Hub Interdiction & Hub Protection problems: Model formulations & Exact Solution methods

Prasanna Ramamoorthy
Sachin Jayaswal
Ankur Sinha
Navneet Vidyarthi

W.P. No.2016-10-01
October 2016

The main objective of the working paper series of the IIMA is to help faculty members, research staff and doctoral students to speedily share their research findings with professional colleagues and test their research findings at the pre-publication stage. IIMA is committed to maintain academic freedom. The opinion(s), view(s) and conclusion(s) expressed in the working paper are those of the authors and not that of IIMA.
Hub Interdiction & Hub Protection problems: 
Model formulations & Exact Solution methods

Prasanna Ramamoorthy 
Sachin Jayaswal 
Ankur Sinha 
Indian Institute of Management Ahmedabad, India. 
Navneet Vidyarthi 
John Molson School of Business 
Concordia University, Canada.

Abstract

In this paper, we present computationally efficient formulations for the hub interdiction and hub protection problems, which are bi-level and tri-level mixed integer linear programs, respectively. In the hub interdiction problem, the aim is to identify a set of \( r \) critical hubs from an existing set of \( p \) hubs that when interdicted results in the greatest disruption cost to the hub-and-spoke network. Reduction of the bi-level interdiction model to single level is straightforward using Karush-Kuhn-Tucker (KKT) conditions corresponding to the lower level problem; however, this turns out to be computationally inefficient in this context. Therefore, we exploit the structure of the problem using various closest assignment constraints to reduce the hub interdiction problem to single level. The modifications lead to computational savings of almost an order of magnitude when compared against the only model existing in the literature. Further, our proposed modifications offer structural advantages for Benders decomposition, which lead to substantial savings, particularly for large problems. Finally, we study and solve the hub protection problem exactly by utilizing the ideas developed for the hub interdiction problem. The tri-level protection problem is otherwise intractable, and to our best knowledge, has not been solved in the literature.

Keywords: Location, hub-and-spoke network, interdiction, protection, Benders decomposition.
Hub Interdiction & Hub Protection problems:  
Model formulations & Exact Solution methods

1 Introduction

Certain infrastructural assets are critical to the functioning of a nation’s economy and societal well being. The United States’ Department of Homeland Security identifies 16 infrastructure sectors as critical, such that their incapacitation or destruction can be debilitating to the national security, economy, and public health. Three out of these sixteen critical infrastructure sectors, namely transportation systems, communications, and energy employ hub-and-spoke as a dominant network structure because of its operational advantage. Hub-and-spoke networks exploit the economies of scale arising from consolidating at hubs the traffic originating from different sources and/or those destined to different demand points, instead of serving each origin-destination pair directly. Flows from the same origin with different destinations in a hub-and-spoke network are consolidated on their route at the hub where they are combined with flows that have different origins but the same destination (Campbell 1996). In multi-hub networks, traffic concentrated at a hub is directed to a second hub, which distributes it to the final destinations, thereby exploiting the economies of scale on the inter-hub flows. Another advantage of a hub-and-spoke network is that it results in fewer links, which makes the network construction cheaper and its maintenance easier, compared to an alternate network with direct connections between all sources and destinations.

O’Kelly (1986) was the first paper to study locating hubs between interacting cities. Campbell (1994) gave the first integer programming formulations for the \( p \)-hub median problem, uncapacitated hub location problem, \( p \)-hub center and hub covering problems. These models are largely inspired by their facility location counterparts: \( p \)-median problem, facility location problem, \( p \)-center problem and maximal covering problem. Since then research papers have been published in both single allocation (a non-hub is allocated to only one hub) or multiple allocation (a non-hub is allocated to one or more than one hub), capacitated (hubs have a fixed capacity) or uncapacitated (no limit on hub capacity) median location problems. Skorin-Kapov et al. (1996), Ebery et al. (2000), Ernst and Krishnamoorthy (1996), Hamacher et al. (2004) are some of the important works in this area. In recent years several variations of the hub location problems have also appeared in literature. Notable among those are, hubs with congestion (Elhedhli and Hu 2005), (Jayaswal and Vidyarthi 2013), cycle hub location problem, where hubs are connected in a cycle (Contreras et al. 2016), tree of hubs location problem, where hubs are connected by a tree structure (Contreras et al. 2010), flow dependent economies of scale (O’Kelly and Bryan 1998), stochastic demands (Contreras et al. 2011b), and hub location over a time period (Contreras et al. 2011a). A review of research papers in hub location can be found in: Alumur and Kara (2008), Campbell and O’Kelly (2012) and Farahani et al. (2013).

While a hub-and-spoke network structure is attractive due to its cost effectiveness, it is prone to severe disruptions in the event of a failure of any of its hubs. This is because failure of any hub in the network disrupts the flows from all the origin and destination points that it serves. A study states that it is possible to disrupt the entire United States’ air network by interdicting just 2% of its all

\(^1\)https://www.dhs.gov/what-critical-infrastructure
airports (Lewis 2006). A related real incident is the June’16 attack on Ataturk International airport in Turkey. Ataturk International airport is one of the busiest airports in the world and it serves as a hub for Turkish Airlines, Onur Air and Atlas Global. It was attacked on 28th June 2016 by some gunmen, which left 45 people dead and more than 230 injured. Following the attack, flights destined for Istanbul were diverted to other hubs in the vicinity. The Federation Aviation Administration (FAA) and the Transportation Security Administration (TSA) of the United states’ government grounded all passenger and commercial flights to and from Turkey for several hours post the attack, which resulted in traffic disruptions throughout the world. Incidents like this, make it necessary to identify critical hubs in a hub-and-spoke network so that resources may be focused towards their fortification (protection), and this forms the motivation of our study. We study the problem of identifying the critical hubs, which when disrupted will cause the maximum disruption. We call this problem as the hub interdiction problem (HIP). We further study the hub protection problem (HPP), which identifies the hubs to protect, taking into account the reaction of the interdictor (attacker). The hub interdiction and protection problems are modeled as bi-level and tri-level mixed integer programs (MIPs), which are challenging to solve. We present efficient solution methods that are capable of solving large instances of the problems in a reasonable time.

Following are the major contributions of the paper:

- This is among the first few papers to study the bi-level interdiction and tri-level protection problems in the context of hub-and-spoke network design.
- We present alternate ways to exploit the structure of the bi-level interdiction problem to reduce it to a tractable single-level optimization problem.
- We further present Benders decomposition for the different single-level formulations to efficiently solve large instances of the hub interdiction problem.
- To the best of our knowledge, this is the only paper to efficiently solve large instances of the hub interdiction and protection problems to optimality.

In Section 2, we present a literature review on interdiction and protection problems. Section 3 presents the model formulation of the bi-level interdiction problem, followed by two alternate ways of reducing it to a single level problem to make it tractable. The first approach uses the well-known Karush-Kuhn-Tucker (KKT) conditions for the lower level problem, while the second approach exploits the structure of the solution to the lower level problem to replace it by what we call as the closest assignment constraints (CACs). We present alternate forms of CACs, and present their relative merits and computational performances in Sections 4 and 5, respectively. In Section 6, we exploit Benders decomposition of the reduced single level formulation of the hub interdiction problem to solve it more efficiently, and present its computational results. In Section 7, we present the tri-level hub protection problem, followed by its solution using Implicit enumeration in combination with Benders decomposition. We conclude by providing possible future research directions in Section 8.

2 Literature Review

Interdiction problems have been widely studied with respect to network flows (network interdiction) and facility location (facility interdiction) problems. The decision maker in an interdiction problem is interested in identifying the set of nodes/arcs (in network interdiction) or facilities (in facility
interdiction) that when interdicted causes the maximum loss to her. The problem is modeled as a Stackelberg game in which the attacker is the leader and the network operator (defender) is the follower.

2.1 Network Interdiction

Network interdiction problems identify critical nodes or arcs in a network. The defender operates on the network to optimize her objective that can be one of the following: (i) to pass through the network as fast as possible (shortest path network interdiction) (Corley and Sha, 1982; Israeli and Wood, 2002; Cappanera and Scaparra, 2011) (ii) to move through the network without getting caught (most reliable path interdiction) (Shimizu et al., 2012) or (iii) to maximize the amount of flow passing through the network (maximum flow network interdiction) (Wood, 1993; Cormican et al., 1998). The objectives of the attacker in these models are: (i) to intercept or destroy the arc(s)/node(s) so as to maximize the length of the shortest path, or (ii) to minimize the maximum flow in the network, or (iii) to maximize the probability of detection in the network. These models find applications in disrupting enemy flows (McMasters and Mustin, 1970), infectious disease control (Assimakopoulos, 1987), counter-terrorism (Farley, 2003), interception of nuclear material (Morton et al., 2007) and contraband smuggling (Washburn and Wood, 1995). A review of network interdiction models with applications can be found in Collado and Papp (2012).

