8-2016

A Benders Based Rolling Horizon Algorithm for a Dynamic Facility Location Problem

Mohammad Marufuzzaman
Mississippi State University

Ridvan Gedik
University of New Haven, RGedik@newhaven.edu

Mohammad S. Roni
Idaho National Laboratory

Publisher Citation

Marufuzzaman, M., Gedik, R., & Roni, M. S. (2016). A Benders based rolling horizon algorithm for a dynamic facility location problem. Computers & Industrial Engineering, 98, 462-469

Comments

This is the authors' accepted manuscript of the article published in *Computers & Industrial Engineering*. The version of record can be found at http://dx.doi.org/10.1016/j.cie.2016.06.029 .

A Benders Based Rolling Horizon Algorithm for a Dynamic Facility Location Problem

[1]Mohammad Marufuzzaman*, [2]Ridvan Gedik, [3]Mohammad S. Roni

[1]Department of Industrial and Systems Engineering, Mississippi State University, USA

[2]Department of Mechanical and Industrial Engineering, University of New Haven, USA

[3]Biofuels & Renewable Energy Technologies, Idaho National Laboratory, USA

Abstract

This study presents a well-known capacitated dynamic facility location problem (DFLP) that satisfies the customer demand at a minimum cost by determining the time period for opening, closing, or retaining an existing facility in a given location. To solve this challenging $\mathcal{NP}$-hard problem, this paper develops a unique hybrid solution algorithm that combines a rolling horizon algorithm with an accelerated Benders decomposition algorithm. Extensive computational experiments are performed on benchmark test instances to evaluate the hybrid algorithm's efficiency and robustness in solving the DFLP problem. Computational results indicate that the hybrid Benders based rolling horizon algorithm consistently offers high quality feasible solutions in a much shorter computational time period than the stand-alone rolling horizon and accelerated Benders decomposition algorithms in the experimental range.

Keywords: Dynamic facility location problem, Benders decomposition algorithm, rolling horizon heuristics, hybrid Benders based rolling horizon algorithm.

1 Introduction

The problem of locating a set of facilities to serve customers has received extensive attention from researchers, managers, and practitioners due to the problem's presence in almost any supply chain. Therefore, various types of facility location problems have been investigated in order to determine which facilities should be opened, closed or relocated to serve select customers to minimize the total cost [1]. This paper examines a version of the capacitated facility location problem (CFLP) in which facilities are assumed to provide a finite amount of goods to meet time-dependent and deterministic customer demand subject to time-dependent cost parameters in a multi-period planning horizon. This problem is referred to as the capacitated *Dynamic Facility Location Problem* (DFLP)

*Corresponding author. E-mail: maruf@ise.msstate.edu; Tel:(662)325-9220. Fax:(662)325-7012.

[2, 3]. In order to be able to respond to varying demand, the decision maker must determine whether to open new facilities, keep the existing facilities open or closed, or relocate them at any time period. In addition, the portion of customer demand needs to be satisfied by each operating facility must be decided. The ultimate objective is to minimize the total cost, which may include transportation and operating costs, facilities opening and closing expenses, or other costs during all planning periods.

Arabani and Farahani [3] categorize the facility location problem into two main groups based on whether the (re)location decisions vary by time. The static facility location problem is referred to as single-period facility location problem in which the facility location decisions and their parameters are independent of time. Since the dynamic counterpart relaxes this assumption, dynamic model variants are more suitable to reflect the impacts of vital factors that cannot be represented by static models, such as incentives, energy prices, and market growth. Thus, dynamic model variants have many application areas, including, but not limited to, combat logistics [4], electronics logistics [5], and healthcare [6]. Current et al., [7] further apply another classification criteria for the DFLP based on facility (re)location decisions. The *explicitly* DFLP controls the opening and closing of a facility in a planning horizon, whereas the parameters may change over time, but the (re)location decisions can be made only at the beginning of the time horizon in the *implicitly* DFLP. Mirchandani and Odoni [8] study a version of the implicitly DFLP in which the travel times are treated as random variables with known discrete probability distributions. Drezner and Wesolowsky [9] demonstrate an optimal solution method for the single facility location problem with a single (re)location option with known demand of each serving point and a continuous linear function of time. Farahani et al., [10] extend this work by including multiple relocation opportunities and proposing an exact algorithm to make optimal relocation decisions. The implicitly DFLP proposed by Drezner [11] develops a progressive p-median problem that does not consider (re)location of the existing facilities but time periods are known when new facilities are added to the network. The common property of these studies is, although the demand is assumed to be dynamic and deterministic as a function of time, facilities can only be opened at the beginning of the planning period.

This study limits attention to the explicitly DFLP that represents the impacts of time-dependent parameters on time-dependent (re)location decisions. Even in this subgroup of facility location problems, differences exist due to several assumptions, or limitations, on the ways facility capacities can be dynamically adjusted to correspond to the dynamic structure of demand. Thus, some researchers assume that once a facility is located during a time period, it will remain open until the end of the planning horizon [12]. Some others consider the case that opening new facilities or expanding current capacities and closing existing ones can occur throughout the entire planning horizon [13, 14, 15, 16]. Klose and Drexl [17] underline the exponentially increasing complexity of the dynamic models over time. We also show that this problem is $\mathcal{NP}$-hard. Despite these facts, the DFLP has received extensive attention due to recent computational advancements and in the problems applicability to real -life applications. Researchers have presented numerous intelligent solution ways for different versions of this problem.

Jena et al., [18] develop several valid inequalities to strengthen the DFLPs formulations separately with decisions about capacity expansion or reduction and facility closing and reopening. Scott [12] proposes a near optimal dynamic programming approach for the DFLP in which multiple facilities can be located over equally distributed discrete time periods. Roy and Erlenkotter [13] propose an exact dual ascent method embedded in a branch-and-bound search for the uncapacitated DFLP that solves the problem instances within one second and considers 25 facility and 50 customer locations, as well as 10 time periods. Later on, Lim and Kim [14] consider the capacitated facilities for the same problem and develop a Lagrangian relaxation based branch-and-bound approach supported by Gomory cuts. Their technique finds good quality lower bounds by employing a subgradient optimization method. Canel et al., [15] further extend this work by considering multi-commodity items. In the first two stages of their algorithm, a branch-and-bound procedure is adopted to make the facility opening and closing decisions for each time period. At the final stage, the optimal configuration of facilities is identified by dynamic programming. Melo et al., [16] introduce modular capacity concept that enables facilities to exchange capacities. In addition, their capacitated multi-commodity DFLP problem considers inventory activities and external supply of goods. They investigate the complexity of each DFLP attribute by reporting the solution quality of the mathematical models solved by a commercial branch-and-bound solver. Jena et al., [19] study the multi-commodity DFLP with generalized modular capacities in which facility closing, reopening, capacity reductions, and expansions are taken into account. They present a Langrangian based algorithm that finds good quality solutions within reasonable CPU times. Their technique consistently obtains solutions within 4% from the best known lower bound, even for the problem instances the commercial solver fails to report any solution due to memory limitation.

