Journal of Graph Algorithms and Applications

http://www.cs.brown.edu/publications/jgaa/

vol. 5, no. 5, pp. 3–16 (2001)

Carrying Umbrellas: An Online Relocation Game on a Graph

Jae-Ha Lee

Max-Planck-Institut für Informatik
Saarbrücken, Germany
http://www.mpi-sb.mpg.de
lee@mpi-sb.mpg.de

Chong-Dae Park *Kyung-Yong Chwa*

Department of Computer Science
Korea Advanced Institute of Science and Technology
http://jupiter.kaist.ac.kr
{cdpark,kychwa}@jupiter.kaist.ac.kr

Abstract

We introduce an online relocation problem on a graph, in which a player that walks around the vertices makes decisions on whether to relocate mobile resources, while not knowing the future requests. We call it *Carrying Umbrellas*. This paper gives a necessary and sufficient condition under which a competitive algorithm exists. We also describe an online algorithm and analyze its competitive ratio.

Communicated by T. Nishizeki, R. Tamassia and D. Wagner:
submitted December 1998; revised March 2000 and July 2000.

Research supported in part by the KOSEF (Korea Science and Engineering Foundation) under grant 98-0102-07-01-3.

1 Introduction

"To carry an umbrella or not?" This is an everyday dilemma. This dilemma does not vanish even when correct short-term weather forecasts are available. For example, if it is sunny now but there is no umbrella at the current destination, one has to carry an umbrella so as not to get wet when leaving there later. To illustrate some of our concepts, we describe a detailed scenario.

Picture a person who walks around N places. At each place, he is told where to go next and then must go there. As a usual person, he dislikes being caught in the rain with no umbrella. Since today's weather forecast is correct, he carries an umbrella whenever it is rainy. However, he does not know the future destinations and weather, which might be controlled by the malicious adversary. We say a person is *safe* if he never gets wet. There exists a trivial safe strategy: $'$Always carry an umbrella$'$. However, carrying an umbrella in sunny days is annoying. As an alternative, he has placed several umbrellas in advance and thinks about an *efficient* strategy; he hopes, through some cleverness, to minimize the number of sunny days on which he carries an umbrella. We say a person's strategy is *competitive* if he carries an umbrella in a small portion of sunny days (for details, see Section 1.1). The following questions are immediate: (1) What is the minimum number of umbrellas with which the person can be safe and competitive? (For example, is placing one umbrella per place sufficient or necessary?) (2) What is a safe and competitive strategy? We formally define the problem next.

1.1 A Game

Let $G=(V,E)$ be a simple graph with N vertices and M edges. An integer $u(v)$ is associated with each vertex $v \in V$, indicating the number of umbrellas placed on the vertex v. We use $u(G)$ to represent the total number of umbrellas in G. Consider an on-line game between a player and the adversary, assuming that $u(G)$ is fixed in advance.

Game Carrying-Umbrellas($G, u(G)$)
1. **Initialization**
   `Player` determines the initial configuration of $u(G)$ umbrellas;
   `Adversary` chooses the initial position $s \in V$ of `Player`;
2. **for** $i = 1$ **to** L **do**
   (a) `Adversary` specifies (v, w), where w is a boolean value and $v \in V$ is adjacent to the current vertex of `Player`;
   (b) `Player` goes to v; If $w=1$, he must carry at least one umbrella;

As an initialization, we assume that the player first determines the initial configuration of $u(G)$ umbrellas and later the adversary chooses the initial position $s \in V$ of the player, although our results of the present paper also hold even if the initialization is done in reverse order. A *play* consists of L *phases*, where L is determined by the adversary and unknown to the player. In each phase, the adversary gives a request (v, w), where v is adjacent to the current vertex of the player and w is a boolean value that represents the weather; $w=1$ indicates that it is *rainy*, and $w=0$ *sunny*. For this request (v, w), the player must go to the specified vertex v and decide whether to carry umbrellas or not. If $w=1$, he must carry at least one umbrella; only in sunny days, he may or may not carry umbrellas. Notice that the player can carry an arbitrary number of umbrellas, if available.

The player *loses* if he cannot find any umbrella at the current vertex in a rainy phase. A strategy of the player is said to be *safe* if it is guaranteed that whenever $w = 1$ the player carries an umbrella. Moreover, a strategy of the player should be *efficient*.