2.2 Facility Interdiction and Protection

Facility interdiction problems study the identification of critical facilities in a supply network. Church et al. (2004) proposed the $r$-interdiction median problem ($r$-IMP) and $r$-interdiction covering problem ($r$-ICP) to study interdiction of facilities under different location scenarios. The $r$-IMP identifies the set of $r$ facilities to remove from the existing ones to maximize the overall demand weighted transportation cost of serving customers from remaining facilities, whereas $r$-ICP identifies the set of $r$ facilities that when removed minimizes the total demand that can be covered within a specific distance or time.

Different variants of r-IMP are studied in the literature. Church and Scaparra (2007a) studied an extension of the problem where the success of the attack is uncertain. The authors assumed that the attacks are successful with a given probability. Losada et al. (2012) studied another type of uncertainty in r-IMP which is the uncertainty of the degree of impact created by the attack. This problem identifies disruption scenarios that result in the maximum overall traveling distance for serving all customers when the impact on a facility after an attack is uncertain. A key assumption here is that the degree of interdiction impact on a facility is proportional to the amount of resources employed.

The problems described above assume no restrictions on the capacity of the facilities. Aksen et al. (2014) studied the partial interdiction of capacitated $r$-IMP, wherein facilities operate with a fixed capacity, which when interdicted, operate with a reduced capacity. The amount of capacity reduction is directly proportional to the interdiction resources deployed. Though various versions of $r$-IMP are studied (capacitated and uncapacitated, partial and full interdiction), its counterpart ($r$-ICPs) have received only limited attention in literature.

Church and Scaparra (2007b) studied an extension of $r$-IMP, known as the $r$-interdiction median problem with fortification ($r$-IMF). This problem identifies optimal fortification/protection strategies against interdiction. According to this model, when a facility is fortified/protected, it is completely immune to an attack. Scaparra and Church (2008a) formulated the $r$-IMF as a bi-level MIP, which is solved using an Implicit enumeration algorithm. Scaparra and Church (2008b) proposed an alternate
method for $r$-IMF. The idea is to reformulate the problem as a maximal covering problem with precedence constraints, which is solved using an approximate heuristic. This helps in identifying the upper and lower bounds of the problem, which is used to reduce the size of the original problem. This reduced problem is then solved to optimality using a standard MIP solver. Losada et al. (2010), Scaparra and Church (2012), Aksen et al. (2010), Aksen and Aras (2012), Aksen et al. (2013) and Liberatore et al. (2012) are other related works in this area.

### 2.3 Hub interdiction

Interdiction of hubs in a hub-and-spoke network has received scarce attention in the literature, despite its many useful applications as discussed in Section 1. However, there have been a few studies in closely related areas. An et al. (2015) and Azizi et al. (2016), for example, studied the reliable hub-and-spoke network design problem, which includes the possibility of re-routing flows through backup hubs when the active hubs are disrupted. However, the objective in both these papers is to minimize the weighted sum of pre-disruption and the expected value (over all disruption scenarios) post-disruption transportation cost. HIPs by contrast, study the worst-case loss to the defender.

To the best of our knowledge, Lei (2013) is the only paper on HIP/HPP. The author presented bi-level and tri-level MIP for HIP and HPP, respectively. However, due to the complexity of the problem, computational results are presented only for small instances of HIP, whereas no solution method for HPP is presented. The objective of this paper is to present efficient solution methods capable of solving large instances of HIP and HPP.

### 3 Problem Description and Model formulation

We consider a hub-and-spoke network with a set of flows $(W_{ij})$ associated with every source node $i \in N$ and destination node $j \in N$. The flows are always routed through one or two of the hubs from the set $H \subseteq N$ of $p$ hubs to benefit from economies of scale in transportation. The objective of the operator of the network (called defender) is to identify the set of $r$ hubs, which when destroyed by an attacker causes her the maximum cost from rerouting of flows that are affected because of the interdicted hubs. We make a reasonable assumption of $r < p$ since the attacker (typically a terrorist organization) might not have resources to interdict all the hubs. This is the context of HIP, which is modeled as a Stackelberg game. In HIP, the attacker makes the first move by choosing the $r$ hubs to interdict, followed by the defender who decides how to route the flows through the remaining $p - r$ hubs with minimum cost. This is represented as a bi-level MIP. The hierarchical structure of the problem is shown through Figure ??.

#### 3.1 Model Formulation

In this section, we provide a mathematical formulation for the HIP. To begin with, we introduce the notations used, and then move on to the formulation.

#### 3.1.1 Notations

To model the problem, we define the following indices and parameters:
\( i \): Index for source nodes, \( i \in N; \)
\( j \): Index for destination nodes, \( j \in N; \)
\( k \): Index for hub which is connected to \( i \), \( k \in H; \)
\( m \): Index for hub which is connected to \( j \), \( m \in H; \)
\( \alpha \): Discount factor for collection (source to hub), \((i \rightarrow k)\)
\( \delta \): Discount factor for transhipment (hub to hub), \((k \rightarrow m)\)
\( \chi \): Discount factor for distribution (hub to destination), \((m \rightarrow j)\)
\( H \): set of all hubs, \( H \subseteq N; \)

\( W_{ij} \): Flow from source \( i \) to destination \( j; \)
\( d_{ij} \): Cost of travelling from node \( i \) to node \( j; \)
\( d_{ijkm} \): Cost of traversing from source \( i \) to destination \( j; \)
\( \alpha d_{ik} + \delta d_{km} + \chi d_{mj}; \)
\( p \): No. of hubs present in the system;
\( r \): No. of hubs to interdict:

The decision variables are defined as follows:

\( X_{ijkm} \): Fraction of flows from source \( i \) to destination \( j \) through hubs \( k \) and \( m \) after interdiction;
\( z_k \): \( 1 \), if hub \( k \) remains open after interdiction, \( 0 \) otherwise.

With the above notation, the hub interdiction problem can mathematically be stated as the following bi-level MIP.

\[
\text{[HIP}_{2L}\text{]} : \max \ Z
\]
\[ \text{s.t. } \sum_{k \in H} z_k = p - r \] (1)
\[ z_k \in \{0, 1\} \quad \forall k \in H \] (2)
\[ Z = \min \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} W_{ij} d_{ijkm} X_{ijkm} \] (3)
\[ \text{s.t. } \sum_{k \in H} \sum_{m \in H} X_{ijkm} = 1 \quad \forall i, j \in N \] (4)
\[ \sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} \leq z_k \quad \forall i, j \in N; k \in H \] (5)
\[ X_{ijkm} \geq 0 \quad \forall i, j \in N; k, m \in H \] (6)

Attacker’s objective function (1) maximizes the defender’s objective of minimizing the weighted transportation cost. Constraint (2) ensures that \( p - r \) hubs remain open after interdiction. Constraints (4) to (7) form the lower level routing problem. Constraint (5) states that the fractional sum of flows between source \( i \) and destination \( j \) through all possible combinations of hubs \( k \) and \( m \) should be equal to \( 1 \). Constraint (6) models the condition that a flow can happen through and out of the hub only if the hub remains open. The same condition can be alternatively represented by the following set of constraints, as done by Lei (2013):

\[
\sum_{k \in H} X_{ijkm} \leq z_m \quad \forall i, j \in N; m \in H
\]
\[
\sum_{m \in H} X_{ijkm} \leq z_k \quad \forall i, j \in N; k \in H
\]
However, the constraint set of the form (6) is proven to be facet defining (Hamacher et al., 2004). Hence, constraint set (6) provides a tighter LP relaxation, which is effective in solving large instances of the hub interdiction problem.

3.2 Reduction to Single level

Bi-level problems, even with linear programs at both levels, are NP-hard problems (Audet et al., 1997; Frangioni, 1995). HIP, which is a bi-level MIP, is even more difficult. Bi-level problems in the literature are traditionally solved by reducing the problem to single level using reduction techniques. We present two alternate ways of reducing HIP to a single level MIP to make it tractable. The first approach is based on the use of the well-known KKT conditions for the lower level problem, while the second approach exploits the structure of the solution to the lower level problem to replace it by CACs.