The multi-period international facility location problem (IFLP), introduced to the literature by Canel and Khumawala [20], is a variant of the DFLP and seeks either to minimize the total cost of dynamically opening facilities in domestic/foreign countries or maximize the after-tax profits. Opening new facilities is the only facility related decision in the IFLP. However, the optimal time of the location decisions, the total quantities that need to be produced in each location and the shipment amounts from facilities to customers are taken into account. Canel and Khumawala [20] further develop few mixed integer programs (MIP) for both the capacitated and uncapacitated IFLP, and by solving these problems in a commercial solver, they demonstrate how sensitive the location decisions are for specific problem parameters, such as with/without demand shortages. In a follow-up study, Canel and Khumawala [21]tackle the uncapacitated IFLP with a branch-and-bound algorithm that is shown to be faster than the MIP formulation by a factor of 50 on some problem instances. Finally, a heuristic proposed by Canel and Khumawala [22] demonstrates significant computational time gains for a similar IFLP problem.

Torres-Soto and Uster [2] study two versions of the DFLP. In the first variant, they allow the facility opening and closing decisions throughout each period, whereas the second variant assumes located facilities are open during the entire planning period.

After presenting a MIP for each, they develop only the Benders decomposition algorithm for the second problem and a Benders and a Lagrangian relaxation based algorithm for the first problem. This study presents the same problem as the first DFLP variant in [2]. No assumption is made on the demand structures, and the facility opening/closing decisions can be made during any time period. The major contribution of this study is two-fold. First, it proposes three main solution approaches: (i) a rolling horizon (RH) heuristic, (ii) an accelerated Benders decomposition algorithm, and (iii) a hybrid (RH-Benders) decomposition algorithm. Second, in addition to the largest set of problem instances introduced by Torres-Soto and Uster [2], we introduce larger problem sets and compare both their methods with our novel algorithms in terms of solution quality and time.

The rest of this paper is organized as follows: Section 2 introduces the mathematical model formulation of the DFLP and discusses some basic properties. The proposed solution methods including rolling horizon approximation, accelerated, and hybrid Benders decomposition algorithms are presented in Section 3. A comparative discussion of these algorithms over some benchmark instances from the literature is demonstrated in Section 4. Finally, Section 5 concludes this paper by providing possible future research directions.

2 Problem Formulation

This section introduces the mathematical formulation of the **[DFLP]** that was proposed by Torres-Soto and Uster [2]. Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a complete directed graph where $\mathcal{N}$ denotes the set of nodes and $\mathcal{A}$ denotes the set of arcs. Set $\mathcal{N}$ consists of set of customers $\mathcal{I}$ and set of facilities $\mathcal{J}$ i.e., $\mathcal{N} = \mathcal{I} \cup \mathcal{J}$ and set $\mathcal{A}$ represents the transportation arcs between the facilities to customers. In **[DFLP]**, we allow the facilities to open, close or remain operational in a given time period to meet the customer demand. The ultimate goal is to determine the optimum locations of capacitated facilities in each time period that will satisfy the customer demand at a minimum possible total cost. We note that when $\{V_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}} = \{U_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}} = 0$ and $\{q_j\}_{j\in\mathcal{J}} \to +\infty$, the **[DFLP]** becomes the classical uncapacitated fixed-charge location problem which is known to be an $\mathcal{NP}$-hard problem. Thus, **[DFLP]** is also an $\mathcal{NP}$-hard problem.

The major cost components in **[DFLP]** are the cost related to opening, closing and operating facilities and transportation costs across all time periods. The sets, input parameters, and decision variables used in this study are summarized in Table 1.

The **[DFLP]** can be formulated as follows:

$$\textbf{[DFLP]}\ \mathit{Minimize}\ \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\left(\psi_{jt}Y_{jt} + \eta_{jt}U_{jt} + \mu_{jt}V_{jt} + \sum_{i\in\mathcal{I}} c_{ijt}X_{ijt}\right)$$

Table 1: Description of the sets and parameters

Symbol	Description
Sets	
$\mathcal{I}$	set of customer locations
$\mathcal{J}$	set of facility locations
$\mathcal{T}$	set of time periods
Parameters	
ψ_{jt}	fixed cost of having a facility open in location j of period t
η_{jt}	fixed cost of opening a facility in location j at the beginning of period t
μ_{jt}	fixed cost of closing a facility in location j at the beginning of period t
d_{ij}	distance between customer i and facility j
α	per unit distance per unit demand cost
b_{it}	demand of customer in location i during period t
c_{ijt}	total cost of shipping demand from location i to j in period t; $c_{ijt} = \alpha b_{it} d_{ij}$
q_j	capacity available for a facility at location j
Decision Variables	
Y_{jt}	1 if a facility remain open in location j at the beginning of period t, 0 otherwise
U_{jt}	1 if a new facility is opened in location j at the beginning of period t, 0 otherwise
V_{jt}	1 if an existing facility is closed in location j at the beginning of period t, 0 otherwise
X_{ijt}	fraction of demand shipped from customer i to facility j in period t

Subject to

$$Y_{jt} + V_{jt} = Y_{j,t-1} + U_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{1}$$

$$\sum_{j \in \mathcal{J}} X_{ijt} = 1 \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{2}$$

$$X_{ijt} \leq Y_{jt} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \tag{3}$$

$$\sum_{i \in \mathcal{I}} b_{it} X_{ijt} \leq q_j Y_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{4}$$

$$Y_{jt}, U_{jt}, V_{jt} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \tag{5}$$

$$X_{ijt} \geq 0 \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \tag{6}$$

The objective function minimizes the fixed cost for opening, operating and closing a facility as well as transportation cost between facilities to customers. Constraints (1) ensure the correct assignment of opening and closing the facilities. This set of constraints can be viewed as network flow constraints which guarantee that for a given value $j \in \mathcal{J}$, the polytope $\{(Y_{jt}, V_{jt}, U_{jt}) \in [0,1]^{3|T|} : Y_{j,t-1} + U_{jt} = Y_{jt} + V_{jt}, \forall t \in \mathcal{T}\}$ will provide the integrality property. Therefore, the resulting mechanism will generate a tight linear programming formulation for model **[DFLP]**. Constraints (2) ensure that the demand must be met for each customer. On the other hand, constraints (3) guarantee that the demand can only be fulfilled from open facilities. Constraints (4) make sure that that no facility can supply more than its capacity. Finally, constraints (5) and (6) are the integrality and non-negativity constraints, respectively.

3 Solution Methods

Since the [**DFLP**] is $\mathcal{NP}$-hard, commercial solvers, such as CPLEX, cannot solve large-scale instances of this problem. In this section we propose the following approaches to solve [**DFLP**]: a rolling horizon heuristic, an accelerated Benders decomposition algorithm, and a Benders-based rolling horizon algorithm. The aim is to produce a near optimal solution for [**DFLP**] in a reasonable amount of time.

