Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-18
https://doi.org/10.22049/cco.2025.30341.2426

CCO
Commun. Comb. Optim.

Research Article

A novel procedure for identification of chief master regulatory genes in weighted gene regulatory networks

Somayeh Bakhteh[1], Alireza Ghaffari-Hadigheh[1,*], Nader Chaparzadeh[2]

[1]Department of Applied Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran
bakhtehs@yahoo.com
*Hadigheha@azaruniv.ac.ir, hadigheha@gmail.com

[2]Department of Biology-Physiology, Azarbaijan Shahid Madani University, Tabriz, Iran
nchapar@azarunive.ac.ir

Received: 7 February 2025; Accepted: 30 June 2025
Published Online: 9 July 2025

Abstract: Identifying master regulatory genes is crucial for analyzing gene regulatory networks. Various optimization-based approaches have been developed to identify potential sets of master regulatory genes. In a weighted gene regulatory network, each interaction between gene pairs is assigned a weight. In such networks, not only direct interactions between genes significant, but indirect influences also play an important role. In this study, an indirect relationship between two genes is considered to exist when, in addition to a potential direct link, there is at least one additional pathway through which they influence each other. An influence value between two genes is calculated using an algorithm inspired by the K-shortest path approach. Furthermore, each gene is assigned an impact factor based on its overall influence within the weighted network. These tools allow us to introduce a new method based on a modified version of the well-known uncapacitated facility location problem. This method can identify the most significant genes among those detected by other approaches and also determine a master regulatory gene that controls a specific target gene. The proposed approach has been applied to several gene regulatory networks, and the results are reported and compared against two existing models.

Keywords: master gene, gene regulatory network, uncapacitated facility location problem, highest effect pathway.

AMS Subject classification: 90C27, 92C42, 92D20, 68Q25, 68W40.

* *Corresponding Author*

1. Introduction

One of the primary objectives in systems biology is the analysis of gene regulatory networks (GRNs). These networks consist of collections of genes and their interactions [17]. Insights gained from such analyses have broad applications across various fields. These fields include medicine, the identification of genetic disorders, and the development of advanced treatments for complex diseases such as cancer [2, 12, 18]. GRNs contain specific sets of genes known as master regulatory genes and key driver genes. Key driver genes regulate the state of other genes [16, 24] and control their expression [32]. Master regulatory genes, on the other hand, occupy the highest level of the regulatory hierarchy and are not influenced by any other genes.

Over the past decade, numerous mathematical optimization models have been introduced to study GRNs. For instance, one model was developed to identify a smaller subset of influential regulatory genes within a GRN [25]. These candidate regulatory genes can either activate or inhibit the expression of other genes. The authors of this model also designed and implemented a meta-heuristic algorithm to address large-scale GRNs. Another integer optimization approach was specifically created to construct a Weighted Gene Regulatory Networks (WGRN), where a reduced set of candidate regulatory elements is identified as activators or inhibitors. This network is generated by assigning weights to its edges based on an activation-inhibition index [9].

To identify key and master regulatory genes in GRNs, several optimization problems have been formulated and solved using the concept of the Minimum Dominating Set (MDS) [27, 29, 31, 36]. In biological networks, an MDS represents an optimized subset of genes (or proteins) where each gene is either part of this subset or directly connected to at least one of its members [36]. For undirected networks, an equivalent optimization problem to the MDS has also been proposed [30].

To address the heterogeneity in degree and betweenness centralities of proteins, a Centrality-Corrected Minimum Dominating Set (CC-MDS) model was introduced to identify the minimum set of driver nodes in protein-protein interaction networks [39]. The heterogeneity refers to the diversity in the features of centrality within the protein-protein interaction network, specifically degree centrality and betweenness centrality. The heterogeneity in these centrality measures helps the CC-MDS model to identify the minimal set of critical nodes (or key proteins) needed to control and guide the functioning of the network effectively.

Additionally, a Collective-Influence Minimum Dominating Set (CI-MDS) model was developed by extending the standard MDS framework to account for the heterogeneity in the collective influence of proteins in protein-protein networks [38]. Furthermore, a weighted GRN problem with dual objectives, formulated as a variant of the MDS problem, was proposed to determine an MDS in GRNs [1]. To enhance the interpretability of results and reduce computational time, the authors employed linear parametric programming and logistic regression techniques.

In an MDS model, only the direct effects of a gene on neighboring genes are considered; indirect effects are ignored. Additionally, these models cannot identify the specific master regulatory genes that control a particular gene. To address these limitations, this paper introduces a new model that accounts for both direct and indirect effects of each gene on others. Furthermore, the proposed model can determine which specific gene is controlled by which identified master regulatory genes.

In the WGRNs studied in this research, the interaction between a pair of genes is assigned a non-negative weight ranging from zero to one. A zero weight indicates no interaction, while positive weights reflect the strength or reliability of the interaction. A minimum set of master regulatory genes is identified within these WGRNs. To determine these weights, we introduce the concept of K-highest effect pathways, inspired by the K-shortest path algorithm. This approach allows for the consideration of both direct and indirect effects of each gene on others.

Additionally, we propose an integer binary programming model based on a modified version of the Uncapacitated Facility Location Problem (UFLP) to identify a minimum set of master regulatory genes in these WGRNs. In this model, the collective influence of a gene is considered its impact factor on other genes.