3.2.1 Single level reduction using lower level KKT conditions

The lower level problem in HIP is a linear program with continuous variables. This makes its reduction to a single level using KKT conditions straightforward. For a given upper level variable $\bar{z}_k$, taking dual variables $\phi_{ij}$ and $\lambda_{ijk}$ for constraints (5), (6) we get the following Lagrangian relaxation:

$$
\sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} W_{ij} d_{ijkm} X_{ijkm} + \sum_{i \in N} \sum_{j \in N} \phi_{ij} \left( \sum_{k \in H} \sum_{m \in H} X_{ijkm} - 1 \right) + \\
\sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \lambda_{ijk} \left( \sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} - \bar{z}_k \right)
$$

Differentiating the expression with respect to $X_{ijkm}$ we get:

$$
W_{ij} d_{ijkk} + \phi_{ij} + \lambda_{ijk} \geq 0 \quad \forall i, j \in N, k, m \in H, k = m
$$

$$
W_{ij} d_{ijkm} + \phi_{ij} + \lambda_{ijk} + \lambda_{ijm} \geq 0 \quad \forall i, j \in N, k, m \in H, k \neq m
$$

The single level problem with KKT conditions can be written as:

$$
[HIP_{KKT}]: \max_{z_k, X_{ijkm}} \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} W_{ij} d_{ijkm} X_{ijkm} \quad (8)
$$

s.t. $\sum_{k \in H} z_k = p - r \quad (9)$

$$
W_{ij} d_{ijkm} + \phi_{ij} + \lambda_{ijk} \geq 0 \quad \forall i, j \in N; k, m \in H, k = m \quad (10)
$$

$$
W_{ij} d_{ijkm} + \phi_{ij} + \lambda_{ijk} + \lambda_{ijm} \geq 0 \quad \forall i, j \in N; k, m \in H, k \neq m \quad (11)
$$

$$
\lambda_{ijk} \left( \sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} - \bar{z}_k \right) = 0 \quad \forall i, j \in N; k \in H \quad (12)
$$

$$
\sum_{k \in H} \sum_{m \in H} X_{ijkm} = 1 \quad \forall i, j \in N \quad (13)
$$

$$
\sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} \leq z_k \quad \forall i, j \in N; k \in H \quad (14)
$$

$$
X_{ijkm}, \lambda_{ijk} \geq 0, -\infty \leq \phi_{ij} \leq \infty \quad \forall i, j \in N; k, m \in H \quad (15)
$$
This resulting single level problem contains non-linear complementary slackness constraint (12), that is linearized using the standard method (Fortuny-Amat and McCarl [1981]). The linearized constraints are:

\[ \lambda_{ijk} \leq M \alpha_{ijk} \quad \forall \, i, j \in N; k \in H \]  

\[ \sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} - z_k \geq -M(1 - \alpha_{ijk}) \quad \forall \, i, j \in N; k \in H \]  

\[ \alpha_{ijk} \in \{0, 1\} \quad \forall \, i, j \in N; k \in H \]  

The objective function (8) along with constraints (9) - (11), (13) - (15) and (16) - (18) forms the linearized problem. The problem contains \(n^2p^2 + 3n^2p + n^2 + 1\) constraints and \(n^2p^2 + 2n^2p + n^2 + p\) variables, out of which \(p + n^2p\) are binary variables. For a 25-node 10-hub problem, this results in 81,876 constraints and 6,260 binary variables out of a total of 75,635 variables, which makes it a fairly difficult problem to solve. This enormous size is due to the addition of binary variables to convert the mixed integer non-linear program to mixed integer linear program. Given that KKT based reduction might not be suitable to solve large scale HIPs, we look at an alternative formulation that exploits the properties of the problem to come up with a more tractable formulation.

### 3.2.2 Single level reduction using closest assignment constraints

The attacker in the upper level of the bi-level HIP decides upon his optimal set of \(r\) hubs to attack, after which the defender routes the disrupted flows optimally through the remaining hubs. Since the lower level problem contains just the routing decision, we can reduce the lower level problem using a closest assignment constraint by which the flows are allocated to the cheapest cost routes.

Closest assignment constraints are used in facility location problems to allocate customers to their nearby facilities. It is because in facility location problems cost is not always proportional to the distance between customer and the facility. The system might assign a customer to some facility farther from him, while he might want to be assigned to the nearest open facility. The closest assignment constraint captures this requirement. Espejo et al. (2012) and Gerrard and Church (1996) compare different closest assignment constraints used in location problems and study their theoretical properties. These constraints find applications in hazardous facility location (Song et al., 2013), facility location under competition (Dobson and Karmarkar, 1987), and interdiction problems (Liberatore et al., 2011). Lei (2013) converted the bi-level hub interdiction problem to single level using the following set of closest assignment constraints:

\[ \sum_{q \in C_{ijkm}} X_{ijqs} + X_{ijkm} \geq z_k + z_m - 1 \quad \forall \, i, j \in N; k, m \in H \]  

(CAC1)

where, \(C_{ijkm} = \{(q, s) \mid d_{ijqs} < d_{ijkm} \text{ or } (d_{ijqs} = d_{ijkm} \text{ and } (q < k \text{ or } (q = k \text{ and } s < m)))\}\).

For a given source - destination (s - d) pair \((i, j)\), CAC1 ensures that the flow between them happens only through a path that is no costlier than the path via hubs \(k\) and \(m\) as long as they are open. This is an extension of its counterpart for facility location problems given by Church and Cohon (1976). CAC1 arbitrarily breaks any tie between paths having the same cost. Breaking ties for HIP is not necessary, unlike in facility location problem without which it becomes infeasible. For ease of discussion, we redefine CAC1 as:

\[ \sum_{q \mid d_{ijqs} \leq d_{ijkm}} X_{ijqs} + X_{ijkm} \geq z_k + z_m - 1 \quad \forall \, i, j \in N; k, m \in H \]  

(CAC1')
It is noteworthy that the use of \( CAC1' \) in place of \( CAC1 \) makes little computational difference.

We propose two alternative CACs, which are presented below. These constraints are designated as \( CAC2 \) and \( CAC3 \). \( CAC2 \) forbids assignment of flows from any \( s - d \) pair \((i,j)\) to a path costlier than the path \( X_{ijkm} \) \((i \rightarrow k \rightarrow m \rightarrow j)\) when hubs \( k \) and \( m \) are open \((z_k \text{ and } z_m = 1)\). \( CAC2 \) is given below:

\[
\sum_{q \in E_{ijkm}} X_{ijqs} \leq 2 - z_k - z_m \quad \forall \ i, j \in N; k, m \in H \quad (CAC2)
\]

where, \( E_{ijkm} = \{(q,s)| d_{ijqs} > d_{ijkm}\} \). \( CAC2 \) is similar to the constraint devised by Wagner and Falkson (1975) for facility center problems. \( CAC3 \) ensures closest assignment by allocating flows from any \( s - d \) pair \((i,j)\) through all the paths \( X_{ijqs} \) \((i \rightarrow q \rightarrow s \rightarrow j)\) not greater than the current path \( X_{ijkm} \) \((i \rightarrow k \rightarrow m \rightarrow j)\) when hubs \( k \) and \( m \) are not interdicted. This is presented below:

\[
\sum_{q \in H} \sum_{s \in H} d_{ijqs} X_{ijqs} + (M - d_{ijkm})(z_k + z_m - 1) \leq M \quad \forall \ i, j \in N; k, m \in H \quad (CAC3)
\]

where, \( M = \max \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} d_{ijkm}. \)

In the above inequality, by fixing \( z_k \) and \( z_m \) to 1, the allocations \( X_{ijqs} \) will be on paths shorter than \( d_{ijkm} \). \( CAC3 \) is an adaptation of the closest assignment constraint from Berman et al. (2009) for hub location problems.

The single level HIP with the addition of closest assignment constraint takes the following form:

\[
[HIP_{1L}] : \max_{y_k,X_{ijkm}} \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} W_{ij} d_{ijkm} X_{ijkm}
\]

s.t. \((2), (3), (5) - (7) \quad CAC1 \quad or \quad CAC2 \quad or \quad CAC3 \)

This single level problem is smaller in size compared to the single level problem formulation obtained using KKT conditions. The current problem has the same number of variables as the original bi-level problem, while the latter one has a very large model size due to the extra binary variables that were required to linearize it. For a 25-node 10-hub problem, the reduced single level problem using the closest assignment constraint contains 62,510 variables and 69,376 constraints compared to 75,635 variables and 81,876 constraints for the single level problem using KKT conditions. This makes the KKT approach computationally inefficient when compared with the CACs. Hence going forward, we focus on reduction using closest assignment constraints for solving the HIP.

### 3.3 Dominance relationship between CACs

In order to find the best closest assignment constraint among the proposed constraints for reduction, we study the dominance relationships between the constraints. A constraint which dominates all the other alternate constraints is the one with the tightest LP relaxation for the problem. Espejo et al. (2012) proposed the rules for dominance relationship between constraints as follows: A constraint dominates the other if the former constraint implies the latter. If both constraints imply one another we say that the constraints are equivalent.