As a measure of efficiency, we adopt the *competitive ratio* [9], which has been widely used in analyzing the performance of online algorithms. Let $\sigma=\sigma_1 \cdots \sigma_L$ be a *request sequence* of the adversary, where $\sigma_i=(v_i, w_i)$ is the request in the i-th phase. The *cost* of a strategy $\mathcal{A}$ for σ, written $C_{\mathcal{A}}(\sigma)$, is defined to be the number of phases in which the player carries an umbrella by the strategy $\mathcal{A}$. Then, the *competitive ratio* of the strategy $\mathcal{A}$ is

$$\frac{C_{\mathcal{A}}(\sigma)}{C_{Opt}(\sigma)}$$

where *Opt* is the optimal off-line strategy of the player. (Since *Opt* knows the entire σ in advance, it pays the minimum cost. However, *Opt* cannot be implemented by any player and is used only for comparison.) A strategy whose competitive ratio is bounded by c is termed *c-competitive*; it is simply said to be *competitive*, if it is c-competitive for some bounded c irrespective of the length of σ. The player *wins*, if he has a safe and competitive strategy; he *loses*, otherwise.

Not surprisingly, whether the player has a winning strategy depends on the number of umbrellas. In this paper, we are interested in that number.

Definition 1 *For a graph G, we define $u^*(G)$ to be the minimum number of umbrellas with which the player has a winning strategy in G.*

1.2 Summary of our results

Let $G=(V, E)$ be a connected simple graph (having no parallel edges) with N vertices and M edges. We show that $u^*(G)=M+1$. That is, no strategy of the player is safe and competitive if $u(G)<M+1$ (Section 3) and there exists a competitive strategy of the player if $u(G)\geq M+1$ (Section 4). The competitive ratio of our strategy is $b(G)$, the number of vertices in the largest biconnected component in G. Moreover, the upper bound is attained by the *weak player* that carries at most one umbrella in a phase, and the competitive ratio of $b(G)$ is optimal for some graphs when $u(G)=M+1$. These results can be easily extended to the case that G is not connected. For a simple graph G with M edges and k connected components, $u^*(G) = M + k$. Throughout this paper, we assume G is connected and simple.

1.3 Related works

Every online problem can be viewed as a game between an online algorithm and the adversary [8, 3, 5]. In this prospect, Chrobak and Larmore [6] introduced online game as a general model of online problems. Clearly, the problem of the present paper is an example of the online game. Many researchers have studied online problems on a graph. Such examples include the k-server problem [1, 7] and graph coloring [2, 4].

2 Examples

In order to introduce the reader to the ideas, we explain two examples.

Example 1. Consider $K_2 = (\{v_1, v_2\}, \{(v_1, v_2)\})$ that consists of two vertices v_1 and v_2 and an edge (v_1, v_2). Let $u(K_2) = 1$. We assume without loss of generality that the unique umbrella is initially placed at v_1 (see Figure 1). We claim that no strategy of the player can be safe and competitive in K_2. Although the adversary can maliciously determine the initial position s of the player, we explain both cases.

Figure 1: K_2

Case A. If $s = v_2$, the adversary only has to choose $\sigma_1 = (v_1, \mathit{rainy})$. The player at v_2 has no umbrella and so must get wet in the first phase. Thus it is not safe.

Case B. If $s = v_1$, the adversary first chooses $\sigma_1 = (v_2, \mathit{sunny})$. For this request σ_1, the player *must* carry an umbrella because otherwise, the resulting configuration is isomorphic to that of Case A; afterwards, the request $\sigma_2 = (v_1, \mathit{rainy})$ makes the player not safe. Therefore, at the end of the first phase, the player must be at v_2 and $u(v_1) = 0$ and $u(v_2) = 1$. Note that this is isomorphic to the initial configuration. Thus, in a similar fashion, the subsequent requests such as $\sigma_i = (v_1, \mathit{sunny})$ if i is even and $\sigma_i = (v_2, \mathit{sunny})$ if i is odd forces the player to always carry an umbrella. Therefore, $C_A(\sigma) = L$. However, since the weather is always *sunny*, *Opt* does not carry an umbrella and $C_{Opt}(\sigma) = 0$. Therefore, any safe strategy of the player is not competitive.

It is easily seen that if $u(K_2) = 2$, then the player has a simple 2-competitive strategy: when it is sunny, the player carries an umbrella if and only if he is going from a vertex with two umbrellas to one with zero.

Example 2. K_3 is a triangle, that is, $K_3 = (\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_2, v_3), (v_3, v_1)\})$. Let $u(K_3) = 3$. A natural initial configuration of the umbrellas is that $u(v_i) = 1$ for $i = 1, 2, 3$ (Figure 2).