3.1 Rolling Horizon (RH) Algorithm

In this section, we introduce a heuristic approach proposed by Balasubramanian and Grossman [23] and Kostina et al.[24]. This approach decomposes problem [**DFLP**] into a series of small subproblems where each subproblem includes few consecutive time periods which are drawn from the overall planning horizon. The algorithm terminates when all the subproblems are investigated. The solution of this heuristic provides an upper bound for problem [**DFLP**]. The overall algorithm is shown in **Algorithm 1**.

Let t_0^s denote the starting time period of subproblem s. Let M^s denote the number of time periods comprised in subproblem s. We can set a fixed or variable size of M^s for each subproblem. Each approximate subproblem of the rolling horizon algorithm is denoted by [**DFLP(s)**]. Now, each approximate subproblem is solved by setting the variables as: (i) $\{Y_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}} \in \{0,1\}$, $\{V_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}} \in \{0,1\}$, and $\{U_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}} \in \{0,1\}$ for $t_0^s \leq t \leq t_0^s + M^s$ and (ii) $0 \leq Y_{jt} \leq 1$, $0 \leq V_{jt} \leq 1$, and $0 \leq U_{jt} \leq 1$ for $t > t_0^s + M^s$. Once a subproblem is solved, we fix the values of Y_{jt}, V_{jt}, and U_{jt} for $t < t_0^s$ and update the step size s. The process terminates when all the subproblems are solved. **Figure 1** shows an example of using the rolling horizon approach to solve a three time period problem.

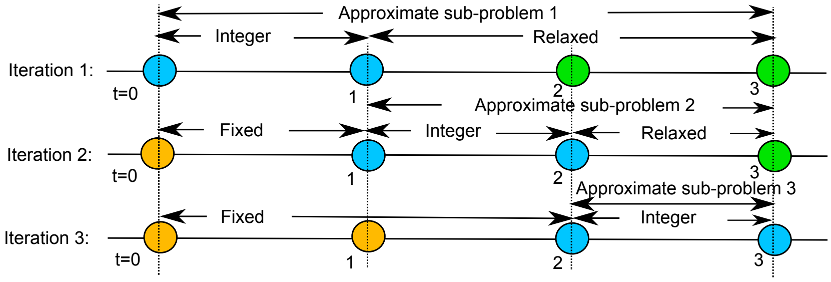

Figure 1: Application of a rolling horizon strategy for a three time period problem

3.2 Benders Decomposition Algorithm

Based on the structure of the model [**DFLP**], we develop an algorithm using the Benders decomposition method [25], which is a well-known partitioning method to solve mixed integer linear programs. Benders decomposition helps separating the original problem into two subproblems: an integer master problem and a linear subproblem. In

Algorithm 1: Rolling Horizon (RH) Heuristic

```
$t_0^s = 0$, $s \leftarrow 1$, $M^s$, terminate $\leftarrow$ false
while (terminate = false) do
    Set:
    $Y_{jt} \in \{0,1\}$, $V_{jt} \in \{0,1\}$ and $U_{jt} \in \{0,1\}$ for $t_0^s \leq t \leq t_0^s + M^s$
    $0 \leq Y_{jt} \leq 1$, $0 \leq V_{jt} \leq 1$ and $0 \leq U_{jt} \leq 1$ for $t > t_0^s + M^s$
    Solve the approximate sub-problem [DFLP(s)] using CPLEX
    if($t_0 > |\mathcal{T}|$) then
        stop $\leftarrow$ true
    end if
    $s \leftarrow s + 1$
    Fixing the values of $Y_{jt}$, $V_{jt}$ and $U_{jt}$ for $t < t_0^s$
end while
```

model [**DFLP**], for fixed values of binary location variables, the resulting model can be decomposed into a linear multi-time period transportation problem. The underlying Benders reformulation for model [**DFLP**] is given below:

$$Minimize \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \left(\psi_{jt} Y_{jt} + \eta_{jt} U_{jt} + \mu_{jt} V_{jt} \right) + [\mathbf{SP}](X|\hat{Y}, \hat{U}, \hat{V})$$

Subject to (1)-(6). $[\mathbf{SP}](X|\hat{Y}, \hat{U}, \hat{V})$ represents the Benders subproblem which is described below.

For given values of the $\mathbb{Y} := \{Y_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}}$, $\mathbb{U} := \{U_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}}$, and $\mathbb{V} := \{V_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}}$ variables satisfying the integrality constraints (5), the model [**DFLP**] reduces to the following primal subproblem involving only the continuous variables $\mathbb{X} := \{X_{ijt}\}_{i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T}}$.

$$[\mathbf{SP}] \ Minimize \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} c_{ijt} X_{ijt}$$

Subject to

$$\sum_{j \in \mathcal{J}} X_{ijt} = 1 \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{7}$$

$$X_{ijt} \leq \hat{Y}_{jt} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \tag{8}$$

$$\sum_{i \in \mathcal{I}} b_{it} X_{ijt} \leq q_j \hat{Y}_{jt} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{9}$$

$$X_{ijt} \geq 0 \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \tag{10}$$

Let $\lambda = \{\lambda_{it} \mid i \in \mathcal{I}, t \in \mathcal{T}\}$, $\delta = \{\delta_{ijt} \geq 0 \mid i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T}\}$ and Let $\gamma = \{\gamma_{jt} \geq 0 \mid j \in \mathcal{J}, t \in \mathcal{T}\}$ be the dual variables associated with constraints (7)-(9), respectively. The dual of the primal subproblem, called the dual subproblem **[DSP]**, can be written as:

$$[\mathbf{DSP}]\ \mathit{Maximize}\ \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}}\lambda_{it} - \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\hat{Y}_{jt}\delta_{ijt} - \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}q_j\hat{Y}_{jt}\gamma_{jt}$$

Subject to

$$\begin{array}{rcll}
\lambda_{it} - \delta_{ijt} - b_{it}\gamma_{jt} & \leq & c_{ijt} & \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \qquad (11) \\
\delta_{ijt}, \gamma_{jt} & \geq & 0 & \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} \qquad (12) \\
\lambda_{it} & \in & \mathbb{R} & \forall i \in \mathcal{I}, t \in \mathcal{T} \qquad (13)
\end{array}$$

Introducing an extra variable θ, the underlying Benders reformulation can be equivalently written as the following Benders master problem **[MP]**:

$$[\mathbf{MP}]\ \mathit{Minimize}\ \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\Big(\psi_{jt}Y_{jt} + \eta_{jt}U_{jt} + \mu_{jt}V_{jt}\Big) + \theta$$