In the following, we state dominance relationships between the closest assignment constraints introduced above.
**Proposition 1.** CAC2 is equivalent to CAC1’

*Proof.* CAC2 can be written as:

\[
1 - \sum_{(qs)|d_{ijqs} \leq d_{ijkm}} X_{ijqs} \leq 2 - z_k - z_m \quad \forall \ i, j \in N; k, m \in H
\]

\[
\iff \sum_{(qs)|d_{ijqs} \leq d_{ijkm}} X_{ijqs} \geq z_k + z_m - 1 \quad \forall \ i, j \in N; k, m \in H
\]

Separating \(X_{ijkm}\) term we get:

\[
\sum_{qs|d_{ijqs} \leq d_{ijkm}} X_{ijqs} + X_{ijkm} \geq z_k + z_m - 1 \quad \forall \ i, j \in N; k, m \in H
\]

Hence, CAC2 \(\iff\) CAC1’. Similarly CAC1’ \(\iff\) CAC2 can be proved. Therefore, CAC2 is equivalent to CAC1’.

**Proposition 2.** CAC2 dominates CAC3

*Proof.* CAC2 can be written as:

\[
\sum_{(qs)\in E_{ijkm}} X_{ijqs} + z_k + z_m \leq 2 \quad \forall \ i, j \in N; k, m \in H
\]

CAC2 can be relaxed and written as:

\[
X_{ijqs} + z_k + z_m \leq 2 \quad \forall \ i, j \in N; k, m \in H; (q, s) \in E_{ijkm}.
\] (CAC2-rel)

Now it is evident that, CAC2 \(\implies\) CAC2-rel, while CAC2-rel \(\implies\) CAC2. Therefore CAC2 dominates CAC2-rel. In order to show that CAC2 dominates CAC3, we just need to prove that CAC2-rel implies CAC3.

Multiplying by \(d_{ijqs}\) on both sides of CAC2-rel and summing it up over \((q, s) \in E_{ijkm}\) we get,

\[
\sum_{(qs)\in E_{ijkm}} d_{ijqs}X_{ijqs} + \sum_{(qs)\in E_{ijkm}} d_{ijqs}z_k + \sum_{(qs)\in E_{ijkm}} d_{ijqs}z_m \leq 2 \sum_{(qs)\in E_{ijkm}} d_{ijqs} \quad \forall \ i, j \in N; k, m \in H
\]

Adding \(\sum_{(qs)|d_{ijqs} \leq d_{ijkm}} d_{ijqs}X_{ijqs} + (M - d_{ijkm}) - \sum_{(qs)\in E_{ijkm}} d_{ijqs}(z_k + z_m - 1)\) to both sides of the above inequality, we get

\[
\sum_{q \in H, s \in H} d_{ijqs}X_{ijqs} + (M - d_{ijkm})(z_k + z_m - 1) \leq \sum_{(qs)\in E_{ijkm}} d_{ijqs} + \sum_{(qs)|d_{ijqs} \leq d_{ijkm}} d_{ijqs}X_{ijqs}
\]

\[
+ (M - d_{ijkm})(z_k + z_m - 1) + \sum_{(qs)\in E_{ijkm}} d_{ijqs}(z_k + z_m - 1) \quad \forall \ i, j \in N; k, m \in H.
\]

In the above constraint, the right hand side takes the maximum value when both \(z_k\) and \(z_m\) are one. This value is always bounded by \(M\) since \(\sum_{(qs)|d_{ijqs} \leq d_{ijkm}} d_{ijqs}X_{ijqs} \leq d_{ijkm}\). Therefore, we get the following

\[
\sum_{q \in H, s \in H} d_{ijqs}X_{ijqs} + (M - d_{ijkm})(z_k + z_m - 1) \leq M \quad \forall \ i, j \in N; k, m \in H,
\]

which proves that CAC2-rel \(\implies\) CAC3. Hence, CAC2 dominates CAC3.

In the following section, we suggest refinements of CAC1 and CAC2, and present two additional CAC sets that lead to fewer constraints.
3.4 Reduced formulation

In this section, we propose a reduced formulation for CAC1 and CAC2 based on constraint dominance principles.

**Proposition 3.** For a given s - d pair \((i, j)\) and hubs \((k, m \neq k)\) between them, CAC1\(_{ijkm}\) dominates CAC1\(_{ijkm}\) when \(d_{ijkm} < d_{ijmk}\).

Proof. For given CAC1\(_{ijkm}\), CAC1\(_{ijmk}\) and \(d_{ijkm} < d_{ijmk}\), comparing CAC1\(_{ijkm}\) and CAC1\(_{ijmk}\) we see that RHS of both the constraints are the same and LHS of CAC1\(_{ijkm}\) contains the terms in the LHS of CAC1\(_{ijkm}\) (since \(d_{ijkm} < d_{ijmk}\)) and additional \(X_{ijqs}\) variables.

CAC1\(_{ijkm}\) and CAC1\(_{ijmk}\) constraints are binding when \(z_k\) and \(z_m = 1\). The additional \(X_{ijqs}\) variables in CAC1\(_{ijkm}\) are set to zero because LHS of CAC1\(_{ijkm}\) is equal to 1, thereby making CAC1\(_{ijmk}\) redundant. Therefore, CAC1\(_{ijkm}\) dominates CAC1\(_{ijmk}\).

Based on Proposition 3 we propose a new formulation for CAC1 which is given below: We define a set, \(H'_{ijkm} = \{(k, m)|d_{ijkm} \leq d_{ijmk}; (m, k)|d_{ijmk} < d_{ijkm}\} \forall i, j \in N, k, m \geq k \in H\} \) Next we define the set \(C'_{ijkm}\) which eliminates the closest assignment constraint corresponding to the longest path as follows: \(C'_{ijkm} = \{(q, s)| d_{ijqs} < d_{ijkm} \) or \(d_{ijqs} = d_{ijkm}\) and \((q < k \) or \(q = k \) and \(s < m\))\) \(\forall i, j \in N, (k, m) \in H'_{ijkm}\). The new closest assignment constraint can then be written as follows:

\[
\sum_{q \in C'_{ijkm}} X_{ijqs} + X_{ijkm} \geq z_k + z_m - 1 \forall i, j \in N; k, m \in H
\]

(rCAC1)

**Proposition 4.** For a given s - d pair \((i, j)\) and hubs \((k, m \neq k)\) between them, CAC2\(_{ijkm}\) dominates CAC2\(_{ijmk}\) when \(d_{ijkm} < d_{ijmk}\).

Proof. Given CAC2\(_{ijkm}\) and CAC2\(_{ijmk}\) and \(d_{ijkm} < d_{ijmk}\), comparing CAC2\(_{ijkm}\) and CAC2\(_{ijmk}\) we see RHS of both the constraints are the same and LHS of CAC2\(_{ijkm}\) contains the terms in the LHS of CAC2\(_{ijmk}\) (since \(d_{ijkm} < d_{ijmk}\)) and additional \(X_{ijqs}\) variables.

CAC2\(_{ijkm}\) and CAC2\(_{ijmk}\) are binding when \(z_k\) and \(z_m = 1\). CAC2\(_{ijkm}\) sets the variables in LHS of CAC2\(_{ijkm}\) and additional \(X_{ijqs}\) variables to zero, making CAC2\(_{ijkm}\) redundant. Therefore, CAC2\(_{ijkm}\) dominates CAC2\(_{ijmk}\).

Based on Proposition 4 we propose a new formulation for CAC2 which is given below: We define a set \(S'_{ijkm} = \{(k, m)|d_{ijkm} \leq d_{ijmk} \) or \(d_{ijmk} < d_{ijkm}\} \forall i, j \in N, k, m \geq k \in H\} \) Next we define the set \(E'_{ijkm}\) which eliminates the closest assignment constraint corresponding to the longest path as follows: \(E'_{ijkm} = \{(q, s)|d_{ijqs} > d_{ijkm}; \forall i, j \in N, (k, m) \in S'_{ijkm}\}\). The new closest assignment constraint can then be written as:

\[
\sum_{q \in E'_{ijkm}} X_{ijqs} \leq 2 - z_k - z_m \forall i, j \in N; k, m \geq k \in H.
\]

(rCAC2)

The reduced constraint sets rCAC1 and rCAC2 have \(|N|^2(p^2 + p)/2\) constraints, while their parents CAC1 and CAC2 have \(|N|^2p^2\) constraints.
4 Advantages of CAC2 over CAC1

In this section, we outline the advantages of CAC2 over CAC1 in different stages of the solution process. CAC2 has certain structural properties that help in solving the single level problem faster than the one with CAC1. These properties are also valid for rCAC2 since it is a tighter version of CAC2.