Though every vertex in K_3 initially has an umbrella, we claim that no strategy of the player is safe and competitive. Without loss of generality, let v_1 be the start vertex. The adversary first chooses $\sigma_1 = (v_2, \mathit{rainy})$. For this input, the player must carry at least one umbrella, making $u(v_1) = 0$ and $u(v_2) = 2$. Next, let $\sigma_2 = (v_3, \mathit{sunny})$. For this input, the player must carry an umbrella because otherwise, the resulting subgraph induced by $\{v_1, v_3\}$ contains only one umbrella and is isomorphic to K_2 in Example 1; afterwards, the adversary can make the player unsafe or not competitive in K_2. Hence, at the end of the phase 2, there are two cases depending on the number of umbrellas the player has carried: $u(v_2) = 1$ and $u(v_3) = 2$ or $u(v_2) = 0$ and $u(v_3) = 3$.

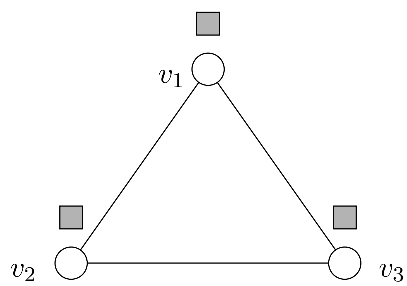

Figure 2: K_3

In either cases, the next request is $\sigma_3 = (v_2, \mathit{sunny})$. Then the player must carry enough umbrellas to make $u(v_2) \geq 2$, because otherwise, K_2 consisting of v_1 and v_2 is deficient in umbrellas. Thus, the resulting configuration is isomorphic to that at the end of the phase 2. Similarly, the subsequent request sequence $\sigma_i = (v_3, \mathit{sunny})$ if i is even and $\sigma_i = (v_2, \mathit{sunny})$ if i is odd forces the player to always carry umbrellas. Therefore, $C_A(\sigma) = L$. However, since *Opt* only has to carry an umbrella in the first phase, the competitive ratio can be arbitrarily large as L becomes large.

As we shall see later, the player has a 3-competitive strategy, if $u(K_3) \geq 4$. It means $u^*(K_3) = 4$.

3 Lower Bound

In this section, we show the lower bound on $u^*(G)$. To simplify the explanation, we first consider the weak player that can carry at most one umbrella in a phase, and show that $u^*(G) \geq M+1$ for the weak player. Later the same bound is shown for the general player that can carry multiple umbrellas.

Theorem 1 *Let G be a simple connected graph with N vertices and M edges. Under the constraint that the player can carry at most one umbrella in a phase, $u^*(G) \geq M+1$.*

Proof: We show that if $u(G) \leq M$ then no strategy of the player is safe and competitive in G. Equivalently, it suffices to show that any safe strategy of the player is not c-competitive for any fixed c ($< \infty$). For convenience, imagine we are the adversary that would like to defeat the player.

Let $\sigma = \sigma_1 \sigma_2 \cdots \sigma_L$ denote the request sequence that we generate. Recall that the starting vertex s of the player is determined by the adversary. We fix s as a vertex such that $G - \{s\}$ is connected; we call such a vertex *non-cut vertex* of G. (A non-cut vertex of G is easily obtained by taking a leaf vertex in an arbitrary spanning tree of G.) We say that a graph G' is *deficient* in umbrellas, if $u(G')$ is no greater than the number of edges in it.

The basic idea is to force the player to go into a deficient subgraph of G and recursively defeat the player in it. Suppose we were somehow able to make the player be at s and at the same time, make $u(s) = d+1$, where d is the degree of s in G. This would tell us that $u(G - \{s\})$ is strictly less than the number of edges in $G - \{s\}$. In the next step, we make the player go into $G - \{s\}$. Even if the player has carried an umbrella, $G - \{s\}$ is deficient and the player is in it. Now we use recursion: after making the player go to a non-cut vertex in G–$\{s\}$, we can make the player not safe or not competitive in $G - \{s\}$ recursively. So our subgoal is to make $u(s)$ increase to $d+1$.

Let $v_1, \cdots, v_d$ be the adjacent vertices of s. We begin with explaining how to increase $u(s)$. Suppose that at the start of the $(2i-1)$-th phase, the player is located at s and $u(s) = j$. In the subsequent two phases, the adversary makes the player go to v_j and return to s. That is, $\sigma_{2i-1} = (v_j, \mathit{sunny})$ and $\sigma_{2i} = (s, *)$. The $2i$-th weather w_{2i} depends on the player's decision in the $(2i$–$1)$-th phase; if the player carried an umbrella in the $(2i-1)$-th phase, then w_{2i} is set to *sunny*; otherwise *rainy*. Before the generation of σ_{2i}, the adversary tests if $G - \{s\}$ is deficient in umbrellas. If it is, the adversary makes the player move to a non-cut vertex of G–$\{s\}$ and calls the recursive procedure. The strategy of the adversary is summarized in Algorithm 1. Initially, the adversary calls `Adversary`$(G, s, 1)$. The *lines 2–3* are the generation of σ_{2i-1} and the *lines 9–13* are that of σ_{2i}. The *lines 5–8* are the recursive procedure. In `MOVE-TO-NON-CUT-VERTEX`(G'), the adversary picks an arbitrary non-cut vertex s' in G', makes the player go to s' while setting the weather to *sunny*, and returns s'.