Subject to

$$\begin{array}{rcll}
Y_{jt} + V_{jt} & = & Y_{j,t-1} + U_{jt} & \forall j \in \mathcal{J}, t \in \mathcal{T} \qquad (14) \\
\theta & \geq & \displaystyle\sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}}\lambda_{it} - \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\delta_{ijt}Y_{jt} - \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}q_j\gamma_{jt}Y_{jt} & \forall(\lambda,\delta,\gamma) \in \mathcal{P}_D (15) \\
\displaystyle\sum_{j\in\mathcal{J}}q_jY_{jt} & \geq & \displaystyle\sum_{i\in\mathcal{I}}b_{it} & \forall t \in \mathcal{T} \qquad (16) \\
Y_{jt}, U_{jt}, V_{jt} & \in & \{0,1\} & \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T}(17) \\
\theta & \geq 0 & & \qquad (18)
\end{array}$$

In **[MP]**, constraints (15) are referred to as optimality cut constraints where $\mathcal{P}_D$ is the set of the extreme points in the feasible region of **[DSP]**. Constrains (16) are served as surrogate constrains which are added to the master problem to ensure that enough capacity plants are opened for **[DSP]** to have a feasible solution. The overall Benders decomposition algorithm is described as follows:

Let UB^n and LB^n denote an upper and lower bound of the original problem **[DFLP]** at iteration n. In each iteration, the solution of the master problem (z^n_{MP}) provides a lower bound for the original problem. We now fix the values of the binary variables $\{\hat{Y}^n_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}}$, $\{\hat{U}^n_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}}$, and $\{\hat{V}^n_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}}$, obtained from the master problem **[MP]**, and use these values to solve the dual subproblem [**DSP**]. The solution of the dual subproblem is denoted by z^n_{SUB}. In iteration n, solving the dual subproblem [**DSP**] generates a new extreme point $p \in \mathcal{P}_D$ which is added to the master problem [**MP**] by updating set $\mathcal{P}_D$ as $\mathcal{P}^n_D = \mathcal{P}^{n-1}_D \bigcup p$. Let $z^n_{MAS} = \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}} \Bigg(\psi_{jt} Y_{jt} + \eta_{jt} U_{jt} + \mu_{jt} V_{jt} \Bigg)$. Therefore, the upper bound on the optimal solution value of the [**DFLP**] can be determined as: $UB^n = z^n_{MAS} + z^n_{SUB}$. As in Geoffrion and Graves [26], at the end of each iteration, we check if the gap between the upper bound and lower bound falls below a threshold value ϵ. If this happens, we terminate the algorithm; otherwise, $\mathcal{P}_D$ is updated by adding an optimality cut in the form (15) in [**MP**]. A pseudo-code of the basic Benders decomposition algorithm is provided in **Algorithm 2**.

Algorithm 2: Benders decomposition

```
$UB^n \leftarrow +\infty$, $LB^n \leftarrow -\infty$, $n \leftarrow 1$, $\epsilon$, $\mathcal{P}_D \leftarrow 0$
$terminate \leftarrow$ false
while ($terminate$ = false) do
        Solve [MP] to obtain $\{Y^n_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}}$, $\{U^n_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}}$, $\{V^n_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}}$, $z^n_{MP}$, $z^n_{MAS}$
        if ($z^n_{MP} > LB^n$) then
            $LB^n \leftarrow z^n_{MP}$
        end if
        Set:
            $\hat{Y}^n_{jt} = Y^n_{jt}; \forall j \in \mathcal{J}, t \in \mathcal{T}$
            $\hat{U}^n_{jt} = U^n_{jt}; \forall j \in \mathcal{J}, t \in \mathcal{T}$
            $\hat{V}^n_{jt} = V^n_{jt}; \forall j \in \mathcal{J}, t \in \mathcal{T}$
        Solve [DSP] to obtain $(\lambda_{it}, \delta_{ijt}, \gamma_{jt}) \in \mathcal{P}_D$ and $z^n_{SUB}$
        if ($z^n_{SUB} + z^n_{MAS} < UB^n$) then
            $UB^n \leftarrow z^n_{SUB} + z^n_{MAS}$
        end if
        if ($(UB^n - LB^n)/UB^n \leq \epsilon$) then
            $terminate \leftarrow$ true
        else
            $\mathcal{P}^{n+1}_D = \mathcal{P}^n_D \cup \{\lambda_{it}, \delta_{ijt}, \gamma_{jt}\}$
        end if
        $n \leftarrow n + 1$
end while
```

The only difference between the Benders decomposition algorithm proposed by Torres-Soto and Uster [2] and ours is that we have added integer cuts (see Section 3.3.4) and set branching priorities (see Section 3.3.5) in addition to the enhancement strategies proposed by Torres-Soto and Uster [2].

3.3 Accelerating Benders Decomposition Algorithm

This section presents some accelerating techniques to improve the computational performance of the basic Benders decomposition algorithm in solving model **[DFLP]**.

3.3.1 Multi-cuts:

We observe that **[DSP]** can be further decomposed into $|\mathcal{T}|$ independent dual subproblems, one for each time period $t \in \mathcal{T}$. Therefore, instead of adding one optimality cut we now add $|\mathcal{T}|$ number of cuts in each iteration of the Benders master problem **[MP]**. The information obtained from solving $|\mathcal{T}|$ independent dual subproblem is now used to generate cut (15). Let $\mathcal{P}_D$ be the set of extreme points of the dual polyhedron $\mathcal{P}_{D^t}$ associated with subproblem t. We thus obtain the following revised master problem **[MMP]**:

$$[\mathbf{MMP}]\ Minimize\ \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}\Big(\psi_{jt}Y_{jt}+\eta_{jt}U_{jt}+\mu_{jt}V_{jt}\Big)+\sum_{t\in\mathcal{T}}\theta_t$$

Subject to: (14), (16), (17), and

$$\theta_t \geq \sum_{i\in\mathcal{I}}\lambda_{it}-\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\delta_{ijt}Y_{jt}-\sum_{j\in\mathcal{J}}q_j\gamma_{jt}Y_{jt} \quad \forall t\in\mathcal{T}, (\lambda,\delta,\gamma)\in\mathcal{P}_{D^t} \tag{19}$$

Note that in formulation **[MMP]** we add multiple terms θ_t instead of single θ presented in equation (15). Additionally, notice that we now have the cuts defined for each time period $t \in \mathcal{T}$, with the dual information used to generate the cuts being indexed accordingly. This approach is expected to take fewer number of iterations to reach the optimality gap; however, each iteration is likely to take longer time to solve compared to **[MP]** [27].

