The Complexity of the Infinity Replacement Problem in the Cyber Security Model

Supachai Mukdasanit
The Theory of Computation Group
Computer Engineering Department
Chiang Mai University
Chiang Mai, 50200, Thailand
Email: supachai_m@cmu.ac.th

Sanpawat Kantabutra
The Theory of Computation Group
Computer Engineering Department
Chiang Mai University
Chiang Mai, 50200, Thailand
Email: sanpawat@alumni.tufts.edu

***Abstract*—In this article we present a new defensive problem on a security system. Let M be a security model (T, C, P), where $T = (V, E)$ is a rooted tree rooted at r, C is a multiset of $|E(T)|$ costs and P is a multiset of $|V(T)| - 1$ prizes and let (T, c, p) be a security system, where c and p are bijections $c : E(T) \to C$ and $p : V(T) \setminus \{r\} \to P$, respectively. Given a budget $B \in \mathbb{Z}^+$, we consider the problem of determining the existence of an edge $e \in E(T)$, where $c(e) = i$ such that the maximum total of prizes obtained from an optimal attack in the security system (T, c, p) is minimum when i is replaced by ∞ . We define the decision and optimization versions of the problem and examine their computational complexities. We prove that the decision problem is *NP*-hard and provide a pseudo-polynomial time algorithm for the optimization version of the problem. Additionally, we show that some restricted instances can be solved in polynomial time. In the end conclusion and an example of the application of the algorithm are given.**

Keywords*—*cyber security, defensive strategy, infinity replacement, cyber attack, NP-hardness, pseudo-polynomial time

I. Introduction

The number of cyber attacks has been increasing rapidly. The cyber crimes statistics in 2016 showed that the number of attacks was 1061, increasing from 1017 in 2015 [1]. Moreover, the cyber crimes continue to be very costly for the organizations. Ponemon, in 2016, [2] reported that the mean annualized cost of cyber crimes for 58 benchmarked organizations is 15 million US dollars per year. Therefore, it is important that cyber systems are provably secure. However, the present practice of building a security system is based on experience [3]. For this reason, Schneider suggested the need for the science of cyber security [4]. Armstrong et al. also discussed that one of the tools for using to understand problems in the cyber security domain is the complexity science [5].

Agnarsson et al. were the first to propose a theoretical cyber security model [3]. They defined the problems of attacking a system within the model and studied their complexities. They proved that the decision problems of attacking a given system is *NP*-complete and their optimization version is *NP*-hard and also provided the pseudo-polynomial time algorithm for solving the problems. In the other article Agnarsson et al. presented defensive strategies for the cyber security model [6]. They showed that in general it is not possible to develop an optimal security system for a given cyber security model. However, they showed that if a tree in a given model is a rooted path or a rooted star, there is an optimal security system for a given cyber security model. In addition, they proved that the cyber security P-model and C-model have optimal security systems for exactly the same types of trees consisting of a rooted path, a rooted star, a rooted 3-caterpillar, and a rooted 4-spider. Finally, they defined a good security system to defend an optimal attack and also showed that a good security system can be constructed in polynomial time.

This research is motivated by [3] and [6]. We define a new defensive strategy on a cyber security system that was presented in [6]. The problem is described as follows. Let M be a security model (T, C, P), where $T = (V, E)$ is a rooted tree rooted at r, C is a multiset of $|E(T)|$ costs and P is a multiset of $|V(T)| - 1$ prizes and let (T, c, p) be a security system, where c and p are bijections $c : E(T) \to C$ and $p : V(T) \setminus \{r\} \to P$, respectively. Given a budget $B \in \mathbb{Z}^+$, we consider the problem of determining the existence of an edge $e \in E(T)$, where $c(e) = i$ such that the maximum total of prizes obtained from an optimal attack in the security system (T, c, p) is minimum when i is replaced by ∞ . We define the decision and optimization versions of the problem to examine their complexities. Moreover, we provide a pseudo-polynomial time algorithm for the Optimization Problem. Finally, we show that some restricted problem instances can be solved in polynomial time.

The outline of this article is as follows: in section 2 we describe some definitions and notations that will be used throughout this article; we examine the complexities of the related problems in section 3; conclusions and an example of the application of the algorithm are discussed in section 4.