To see why this request sequence makes $u(s)$ increase, we define a weighted sum Φ of the umbrellas placed in vertex s and its neighbors, where j umbrellas in s weighs $0.5, 1.5, \cdots, j$–0.5, respectively and each umbrella in vertex v_i weighs i. Specifically,

$$\Phi = \sum_{i=1}^{d} i \cdot u(v_i) + \sum_{j=1}^{u(s)} (j - 0.5)$$

Let Φ_k denote Φ at the end of the k-th phase. We are interested in the change

Algorithm 1 `Adversary`(G, s, i)

while TRUE **do**
  $j = u(s)$;
  $\sigma_{2i-1} = (v_{\min(j,d)}, sunny)$;
  { `Player`'s move for σ_{2i-1}}
  if $u(G-\{s\}) \leq |E(G-\{s\})|$ **then**
    $s' =$ `MOVE-TO-NON-CUT-VERTEX`$(G-\{s\})$;
    `Adversary`$(G-\{s\}, s', i+1)$;
  end if
  if the player carried an umbrella in phase $(2i-1)$ **then**
    $\sigma_{2i} = (s, sunny)$;
  else
    $\sigma_{2i} = (s, rainy)$;
  end if
  i++;
end while

between Φ_{2k-2} and Φ_{2k}. Suppose that $u(s)$ is j at the end of the $(2k-2)$-th phase. Depending on the decision of the player, there are four cases to consider.

Case A. If the player carried an umbrella throughout the $(2k-1)$-th and the $2k$-th phase, we have $\Phi_{2k-2} = \Phi_{2k}$ because the number of umbrellas is unchanged. However, the weather must be *sunny* from `Adversary`(G, s, i). Therefore, the player did *useless carrying*.

Case B. If the player carried an umbrella from s to v_j only, $\Phi_{2k} = \Phi_{2k-2} + 0.5$. This is because the umbrella moved had weight $j-0.5$ at s and has weight j at v_j.

Case C. If the player carried an umbrella from v_j to s only, $\Phi_{2k} = \Phi_{2k-2} + 0.5$. This is because the weight of the umbrella moved changes from j to $j+0.5$.

Case D. The remaining case is that the player never carried an umbrella. However, if the player didn't carry an umbrella in the $(2k-1)$-th phase, the adversary chooses the weather w_{2k} *rainy*. Since we are considering a safe strategy, the player must carry an umbrella. Hence this case is impossible.

In summary, over the two phases $2k-1$ and $2k$, Φ increases or is unchanged; if Φ is unchanged, the player did *useless carrying* in sunny days. Note that successive *useless carryings* make the player's strategy not competitive, because *Opt* would not carry an umbrella over two *sunny* days. Hence, in order to be competitive, the player must sometimes increase Φ.

How many times can Φ increase before $u(s)$ becomes $d+1$? Recall that once $u(s)$ becomes $d+1$, we can make $G-\{s\}$ deficient in umbrellas and start the recursive procedure. While $u(s) \leq d$, each umbrella can have $2d$ different weights. Moreover, since $u(G)$ is less than or equal to the number of edges in G, Φ can increase in at most $2d \times \frac{N^2}{2} < N^3$ phases before $u(s)$ becomes $d+1$. Thus, Φ can increase in at most N^3 phases, before the recursive procedure is called.

In `Adversary(`$G-\{s\}, s', i+1$`)`, we recursively define the request sequence and weight sum Φ'. Thus, in $G-\{s\}$, Φ' can increase in at most $(N-1)^3$ phases, before the player moves into some deficient subgraph. In all recursive procedures, the weight sum can increase in at most $N^3+(N-1)^3+\cdots+2^3<N^4$ phases. This means that the player carries an umbrella in all except at most N^4 phases, and that for the same input, *Opt* can carry an umbrella in at most N^4 phases. At the final recursive call, the graph becomes K_2 and the number of umbrellas in it is at most 1. If there is no umbrella, the player is not safe, which is a contradiction. Otherwise, the situation is the same as in Example 1 in the previous section; we can continue this game forever with no cost for *Opt* and the cost of 1 for the player in every phase.