3.3.2 Pareto-optimality cuts:

One way to improve the convergence of the Benders decomposition algorithm is to construct stronger, non-dominated cuts, commonly named as *pareto-optimality cuts* [28]. From the context of our problem, we can say that a pareto-optimal cut is generated when the cut produced from an extreme point $(\lambda^1,\delta^1,\gamma^1)$ dominates the cut produced from another extreme point $(\lambda^2,\delta^2,\gamma^2)$, i.e.,

$$\sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}}\lambda^1_{it}-\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}Y_{jt}\delta^1_{ijt}-\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}q_jY_{jt}\gamma^1_{jt} \geq \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}}\lambda^2_{it}-\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}Y_{jt}\delta^2_{ijt}-\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}}q_jY_{jt}\gamma^2_{jt}$$

with strict inequality for at least one point $\{Y_{jt}\}_{j\in\mathcal{J},t\in\mathcal{T}} \in \mathbb{Y}$. In this study, we have used the following subproblem independent pareto-optimal cuts proposed by Papadakos [29]. We refer to this subproblem as **[DSP(MMW)]**. Let $\mathbb{Y}^{LP}$ be the polyhedron defined by (14), (16), and $0 \leq \{Y_{jt}\}_{\forall j \in \mathcal{J}, t \in \mathcal{T}} \leq 1$. Let $ri(\mathbb{Y}^{LP})$ denote the relative interior of $\mathbb{Y}^{LP}$. A pareto-optimal cut can be obtained by solving the following subproblem where $Y^0_{jt} \in ri(\mathbb{Y}^{LP})$; $\forall j \in \mathcal{J}, t \in \mathcal{T}$.

$$[\mathbf{DSP(MMW)}] \; \mathit{Maximize} \; \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \lambda_{it} - \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} Y^0_{jt} \delta_{ijt} - \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} q_j Y^0_{jt} \gamma_{jt}$$

Subject to

$$\begin{aligned} \lambda_{it} - \delta_{ijt} - b_{it}\gamma_{jt} &\leq c_{ijt} && \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} && (20) \\ \delta_{ijt}, \gamma_{jt} &\geq 0 && \forall i \in \mathcal{I}, j \in \mathcal{J}, t \in \mathcal{T} && (21) \\ \lambda_{it} &\in \mathbb{R} && \forall i \in \mathcal{I}, t \in \mathcal{T} && (22) \end{aligned}$$

In this formulation Y^0_{jt} are core points which can be updated as follows: $Y^0_{jt} = \tau Y^0_{jt} + (1-\tau)\hat{Y}_{jt}$; $\forall j \in \mathcal{J}, t \in \mathcal{T}$. $\{\hat{Y}_{jt}\}_{j \in \mathcal{J}, t \in \mathcal{T}}$ is obtained from solution of the current master problem. Experimental results indicate that setting $\tau = 0.5$ provides the best empirical results. We note that the auxiliary subproblem [**DSP(MMW)**] is independent from the solutions of the dual subproblem [**DSP**] which helps the Benders master problem to be one step closer to the optimal solution from the very first iteration [29].

3.3.3 Knapsack Inequalities:

Santoso et al. [30] show that when a good upper bound is available from the Benders decomposition algorithm, then adding knapsack inequalities of the following forms will help the commercial solvers such as CPLEX to derive a varieties of valid inequalities from it. This will speed up the branch and bound process of the solver and eventually will expedite the convergence of the Benders decomposition algorithm. Let UB^n and LB^n denote the best upper and lower bound obtained so far. Therefore, the following valid inequalities are added to the master problem [**MP**] in iteration $n+1$:

$$LB^n \leq \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \Big(\psi_{jt} Y_{jt} + \eta_{jt} U_{jt} + \mu_{jt} V_{jt} \Big) + \theta \qquad (23)$$

$$UB^n \geq \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \Big(\psi_{jt} - \delta_{ijt} - q_j \gamma_{jt} \Big) Y_{jt} + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \eta_{jt} U_{jt} + \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \mu_{jt} V_{jt} + \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \lambda_{it} \qquad (24)$$

3.3.4 Integer Cuts:

In the earlier stage of the the Benders decomposition algorithm, the master problem often produces same values for some of the integer variables over the iterations. This does not help the convergence of the Benders decomposition algorithm and at the same time increases the running time of the overall algorithm. To reduce the search space and expedite the running time of the overall algorithm, the following integer cut is added in each iteration of the Benders master problem [31]. Let $\mathcal{Y}_1^n = \{(j,t)|\hat{Y}_{jt}^n = 1, \forall j \in \mathcal{J}, t \in \mathcal{T}\}$ where $\hat{Y}_{jt}^n$ for $j \in \mathcal{J}, t \in \mathcal{T}$ be the solutions obtained from solving the master problem in iteration n. We add the following constraints to the master problem [**MP**] in iteration $n+1$:

$$\sum_{(j,t)\in\mathcal{Y}_1^n} (1 - Y_{jt}) + \sum_{(j,t)\notin\mathcal{Y}_1^n} Y_{jt} \quad \geq \quad 1 \tag{25}$$

3.3.5 Heuristics Improvements:

Obtaining Good Solutions before Convergence: In the initial stage of the Benders decomposition algorithm, the master problem typically produces low-quality solutions. The process continues until sufficient information from subproblem is passed to the master problem via constraints (15). Additionally, the master problem is an integer problem for which generating an optimal solution even for a moderate size network problem, is a challenging task. In order to alleviate this problem, we initially set a large optimality gap which is gradually reduced as the algorithm progresses. For instance, initially an optimality gap is set at 5%, which is reduced to 1% when the gap between the upper and lower bound of the Benders decomposition algorithm falls below 10%.

Setting Branching Priorities: We set branching priorities explicitly to help CPLEX decides the order in which the solver branch on variables. Our numerical analysis indicates that branching on variables Y_{jt} first, followed by U_{jt} and V_{jt} save some computational time in solving the Benders master problem [**MP**].

3.4 The Hybrid Solution Algorithm

This hybrid solution algorithm ([**RH-Benders**]) combines rolling horizon algorithm with the accelerated Benders decomposition algorithm. In this approach, the accelerated Benders decomposition algorithm is used to solve the subproblems obtained by the rolling horizon algorithm. This hybrid solution algorithm is particularly useful when the network size is sufficiently large (i.e., a large set of $|\mathcal{J}|$ and $|\mathcal{T}|$) and CPLEX finds it difficult to solve the first few subproblems of the rolling horizon algorithm. Our computational results indicate that as soon as the first few subproblems of the rolling horizon algorithm are solved (s^n), the remaining subproblems can be tackled fairly by CPLEX in a reasonable amount of time. This motivates us to apply the accelerated Benders decomposition algorithm (discussed in Section 3.3) to solve the first few subproblems (i.e.,$s^n = 3$) of the rolling horizon algorithm ($s \leq s^n$), and solve the remaining subproblems ($s > s^n$) using CPLEX. The algorithm terminates when all the subproblems are investigated. Note that, the aim of this approach is to provide a high quality feasible solution for model [**DFLP**] in a reasonable amount of time. The overall algorithm is shown in **Algorithm 3**.