The core innovation of this paper lies in the introduction of the Adapted Uncapacitated Facility Location Problem (AUFLP) model, which is designed to analyze Weighted Gene Regulatory Networks (WGRNs) by considering both direct and indirect effects. This involves novel algorithms (HEP and K-HEP), unique methods to compute the aggregate influence of genes through pathways, and a modified UFLP to identify master genes. The algorithms and constraints are specifically tailored for WGRNs, and the paper demonstrates that the AUFLP can help identify master regulatory genes that control other genes and pathways. By considering these collective influences, this model goes beyond traditional models that focus on direct relationships.

The paper's innovation is a combination of novel elements: the AUFLP model, its focus on indirect influences, the HEP and K-HEP algorithms, and the comprehensive understanding it enables of gene networks. List of notations and symbols are appeared in Table 1.

The paper is organized as follows: Section 2 introduces basic concepts of graph theory and provides a brief overview of the single-source shortest path and the single-source K-shortest paths algorithms. Section 3 presents the UFLP and discusses an optimization problem for solving it. Section 4 proposes the highest effect pathway and the K-highest effect pathway algorithms. Additionally, a variant of the UFLP is introduced, and an integer binary programming formulation is proposed to identify chief master regulatory genes in the considered WGRN. Computational results are reported and interpreted in Section 5, along with examples to illustrate the findings. The final section provides concluding remarks.

2. Some Necessary Concepts from Graph Theory

This section briefly reviews some essential concepts and algorithms from graph theory needed for this study.

An undirected graph $G = (V, E)$ consists of a finite set of vertices V and edges

Table 1. **List of Notations and Symbols**

Symbol	**Definition**
GRN	Gene regulatory network.
$WGRN$	Weighted gene regulatory network.
MDS	Minimum dominating set
CC-MDS	Centrality-corrected minimum dominating set
CI-MDS	Collective-influence minimum dominating set
UFLP	Uncapacitated facility location problem
HEP	the *The highest effect pathway*
$G(V, E)$	A graph consisting of vertices V and edges E.
w_{ij}	The weight of the edge between vertices i and j.
d_i	The degree of vertex i in an unweighted graph.
$C_D^w(i)$	The degree centrality of vertex i for weighted graphs.
α	A parameter balancing degree and weight importance in centrality measures.
$\partial Ball(i, \ell)$	The set of nodes at distance ℓ from node i.
$d(i, j)$	the distance between nodes i and j.
b_j	Betweenness centrality of vertex j.
σ_{ik}	Total number of shortest paths from vertex i to vertex k.
$N = (V, E, w)$	A weighted graph representing a gene regulatory network, where w is the weight function.

$E \subset V \times V$ [7]. In this paper, the set of vertices V is denoted as $\{1, 2, 3, \ldots, n\}$. An edge $e = \{i, j\} \in E$ in an undirected graph connects two vertices i and j ($\{i, j\}$=$\{j, i\}$). If all edges of the graph $G = (V, E)$ are ordered pairs with entries in V, the graph is referred to as directed.

In a graph, weights can be assigned to vertices or edges. In our model, we focus on weights assigned to edges. Thus, a weight function $w : E \to \mathbb{R}$ assigns a weight $w(e)$ to each edge e. A graph G along with a weight function w is called a network and is denoted by $N = (G, w)$. This study assumes that for all $e \in E$, $w(e) \geq 0$.

A Dominating Set in a graph $G = (V, E)$ is a set $S \subset V$ such that every node $v \in V$ is either an element of S or adjacent to an element of S [29]. An MDS $S \subset V$ has the smallest cardinality among all dominating sets. The domination number of a graph G is the number of nodes in an MDS.

A path in a graph G is an alternating sequence of distinct vertices $P = \{i_0, i_1, \ldots, i_{n-1}, i_n\}$ such that there exists an edge between two vertices i_k and i_{k+1}, $k = 0, 1, \ldots, n-1$. A graph G is called connected if each pair of vertices are joined by a path. The weight of a path P is defined as $w(P) = \sum_{e \in P} w(e)$. Typically, there might exist several paths between two vertices v and u on a given network $N = (G, w)$. In weighted graphs, the path with minimum weight between nodes $i, j \in V$ is called the shortest path between these nodes. The weight of the shortest path between nodes i and j is defined as the distance between these nodes and is denoted by $d(i, j)$. If there is no path between nodes $i, j \in V$, then $d(i, j) = \infty$.

The single-source shortest path problem finds the shortest paths from a given source vertex v to all other vertices. Several algorithms exist for solving this problem, such as Dijkstra's algorithm [10] and the Bellman-Ford algorithm [6]. Dijkstra's algorithm

solves this problem on networks with non-negative weights, whereas the Bellman-Ford algorithm handles both positive and negative weights. Furthermore, the Bellman-Ford algorithm is more general and simpler than Dijkstra's algorithm, making it suitable for distributed systems. However, Dijkstra's algorithm has a time complexity of $O(|V| \log |V|)$ (with the use of Fibonacci heap), compared to the Bellman-Ford algorithm's time complexity of $O(|V|\ |E|)$ [26]. Given the underlying networks with nonnegative weights and the time complexity of both algorithms, Dijkstra's algorithm with some modifications is used for finding the shortest path between two nodes in this study.
The degree of a vertex $i \in V$ in an unweighted graph is defined as the number of edges incident to i, denoted as d_i.