II. Definitions and Notations

In this section we describe some definitions and notations that are used throughout this article. We borrow the cyber security model and the security system definitions from [6]. The definitions are as follows.

Definition 2.1. [Cyber Security Model] *A cyber security model M is given by a three-tuple $M = (T, C, P)$, where $T = (V, E)$ is a tree rooted at r having $n \in \mathbb{N}$ non-root vertices, C is a multiset of penetration costs $c_1, \ldots, c_n \in \mathbb{Z}^+$ and P is a multiset of prizes $p_1, \ldots, p_n \in \mathbb{Z}^+$.*

Definition 2.2. [SECURITY SYSTEM] *A security system (T, c, p) with respect to a cyber security model $M = (T, C, P)$ is given by two bijections $c : E(T) \to C$ and $p : V(T) \backslash \{r\} \to P$. A system attack in (T, c, p) is given by a subtree τ of T that contains the root r of T. The cost of a system attack τ with respect to (T, c, p) is given by $cost(c, p, \tau) = \sum_{e \in E(\tau)} c(e)$. The prize of a system attack τ with respect to (T, c, p) is given by $prize(c, p, \tau) = \sum_{u \in V(\tau)} p(u)$. For a given budget $B \in \mathbb{Z}^+$ the maximum prize $prize^*(c, p, B)$ with respect to B is defined by $prize^*(c, p, B) = max\{prize(c, p, \tau)|$ for all $\tau \subseteq T$, where $cost(c, p, \tau) \leq B\}$. A system attack τ whose prize is maximum with respect to a given budget B is called an optimal attack.*

From the defense perspective, given (T, c, p) and a budget B, we want an attacker to obtain the least possible maximum prize [6]. In this article, given (T, c, p) and a budget B, we want to replace an edge cost with ∞ to minimize the maximum total prize that an attacker can obtain. We study this problem from the complexity theory standpoint. The decision and optimization versions of the problem is defined as follows.

Definition 2.3. [INFINITY REPLACEMENT OPERATION] *Let (T, c, p) be a security system and (T, c', p) be a security system after the infinity replacement operation. We say that an infinity replacement operation replaces an edge cost $c(e)$ with ∞ if for all edges $e' \in E(T) \setminus \{e\}$, $c'(e') = c(e')$ and $c'(e) = \infty$.*

Definition 2.4. [INFINITY REPLACEMENT DECISION PROBLEM (IR-DP)]
INSTANCE*: A security system (T, c, p) , a budget $B \in \mathbb{Z}^+$ and a prize target $K \in \mathbb{Z}^+$.*
QUESTION*: Is there an edge $e' \in E(T)$ such that $prize^*(c', p, B) \leq K$ after the infinity replacement operation has been performed?*

Definition 2.5. [INFINITY REPLACEMENT OPTIMIZATION PROBLEM (IR-OP)]
INSTANCE*: A security system (T, c, p) and a budget $B \in \mathbb{Z}^+$.*
INSTRUCTION*: Find an edge $e' \in E(T)$ such that $prize^*(c', p, B)$ is minimized after the infinity replacement operation has been performed.*

III. COMPLEXITY

In this section we show that IR-DP is *NP*-hard, which implies the *NP*-hardness of the IR-OP. We next show that IR-OP can be solved in pseudo-polynomial time. Additionally, if we consider some instances with certain restrictions, we show that they can be solved in polynomial time.

A. Complexity of the problems

If a decision problem is *NP*-complete or *NP*-hard, the complement of the decision problem must be *NP*-hard [7]. For this reason, we use the complement of the well-known *NP*-hard PARTITION PROBLEM in our reduction. The definition of the COMPLEMENT OF PARTITION PROBLEM is described below.

Definition 3.1. [COMPLEMENT OF PARTITION PROBLEM]
INSTANCE*: A finite set A and a "size" $s(a) \in \mathbb{Z}^+$ for each $a \in A$.*
QUESTION*: For each nonempty subset $A' \subseteq A$, $\sum_{a \in A'} s(a) \neq \sum_{a \in A \setminus A'} s(a)$?*

Theorem 3.1. *The* IR-DP *is NP-hard.*

Proof: Given a subset $A' \subseteq A$, we let the total size of all elements in A' be $s(A') = \sum_{a \in A'} s(a)$. We make a reduction from the COMPLEMENT OF PARTITION PROBLEM. We construct a security system (T, c, p), a budget $B \in \mathbb{Z}^+$ and a prize target $K \in \mathbb{Z}^+$ such that for each nonempty subset $A' \subseteq A$, $s(A') \neq s(A \setminus A')$ if and only if there is an edge $e' \in E(T)$ such that $prize^*(c', p, B) \leq K$ after the infinity replacement operation has been performed.

