, we deduce that $\varrho(c_0) \leq r(\gamma_0)$. By Lemma 4, there exists a chain c'_0 in $G^\dagger$ such that c_0 is a suffix of c_0', and $\varrho(c_0') = r(\gamma_0)$. It follows that c_0' is in $\mathcal{C}(\to i, G^\dagger)$, $\lambda(c_0') \geq \lambda(\gamma_0)$, and $\varphi^+(c_0') = k$. Then $\varphi^-(c_0') = k-1$ or $\varphi^-(c_0') = k$.

Now we show that $\varphi^-(c_0')$ is necessarily equal to k. Suppose for contradiction that $\varphi^-(c_0') = k-1$. By definition of $\varphi^-(c_0')$, there exists a chain γ_0' in $T^-(c_0')$ such that $r(\gamma_0') = k-1$. Therefore, we have

$$r(\gamma_0') = k-1 \ \text{ and } \lambda(\gamma_0') = \lambda(c_0').$$

Since $\lambda(c_0') \geq \lambda(\gamma_0)$ and $\lambda(\gamma_0) = \theta_i - 1$, we conclude that

$$\max\{\lambda(\gamma)\colon \gamma \in \mathcal{C}(\to {}^-[i], H) \wedge r(\gamma) = k-1\} \geq \theta_i - 1$$

which by Theorem 3, is in contradiction with $\theta_i > \theta_{-[i]}$.

Hence, $\varphi^-(c_0') = k$ and so $\Phi(c_0') = 2k$. Thereby, the set $\{c \in \mathcal{C}(\to i, G^\dagger)\colon \Phi(c) = w_i - 1\}$ is non empty, and since $\lambda(c_0') \geq \lambda(\gamma_0) = \theta_i - 1$

$$\theta_i \leq \max\{\lambda(c)\colon c \in \mathcal{C}(\to i, G^\dagger) \wedge \Phi(c) = w_i - 1\} + 1.$$

Conversely, since $\{c \in \mathcal{C}(\to i, G^\dagger)\colon \Phi(c) = w_i - 1\}$ is non empty, there exits a chain c_1 in $\mathcal{C}(\to i, G^\dagger)$ such that $\Phi(c_1) = 2k$, and

$$\lambda(c_1) = \max\{\lambda(c)\colon c \in \mathcal{C}(\to i, G^\dagger) \wedge \Phi(c) = 2k\}.$$

By Theorem 6, $\Phi(c_1) = 2\varphi^+(c_1) - \mathit{Res}(c_1)$. Since $\mathit{Res}(c_1) = 0$ or 1 and $\Phi(c_1)$ is even, it follows that $\mathit{Res}(c_1)$ is actually null, and $\varphi^+(c_1) = k$. Hence, there is a chain γ_1 in $T^+(c_1)$ such that $r(\gamma_1) = k$. By definition of $T^+(c_1)$, $\lambda(\gamma_1) = \lambda(c_1)$. It follows that

$$\lambda(c_1) \leq \max\{\lambda(\gamma)\colon \gamma \in \mathcal{C}(\to [i], H) \wedge r(\gamma) = k\},$$

and so

$$\max\{\lambda(c)\colon c \in \mathcal{C}(\to i, G^\dagger) \wedge \Phi(c) = w_i - 1\} \leq \theta_i - 1.$$

□