Algorithm 3: The Hybrid Solution Algorithm (RH-Benders)

```
$t_0^s = 0$, $s \leftarrow 1$, $M^s$, $s^n \leftarrow 3$, $terminate \leftarrow$ false
while ($terminate$ = false) do
    if ($s \leq s^n$) then
        Set:
        $Y_{jt} \in \{0,1\}$, $V_{jt} \in \{0,1\}$ and $U_{jt} \in \{0,1\}$ for $t_0^s \leq t \leq t_0^s + M^s$
        $0 \leq Y_{jt} \leq 1$, $0 \leq V_{jt} \leq 1$ and $0 \leq U_{jt} \leq 1$ for $t > t_0^s + M^s$
        Use Algorithm 2 to solve the approximate sub-problem [DFLP(s)]
    end if
    if ($s > s^n$) then
        Set:
        $Y_{jt} \in \{0,1\}$, $V_{jt} \in \{0,1\}$ and $U_{jt} \in \{0,1\}$ for $t_0^s \leq t \leq t_0^s + M^s$
        $0 \leq Y_{jt} \leq 1$, $0 \leq V_{jt} \leq 1$ and $0 \leq U_{jt} \leq 1$ for $t > t_0^s + M^s$
        Use CPLEX to solve the approximate sub-problem [DFLP(s)]
    end if
        if($t_0 > |\mathcal{T}|$) then
            $stop \leftarrow$ true
        end if
        $s \leftarrow s + 1$
        Fixing the values of $Y_{jt}$, $V_{jt}$ and $U_{jt}$ for $t < t_0^s$
end while
```

4 Computational Experiments

This section presents a comprehensive analysis of the solution algorithms introduced in the previous section for the DFLP. All the algorithms, including the one developed by Torres-Soto and Uster [2], are coded in GAMS 24.2.1 [32] and executed on a desktop computer with Intel Core i7 3.50 GHz processor and 32.0 GB RAM. The optimization solver used is ILOG CPLEX 12.6.

Table 2 summarizes the characteristics of the input parameters used in the problem instances, referred to as *cases*. The problem instances used in the computational experiments are generated by following the same procedure as described in Torres-Soto and Uster [2]. Accordingly, the total demand of customers during the time horizon follows one of the three patterns: (i) increasing, (ii) decreasing, or (iii) steady. They use four values for the number of customer and facility locations $|\mathcal{I}| = |\mathcal{J}| = \{50, 100, 150, 200\}$, and two values for the number of periods in the planning horizon, $|\mathcal{T}| = \{5, 10\}$. We add three different values to the list of values for the number of customer and facility locations ($|\mathcal{I}| = |\mathcal{J}| = \{50, 100, 150, 200\} \cup \{250, 300, 350\}$) to create more challenging problem instances. The corresponding number of binary and continuous decision variables in each problem instance are also given in Table 2. The cost parameters considered in this model are assumed to be computed in terms of their present values. The fixed cost of operating a facility (ψ_{jt}) is generated randomly from a discrete uniform distribution $U[\overline{u}, \underline{u}]$, where, $0 < \underline{u} < \overline{u}$. Let $\theta = (\overline{u} + \underline{u})/2$. The fixed cost of opening a facility (η_{jt}) is generated randomly from a uniform distribution $U[0.75\theta, 0.85\theta]$, like the fixed cost of closing a facility (μ_{jt}) is generated randomly from a uniform distribution $U[0.10\theta, 0.15\theta]$. We assume that no facilities operate at the beginning of the first period i.e., $Y_{j0} = 0; \forall j \in \mathcal{J}$. The capacity of the facilities (q_j) at location $j \in \mathcal{J}$, the demand of customer (b_{it}) in location $i \in \mathcal{I}$, and the distance (d_{ij}) between customer $i \in \mathcal{I}$ and facility $j \in \mathcal{J}$ can be obtained from `http://ise.tamu.edu/LNS/dcflp-data.html`. For all problem instances, $\alpha = 1$.

Table 2: Problem size of the test instances

Case	$\|\mathcal{I}\|$	$\|\mathcal{J}\|$	$\|\mathcal{T}\|$	Binary Variables	Continuous Variables	No. of Constraints
1	50	50	5	750	12,500	13,251
2	50	50	10	1,500	25,000	26,501
3	100	100	5	1,500	50,000	51,501
4	100	100	10	3,000	100,000	103,001
5	150	150	5	2,250	112,500	114,751
6	150	150	10	4,500	225,000	229,501
7	200	200	5	3,000	200,000	203,001
8	200	200	10	6,000	400,000	406,001
9	250	250	5	3,750	312,500	316,251
10	250	250	10	7,500	625,000	632,501
11	300	300	5	4,500	450,000	454,501
12	300	300	10	9,000	900,000	909,001
13	350	350	5	5,250	612,500	617,751
14	350	350	10	10,500	1,225,000	1,235,501

Tables 3, 4, and 5 assess the performance of the three proposed algorithms for each demand pattern discussed above. In addition, the performance of CPLEX and the Benders algorithm proposed in Torres-Soto and Uster [2] is reported. For each pattern, results include the percent optimality gap, the solution time in seconds, and the number of iterations applied to the solution techniques. We use one of the following criteria to terminate the algorithms: (i) the optimality gap between the upper (UB) and lower bound (LB) falls below a threshold value ϵ, i.e., $\epsilon = |UB - LB|/UB = 0.001$; (ii) the maximum time limit is reached (10,800 CPU seconds); or (iii) the maximum number of iteration is reached (Iter = 500).

From the results presented in Tables 3, 4, and 5, both the three algorithms introduced here and the Benders decomposition algorithm in Torres-Soto and Uster [2] perform better than the branch-and-cut procedure of CPLEX in terms of solution time and percent optimality gap. CPLEX cannot terminate with a solution within the 1% optimality gap for the problem instances with $|\mathcal{I}| = |\mathcal{J}| > 50$ in 10,800 seconds time limit. The only exception is case 3 in Table 4. Even for the smaller cases 1 and 2, CPLEX is outperformed, in terms of solution time, by the solution methods presented here and the Benders decomposition algorithm in Torres-Soto and Uster [2]. The accelerated Benders algorithm developed in this study and its counterpart in Torres-Soto and Uster [2] terminate with a near optimal solution ($< 0.1\%$) for the first nine problem cases.