Given a problem instance of the COMPLEMENT OF PARTITION PROBLEM, we first construct a corresponding model $M = (T, C, P)$ as follows. We construct a rooted star T by $V(T) = A \cup \{r, x\}$ and $E(T) = \{\{r, v\}|$ for each $v \in V(T) \setminus \{r\}\}$. Let C be $\{s(a)|$ for each $a \in A\} \cup \{1\}$ and P be $\{s(a)|$ for each $a \in A\} \cup \{\frac{s(A)}{2} + 1\}$. We construct the assignments c and p in such a way that for each $a \in V(T) \setminus \{r, x\}$, $c(\{r, a\}) = p(a) = s(a)$, $c(\{r, x\}) = 1$ and $p(x) = \frac{s(A)}{2} + 1$. Finally, we let the budget $B = \frac{s(A)}{2}$ and the prize target $K = \frac{s(A)}{2} - 1$. The whole construction of the problem instance can be computed in $O(n)$, where $n = |A|$.

We will show that for each nonempty subset $A' \subseteq A$, $s(A') \neq s(A \setminus A')$ if and only if there is an edge $e' \in E(T)$ such that $prize^*(c', p, \frac{s(A)}{2}) \leq \frac{s(A)}{2} - 1$ after the infinity replacement operation has been performed.
($\rightarrow$) Suppose for each nonempty subset $A' \subseteq A$, $s(A') \neq s(A \setminus A')$. We show that there is an edge $e' \in E(T)$ such that $prize^*(c', p, \frac{s(A)}{2}) \leq \frac{s(A)}{2} - 1$ after the infinity replacement operation has been performed. After the infinity replacement operation is performed, $c'(\{r, x\}) = \infty$. WLOG, assume that $s(A') < s(A \setminus A')$. Since $B = \frac{s(A)}{2}$ and $s(A') < \frac{s(A)}{2} < s(A \setminus A')$ and $c'(\{r, a\}) = p(a) = s(a)$ for each $a \in V(T) \setminus \{r, x\}$, $prize^*(c', p, \frac{s(A)}{2}) < \frac{s(A)}{2}$ which implies $prize^*(c', p, \frac{s(A)}{2}) \leq \frac{s(A)}{2} - 1$.
($\leftarrow$) Suppose there is an edge $e' \in E(T)$ such that $prize^*(c', p, \frac{s(A)}{2}) \leq \frac{s(A)}{2} - 1$ after the infinity replacement operation has been performed. We show that for each nonempty subset $A' \subseteq A$, $s(A') \neq s(A \setminus A')$. Let $A^* \subseteq V(T)$ be the set that yields $prize^*(c', p, \frac{s(A)}{2}) \leq \frac{s(A)}{2} - 1 < \frac{s(A)}{2}$ and $A^* \cup A^{**} = A$. Because A^* yields the maximum prize $prize^*(c', p, \frac{s(A)}{2})$ and $prize^*(c', p, \frac{s(A)}{2}) < \frac{s(A)}{2}$, $prize(c', p, \tau) > \frac{s(A)}{2}$, where τ is the system attack that contains all prizes in A^{**}. This implies $s(A^*) \neq s(A^{**})$. Observe that, given a budget $\frac{s(A)}{2}$, all system attacks $\tau_1, \tau_2, ..., \tau_q$ yield at most $prize^*(c', p, \frac{s(A)}{2}) < \frac{s(A)}{2}$ and each such τ_i corresponds to a subset A_i of vertices (or prizes). Therefore, $s(A_i) \neq s(A \setminus A_i)$ for all i. Because each nonempty subset A' of A is either A_i or $A \setminus A_i$, we conclude that for each nonempty subset $A' \subseteq A$, $s(A') \neq s(A \setminus A')$. Hence, the theorem holds. ■

Thoerem 3.1 implies that IR-OP is *NP*-hard. We next show that there is a pseudo-polynomial time algorithm for IR-OP.