4.1 Advantage at Presolve

Presolve procedure is executed by the solver, prior to solving the optimization problem to reduce the size of the given problem by removing the redundant variables and constraints. Probing is a process that is carried out at the presolve step wherein logical consequences are investigated by setting the binary variables at their bounds (Savelsbergh, 1994). In this subsection, we show that CAC2 and rCAC2 together with constraint (6) eliminate a lot of variables by probing.

Proposition 5. For a given s - d pair (i,j) and hub k, $X_{ijkm}$ variables that appear common in constraint (6) and CAC2,ijkk can be fixed to zero.

Proof. Given CAC2,ijkk, let the set A = \{X_{ijkm} | X_{ijkm} \in CAC2,ijkk; (6) for i, j, k\}

Case 1: $z_k = 0$ Variables in set A are reduced to zero by constraint (6) for i, j, k.

Case 2: $z_k = 1$ Variables in set A are reduced to zero by CAC2,ijkk.

Since the variables in set A are reduced to zero eitherwise, they can be eliminated from the model.

Thus, CAC2 and rCAC2 formulations eliminate a lot of variables by probing procedure. Despite the constraints being equivalent, probing reduction using CAC1 formulation is not straightforward. It will be obvious from the results provided in the later section, where we will observe that there is little advantage at the presolve stage with the CAC1 formulation. In the following subsection, we present the advantage provided by CAC2 in a branch-and-bound procedure.

4.2 Advantage at Branch-and-Bound step

In a branch-and-bound procedure, the given MIP is relaxed and the linear relaxation is solved at the root node. Further branching is done by setting the integer variables to its bounds that have taken a fractional value in the optimal solution to the relaxed problem. In our problem, branching is done by setting $z_k$ variables to zero and one. When a $z_k$ variable is set to one, some $X_{ijkm}$ variables are set to zero because of the CAC2 formulation. These variables can be eliminated from the model to reduce the model size. Alternatively, when $z_k$ is set to zero some $X_{ijkm}$ variables are eliminated because of constraint (6) which again reduces the model size. This is elaborated in the example below:

Consider a hub interdiction problem with $N = 5$, $p = 3$ and $r = 1$. Let the located hubs be 1,2 and 3. For a given s - d pair (0,4) of the problem we have the following distance matrix:

\[
\begin{array}{cccc}
   & 1 & 2 & 3 \\
0 & 373.8127 & 1006.071 & 2642.653 \\
1 & 1527.739 & 1375.603 & 1696.158 \\
2 & 2266.804 & 1745.126 & 1696.158 \\
3 & 1696.158 & 1696.158 & \\
\end{array}
\]

Now writing constraint (6) and CAC2 constraint for the path 0411 we have:

\[
\begin{align*}
X_{0411} + X_{0412} + X_{0413} + X_{0421} + X_{0431} & \leq z_1 \\
X_{0412} + X_{0413} + X_{0421} + X_{0422} + X_{0423} + X_{0431} + X_{0432} + X_{0433} & \leq 2 - 2z_1
\end{align*}
\]
As discussed above, in the branch-and-cut tree two branches are created with $z_1 = 1$ and $z_1 = 0$. For the branch with $z_1 = 0$, the variables $X_{0411}, X_{0412}, X_{0413}, X_{0421}, X_{0431}$ are set to zero. For the branch with $z_1 = 1$, the variables $X_{0412}, X_{0413}, X_{0421}, X_{0422}, X_{0423}, X_{0431}, X_{0432}, X_{0433}$ are set to zero. Thus $CAC2$ and $rCAC2$ formulations reduce the size of the problem further for each subproblem in the branch-and-cut tree. This advantage of $CAC2$ and $rCAC2$ further results in improved performance while solving the hub interdiction problem.

5 Computational Results

In the previous section, we studied the relative merits of different CACs in solving the hub interdiction problem. In this section, we present the results of our computational experiments to highlight the degree of computational advantage gained through the use of one CAC vis-a-vis others. For our experiments, we use instances derived from the Civil Aeronautics Board (CAB) dataset containing $|N| = 25$ nodes, and Australian Post (AP) dataset containing $|N| = 200$ nodes. The hub locations used for each of these instances are the optimal hub locations obtained by solving a corresponding uncapacitated p-hub median problem [Ebery et al., 2000]. All the computational experiments are performed on a workstation with a 2.60GHz Intel Xeon - e5 processor and 24GB memory, and all the instances are solved using Cplex 12.6.

In Table 1, we present the results of the experiments for CAB dataset with $|N| = 25$ nodes and $p \in \{7, 10\}$ hubs. The discount factors for collection ($\alpha$) and distribution ($\chi$) are set at 1.0 in the CAB dataset, while the discount factor for transhipment ($\delta$) is varied in the experiments. For $N = 25$ and $p = 7$ hubs, HIP with $CAC1$ or $CAC2$ or $CAC3$ contains 35,626 constraints and 30,633 variables. Similarly, $N = 25$ and $p = 10$ hubs, the number of constraints and variables are 69,376 and 62,511, respectively, which are reported under the column “Original size”. The number of constraints and variables remaining in the model after the presolve operation and the CPU time to solve the model are reported under the columns “Cons.”, “Vars.” and “CPU”, respectively for the different CACs. These results clearly show the computational inferiority of $CAC3$ compared to the other CACs, as highlighted by its relatively high CPU times. This is mainly due to a weak LP relaxation of the resulting single-level MIP model, besides relatively very few variables and constraints getting eliminated at the presolve stage. Between $CAC1$ and $CAC2$, the former speeds up the computations by a factor of 2 to 4 for $p = 7$ and by a factor of 2 to almost 7 for $p = 10$, as highlighted by their relative CPU times. This gain comes largely from the elimination of a substantially larger proportions of the variables and constraints in the model using $CAC2$ compared to $CAC1$. With the reduced version of $CAC1$, namely $rCAC1$, the presolve operation is further able to eliminate a significant proportion of the variables remaining in the model with $CAC1$, leading to further savings in CPU times. $rCAC2$, on the other hand, is able to further eliminate only constraints but not variables remaining in the model with $CAC2$. Still, use of $rCAC2$ results in savings in CPU time by a factor of 2 to 4 compared to $rCAC1$, which itself provides savings in CPU time by a factor of 1.5 to 2 compared to $CAC1$. However, compared to $CAC2$, its reduced version, namely $rCAC2$, gives only marginal savings in CPU times since post presolve, $CAC2$ and $rCAC2$ have almost the same problem size.