2.1. Strength, Centrality Measure, and Collective Influence

In a weighted graph, the degree of a vertex extends to the sum of the weights of the edges connected to that vertex [3], known as the node strength. This measure for a node i is formalized as follows:

$$C_D^w(i) = \text{strength}(i) = \sum_{(i,j)\in E} w_{ij}, \qquad i \in V,$$

where $w_{ij} \geq 0$ represents the weight of the edge (i, j). Thus, $w_{ij} > 0$ indicates that node i is connected to node j. In [33], an adjusted degree centrality measure is proposed that combines both degree and strength using a positive parameter α to determine the relative importance of the number of degrees compared to the degree weights:

$$C_D^{w\alpha}(i) = d_i^{1-\alpha} \times C_D^w(i)^\alpha, i \in V. \tag{2.1}$$

The parameter α can be adjusted according to the research setting. The authors in [33] further elaborated on different levels of α. If $\alpha = 1$, then $C_D^{w\alpha}(i) = C_D^w(i)$, and if $\alpha = 0$, then $C_D^{w\alpha}(i) = d_i$. Moreover, if $\alpha \in (0, 1)$, then $C_D^{w\alpha}(i)$ increases with both d_i and $C_D^w(i)$, whereas if $\alpha > 1$, then $C_D^{w\alpha}(i)$ increases by decreasing d_i and increasing $C_D^w(i)$.
The Collective Influence (CI) of each node is the product of its reduced degree and the sum of the reduced degrees of all nodes at a distance ℓ from it [28]. In mathematical terms, CI of a node $i \in V$ is defined as:

$$CI_\ell(i) = (d_i - 1) \sum_{j\in\partial\text{Ball}(i,\ell)} (d_j - 1),$$

where $\partial\text{Ball}(i, \ell)$ denotes the set of nodes at distance ℓ from node i. Nodes with higher collective influence play significant roles in the network [20]. Note that CI has a free parameter ℓ that needs to be adjusted. At $\ell = 0$, $CI_0(i) = (d_i - 1)^2$ represents the square of its reduced degree. In [28], the authors chose a non-zero ℓ but not too large (e.g., $\ell = 1, 2, 3$), as the network boundaries are reached and CI of all nodes approaches zero for larger values of ℓ.

Since this study considers networks with non-negative weighted edges, we define the collective influence of each node as:

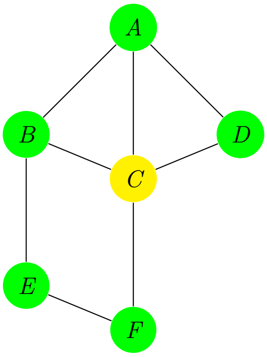
$$CI_{\ell}^{w\alpha}(i) = (C_D^{w\alpha}(i) - 1) \sum_{u \in \partial \mathrm{Ball}(i,\ell)} (C_D^{w\alpha}(j) - 1). \tag{2.2}$$

Recall that a vertex may lie on the shortest paths between some nodes. Therefore, the betweenness centrality b_j of a node j is defined as the number of shortest paths between two other nodes that pass through the vertex j [8], and is determined as:

$$b_j = \sum_{i \neq j \neq k \in V} \frac{\sigma_{ik}(j)}{\sigma_{ik}}, \tag{2.3}$$

where σ_{ik} is the number of shortest paths between nodes i and k, and $\sigma_{ik}(j)$ is the number of those paths that pass through node j.

Example 1. Consider the following simple undirected graph. In this graph, we calculate the betweenness centrality of node C.



Step 1: **Identifying all shortest paths:** We need to consider all pairs of nodes (excluding C) and identify the shortest paths between them, which is summarized in Tabel 2.

Step 2: **Calculating betweenness centrality for node C.** Let us apply Equation (2.3) to node C:

$$\begin{aligned} b(v) &= \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \\ &= \frac{0}{1} + \frac{1}{1} + \frac{0}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{0}{1} + \frac{0}{1} + \frac{1}{1} + \frac{0}{1} \\ &= 0 + 1 + 0 + 1 + 1 + 1 + 0 + 0 + 1 + 0 = 5. \end{aligned}$$

The betweenness centrality of node C is 5. This high value indicates that node C is a critical node in this network.

Table 2. Shortest paths and betweenness centrality calculations for node C

Node pair (s, t)	Total shortest paths between nodes s and t	Shortest paths through node C.
(A, B)	A–B	0
(A, D)	A–C–D	1
(A, E)	A–B–E	0
(A, F)	A–C–F	1
(B, D)	B–C–D	1
(B, F)	B–C–F	1
(B, E)	B–E	0
(D, E)	D–C–F–E	0
(D, F)	D–C–F	1
(E, F)	E–F	0

2.2. Single-Source K-Shortest Paths

For finding the first K shortest paths in a network with non-negative weighted edges, the original Yen's algorithm was proposed [37], and it serves as the basis for our algorithm for identifying K highest effect pathways between genes. This problem has applications in various fields, including probabilistic networks [13], sequence alignment and metabolic pathway finding in bioinformatics [34], road networks [23], and multiple object tracking [4]. We will modify this algorithm to address our specific problem.
Yen's algorithm computes these paths in two phases. In the first phase, the 1st shortest path (P^1) from a source node to a destination node is determined using Dijkstra's algorithm. In the second phase, the k-th shortest path (P^k) for $k = 2, \ldots, K$ is identified through a two-step process. This algorithm determines the K-shortest paths in $\mathcal{O}(Kn^3)$ time, where the term $\mathcal{O}(n^2)$ is attributed to the shortest path calculation by Dijkstra's algorithm, and n is the number of vertices [37].

3. Uncapacitated Facility Location Problem

The Uncapacitated Facility Location Problem (UFLP) is one of the optimization problems consisting of some potential facilities and existing customers. In this problem, overall costs of transportation to each customer and the cost associated to facility opening is minimized [11]. This problem has wide applications such as distribution system design [19], self-configuration in wireless sensor networks [14], computer vision [22], and pace segmentation [21].