B. Pseudo-polynomial time algorithm

We present the algorithm FINDMINEDGE for solving IR-OP. Let a security system (T, c, p) and a budget B be inputs to the algorithm FINDMINEDGE. For clarity, we let $T = (V, E)$, where $V(T) = \{r, v_1, v_2, v_3, \ldots, v_n\}$ and $E(T) = \{e_1, e_2, e_3, \ldots, e_n\}$. FINDMINEDGE uses a subroutine called FINDMAXPRIZE$((T, c', p), B)$, which is borrowed from Lemma 5.1 in [3]. FINDMINEDGE is defined as follows.

Algorithm 1 FINDMINEDGE

Input: (T, c, p) and B.
Output: e_{min}.

```
1:  e_min ← ∅
2:  MinPrize ← ∞
3:  for each i from 1 to n do
4:      REPLACECOST(e_i,∞)
5:      currentMaxPrize ← FINDMAXPRIZE((T, c', p), B)
6:      if (currentMaxPrize < MinPrize) then
7:          e_min ← e_i
8:          MinPrize ← currentMaxPrize
9:      end if
10:     REPLACECOST(e_i,c(e_i))
11: end for
12: return e_min
```

A few remarks are in order. First, we assume that $V(T) \neq \emptyset$ and $E(T) \neq \emptyset$. Second, the subroutine REPLACECOST(e,x) simply replaces the cost of edge $e \in E(T)$ with x and leaves everything else unchanged. Given the subroutine FINDMAXPRIZE$((T, c', p), B)$, the algorithm works straightforwardly by replacing the cost of each edge in T with the ∞ and invoking the subroutine FINDMAXPRIZE$((T, c', p), B)$ to find the ∞ replacement that yields the least maximum total prize.

Theorem 3.2. FINDMINEDGE *returns an edge* $e_{min} \in E(T)$ *such that* $prize^*(c', p, B)$ *is minimized after the infinity replacement operation has been performed.*

Proof: We use the loop invariant method. More precisely, we show that at iteration i the variable $MinPrize$ is the minimum $prize^*(c', p, B)$ after the sequence of the i infinity replacement operations that replace the costs of the edges $e_1, e_2, e_3, \ldots, e_i$.

BASE CASE $i = 0$ Before entering the loop, $MinPrize$ is ∞ (line 2). The invariant clearly holds.

INDUCTIVE STEP Assume at iteration i the variable $MinPrize$ is the minimum $prize^*(c', p, B)$ after the sequence of the i infinity replacement operations that replace the costs of the edges $e_1, e_2, e_3, \ldots, e_i$. We show that at iteration $i+1$ the variable $MinPrize$ is the minimum $prize^*(c', p, B)$ after the sequence of the $i+1$ infinity replacement operations that replace the costs of the edges $e_1, e_2, e_3, \ldots, e_i, e_{i+1}$. In line 5 the algorithm has the maximum prize $prize^*(c', p, B)$ from the ∞ replacement operation on edge e_{i+1}. By inductive hypothesis, the variable $MinPrize$ is the minimum $prize^*(c', p, B)$ after the sequence of the i infinity replacement operations that replace the costs of the edges $e_1, e_2, e_3, \ldots, e_i$. Lines 6-9 produce the smaller of the $prize^*(c', p, B)$ at the iterations i and $i+1$. Hence, the invariant holds at the iteration $i+1$.

TERMINATION The loop exits when $i = n+1$. The last execution of the loop is when $i = n$. Hence, at iteration n the variable $MinPrize$ is the minimum $prize^*(c', p, B)$ after the sequence of the n infinity replacement operations that replace the costs of the edges $e_1, e_2, e_3, \ldots, e_n$ and, because line 7 is executed if and only if line 8 is executed, the theorem is true. ∎

We analyze the time complexity of FINDMINEDGE as follows. The running time of the subroutine REPLACECOST(e,x) is $O(1)$ and the running time of the subroutine FINDMAXPRIZE$((T, c', p), B)$ is $O(nB^2)$ [3]. In each iteration the subroutine REPLACECOST(e,x) is invoked twice and the subroutine FINDMAXPRIZE$((T, c', p), B)$ is invoked once and there is a total of n iterations. The total time complexity is then $O(n^2B^2)$.