In Table 2, we present additional results from our computational experiments with the best two CACs, namely $CAC2$ and $rCAC2$, for larger instances of the problem derived using the data related to the first 100, and all 200 nodes in the AP dataset. The observations based on the comparison between the computational performance of $CAC2$ and $rCAC2$ largely carry through from Table 1. However, for many of the AP dataset instances reported in Table 2, Cplex runs out of memory.
Table 1: Computational performance of HIP with different CACs using CAB dataset with $|N| = 25$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p r δ</td>
<td>Cons. Vars.</td>
<td>CPU</td>
<td>Cons. Vars.</td>
<td>CPU</td>
<td>Cons. Vars.</td>
<td>CPU</td>
</tr>
<tr>
<td>7 1 0.9</td>
<td>35626 30633</td>
<td>9911 8619</td>
<td>4 1329 845</td>
<td>2 34986 30058</td>
<td>30 6611 8619</td>
<td>3 1329 845</td>
</tr>
<tr>
<td>2</td>
<td>27593 23775</td>
<td>15 10383 4655</td>
<td>5 35094 30058</td>
<td>47 18377 23479</td>
<td>8 10365 4655</td>
<td>4 35094 30058</td>
</tr>
<tr>
<td>3</td>
<td>27593 23775</td>
<td>17 10383 4655</td>
<td>6 35094 30058</td>
<td>53 18377 23479</td>
<td>12 10365 4655</td>
<td>5 35094 30058</td>
</tr>
<tr>
<td>4</td>
<td>27593 23775</td>
<td>22 10383 4655</td>
<td>5 35094 30058</td>
<td>65 18377 23479</td>
<td>16 10365 4655</td>
<td>5 35094 30058</td>
</tr>
<tr>
<td>5</td>
<td>27593 23775</td>
<td>13 10383 4655</td>
<td>3 35094 30058</td>
<td>54 18377 23479</td>
<td>10 10365 4655</td>
<td>2 35094 30058</td>
</tr>
<tr>
<td>1 0.5</td>
<td>19514 17015</td>
<td>7 3713 2405</td>
<td>3 34964 30058</td>
<td>45 12815 17015</td>
<td>5 3713 2405</td>
<td>3 34964 30058</td>
</tr>
<tr>
<td>2</td>
<td>29809 25835</td>
<td>11 11200 6567</td>
<td>7 35094 30058</td>
<td>48 19645 25529</td>
<td>12 10904 6567</td>
<td>6 35094 30058</td>
</tr>
<tr>
<td>3</td>
<td>29809 25835</td>
<td>30 11200 6567</td>
<td>7 35094 30058</td>
<td>51 19645 25529</td>
<td>19 10904 6567</td>
<td>8 35094 30058</td>
</tr>
<tr>
<td>4</td>
<td>29809 25835</td>
<td>9 11200 6567</td>
<td>3 35094 30058</td>
<td>88 19645 25529</td>
<td>10 10904 6567</td>
<td>2 35094 30058</td>
</tr>
<tr>
<td>5</td>
<td>29809 25835</td>
<td>1 0.5 19519 17015</td>
<td>7 3713 2405</td>
<td>3 34964 30058</td>
<td>45 12815 17015</td>
<td>7 3713 2405</td>
</tr>
<tr>
<td>10 1 0.9</td>
<td>69376 62511</td>
<td>28475 25786</td>
<td>14 1743 1102</td>
<td>7 68135 61285</td>
<td>90 17441 25786</td>
<td>9 1743 1102</td>
</tr>
<tr>
<td>2</td>
<td>58805 53083</td>
<td>48 14045 6741</td>
<td>12 68241 61285</td>
<td>163 35957 52397</td>
<td>25 14027 6741</td>
<td>11 68241 61285</td>
</tr>
<tr>
<td>3</td>
<td>58805 53083</td>
<td>61 14045 6741</td>
<td>13 68241 61285</td>
<td>393 35957 52397</td>
<td>34 14027 6741</td>
<td>12 68241 61285</td>
</tr>
<tr>
<td>4</td>
<td>58805 53083</td>
<td>74 14045 6741</td>
<td>16 68241 61285</td>
<td>425 35957 52397</td>
<td>45 14027 6741</td>
<td>13 68241 61285</td>
</tr>
<tr>
<td>5</td>
<td>58805 53083</td>
<td>67 14045 6741</td>
<td>15 68241 61285</td>
<td>520 35957 52397</td>
<td>40 14027 6741</td>
<td>13 68241 61285</td>
</tr>
<tr>
<td>6</td>
<td>58805 53083</td>
<td>82 14045 6741</td>
<td>12 68241 61285</td>
<td>659 35957 52397</td>
<td>44 14027 6741</td>
<td>12 68241 61285</td>
</tr>
<tr>
<td>7</td>
<td>58805 53083</td>
<td>96 14045 6741</td>
<td>14 68241 61285</td>
<td>2504 35957 52397</td>
<td>33 14027 6741</td>
<td>11 68241 61285</td>
</tr>
<tr>
<td>8</td>
<td>58805 53083</td>
<td>49 14045 6741</td>
<td>8 68241 61285</td>
<td>914 35957 52397</td>
<td>40 14027 6741</td>
<td>7 68241 61285</td>
</tr>
<tr>
<td>1 0.5</td>
<td>45559 41340</td>
<td>22 5301 3430</td>
<td>8 68099 61285</td>
<td>83 27655 41340</td>
<td>14 5301 3430</td>
<td>7 68099 61285</td>
</tr>
<tr>
<td>2</td>
<td>62389 56487</td>
<td>64 17091 9885</td>
<td>18 68099 61285</td>
<td>150 37917 55765</td>
<td>33 16807 9885</td>
<td>17 68099 61285</td>
</tr>
<tr>
<td>3</td>
<td>62389 56487</td>
<td>84 17091 9885</td>
<td>22 68241 61285</td>
<td>288 37917 55765</td>
<td>52 16807 9885</td>
<td>19 68241 61285</td>
</tr>
<tr>
<td>4</td>
<td>62389 56487</td>
<td>115 17091 9885</td>
<td>25 68241 61285</td>
<td>358 37917 55765</td>
<td>93 16807 9885</td>
<td>23 68241 61285</td>
</tr>
<tr>
<td>5</td>
<td>62389 56487</td>
<td>104 17091 9885</td>
<td>21 68241 61285</td>
<td>526 37917 55765</td>
<td>66 16807 9885</td>
<td>20 68241 61285</td>
</tr>
<tr>
<td>6</td>
<td>62389 56487</td>
<td>127 17091 9885</td>
<td>19 68241 61285</td>
<td>518 37917 55765</td>
<td>59 16807 9885</td>
<td>20 68241 61285</td>
</tr>
<tr>
<td>7</td>
<td>62389 56487</td>
<td>99 17091 9885</td>
<td>21 68241 61285</td>
<td>468 37917 55765</td>
<td>53 16807 9885</td>
<td>18 68241 61285</td>
</tr>
<tr>
<td>8</td>
<td>62389 56487</td>
<td>31 17091 9885</td>
<td>11 68241 61285</td>
<td>299 37917 55765</td>
<td>19 21055 14681</td>
<td>10 68241 61285</td>
</tr>
</tbody>
</table>
Hence, in the following section, we further exploit the structure of the resulting single-level model with \(CAC2/rCAC2\) by solving it using Benders decomposition. This is also motivated by the fact that Benders decomposition has been successfully applied to solve large instances of hub location problems (de Camargo et al., 2009, 2008).

6 Benders Decomposition

In this section, we present the Benders decomposition algorithm for the single level MIP model for the hub interdiction problem, obtained through the use of \(CAC2/rCAC2\). Using the usual trick used in Benders decomposition of separating the MIP model into a master problem consisting of constraints only involving integer variables, and a sub-problem consisting of rest of the constraints leads to the following master problem:

\[
\begin{align*}
\text{max} & \quad \theta \\
\text{s.t.} & \quad \sum_{k \in H} z_k = p - r \\
& \quad z_k \in \{0, 1\}
\end{align*}
\]

(19) \hspace{1cm} (20) \hspace{1cm} (21)

Solving the master problem provides the values of the \(z\) variables, say \(\bar{z}\), and an upper bound to the original problem. For values of \(z\) fixed at \(\bar{z}\) by the master problem, the sub-problem for the hub interdiction problem with \(rCAC2\) can be written as:

\[
\begin{align*}
\text{max} & \quad \sum_{i \in N} \sum_{j \in N} \sum_{m \in H} \sum_{p \in B} W_{ij} d_{ijkm} X_{ijkm} \\
\text{s.t.} & \quad \sum_{i \in N} \sum_{j \in N} X_{ijkm} = 1 \quad \forall \ k, m \in H \\
& \quad \sum_{i \in N} \sum_{j \in N} X_{ijkm} \leq \bar{z}_k \quad \forall \ i, j \in N, k \in H \\
& \quad \sum_{q \in E'} X_{ijqs} \leq 2 - \bar{z}_k - \bar{z}_m \quad \forall \ i, j \in N, k, m \in H \\
& \quad X_{ijkm} \geq 0 \quad \forall \ i, j \in N, k, m \in H
\end{align*}
\]

(22) \hspace{1cm} (23) \hspace{1cm} (24) \hspace{1cm} (25)

Associating \(\phi_{ij}, \lambda_{ijk}\) and \(\beta_{ijkm}\) as the dual variables with the constraints (23), (24) and (25), respectively, we get the following dual of the sub-problem:

\[
\begin{align*}
\text{min} & \quad \sum_{i \in N} \sum_{j \in N} \phi_{ij} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \lambda_{ijk} \bar{z}_k + \sum_{i \in N} \sum_{j \in N} \sum_{k, m \geq k} \beta_{ijkm}(2 - \bar{z}_k - \bar{z}_m) \\
\text{s.t.} & \quad \sum_{(q,s) \in B_{ijkm}} \beta_{ijqs} + \lambda_{ijk} + \phi_{ij} \geq W_{ij} d_{ijkm} \quad \forall \ i, j \in N, k, m \in H, k = m \\
& \quad \sum_{(q,s) \in B_{ijkm}} \beta_{ijqs} + \lambda_{ijk} + \lambda_{ijm} + \phi_{ij} \geq W_{ij} d_{ijkm} \quad \forall \ i, j \in N, k, m \in H, k \neq m \\
& \quad \beta_{ijkm} \geq 0 \quad \forall \ i, j \in N, (k, m \geq k) \in B_{ijkm} \\
& \quad \phi_{ij}, \lambda_{ijk} \geq 0 \quad \forall \ i, j \in N, k \in H
\end{align*}
\]