3.1. The UFLP Mathematical Model

Let F and C be finite sets of facilities and customers, respectively. For opening each facility $i \in F$ a non-negative cost f_i is associated. Moreover, let a non-negative transportation cost from each facility $i \in F$ to each customer $j \in C$, c_{ij} be assigned to the edge (i, j). The goal of UFLP is to determine a subset X of facilities to be opened

and the assignment of each customer to only one appropriate facility such that the sum of the opening costs of facilities and the transportation costs is minimized. For opening the facility $i \in F$, a binary variable y_i is defined. If the ith facility is opened, then $y_i = 1$ and $y_i = 0$, otherwise. Moreover, for each customer $j \in C$, a binary variable x_{ij} is defined such that $x_{ij} = 1$ when the demand of client j is satisfied from facility i and $x_{ij} = 0$, otherwise.
The UFLP can be mathematically formulated as the following binary integer linear program:

$$
\begin{aligned}
\text{UFLP}: \qquad \min \quad & \sum_{i \in F} \sum_{j \in C} c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1, \qquad \forall j \in C, && (3.1) \\
& x_{ij} \leq y_i \qquad \forall i \in F\ , j \in C, && (3.2) \\
& x_{ij},\ y_i \in \{0,1\}, \qquad i \in F\ , j \in C. && (3.3)
\end{aligned}
$$

Constraint (3.1) ensures that the demand of each client is satisfied just from one facility, and constraint (3.2) guarantees that clients are supplied only from open facilities. When at least one client is assigned to a facility, then that facility must be opened (i.e. if there exist at least one j with $x_{ij} = 1$, then $y_i = 1$). When $y_i = 1$ and $x_{ij} = 1$, then facility i satisfies the demand of client j. Constraint (3.3) enforces x_{ij} and y_i being binary variables. We recall that the UFLP is one of the well known NP-hard problems [15].

4. The Weighted Gene Regulatory Network

Recall that a WGRN consists of a set of genes and their interactions that are considered as non-negative weights. In mathematical terms, consider a WGRN as a network $N = (V, E, w)$, where the function $w : E \to \mathbb{R}^+$ associates to each edge a non-negative weight. For example here, the weight of an interaction between two genes in a WGRN could be the Pearson correlation coefficient of pair genes expression levels across several experiments that are a value between 0 and 1 [35].
Here, a new optimization model based on the AUFLP is presented that not only determines a minimum set of master genes in a WGRN but also identifies the master genes that control a specific gene. The associated binary integer program is formulated in the sequel.
The goal of AUFLP model is to determine potential master genes that have the highest effect together with the highest impact factor on other genes. Here, the impact factor of each gene i is computed as its collective influence in a weighted graph defined as (2.2). In this model, the direct and indirect effective of each gene on other genes is considered. To compute these effects, motivating from the Dijkstra algorithm, we propose the **Highest Effect Pathway** (HEP) algorithm (see Algorithm 1). The out put of this algorithm, for each vertex $i \in V$ is a label effect(i) as well as a

label previous(i). The label effect(i) represents the effect of a given source s on the sink node i and the label previous(i) shows previous node of vertex i in the optimal pathway from the source.

Algorithm 1 Highest effect pathway algorithm

```
Input: Network $N = (G, w)$ with non-negative weight on edges, and source vertex $s \in V$.
Output: Paths from source vertex $s \in V$ to all other vertices with highest effect.

BEGIN
for each $i \in V - s$ do
    effect($i$) = 0;
    previous($i$) =undefined;
end for
effect($s$) = 1;
$Q = V$;
while $Q \neq \emptyset$ do
    $j$= vertex in $Q$ with maximum effect.
    $Q = Q \setminus u$;
    for all $(j, i) \in E$ do
    if effect($i$) < effect($j$) × $w(j, i)$
      effect($i$) = effect($u$) × $w(j, i)$;
      previous($i$) = $j$;
      end if
    end for
end while
END
```

The HEP algorithm finds the most effective path from given source node to the destination vertex. Let us highlight the differences between the Dijkstra's algorithm and HEP algorithm. Recall that in the Dijkstra's algorithm, dist(s) = 0 and dist(i) = ∞ for $i \in V - s$, while in Algorithm 1, first effect(s) = 1 and effect(i) = 0 for $i \in V - s$. Since the goal is to find the genes with higher effect on other genes, instead of the comparison condition $dist(i) > dist(j) + w(j, i)$ in the Dijkstra's algorithm, the condition effect(i) < effect(j) × $w(j, i)$ is replaced. If a gene is enough far from the given gene, then its effect must be reduced, therefore we set effect(i) = effect(j) × $w(j, i)$ instead of $dist(i) = dist(j) + w(j, i)$ in the Dijkstra's algorithm. By running the Algorithm 1, calculated values would be considered as the indirect effect of each gene on others. Observe that when the vertex i is adjacent to the given vertex s, its effect is just the weight of the edge (s, i). We state that the Algorithm 1 is terminated after finite iterations, because only the calculation and comparison operations differs from the Dijkstra's algorithm. As a result, time complexity of Algorithm 1 is identical with the time complexity of the the Dijkstra's algorithm (i.e. $\mathcal{O}(n^2)$ that n is the number of nodes).