We remark here that FINDMINEDGE can certainly be sped up. Instead of considering every edge $e \in E(T)$ in line 3 for infinity replacement, we only consider the edges $e = \{r, v\}$ for infinity replacement. However, in the worst case, the time complexity does not change. Furthermore, we note that, like the MINIMUM SPANNING TREE PROBLEM [8], the system attack trees τ of the OPTIMIZATION PROBLEM may not be unique. However, the maximum total prize $prize^*(c', p, B)$ is always unique. For example, let $T = (\{v_1, v_2, v_3, r\}, \{\{r, v_1\}, \{r, v_2\}, \{r, v_3\}\})$ be a star rooted at r. The prize and cost functions p and c are respectively defined to be $p(v_1) = p(v_2) = p(v_3) = 2$ and $c(\{r, v_1\}) = c(\{r, v_2\}) = 1$ and $c(\{r, v_3\}) = 2$. Given the security system (T, c, p) and the budget $B = 2$, the infinity replacement can be done at either $\{r, v_1\}$ or $\{r, v_2\}$ that yields the same minimum $prize^*(c', p, 2) = 2$.

C. Polynomial time solutions

In Subsection B we show that IR-OP in general can be solved in pseudo-polynomial time. In this section we show that some instances of IR-OP can be solved in polynomial time if we put certain restrictions on the problem instances.

Our first observation is that if the cost function c is a constant function, the subroutine FINDMAXPRIZE$((T, c', p), B)$ takes $O(n^3)$ time [3], [9]. Hence, the time complexity of FINDMINEDGE is $O(n^4)$.

Secondly, if we restrict T to be a star graph with a constant prize function p, the time complexity of the subroutine FINDMAXPRIZE$((T, c', p), B)$ becomes polynomial in n. More precisely, let $T = (V, E)$, where $V(T) = \{r, v_1, v_2, \ldots, v_n\}$ and $E(T) = \{\{r, v_1\}, \{r, v_2\}, \ldots, \{r, v_n\}\}$ such that $c(\{r, v_1\}) \leq c(\{r, v_2\}) \leq \ldots \leq c(\{r, v_n\})$. We present a greedy-based subroutine FINDMAXPRIZE$((T, c', p), B)$ as follows.

Algorithm 2 GREEDY-BASED FINDMAXPRIZE

Input: (T, c, p) and B.
Output: $maxPrize$.

```
i ← 1, maxPrize ← 0, cost ← 0
while B − cost ≥ c({r, v_i}) and i ≤ n do
    maxPrize ← maxPrize + p(v_i)
    cost ← cost + c({r, v_i})
    i ← i + 1
end while
return maxPrize
```

We will show that the subroutine GREEDY-BASED FINDMAXPRIZE is correct.

Since p is a constant function, the subroutine FINDMAXPRIZE$((T, c', p), B)$ reduces to the problem of maximizing the number of vertices, given a budget B. More precisely, given a security system (T, c, p) and a budget B, we want to find the maximum set of vertices $V^* \subseteq V(T) \setminus \{r\}$ such that $cost(V^*) = \sum_{\forall v_i \in V^*} c(\{r, v_i\}) \leq B$ and the maximum prize is $|V^*|z$, where z is the constant prize. Our greedy choice in this case is the minimum edge cost of all the remaining edges. We prove the correctness of our algorithm by showing the greedy choice and optimal substructure properties.

Lemma 3.1. *The optimal solution $V_{opt} = \{v_i\}$ such that $\sum_{\forall v_{1 \leq i \leq m} \in V_{opt}} p(v_i)$ is maximum has the greedy choice property.*

Proof: Suppose V_{opt} is in ascending order of edge costs. Let $e_g = \{r, v_g\}$ be the first edge selected from $E(T)$ in line 4 of the subroutine GREEDY-BASED FINDMAXPRIZE. If $v_g = v_1$ then we have an optimal solution beginning with a greedy choice. If $v_g \neq v_1$ then we will show that there is an optimal solution V' starting with greedy choice v_g. Let $V' = V_{opt} \setminus \{v_1\} \cup \{v_g\}$. Because $c(\{r, v_g\})$ is minimum of all edges and the same budget B is used, $|V'| \geq |V_{opt}|$ and the fact that $|V_{opt}| \geq |V'|$ implies V' is also optimal. Hence, $\sum_{v \in V'} p(v)$ is maximum. The lemma holds. ∎