One remaining step we have to consider is `MOVE-TO-NON-CUT-VERTEX`, which is executed at the start of each recursive call. Since the i-th call of `MOVE-TO-NON-CUT-VERTEX` is done in G' with $(N-i)$ vertices, at most $(N-i)$ requests are given and thus the cost of *Opt* is at most $(N-i)$. Throughout all executions of `MOVE-TO-NON-CUT-VERTEX`, the cost of *Opt* is at most N^2, because the total number of requests is at most N^2.

Since this game can continue forever as long as the player is safe, we can choose the length L of the input sequence σ to be arbitrarily large. Specifically, we let $L \geq (c+1)\cdot(N^4+N^2)$. Then, the cost of the player's strategy $\mathcal{A}$ is

$$C_{\mathcal{A}}(\sigma) \geq L - N^4 - N^2 \geq c\cdot(N^4+N^2)$$

And, the cost of *Opt* is

$$C_{Opt}(\sigma) < (N^4+N^2)$$

Therefore, the competitive ratio is

$$\frac{C_{\mathcal{A}}(\sigma)}{C_{Opt}(\sigma)} > c$$

for an arbitrary $c(<\infty)$, completing the proof. ⊔⊓

Now we present the main result of this section.

Theorem 2 *Let G be a simple connected graph with M edges. Then, $u^*(G) \geq M+1$.*

Proof: Here, the player can carry an arbitrary number of umbrellas in a phase. The algorithm of the adversary is exactly same as `Adversary`(G,s,i). However, the analysis is slightly more complex. The crucial fact in proving Theorem 1 was that if $u(s) \geq d+1$ then the adversary makes the player go to v_d, resulting that $G-\{s\}$ is deficient even if the player carries an umbrella, and calls the recursive procedure. In this theorem, however, the player can carry an arbitrary number of umbrellas and thus $u(s) \geq d+1$ does not imply that the recursive procedure is called in the next phase. In other words, $u(s)$ may fluctuate, even over $d+1$, without being trapped into the recursive procedure.

Φ is defined as in Theorem 1. We say Φ_{2k} is *stable* if $u(s) < d$ at the end of the $2k$-th phase; *unstable* otherwise. We divide the request sequence into a number of *stages*, each of which starts with stable Φ_{2k} and ends right before the next stable $\Phi_{2k'}$. We are interested in the change between Φ_{2k} and $\Phi_{2k'}$.

If $k' = k+1$ (i.e., no unstable Φ exists in this stage), the change of Φ is the same as in Theorem 1, except that Φ can increase by more than 0.5 when the player carries multiple umbrellas. Assume otherwise, that is, $\Phi_{2k''}$ ($k < k'' < k'$) is unstable. Unless the player carries enough umbrellas to make $G - \{s\}$ not deficient in phase $2k''+1$, the recursive procedure is called. Thus, the player must carry at least one umbrella in the $(2k''+1)$-th phase. Similarly, the player must carry at least one umbrella in phase $2k''$, to make Φ unstable. Therefore, the player always carries an umbrella right before and right after Φ is unstable and so the weather is always *sunny*. Comparing Φ_{2k} and $\Phi_{2k'}$ reveals that some umbrellas have moved from v_j to v_d (because the player went to some $v_j (j < d)$ in the $(2k+1)$-th phase, and returned to s with enough umbrellas to make Φ unstable and afterwards goes to and returns from v_d till Φ becomes stable). Therefore, $\Phi_{2k'} \geq \Phi_{2k} + 1$. Moreover, the player always carries at least one umbrella, except in the $(2k+1)$-th and the $2k'$-th phases.

Combining the cases of $k' = k+1$ and $k' > k+1$, we can conclude that in a stage, either (1) Φ increases at least $0.5 \cdot x$ ($x = 1$ or 2) and the player does not carry umbrellas in at most x phases, or (2) Φ is unchanged and the player always does useless carrying. *Opt* still suffices to carry umbrellas only in x phases in case of (1), depending on the change of Φ. The remaining proof is same as that of Theorem 1. (Only one difference is that we concentrate on stable Φ. Before the recursive procedure is called, stable Φ can increase at most N^3 times, the player does not carry umbrellas in at most N^3 phases, and *Opt* carries umbrellas in at most N^3 phases.) By using the counting arguments in Theorem 1 (ignoring unstable Φ), this theorem is easily seen. $\square$

4 Upper Bound

In this section, we show the upper bound on $u^*(G)$.

Theorem 3 *Let G be a simple connected graph with N vertices and M edges. Then, $u^*(G) \leq M+1$.*

Proof: It suffices to show that if $u(G) = M+1$ then the player has a safe and competitive strategy. Throughout this section, we assume that $u(G) = M+1$. To show this theorem, we only consider the weak player that carries at most one umbrella in a phase. Imagine that we are the player that should be safe and competitive against the adversary.