4.1. K-Highest Effect Pathway Algorithm

Motivating from the KSP algorithm, let us introduce the K-HEP Algorithm to obtain the first K highest effect pathways between given two nodes. (See Algorithm 2). In this algorithm, the HEP algorithm is called to find a path with the highest effect between two nodes. Let us introduce used notations in K-HEP Algorithm. Let $P^k = (1) - (2^k) - (3^k) - \ldots - (Q_k^k) - (t),\ k = 1, \ldots, K$, be the kth highest effect pathway from source node 1 to the destination node t, where (2^k) is the second node of P^k, (3^k) is the third node of P^k, and so on. The path $P_i^k,\ i = 1, \ldots, Q_k$, is a deviation path from P^{k-1} at node i^k and vertex i^k is referred to as the deviation node of that path. If a node i^k is being analyzed, then the subpath $R_i^k = (1) - (2^k) - \ldots - i^k$ is said to be the root path and the subpath $S_i^k = (i^k) - \ldots - (t)$ is said the spur path of P_i^k. The root and the spur paths are joined to form a complete path from the source node 1 to the destination node t. The already-known HEPs are stored in a list $\mathcal{A}$ and candidate paths for the next HEP is stored in the list $\mathcal{B}$.

Algorithm 2 K Highest effect pathway algorithm.

```
Input: Network $N = (G, w)$ with non-negative weight on edges, a source vertex 1, a sink vertex $t$, and $K$.
Output: Determine $K$ paths from source vertex 1 to sink vertex $t$ with highest effect.
BEGIN
Determine the 1st highest effect path ($P^1$) from a source node 1 to a destination node $t$ using HEP algorithm.
$\mathcal{A} = [P^1]$;
$\mathcal{B} = \emptyset$;
for $k = 2, \ldots, K$ do
    for $i = 1 : Q_k^k$ do
     if there exists a subpath $(1) - (2)^{k-1} - \ldots - i$ in list $\mathcal{A}$then
      Choose that subpath as $R_i^k$ and remove the edge $(i, i+1)$.
     end if
    Compute a the highest effect path from $i$ to $t$ applying HEP algorithm
     if a path from $i$ to $t$ is found  do
      Choose that path as $S_i^k$
      Set $P_i^k = R_i^k + S_i^k$ as a candidate path for the next effect path.
     end if
     Add $P_i^k$ to list $\mathcal{B}$.
     Restore removed edges.
    end for
     Choose a path from list $\mathcal{B}$ with maximum effect as $P^k$ and remove it from $\mathcal{B}$ and add it to $\mathcal{A}$.
end for
END
```

Analogous to the KSP algorithm, the K-HEP algorithm terminates after finite iterations because only its difference with KSP algorithm is calling Highest Effective Pathway algorithm. Since time complexity of Highest Effective Pathway is identical with the complexity of the Dijkstra's algorithm, so, the time complexity of the K-HEP algorithm is $\mathcal{O}(Kn^3)$.

After finding all k paths, $k = 1, \ldots, K$, between the source node i and the sink node j, we compute the aggregate effect of the gene i on gene j (AE_{ij}) using the *Aggregate Effect* algorithm. The aggregate effect of gene i is considered as the effect power of the gene i on the gene j.

Algorithm 3 Aggregate Effect Algorithm.

```
Input: Network $N = (G, w)$ with non-negative weights on edges and a value for $K$.
Output: The aggregate effect of each gene.
Begin
Set $i = 1$;
while $i < n$ do
    for $j = i + 1$ to $n$ do
     $i$= source vertex;
      $j$=destination vertex;
      Call $K$-HEP algorithm and search all $K$ highest effect paths from node $i$ to node $j$;
      Compute the weight of paths $P_k,\ k = 1, \dots, K$, by $w(P_k) = \prod_{(i,j)\in E(P_k)} w(i,j)$;
     end for
     Compute aggregate effect of gene $i$ on gene $j$ by $AE_{ij} = \sum_k w(P_k)$;
     Set $i = i + 1$;
end while
End
```

In this study, the networks are considered as undirected graphs and therefore, for each gene $i,\ j \in V$, $AE_{ii} = 0$ and $AE_{ij} = AE_{ji}$. It is not hard to check that all AE_{ij}, $\forall i, j \in V$ are computed by the Aggregate Effect algorithm in $\mathcal{O}(K^{\frac{n(n-1)}{2}} n^3)$.

4.2. The AUFLP Mathematical Model

Recall that the UFLP minimizes the overall transportation cost from facilities to customers, along with the opening costs of facilities. In contrast, the goal of the AUFLP is to identify a minimum set of master genes with the highest effect and impact factor on other genes. Consequently, we set for the gene i, $c_{ij} = \{\max_{\{i,j\}} AE_{ij}\} - AE_{ij}$ and $f_i = \{\max_i CI_\ell^{w\alpha}(i)\} - CI_\ell^{w\alpha}(i)$. Additionally, note that in the UFLP model, the constraint $\sum_{i\in V} x_{ij} = 1$ ensures that each customer's demand is satisfied by only one facility. However, in the AUFLP model, each gene may be controlled by more than one master gene. Therefore, we use the constraint $\sum_{i\in V} x_{ij} \geq 1$ instead. For formulation of the AUFLP, consider the following binary variables:

$$y_i = \begin{cases} 1, & \text{if gene } i \text{ is master gene,} \\ 0, & \text{otherwise,} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if gene } i \text{ controlled gene } \ j, \\ 0, & \text{otherwise.} \end{cases}$$

The AUFLP model is therefore formulated as the binary integer liner programming problem (4.1)-(4.4).