(27) \hspace{1cm} (28) \hspace{1cm} (29) \hspace{1cm} (30) \hspace{1cm} (31)
where, \( B^1_{ijkm} = \{(k,m)|d_{ijkm} \leq d_{ijmk} \text{ or } (m,k)|d_{ijmk} < d_{ijkm}\}. \)

\( B^2_{ijkm} = \{(q,s)|d_{ijqs} \leq d_{ijsg} \text{ and } d_{ijkm} > d_{ijqs} \text{ or } (s,q)|d_{ijqs} < d_{ijsg} \text{ and } d_{ijkm} > d_{ijqs} \forall q, s \geq q\}. \)

The dual for the primal sub-problem with CAC2 added instead of rCAC2 can be similarly written as:

\[
\begin{align*}
\min & \sum_{i \in N} \sum_{j \in N} \phi_{ij} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \lambda_{ijk}z_k + \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} \beta_{ijkm}(2 - z_k - z_m) \\
\text{s.t} & \sum_{q:s|d_{ijqs} < d_{ijkm}} \beta_{ijqs} + \lambda_{ijk} + \phi_{ij} \geq W_i d_{ijkm} & \forall i, j \in N, k, m \in H, k = m \\
& \sum_{q:s|d_{ijqs} < d_{ijkm}} \beta_{ijqs} + \lambda_{ijk} + \lambda_{ijm} + \phi_{ij} \geq W_i d_{ijkm} & \forall i, j \in N, k, m \in H \\
& \beta_{ijkm}, \phi_{ij}, \lambda_{ijk} \geq 0 & \forall i, j \in N, k, m \in H. \tag{35}
\end{align*}
\]

In Benders decomposition, the master and the sub problems are solved iteratively, with (optimality and feasibility) cuts derived from the dual of the subproblem at a given iteration added to the master problem in the subsequent iteration. With \( \bar{\phi}_{ij}, \bar{\lambda}_{ijk} \) and \( \bar{\beta}_{ijkm} \) as the optimal duals associated with the constraints (26), (27) and (28) at a given iteration, we get the following optimality cut:

\[
\theta \leq \sum_{i \in N} \sum_{j \in N} \bar{\phi}_{ij} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \bar{\lambda}_{ijk}z_k + \sum_{i \in N} \sum_{j \in N} \sum_{(k,m) \in B^1_{ijkm}} \bar{\beta}_{ijkm}(2 - z_k - z_m). \tag{36}
\]

[36] is the optimality cut when the bi-level hub interdiction model is reduced to a single level model using rCAC2. If instead, CAC2 is used, then the optimality cut added to the master problem is of the following form:

\[
\theta \leq \sum_{i \in N} \sum_{j \in N} \bar{\phi}_{ij} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \bar{\lambda}_{ijk}z_k + \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} \bar{\beta}_{ijkm}(2 - z_k - z_m). \tag{37}
\]

Solving the master problem with the addition of Benders cut gives the upper bound, while the subproblem gives a lower bound to the original problem. The algorithm terminates when the difference between upper and and the best lower bound falls within a pre-specified tolerance for \( \epsilon \) optimality gap.

**Remark:** As discussed in the previous section, rCAC2 does not enjoy much computational advantage over CAC2 when solving the resulting single-level MIP directly using Cplex since both of them result in almost the same model size (exactly the same number of variables and almost the same number of constraints) post-presolve, as evident from Table I. However, when the MIP is decomposed into a master problem consisting of only the binary variables \( z \) and a subproblem consisting of only the continuous variables \( X \), presolve loses its effect. Nonetheless, with Benders decomposition, the dual subproblem (27)-(31) resulting from the use of rCAC2 has fewer variables \( \beta_{ijkm} \) compared to the dual subproblem (32)-(35) resulting from the use of CAC2. With \(|N|\) nodes and \( p \) hubs, the use of rCAC2 and CAC2 give \(|N|^2(p^2 + p)/2 \) and \(|N|^2p^2 \beta_{ijkm} \) variables, respectively. For example, a problem instance with \(|N| = 200\) and \( p = 10 \) results in \( 2,200,000 \) \( \beta_{ijkm} \) variables with the use of rCAC2, which is otherwise \( 4,000,000 \) with the use of CAC2. Thus, rCAC2 results in a reduction in the size of the dual subproblem by a factor of \( 0.5 + (0.5/p) \), which tends to \( 0.5 \) as \( p \) becomes large. We expect Benders decomposition to exploit this reduction in the size of the dual subproblem with rCAC2 to provide a computational advantage over CAC2.
6.1 Computational results

In this section, we present our computational results for larger instances using CAC2 and rCAC2. For each of the instances, we impose a CPU time limit of 36000 seconds (10 hours), and report the optimality gap for instances that are not solved to optimality within the prescribed time limit. Table 2 gives a comparison of the computation performances between Cplex and Benders decomposition with CAC2 and rCAC2. The results clearly highlight that for the instances that Cplex could solve within the prescribed 10 hour time limit, Benders decomposition is computationally faster than Cplex by a factor of 5–11 and 5–14 with the use of CAC2 and rCAC2, respectively. Of the remaining 7 instances that Cplex could not handle owing to memory restriction, Benders decomposition with CAC2 could solve only 2 of them to optimality within the prescribed time limit, while the Benders decomposition with rCAC2 could solve all of them. Further, for the 2 instances that Benders decomposition with CAC2 could solve to optimality within the prescribed time limit, Benders decomposition with rCAC2 is computationally faster by a factor of 7–8. This observation is consistent with the remark we made in the previous section regarding the computational advantage enjoyed by Benders decomposition with rCAC2.

Table 2: Computational performance of HIP with CAC2 & rCAC2, with and without Benders decomposition

<table>
<thead>
<tr>
<th>Parameters</th>
<th></th>
<th>CPLEX</th>
<th></th>
<th>Benders Decomposition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CAC2</td>
<td></td>
<td>rCAC2</td>
</tr>
<tr>
<td>(</td>
<td>N</td>
<td>p</td>
<td>r</td>
<td>(\alpha,\chi,\delta)</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
<td>3.075,2</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>6</td>
<td>3.075,2</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>7</td>
<td>3.075,2</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>8</td>
<td>3.075,2</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>15</td>
<td>5</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>100</td>
<td>15</td>
<td>6</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>100</td>
<td>15</td>
<td>7</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>100</td>
<td>15</td>
<td>8</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>5</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>6</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>7</td>
<td>3.075,2</td>
<td>*</td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>8</td>
<td>3.075,2</td>
<td>*</td>
</tr>
</tbody>
</table>

* indicates out of memory status

7 Extension to Protection problem

A natural extension of the HIP is the hub protection problem (HPP), wherein the defender has an option to fortify/protect only a subset of \(q\) hubs (due to its limited protection resources like budget, etc.) against interdiction. Figure 1 gives a schematic representation of the tri-level Hub protection problem. While protection problems have been widely studied in the context of facility location (Church and Scaparra, 2007b; Scaparra and Church, 2008a,b, 2012; Aksen et al., 2010; Aksen and Aras, 2012; Aksen et al., 2013), the same has not received much attention in the context of hub location. To the best of our knowledge, Lei (2013) is the only paper on hub protection. The author
Figure 1: Hub protection problem as a tri-level MIP

presents a tri-level MIP formulation for HPP, which is reduced to a bi-level MIP using CAC1 (see Section 3.2.2). However, the author does not present any computational results in absence of any proposed solution method. In this section, we also present a tri-level MIP formulation for HPP. However, we reduce the tri-level MIP to bi-level MIP using rCAC2, which has been shown to perform the best among the different versions of CACs discussed in Section 3.2.2. This allows us to solve large instances of HPP using an efficient algorithm (Implicit enumeration + Benders decomposition) as described in the following subsection.

To model HPP, we define a new set of binary variables \( y_k \) which is 1, if the hub \( k \) is protected, 0 otherwise. With these new variables, the HPP can be modeled as a tri-level MIP.