The heart of our strategy is how to decide whether to carry an umbrella in sunny days. To help this decision, we maintain two things: labels of umbrellas and a subtree of G. First, let us explain the labels, recalling that $u(G) = M+1$. M umbrellas are labeled as $\gamma(e)$ for each edge $e \in E$, and one remaining umbrella is labeled as *Current*. Under these constraints, labels are updated during the game. In addition to the labels, we also maintain a dynamic subtree $\mathcal{S}$ of G, called *skeleton*.

- $\mathcal{S}$ is a subtree of G whose root is the current position of the player.
- The root in $\mathcal{S}$ has the umbrella labeled as *Current*.
- For an edge $e \in \mathcal{S}$, the child-vertex of e has the umbrella labeled as $\gamma(e)$.
- For an edge $e \notin \mathcal{S}$, the umbrella labeled as $\gamma(e)$ lies in one endpoint of e.

Note that all umbrellas labeled as $\gamma(e)$ are placed on one endpoint of e. In an example of Figure 3, the current skeleton $\mathcal{S}$ is enclosed by a dotted line and every edge in $\mathcal{S}$ is directed towards the child, indicating the location of its umbrella.

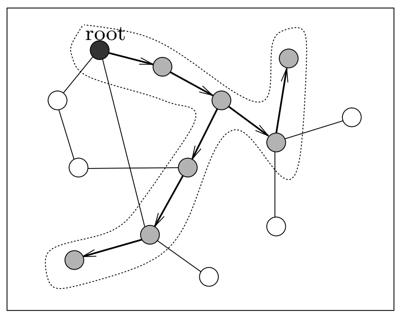

Figure 3: Example of a skeleton (enclosed in a dotted line).

$\mathcal{S}$ is initialized as follows: The player first chooses an arbitrary spanning tree T and places one umbrella in each vertex of T, while their labels undetermined. Then, the number of umbrellas used in T is N. For an edge e not in T, we place an umbrella on *any* endpoint of it; this umbrella is labeled as $\gamma(e)$. Then, the number of umbrellas labeled equals the number of edges not in T that is $M - N + 1$ and so the total number of umbrellas used is $M + 1$. After the adversary chooses the start vertex s, the umbrellas in T are labeled. The umbrella in s is labeled as *Current*, and the umbrella in v ($\neq s$) is labeled as $\gamma(e)$, where e connects v with its parent in T.

Now, we describe the strategy of the player. Suppose that currently, the player is at v_{i-1} and the current request is (v_i, w_i). If the weather is *rainy*, the player has no choice; it has to move to v_i with carrying the umbrella *Current*. In this case, the player resets $\mathcal{S}$ to a single vertex $\{v_i\}$. It is easily seen that the new skeleton satisfies the *invariants*, because only the umbrella *Current* is moved. The more difficult case is when the weather is *sunny*.

Suppose that $w_i = 0$, i.e., *sunny*. Depending on whether v_i belongs to $\mathcal{S}$ or not, there are two cases. If v_i belongs to $\mathcal{S}$ (Figure 4a), then the player moves to v_i without an umbrella. Though no umbrellas are moved, labels must be changed. Let $e_1, e_2, \cdots, e_r$ be the edges encountered when we traverse from v_{i-1} to v_i in $\mathcal{S}$. The umbrella *Current* is relabeled as $\gamma(e_1)$ and the umbrella

$\gamma(e_k)$ is relabeled as $\gamma(e_{k+1})$ for $1 \leq k \leq r-1$ and finally, the umbrella $\gamma(e_r)$ is relabeled as the new *Current*. Observe that new skeleton $\mathcal{S}$ still satisfies the *invariants*.

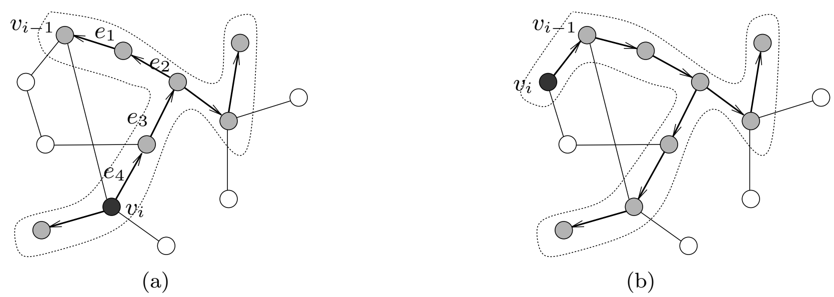

Figure 4: After Figure 3, $(v_i, \textit{sunny})$ is given. (a) v_i belongs to $\mathcal{S}$. (b) v_i does not belong to $\mathcal{S}$.