$$\text{AUFLP}: \quad \min \quad \sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij} + \sum_{i\in V} f_i y_i \tag{4.1}$$

$$\text{s.t.} \quad \sum_{i\in V} x_{ij} \geq 1 \qquad \forall j \in V \tag{4.2}$$

$$x_{ij} \leq y_i \qquad \forall i \in V\ , j \in V, \tag{4.3}$$

$$x_{ij},\ y_i \in \{0, 1\}, \qquad i \in V, (i,j) \in E. \tag{4.4}$$

The objective is to minimize the number of genes with high impact factors and significant effects on others, known as master genes in the literature. Constraint (4.2) ensures that all genes are controlled, with each gene being regulated by at least one determined master gene. Constraint (4.3) guarantees that a gene j can be controlled by a gene i only if gene i is identified as a master gene. In an optimal solution, this constraint also implies that when $y_i = 1$ and $x_{ij} = 1$, gene j is controlled by the master gene i. Condition (4.4) specifies that all variables are binary.
Recall that an integer linear program is NP-hard [5]. However, we are not concerned about computational difficulties and solve this problem using the solver CPLEX [1].

5. Computational Experience

In this section, we report the computational results of applying the AUFLP model to several networks. Five weighted gene networks were selected from the *GeneMANIA* database [35] (see Table 3). In this database, the Pearson correlation coefficient of the expression levels of two genes is assigned as the weight for each interaction, ranging from zero to one. This table includes the number of genes and the number of predicted master regulatory genes identified by the AUFLP model. The collective influence of genes was computed using Equation. (2.2) for $\ell = 0$ and $\alpha = 0.5$. Additionally, AE_{ij} values were calculated using the Aggregate Effect algorithm with different values of $K = 5, 10, 20$. It was observed that the aggregate value increases with K. However, the set of identified master genes is not sensitive to higher values of K. Therefore, the results with $K = 10$ are reported in Table 3.

Table 3. **Characteristics of Gene networks.**

Data Set	# of Genes	# of Interactions	# of master Genes	% Master Genes
Koren-Barkai	13	62	3	0.23
Daniel-Burke	128	214	1	0.01
Finger-Novick	19	89	1	0.05
Berg-Poot	159	241	1	0.01
Breslow-Weissman	32	126	1	0.03

5.1. Comparison of AUFLP with CC-MDS and CI-MDS

Let us compare the results of the AUFLP model with the CC-MDS and CI-MDS models on the given networks. It's important to note that the CC-MDS model was developed to identify driver proteins in protein-protein interaction networks [39]. This

[1] *http://WWW-01.ibm.com/software/commerce/optimization/cplex-optimizer/*

model incorporates heterogeneity in the degree and betweenness centralities of proteins on unweighted graphs and is formulated as follows.

$$\begin{aligned} \min_{i\in V} \quad & \sum_{i\in V} w_i x_i && && (5.1)\\ \text{s.t.} \quad & x_i + \sum_{(j,i)\in E} x_j \geq 1, && \forall j \in V, \\ & x_i \in \{0,1\}, && \forall v \in V, \end{aligned}$$

where $w_j = (d_j b_j)^{-\gamma}$ is associated to the centralities of node j, d_j is the degree centrality, b_j is the betweenness centrality of the node j, and $\gamma \geq 0$ controls the weights. Specifically, x_i is defined as a binary variable that indicates the selection status of gene i; such that:

$$x_i = \begin{cases} 1 & \text{if gene } i \text{ is selected as a master regulatory gene,} \\ 0 & \text{otherwise.} \end{cases}$$

Also, the Collective-Influence-corrected minimum domination set model (CI-MDS) is developed to detect MDS proteins as follows:

$$\begin{aligned} \max \quad & CI_\ell(v) x_i && && (5.2)\\ \text{s.t.} \quad & x_i + \sum_{(j,i)\in E} x_j \geq 1, && \forall j \in V, \\ & \sum_{i\in V} x_i = \gamma(G) && x_i \in \{0,1\}, \quad \forall i \in V, \end{aligned}$$

where $\gamma(G)$ is the domination number of graph G. Since we aim to identify master genes in given weighted networks and the CC-MDS and CI-MDS models operate on unweighted networks, we need to compute w_v and $CI_\ell(i)$ on the given weighted networks. For computing the degree in a weighted network, we set $\alpha = 0.5 \in (0,1)$ in Equation. (2.1), and the betweenness centrality is computed using the Matlab package *MatlabBGL*[2]. Additionally, $CI_\ell(i)$ is computed using Equation. (2.2) with $\alpha = 0.5$. We adjusted the CC-MDS and CI-MDS models for the weighted networks, resulting in the CC-MWDS and CI-WMDS models, respectively. Problems (5.1) with different values $\gamma \in \{0, 0.05, 0.1, \dots, 1\}$ and problem (5.2) were solved for the given networks using the CPLEX solver.

Table 4 shows the number and names of identified master genes in the given networks for the AUFLP, CC-MWDS, and CI-WMDS models. This table also reports the

[2] *http://dgleich.github.io/matlab-bgl/*

non-master genes controlled by the identified master genes using the AUFLP. For instance, in the Koren-Barkai network, genes YBL039C, YKL113C, and YOR241W are identified as master genes by the AUFLP model. Each of these genes controls several non-master genes. For example, the master gene YBL039C controls non-master genes YBR278W, YCL061C, YDR121W, YER070W, and YLR176C.