### 7.1 Model Formulation

\[
[HPP_{3L}]: \min_{y_k} Z_1
\]
\[
s.t. \quad \sum_{k \in H} y_k = q \tag{39}
\]
\[
y_k \in \{0, 1\} \tag{40}
\]
\[
Z_1 = \max_{z_k} Z_2 \tag{41}
\]
\[
s.t. \quad \sum_{k \in H} z_k = p - r \tag{42}
\]
\[
y_k \leq z_k \tag{43}
\]
\[
z_k \in \{0, 1\} \tag{44}
\]
\[
Z_2 = \min_{X_{ijkm}} \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} W_{ij} d_{ijkm} X_{ijkm} \tag{45}
\]
\[
s.t. \quad \sum_{k \in H} \sum_{m \in H} X_{ijkm} = 1 \quad \forall \; i, j \in N \tag{46}
\]
\[
\sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} \leq z_k \quad \forall i, j \in N; k \in H \quad (47)
\]
\[
X_{ijkm} \geq 0 \quad \forall i, j \in N; k, m \in H \quad (48)
\]

In this \(HPP_{3L}\), the defender makes the first move, protecting \(q\) hubs from a set of \(p\) hubs \((39)\), followed by the attacker who attacks \(r\) hubs out of the remaining \(p - q\) unprotected hubs \((42)\). For feasibility, \(q + r \leq p\). In the above model, the defender in the upper level problem minimizes the attacker’s second level objective of maximizing the interdiction cost. The lower two levels form the bi-level HIP, as described in Section 3.1.1, with the additional constraint \((43)\), which ensures that a protected hub cannot be attacked. As discussed in Section 3.2.2, the lower bi-level HIP can be reduced to a single level HIP using the CACs. Using the most efficient CAC, namely \(rCAC\), \(HPP\) can be restated as the following bi-level program:

\[
[HPP_{2L}] : \min Z_1
\]
\[
s.t. \sum_{k \in H} y_k = q \quad (50)
\]
\[
y_k \in \{0, 1\} \quad (51)
\]
\[
Z_1 = \max_{z_k} \sum_{i \in N} \sum_{j \in N} \sum_{k \in H} \sum_{m \in H} W_{ij} d_{ijkm} X_{ijkm} \quad (52)
\]
\[
s.t. \sum_{k \in H} \sum_{m \in H} X_{ijkm} = 1 \quad \forall i, j \in N \quad (53)
\]
\[
\sum_{k \in H} z_k = p - r \quad (54)
\]
\[
\sum_{m \in H} X_{ijkm} + \sum_{m \in H/(k)} X_{ijmk} \leq z_k \quad \forall i, j \in N; k \in H \quad (55)
\]
\[
\sum_{q \in E'_{ijkm}} X_{ijqs} \leq 2 - z_k - z_m \quad \forall i, j \in N; k, m \geq k \in H \quad (56)
\]
\[
y_k \leq z_k \quad \forall k \in H \quad (57)
\]
\[
z_k \in \{0, 1\}, \quad X_{ijkm} \geq 0 \quad \forall i, j \in N; k, m \in H \quad (58)
\]

7.2 Solution method

We present a solution method for HPP, which is inspired by the implicit enumeration algorithm for \(r\)-IMF \([Scaparra and Church 2008a]\). The algorithm is based on the proposition that the optimal solution to \(r\)-IMF will necessarily contain at least one of the facilities interdicted in \(r\)-IMP since any other combination of protected facilities will not prevent the worst scenario for the defender.

Implicit enumeration procedure for HPP is described as follows: At the root of the search tree, the algorithm solves an HIP, giving \(r\) interdicted hubs. The root node is then branched into \(r\) children nodes, each corresponding to protection of a hub \(k\) (by setting \(y_k = 1\)) out of the \(r\) interdicted hubs. At each of these \(r\) nodes, it solves a conditional hub interdiction problem (CHIP), which is an HIP with the restriction that the protected hub \(k\) cannot be interdicted (imposed using constraint set \((57)\)). The solution to each of these CHIPs gives \(r\) interdicted hubs. Each of these nodes is in turn branched into \(r\) children nodes, each corresponding to protection of a hub \(k\) (by setting \(y_k = 1\)) out of the \(r\) interdicted hubs, in addition to the hubs protected at its parent node. This procedure is repeated
till the number of protected hubs on the path starting from the root to the current node is \( q \). Any node at which \( q \) hubs are protected is called a leaf node. When each of the paths from the root node terminates in a leaf node, then the node with the lowest objective function value to its corresponding CHIP provides the solution to HPP. At each node in the search tree, HIP/CHIP is reduced to a single level MIP using \( rCAC2 \), which is solved using Benders decomposition.

To summarize the above procedure, we use \( y \) and \( z \) to denote the optimal solution vector to the protection and interdiction variables, respectively. Further, \( \theta \) denotes the optimal objective function value to HPP. Let \( r \) denote the root node of the search tree, and \( S \) denote the set of nodes in the tree to be visited. We define the following two sets associated with each node \( n \):

- \( C_n \): set of candidate hubs to be protected in the subsequent nodes on the subpath starting from node \( n \).
- \( F_n \): set of hubs protected on the path from root to node \( n \).

We use CHIP\((F_n)\) to denote CHIP with the additional restriction that the hubs in \( F_n \) cannot be interdicted. Using the above notation, the above procedure has been described in Algorithm 1.

**Algorithm 1: Implicit enumeration**

1: procedure IMPPLICIT ENUMERATION
2: \( F_r \leftarrow \phi \)
3: \( \tilde{y}_k \leftarrow 0 \) \( \forall k \in H \)
4: Solve CHIP\((F_r)\). \( \hat{z} \leftarrow \{ k \mid z_k = 0 \} \); \( \hat{\theta} \leftarrow \) objective function value of CHIP\((F_r)\)
5: \( \hat{\theta} \leftarrow \hat{\theta}; C_r = \{ k \mid \hat{z}_k = 0 \}; S = \{ r \} \)
6: while \( S \neq \phi \) do
7: select \( n \in S \)
8: while \( C_n \neq \phi \) do
9: Select \( k \in C_n \)
10: \( C_n \leftarrow C_n \setminus \{k\} \)
11: Generate node \( n_1 \) with \( F_{n_1} = F_n \cup \{k\} \)
12: solve CHIP\((F_{n_1})\). \( \hat{z} \leftarrow \{ k \mid \hat{z}_k = 0 \} \); \( \hat{\theta} \leftarrow \) objective function value of CHIP\((F_{n_1})\)
13: if \( |F_{n_1}| = q \) then
14: if \( \hat{\theta} < \hat{\theta} \) then
15: \( \hat{z} \leftarrow \hat{z}; \hat{\theta} \leftarrow \hat{\theta} \)
16: for \( k \in H \) do
17: if \( k \in F_{n_1} \) then
18: \( \tilde{y}_k = 1 \)
19: else \( \tilde{y}_k = 0 \)
20: end if
21: end for
22: end if
23: else \( C_{n_1} = \{ k \mid \hat{z}_k = 0 \} ; S = S \cup \{ n_1 \} \)
24: end if
25: end while
26: end while
27: return \( \hat{\theta}, \tilde{y}, \hat{z} \)
28: end procedure
The above Implicit enumeration procedure solves \( r^0 + r^1 + r^2 + \ldots + r^q = \frac{(r^{(q+1)} - 1)}{(r - 1)} - 1 \) HIP/CHIPs, as opposed \( \binom{p}{q} \) CHIPs in complete enumeration of the set of \( q \) protected hubs.

### 7.2.1 Results for the protection problem

Table 3 provides a comparison of the computational performance of the Implicit enumeration algorithm versus complete enumeration for different instances derived from AP dataset. Computational results show that Implicit enumeration provides a computational saving of 6% - 50% compared to complete enumeration.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Complete Enumeration</th>
<th>Implicit Enumeration</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>N</td>
<td>)</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>

### 8 Conclusion and future research directions

In this paper, we studied hub interdiction problem (HIP) and hub protection problem (HPP), which are formulated as bi-level and tri-level MIPs. For the bi-level HIP, we explored two alternate ways to reduce it to a single level problem. The first approach used the well-known KKT conditions for the lower level problem, while the second approach exploited the structure of the solution to the lower level problem to replace it using closest assignment constraints (CACs). We further studied alternate forms of CACs and their relative computational performances. Our results indicated that the best among our proposed CAC provided a computational advantage (in terms of reduced CPU times) by a factor of 7 times compared to the CAC proposed in the literature. We further provided reduced versions of the alternate CACs, one of which in conjunction with Benders decomposition helped solve large instances of HIP 7-8 times faster than its parent CAC. The computational advantage gained for HIP with the use of CAC and Benders decomposition allowed us to further solve large instances of an otherwise intractable HPP.

The current work opens up a number of exciting possibilities for future research. In this paper, we studied protection of hubs as one approach used by the decision maker to safeguard against interdiction. Yet another approach against interdiction is to consider interdiction possibility at the design of the hub network itself. This, however, will be a much more challenging problem to solve than HPP. Another
interesting extension of HIP and HPP is to incorporate uncertainties in the problem parameters (like demand, etc.), leading to their robust counterparts (Bertsimas and Sim [2004], Ben-Tal and Nemirovski [1999]). All these problems can further be extended to their capacitated versions, wherein the hubs have limited capacities.
References