If v_i does not belong to $\mathcal{S}$ (Figure 4b), the edge $e = (v_{i-1}, v_i)$ does not lie in $\mathcal{S}$. Remember that the umbrella $\gamma(e)$ was at v_{i-1} or v_i from the *invariants*. If the umbrella $\gamma(e)$ was placed at v_{i-1}, the player moves to v_i carrying the umbrella *Current*, and adds the vertex v_i and the edge (v_{i-1}, v_i) to $\mathcal{S}$. If the umbrella $\gamma(e)$ was placed at v_i, the player moves to v_i without carrying an umbrella and adds the vertex v_i and the edge (v_{i-1}, v_i) to $\mathcal{S}$ and additionally, swaps the labels of *Current* and $\gamma(e)$. In both cases, it is easy to see that $\mathcal{S}$ still satisfies the *invariants* (see Figure 4b). The strategy of the player is summarized in Algorithm 2.

In order to complete the proof, it suffices to show that `Player`(G) is a safe and competitive strategy. First, it is easily seen that the player is safe, because the player always has the umbrella *Current*. Next, we show that `Player`(G) is an N-competitive strategy. Let us divide the request sequence into a number of *stages*, each of which contains exactly one '*rainy*' and starts with '*rainy*'. Remember that the player resets $\mathcal{S}$ to a single vertex at the start of every stage. The cost of `Player`(G) in a stage is at most N, because the cost 1 is paid only when $\mathcal{S}$ is set to a single vertex or becomes larger. The cost of *Opt* in a stage is at least one, because each stage starts with '*rainy*'. The cost of the initial part of the sequence until the first '*rainy*' is zero for `Player`(G) since S has size N. Therefore, the competitive ratio is at most N. ◻

Unfortunately, the competitive ratio of N we obtained above is best possible for some graph G when $u(G) = M+1$.

Theorem 4 *Suppose that G is the N-vertex ring and that $u(G) = N+1$. The competitive ratio of any strategy of the player is at least N.*

Proof: Assume N vertices in G are numbered from 0 to $N-1$. See Figure 5a. As an initial configuration, the player might evenly distribute $M+1$ umbrellas

Algorithm 2 `Player`(G)

$i = 1$;
pick a spanning tree T of G; determine the initial configuration of $u(G)$ umbrellas;
receive s from the adversary;
transform T to the rooted tree $\mathcal{S}$ with root s; label the umbrellas;
while TRUE **do**
    $\{\sigma_i = (v_i, w_i)\}$
    if $w_i = \textit{rainy}$ **then**
        carry an umbrella; $\mathcal{S} = (\{v_i\}, \emptyset)$;
    else
        if $v_i \in V(\mathcal{S})$ **then**
            do not carry an umbrella; new root is v_i; relabel;
        else
            if $\gamma(v_{i-1}, v_i)$ is placed at v_i **then**
                do not carry an umbrella; $V(\mathcal{S}) = V(\mathcal{S}) \cup \{v_i\}$; $E(\mathcal{S}) = V(\mathcal{S}) \cup \{(v_i, v_{i-1})\}$; relabel;
            else
                carry an umbrella; $V(\mathcal{S}) = V(\mathcal{S}) \cup \{v_i\}$; $E(\mathcal{S}) = V(\mathcal{S}) \cup \{(v_i, v_{i-1})\}$; relabel;
            end if
        end if
    end if
    $i{+}{+}$;
end while

(i.e., one umbrella per vertex and the remaining one to an arbitrary vertex). Otherwise, one vertex, say i, has no umbrella and the adversary forces the player to rotate the ring, setting the weather to *sunny*. When entering into i, the player must carry an umbrella, yet Opt never carries umbrellas, which makes the player not competitive.

Thus, we can assume that $u(i) = 2$ and $u(j) = 1$ for each j $(\neq i)$. Then the adversary makes the player go to vertex $i-1$, setting the weather to *sunny*. The player must go to $i-1$ without carrying umbrellas. Next request is $(i, \textit{rainy})$, so the player must carry an umbrella from $i-1$ to i. As a result, $u(i) = 3$ and $u(i-1) = 0$ and $u(j) = 1$ for other j (Figure 5b).