Table 3: Computation results for [**DFLP**] under steady demand

	CPLEX		Torres-Soto & Uster [2]			Accelerated Benders			RH Algorithm		RH-Benders	
	Gap	CPU	Gap	CPU	Iter	Gap	CPU	Iter	Gap	CPU	Gap	CPU
Case	(%)	(sec)	(%)	(sec)		(%)	(sec)		(%)	(sec)	(%)	(sec)
1	0.09	1,031.2	0.08	13.8	14	0.09	10.5	10	0.09	31.3	0.12	7.3
2	0.09	7,203.3	0.09	19.6	12	0.09	23.9	12	0.14	484.1	0.16	12.9
3	1.31	10,800.0	0.08	123.6	28	0.07	109.5	18	0.26	1,057.7	0.21	82.1
4	3.33	10,800.0	0.08	289.2	23	0.09	231.4	16	1.98	2,432.8	0.11	122.6
5	1.72	10,800.0	0.09	414.4	25	0.08	311.8	21	1.05	4,524.8	0.19	110.9
6	2.96	10,800.0	0.07	794.4	26	0.08	427.4	20	1.06	5,527.1	0.20	206.7
7	4.76	10,800.0	0.09	1,423.8	32	0.07	945.3	25	2.14	9,686.6	0.18	573.4
8	*n.a.*[a]	10,800.0	0.09	1,966.7	34	0.08	1,404.4	21	8.78	10,742.1	0.17	701.7
9	*n.a.*[a]	10,800.0	0.08	5,546.8	76	0.09	4,881.7	69	6.64	10,800.0	0.21	1,913.5
10	*n.a.*[a]	10,800.0	4.48	10,800.0	110	1.69	10,800.0	102	12.29	10,800.0	0.34	3,104.8
11	*n.a.*[a]	10,800.0	1.21	10,800.0	144	0.84	10,800.0	134	8.89	10,800.0	0.26	2,411.7
12	*n.a.*[a]	10,800.0	11.74	10,800.0	94	4.52	10,800.0	88	*n.a.*[a]	10,800.0	0.31	3,577.8
13	*n.a.*[a]	10,800.0	5.22	10,800.0	89	1.51	10,800.0	77	9.29	10,800.0	0.61	3,004.1
14	*n.a.*[a]	10,800.0	16.64	10,800.0	62	6.21	10,800.0	54	*n.a.*[a]	10,800.0	0.55	4,112.8

[a] unable to find an integer feasible solution within the time limit

On the other hand, for the same first nine problem cases, the rolling horizon Benders [**RH-Benders**] algorithm obtains solutions faster than these two Benders variants with slightly worse optimality gaps ($\leq 0.34\%$). The rolling horizon heuristic [**RH Algorithm**] has inferior solution quality and time compared to the Benders algorithms. Yet, for all of the problem cases, [**RH Algorithm**] provides better solutions than CPLEX in terms of both average optimality gap and solution time.

Table 4: Computation results for [**DFLP**] under increasing demand

	CPLEX		Torres-Soto & Uster [2]			Accelerated Benders			RH Algorithm		RH-Benders	
Case	Gap	CPU	Gap	CPU	Iter	Gap	CPU	Iter	Gap	CPU	Gap	CPU
	(%)	(sec)	(%)	(sec)		(%)	(sec)		(%)	(sec)	(%)	(sec)
1	0.09	620.9	0.06	19.7	11	0.08	20.8	10	0.27	247.1	0.14	22.1
2	0.01	8,552.5	0.09	26.4	14	0.07	25.9	10	0.18	1,221.8	0.16	26.2
3	0.09	3,936.6	0.07	301.7	32	0.04	279.5	24	0.12	1,427.6	0.17	179.4
4	1.67	10,800.0	0.09	394.2	34	0.07	351.7	28	0.72	7,598.7	0.20	201.5
5	1.00	10,800.0	0.08	822.6	31	0.08	733.3	26	0.51	7,516.5	0.13	312.6
6	2.61	10,800.0	0.09	1,101.9	34	0.09	916.4	28	1.81	7,622.8	0.22	564.3
7	0.73	10,800.0	0.08	1,962.1	47	0.09	1,729.3	37	0.33	8,035.9	0.24	774.8
8	88.16	10,800.0	0.08	2,338.2	44	0.08	2,091.7	35	14.69	10,689.2	0.21	1,089.2
9	*n.a.*[a]	10,800.0	0.09	8,229.6	121	0.09	6,994.1	113	8.86	10,800.0	0.32	1,820.4
10	*n.a.*[a]	10,800.0	3.38	10,800.0	114	1.15	10,800.0	99	16.47	10,800.0	0.38	2,934.5
11	*n.a.*[a]	10,800.0	0.94	10,800.0	147	0.36	10,800.0	132	11.54	10,800.0	0.36	2,422.1
12	*n.a.*[a]	10,800.0	7.93	10,800.0	97	3.34	10,800.0	83	17.78	10,800.0	0.44	3,689.7
13	*n.a.*[a]	10,800.0	4.62	10,800.0	85	2.26	10,800.0	81	9.17	10,800.0	0.84	3,157.4
14	*n.a.*[a]	10,800.0	14.25	10,800.0	61	8.11	10,800.0	59	*n.a.*[a]	10,800.0	0.67	4,058.5

[a] unable to find an integer feasible solution within the time limit

For the larger problem cases (10-14) and all demand patterns in Tables 3, 4, and 5, CPLEX is unable to report an integer feasible solution within the time limit. For the same set of problem cases, the rolling horizon heuristic [**RH Algorithm**], the accelerated Benders algorithm and its counterpart in Torres-Soto and Uster [2] reach the time limit before obtaining an ϵ-optimal solution. Among them, the accelerated Benders achieves the smallest optimality gap for each problem case (10-14) with fewer number of iterations. On the other hand, in terms of solution time, the hybrid Benders decomposition algorithm outperforms all other solution techniques by a factor of two in every problem case. In return, the worst case optimality gap of the hybrid algorithm is only 0.72% across all problem cases.

Table 5: Computation results for **[DFLP]** under decreasing demand

	CPLEX		Torres-Soto & Uster [2]			Accelerated Benders			RH Algorithm		RH-Benders	
Case	Gap	CPU	Gap	CPU	Iter	Gap	CPU	Iter	Gap	CPU	Gap	CPU
	(%)	(sec)	(%)	(sec)		(%)	(sec)		(%)	(sec)	(%)	(sec)
1	0.09	925.4	0.08	24.2	19	0.09	16.7	15	0.27	179.7	0.24	18.8
2	0.09	1,880.2	0.09	26.4	17	0.08	17.4	11	0.22	432.4	0.23	22.3
3	1.64	10,800.0	0.09	144.8	28	0.06	112.7	20	0.19	996.2	0.21	74.6
4	2.61	10,800.0	0.06	151.7	29	0.09	120.9	18	0.20	1,109.4	0.17	89.7
5	2.89	10,800.0	0.08	722.6	33	0.09	609.7	26	0.88	7,379.5	0.17	279.5
6	4.00	10,800.0	0.07	765.8	31	0.08	710.6	25	1.55	7,422.1	0.34	322.1
7	3.41	10,800.0	0.08	1,421.8	35	0.04	1,173.6	26	2.01	7,826.8	0.31	576.8
8	3.76	10,800.0	0.09	2,438.7	42	0.07	1,967.1	33	2.55	8,312.1	0.28	962.1
9	26.67	10,800.0	0.09	7,142.4	84	0.09	6,127.5	71	4.16	10,800.0	0.32	1,801.5
10	*n.a.*[a]	10,800.0	5.12	10,800.0	116	2.27	10,800.0	103	8.66	10,800.0	0.40	3,172.7
11	*n.a.*[a]	10,800.0	1.22	10,800.0	142	0.83	10,800.0	129	5.84	10,800.0	0.31	2,600.2
12	*n.a.*[a]	10,800.0	8.43	10,800.0	101	3.19	10,800.0	84	15.26	10,800.0	0.48	3,477.3
13	*n.a.*[a]	10,800.0	7.57	10,800.0	88	2.21	10,800.0	76	7.24	10,800.0	0.65	3,215.6
14	*n.a.*[a]	10,800.0	17.44	10,800.0	62	9.16	10,800.0	57	*n.a.*[a]	10,800.0	0.72	3,882.5