Implementing the AUFLP model on the given networks revealed that it identifies some master regulatory genes not detected by the CC-MWDS and CI-WMDS models. Additionally, some genes identified by the CC-MWDS and CI-WMDS models were not recognized as master genes by the AUFLP model. For example, in the Koren-Barkai network, only the gene YOR241W is identified by both the CC-MWDS and the AUFLP. Similarly, gene YBL039C is identified by both the CI-MWDS and the AUFLP models. However, the gene YKL113C is identified as a new master regulatory gene by the AUFLP. As another example, in the Daniel-Burke network, the master genes YGL086W, YJL013C, and YJL030W are identified only by the CC-MWDS and CI-MWDS models but are not recognized as master genes in the AUFLP model. Only the gene YOR026W is identified by the AUFLP model. The role of this discrepancy in the identified master regulatory genes by different methods would require elaborate laboratory testing. The AUFLP model determines master regulatory genes with the highest impact factor and the highest effect on other genes. Therefore, it may indicate that those genes identified by both models are of utmost importance.

6. Conclusion

In this study, a Weighted Gene Regulatory Network (WGRN) was introduced by assigning a non-negative weight to each interaction. These weights are represented by the Pearson correlation coefficient of the expression levels of two genes. To identify a minimum set of master regulatory genes in this network, an integer linear programming model was proposed based on a modification of the uncapacitated facility location problem. In this model, both the direct and indirect effects between genes were considered and computed using the K-Highest Effect Pathways algorithm. Additionally, the Aggregate Effect Algorithm was developed to calculate the highest effect of a gene on others. Finally, the impact factor of each gene was computed using its collective influence.

The proposed model identifies master regulatory genes that have the highest impact factor as well as the highest effect on other genes. The optimal solution not only determines a set of master genes with greater effect but also identifies which genes are controlled by the determined master genes. The proposed method was applied to several WGRNs. The computational results were compared with the CC-MWDS and CI-MWDS models and demonstrated that our proposed model could effectively determine the chief master genes.

Conflict of Interest: The authors declare that they have no conflict of interest.

Data Availability: Data sharing is not applicable to this article as no data sets

Table 4. **The results of implementation of AUFLP and CC-MWDS.**

Method	Networks	# of Master gen	Master genes	Controlled genes
AUFLP	Koren-Barkai	3	YBL039C	YBR278W YCL061C YDR121W YER070W YLR176C
			YKL113C	YHR031C YJL071W YLR079W YPR120C
			YOR241W	YER040W
CC-MWDS		1	YOR241W	
CI-MWDS		1	YBL039C	
AUFLP	Daniel-Burke	1	YOR026W	other genes
CC-MWDS, CI-MWDS		4	YGL086W YJL013C YJL030W YOR026W	
AUFLP	Finger	1	YGL233W	other genes
CC-MWDS, CI-MWDS		3	YER008C YGL233W YGR009C	
AUFLP	Berg-Poot	1	ENSMUSG26380	other genes
CC-MWDS, CI-MWDS		5	ENSMUSG21255 ENSMUSG24406 ENSMUSG25056 ENSMUSG26380 ENSMUSG27547	
AUFLP	Breslow-Weissman	1	YHR200W	other genes
CC-MWDS, CI-MWDS		6	YGR135W YHR200W YKL145W YMR263W YNL097C YOR261C	

were generated or analyzed during the current study.

References

[1] S. Bakhteh, A. Ghaffari-Hadigheh, and N. Chaparzadeh, *Identification of minimum set of master regulatory genes in gene regulatory networks*, IEEE/ACM Trans. Comput. Biol. Bioinform. **17** (2018), no. 3, 999–1009. https://doi.org/10.1109/TCBB.2018.2875692.

[2] A.L. Barabási, N. Gulbahce, and J. Loscalzo, *Network medicine: a network-based approach to human disease*, Nat. Rev. Genet. **12** (2011), no. 1, 56–68. https://doi.org/10.1038/nrg2918.

[3] A. Barrat, M. Barthelemy, and A. Vespignani, *The architecture of complex weighted networks: Measurements and models*, Large scale structure and dynamics of complex networks: from information technology to finance and natural science, World Scientific, 2007, pp. 67–92.

[4] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, *Multiple object tracking using k-shortest paths optimization*, IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011), no. 9, 1806–1819. https://doi.org/10.1109/TPAMI.2011.21.

[5] K. Bernhard and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 2008.

[6] Jørgen B.J. and G. Gregory, *Digraphs: Theory, Algorithms and Applications*, Springer, London, 2000.

[7] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, vol. 290, Citeseer, 1976.

[8] S.P. Borgatti, *Centrality and AIDS*, Connections **18** (1995), no. 1, 111–113.

[9] R. Cordone and G. Lulli, *An integer optimization approach for reverse engineering of gene regulatory networks*, Discrete Appl. Math. **161** (2013), no. 4-5, 580–592. https://doi.org/10.1016/j.dam.2012.02.010.

[10] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math. **1** (1959), 269–271.

[11] H. Eiselt and V. Marianov, *Foundations of Location Analysis*, vol. 155, Springer science & business media, 2011.

[12] F. Emmert-Streib, M. Dehmer, and B. Haibe-Kains, *Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks*, Front. Cell Dev. Biol. **2** (2014), Article ID: 38 https://doi.org/10.3389/fcell.2014.00038.

[13] B.L. Fox, *κth shortest paths and applications to the probabilistic networks*, ORSA/TIMS Joint National Mtg. **23** (1975), Artice ID: B263.

[14] C. Frank and K. Römer, *Distributed facility location algorithms for flexible con-*

figuration of wireless sensor networks, Distributed Computing in Sensor Systems (Berlin, Heidelberg) (J. Aspnes, C. Scheideler, A. Arora, and S. Madden, eds.), Springer Berlin Heidelberg, 2007, pp. 124–141.