In any subsequent k-th steps $(1 \leq k \leq N-2)$, the adversary gives request $(i+k, \textit{sunny})$, where $+$ is taken modulo N. In spite of sunny days, the player must carry an umbrella because otherwise, the subgraph induced by the vertices $\{i+k, i+k+1, \cdots, i-1\}$ shall contain $N-k-1$ edges and $N-k-1$ umbrellas and the subgraph is deficient. The final request $(i-1, \textit{sunny})$ brings the player back to the starting point. Since v_{i-1} does not have any umbrella before this request, the player must bring one to be safe. For these requests, $C_A(\sigma)$ is at least N and C_{Opt} is 1 because only one day is rainy. Therefore, the competitive

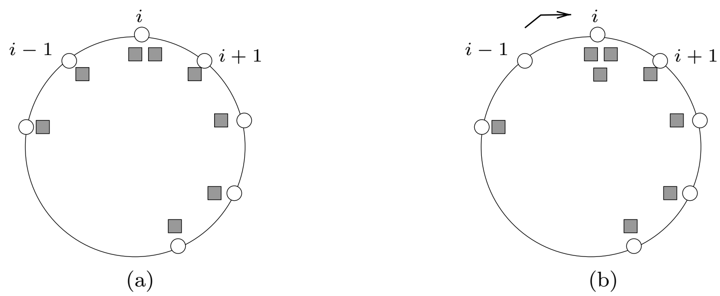

Figure 5: Theorem 4.

ratio is at least N. ◻

Though the competitive ratio of N is optimal for rings, we can easily design a 2-competitive algorithm for trees. Here we slightly modify the algorithm `Player`(G) and show that the new algorithm is $b(G)$-competitive, where $b(G)$ is the number of vertices in the largest biconnected component in G. Note that $b(G)$ is 2 for any tree G.

Theorem 5 *Let G be a simple connected graph with M edges and let $u(G) = M+1$. The player has a $b(G)$-competitive strategy, where $b(G)$ is the number of vertices in the largest biconnected component in G.*

Proof: It suffices to modify `Player`(G) for rainy phases. Suppose it is rainy when the player goes from u to v. In `Player`(G), the player resets S to a single vertex v (i.e. the current vertex) in every rainy phase.

In the new strategy, the player resets S in the biconnected component that contains u and v. More formally, suppose the edges in G are partitioned into l biconnected components $C_1, \cdots C_l$ and C_k contains the edge (u, v). Then the player resets S in C_k to a single vertex v and maintains other edges outside C_k. Note that the skeleton S may be disconnected during the game.

The analysis of the new strategy is nearly the same as that of `Player`(G). Consider the *sunny* phase i in which the player carries an umbrella from u to v. Let C_k be the biconnected component that contains the edge (u, v). This cost 1 is charged to the last rainy phase j $(< i)$ in which u and v are disconnected in S (i.e. the recent rainy phase in which the player was traversing an edge in C_k). The cost of the player in every rainy phase is charged to itself. In this way, all costs of the player are charged to rainy phases in the same biconnected component. It is easily seen that at most $b(G)$ is charged to each rainy phase, because the player resets S in the biconnected component in G containing the current edge, whose size is bounded by $b(G)$. ◻

5 Concluding Remarks

In this paper, we introduced an online game on a graph, *Carrying Umbrellas*. We showed that the player has a safe and competitive strategy if and only if $u(G)$ is at least $M+1$. We also presented an $b(G)$-competitive strategy of the player when $u(G)=M+1$. This paper is the first attempt on this problem, and many questions remain open. First, observe that $u^*(G)$ is independent of the *topology* of the graph, the starting position of the player, and the number of umbrellas that the player can carry in a phase. Finding this number $u^*(G)$ for digraphs seems to be interesting from a graph-theoretical viewpoint. Second, we believe that the problem *Carrying Umbrellas* can be extended to be of practical use. For now, however, it is mainly of theoretical interest. Finally, reducing the competitive ratio with more umbrellas is an open problem.

Acknowledgment We are grateful to Oh-Heum Kwon for motivating the problem of this paper, to Jung-Heum Park for helpful discussions and to anonymous referees for giving constructive comments.

References

[1] N. Alon, R. M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal of Computing*, 24(1):78–100, 1995.

[2] R. Beigel and W. Gasarch. The mapmaker's dilemma. *Discrete Applied Mathematics*, 34(1):37–48, 1991.

[3] S. Ben-David, A. Borodin, R. Karp, G. Tardos, and A. Widgerson. On the power of randomization in online algorithms. In *Proc. of 22nd STOC*, pages 379–386, 1990.

[4] H. L. Bodlaender. On the complexity of some coloring games. *Int. J. of Foundation of Computer Science*, 2:133–147, 1991.

[5] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 1998.

[6] M. Chrobak and L. Larmore. The server problem and online games. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 7, pages 11–64. AMS-ACM, 1992.

[7] M. S. Manasse, L. A. McGeoch, and D. D. Sleator. Competitive algorithms for server problem. *Journal of Algorithms*, 24(1):78–100, 1990.

[8] P. Raghavan and M. Snir. Memory versus randomization in online algorithms. In *Proc. of 16th ICALP*, pages 687–703, 1987.

[9] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of ACM*, 2(28):202–208, 1985.