[a] unable to find an integer feasible solution within the time limit

5 Conclusions

This paper presents two novel Benders decomposition algorithms and one rolling horizon heuristics to efficiently solve the DFLP in which facility (re)location decisions are made for each time period across the entire planning horizon. In order to assess the performance of these techniques, the solution quality and time of the algorithms developed here are compared with both CPLEX and the Benders algorithm developed by Torres-Soto and Uster [2] over their expanded test cases. Careful analysis of the results shows that both the accelerated Benders algorithm presented in this paper and its counterpart in Torres-Soto and Uster [2] provide better quality solutions than the other algorithms for the smaller problem cases (1-9). However, our hybrid Benders algorithm **[RH-Benders]** obtains near optimal solutions for all problem cases in much shorter computational time. Moreover, the rolling horizon algorithm developed in this paper is demonstrated to be more efficient and effective than CPLEX for almost every problem case, although it is outperformed by all three Benders algorithms in terms of average solution time and optimality gap.

Considering the computational gains via our hybrid Benders algorithm, further research should be directed toward the development of another hybrid Benders algorithm for the DFLP that considers multi-commodities and stochastic demand arrivals. Multi-echelon problems that include inventory control policies can also be embedded into this future application. Another extension of this work is to model and solve a dynamic facility location problem that accounts for customer preferences, market movements, competition, and other dynamics of supply chain management.

References

1. Melo M.T., Nickel S., Saldanha-da-Gama F., Facility location and supply chain management - A review., European Journal of Operational Research 196 (2) (2009) 401 – 412.

2. Torres-Soto J.E., Uster H., Dynamic-demand capacitated facility location problems with and without relocation., International Journal of Production Research 49 (13) (2011) 3979–4005.

3. Arabani A.B., Farahani R.Z., Facility location dynamics: An overview of classifications and applications., Computers & Industrial Engineering 62 (1) (2012) 408–420.

4. Gue K.R., A dynamic distribution model for combat logistics., Computers & Operations Research 30 (3) (2003) 367–381.

5. Manzini R., Gebennini E., Optimization models for the dynamic facility location and allocation problem., International Journal of Production Research 46 (8) (2008) 2061–2086.

6. Ghaderi A., Jabalameli M.S., Modeling the budget-constrained dynamic uncapacitated facility locationnetwork design problem and solving it via two efficient heuristics: A case study of health care., Mathematical and Computer Modelling 57 (3-4) (2013) 382–400.

7. Current J., Ratick S., ReVelle C., Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach., European Journal of Operational Research 110 (3) (1998) 597–609.

8. Mirchandani P.B., Odoni A.R., Locations of Medians on Stochastic Networks., Transportation Science 13 (2) (1979) 85–97.

9. Drezner Z., Wesolowsky G.O., Facility location when demand is time dependent., Naval Research Logistics 38 (5) (1991) 763–777.

10. Farahani R., Drezner Z., Asgari N., Single facility location and relocation problem with time dependent weights and discrete planning horizon., Annals of Operations Research 167 (1) (2009) 353–368.

11. Drezner Z., Dynamic facility location: The progressive p-median problem., Location Science 3 (1) (1995) 1–7.

12. Scott A.J., Dynamic location-allocation systems: some basic planning strategies., Environment and Planning 3 (1) (1971) 73–82.

13. Roy T.J.V., Erlenkotter D., A dual-based procedure for dynamic facility location., Management Science 28 (10) (1982) 1091–1105.

14. Lim S., Kim Y., An integrated approach to dynamic plant location and capacity planning., Journal of the Operational Research society 50 (12) (1999) 1205–1216.

15. Canel C., Khumawala B.M., Law J., Loh A., An algorithm for the capacitated, multi-commodity multi-period facility location problem., Computers & Operations Research 28 (5) (2001) 411–427.

16. Melo M.T., Nickel S., da Gama F.S., Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning., Computers & Operations Research 33 (1) (2006) 181–208.

17. Klose A., Drexl A., Facility location models for distribution system design., European Journal of Operational Research 162 (1) (2005) 4–29.

18. Jena S.D., Cordeau J.F., Gendron B., Dynamic facility location with generalized modular capacities, Transportation Science 49 (3) (2015) 484–499.

19. Jena S.D., Cordeau J.F., Gendron B., Lagrangian heuristics for large-scale dynamic facility location with generalized modular capacities., Tech. rep., CIRRELT-2014-21, Montreal (2014).

20. Canel C., Khumawala B.M., A mixed-integer programming approach for the international facilities location problem., International Journal of Operations & Production Management 16 (4) (1996) 49–68.

21. Canel C., Khumawala B.M., Multi-period international facilities location: An algorithm and application., International Journal of Production Research 35 (7) (1997) 1891–1910.

22. Canel C., Khumawala B.M., International facilities location: A heuristic procedure for the dynamic uncapacitated problem., International Journal of Production Research 39 (17) (2001) 3975–4000.

23. Balasubramanian J., Grossmann I., Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty., Industrial & Engineering Chemistry Research 43 (14) (2004) 3695–3713.

24. Kostina A.M., Guillen-Gosalbeza G., Meleb F.D., Bagajewiczc M.J., Jimeneza L., A novel rolling horizon strategy for the strategic planning of supply chains. Application to the sugar cane industry of Argentina., Computers & Chemical Engineering 35 (2011) 2540–2563.

25. Benders J.F., Partitioning procedures for solving mixed-variables programming problems., Numerische Mathematik 4 (1962) 238–252.

26. A. M. Geoffrion, G. W. Graves, Multicommodity distribution system design by benders decomposition, Management science 20 (5) (1974) 822–844.

27. Birge J.R., Louveaux F.V., A multicut algorithm for two-stage stochastic linear programs., European Journal of Operational Research 34 (3) (1988) 384–392.

28. Magnanti T.L., Wong R.T., Acclerating Benders Decomposition: Algorithmic enhancement and model selection criteria., Operations Research 29 (1981) 464–484.

29. Papadakos N., Practical enhancements to the Magnanti-Wong method., Operations Research Letters 36 (2008) 444–449.

30. Santoso T., Ahmed S., Goetschalckx M., Shapiro A., A stochastic programming approach for supply chain network design under uncertainty., European Journal of Operational Research 167 (2005) 96–115.

31. Fischetti M., Lodi A., Local branching., Mathematical Programming 98 (2003) 23–47.

32. General Algebraic Modeling System (GAMS), available from: `http://www.gams.com/` (2013).