Lemma 3.2. *The optimal solution $V_{opt} = \{v_i\}$ such that $\sum_{\forall v_{1 \leq i \leq m} \in V_{opt}} p(v_i)$ is maximum has the optimal substructure property.*

Proof: Observe that $\sum_{\forall v_{1 \leq i \leq m} \in V_{opt}} p(v_i) = |V_{opt}|z$, where z is a constant prize. We consider only the number of vertices in our proof. Given a budget B, suppose V_{opt} = $\langle v_1, v_2, v_3, \ldots, v_m \rangle$ is an optimal solution to the problem of size $|V(T) \setminus \{r\}|$ and is in ascending order of edge costs. We will show that $V' = V_{opt} \setminus \{v_1\}$ is an optimal solution to the problem of size $|V(T) \setminus \{r, v_1\}|$, given the budget $B' = B - c(\{r, v_1\})$. Suppose V' is not an optimal solution to the problem of size $|V(T) \setminus \{r, v_1\}|$. Therefore, there must exist another solution V'' such that $|V''| > |V'|$ under the same budget B'. Because $B = B' + c(\{r, v_1\})$, $V'' \cup \{v_1\}$ is a solution to the problem size $|V(T) \setminus \{r\}|$ but $|V'' \cup \{v_1\}| > |V_{opt}|$. This is a contradiction. Hence, the lemma is true. ∎

Theorem 3.3. *The subroutine* GREEDY-BASED FINDMAXPRIZE *is correct.*

Proof: The proof is by induction on the number of greedy choices and the application of the optimal substructure property. ∎

The time complexity of the subroutine GREEDY-BASED FINDMAXPRIZE is $O(n)$, where $|E(T)| = n$. But if we include the running time of sorting the edge costs in E, the time complexity becomes $O(n \log n)$. If we use the subroutine GREEDY-BASED FINDMAXPRIZE in FINDMINEDGE, the time complexity of FINDMINEDGE becomes $O(n^2 \log n)$.

IV. CONCLUSION

In this article we presented a new defensive strategy problem based on the cyber security model. We defined the IR-DP and IR-OP and examined the complexities of problems. We showed that the IR-DP and IR-OP are *NP*-hard. We also showed that the IR-OP can be solved in pseudo-polynomial time $O(n^2B^2)$. We discussed the time complexities for the case where the cost function is constant and the case where the prize function is constant and the graph T is a star. The time complexities for the two cases are $O(n^4)$ and $O(n^2 \log n)$, respectively.

In real-life applications security measures could be firewalls or encryption, which have their own costs of breaking and the valuable prizes could be databases or passwords. In our context we want to find a place in a computer network (i.e., tree graph) to install the strongest security measure (i.e., infinity) so that the attacker can only accumulate the minimum total value of prizes.

REFERENCES

[1] P. Passeri. (2017, jan) 2016 cost of cyber crime study: United states. [Online]. Available: http://www.hackmageddon.com/2017/01/19/2016-cyber-attacks-statistics/

[2] P. Institute. (2016, may) 2016 cost of cyber crime study: United states. [Online]. Available: http://www.ponemon.org/library/2016-cost-of-cyber-crime-united-states

[3] G. Agnarsson, R. Greenlaw, and S. Kantabutra, "On cyber attacks and the maximum-weight rooted-subtree problem." *Acta Cybern.*, vol. 22, no. 3, pp. 591–612, 2016.

[4] F. B. Schneider, "Blueprint for a science of cybersecurity," Tech. Rep., 2011.

[5] R. Armstrong, J. Mayo, and F. Siebenlist, "Complexity science challenges in cybersecurity," *Sandia National Laboratories SAND Report*, 2009.

[6] G. Agnarsson, R. Greenlaw, and S. Kantabutra, "The structure of rooted weighted trees modeling layered cyber-security systems." *Acta Cybern.*, vol. 22, no. 4, pp. 735–769, 2016.

[7] R. G. Michael and S. J. David, *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms second edition*. The MIT Press, 2001.

[9] S. Coene, C. Filippi, F. C. Spieksma, and E. Stevanato, "Balancing profits and costs on trees," *Networks*, vol. 61, no. 3, pp. 200–211, 2013.