[15] S.L. Hakimi, *Optimum locations of switching centers and the absolute centers and medians of a graph*, Oper. Res. **12** (1964), no. 3, 450–459.
https://doi.org/10.1287/opre.12.3.450.

[16] M. Hamed, C. Spaniol, A. Zapp, and V. Helms, *Integrative network-based approach identifies key genetic elements in breast invasive carcinoma*, BMC Genomics **16** (2015), no. 5, 1–14.
https://doi.org/10.1186/1471-2164-16-S5-S2.

[17] B.H. Junker and F. Schreiber Falk, *Analysis of Biological Networks*, John wiley & sons, 2011.

[18] H. Kitano, *A robustness-based approach to systems-oriented drug design*, Nat. Rev. Drug Discov. **6** (2007), no. 3, 202–210.
https://doi.org/10.1038/nrd2195.

[19] A. Klose and A. Drexl, *Facility location models for distribution system design*, Eur. J. Oper. Res. **162** (2005), no. 1, 4–29.
https://doi.org/10.1016/j.ejor.2003.10.031.

[20] I. A. Kovács and A. L. Barabási, *Network science: Destruction perfected*, Nature **524** (2015), 38–39.

[21] N. Lazic, I. Givoni, B. Frey, and P. Aarabi, *Floss: Facility location for subspace segmentation*, 2009 ieee 12th international conference on computer vision, IEEE, 2009, pp. 825–832.

[22] H. Li, *Two-view motion segmentation from linear programming relaxation*, 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[23] Y. Lim and H. Kim, *A shortest path algorithm for real road network based on path overlap*, J. East Asia Soc. Transp. Stud. **6** (2005), 1426–1438.
https://doi.org/10.11175/easts.6.1426.

[24] Y.Y. Liu, J.J. Slotine, and A.L. Barabási, *Controllability of complex networks*, Nature **473** (2011), no. 7346, 167–173.
https://doi.org/10.1038/nature10011.

[25] G. Lulli and M. Romauch, *A mathematical program to refine gene regulatory networks*, Discrete Appl. Math. **157** (2009), no. 10, 2469–2482.
https://doi.org/10.1016/j.dam.2008.06.044.

[26] K. Mehlhorn and P. Sanders, *Algorithms and Data Structures: The Basic Toolbox*, Springer Science & Business Media, 2008.

[27] T. Milenković, V. Memišević, A. Bonato, and N. Pržulj, *Dominating biological networks*, PloS one **6** (2011), no. 8, Artice ID: e23016.
https://doi.org/10.1371/journal.pone.0023016.

[28] F. Morone and H.A. Makse, *Influence maximization in complex networks through optimal percolation*, Nature **524** (2015), no. 7563, 65–68.
https://doi.org/10.1038/nature14604.

[29] J.C. Nacher and T. Akutsu, *Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control*, New J. Phys. **14**

(2012), no. 7, Article ID: 073005.
https://doi.org/10.1088/1367–2630/14/7/073005.

[30] ———, *Structural controllability of unidirectional bipartite networks*, Sci. Rep. **3** (2013), no. 1, Article ID: 1647.
https://doi.org/10.1038/srep01647.

[31] M. Nazarieh, A. Wiese, T. Will, M. Hamed, and V. Helms, *Identification of key player genes in gene regulatory networks*, BMC Syst. Biol. **10** (2016), no. 1, Article ID: 88.
https://doi.org/10.1186/s12918-016-0329-5.

[32] S. Ohno, *Major Sex-Determining Genes*, Springer-Verlag, Berlin, Germany, 1979.

[33] T. Opsahl, F. Agneessens, and J. Skvoretz, *Node centrality in weighted networks: Generalizing degree and shortest paths*, Soc. Netw. **32** (2010), no. 3, 245–251.

[34] R.Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson, *Alignment of metabolic pathways*, Bioinformatics **21** (2005), no. 16, 3401–3408.
https://doi.org/10.1093/bioinformatics/bti554.

[35] D. Warde-Farley, S.L. Donaldson, O. Comes, K. Zuberi, R. Badrawi, P. Chao, M. Franz, C. Grouios, F. Kazi, and Christian T. Lopes, *The genemania prediction server: biological network integration for gene prioritization and predicting gene function*, Nucleic Acids Res. **38** (2010), no. suppl_2, W214–W220.
https://doi.org/10.1093/nar/gkq537.

[36] S. Wuchty, *Controllability in protein interaction networks*, Proc. Natl. Acad. Sci. **111** (2014), no. 19, 7156–7160.
https://doi.org/10.1073/pnas.1311231111.

[37] J.Y. Yen, *Finding the k shortest loopless paths in a network*, Mgmt. Sci. **17** (1971), no. 11, 712–716.

[38] X.F. Zhang, L. Ou-Yang, D.Q. Dai, M.Y. Wu, Y. Zhu, and H. Yan, *Comparative analysis of housekeeping and tissue-specific driver nodes in human protein interaction networks*, BMC Bioinform. **17** (2016), no. 1, Article ID: 358.
https://doi.org/10.1186/s12859-016-1233-0.

[39] X.F. Zhang, L. Ou-Yang, Y. Zhu, M.Yu. Wu, and D.Q. Dai, *Determining minimum set of driver nodes in protein-protein interaction networks*, BMC Bioinform. **16** (2015), no. 1, Article ID: 146.
https://doi.org/10.1186/s12859-015-0591-3.