Global Optimization Algorithm for Multi-period Design and Planning of Centralized and Distributed Manufacturing Networks

Cristiana L. Lara, David E. Bernal, Can Li, Ignacio E. Grossmann

Department of Chemical Engineering, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15232, USA

Abstract

This paper addresses the design and planning of manufacturing networks considering the option of centralized and distributed facilities, taking into account the potential trade-offs between investments and transportation. The problem is formulated as an extension of the Capacitated Multi-facility Weber Problem, which involves the selection of which facilities to build in each time period, and their location in the continuous two-dimensional space, in order to meet demand and minimize costs. The model is a multi-period GDP, reformulated as a nonconvex MINLP. We propose an accelerated version of the Bilevel Decomposition by Lara et al. [1] that finds stronger bounds in the decomposition scheme. We benchmark the performance of our algorithm against the original Bilevel Decomposition and commercial global solvers and show that our approach outperforms the others in all instances tested. Additionally, we illustrate the applicability of the proposed model and solution framework with a biomass supply chain case study.

Keywords: Distributed manufacturing, Weber problem, Global optimization

1. Introduction

Advances in technology have led to the rethinking of traditional manufacturing. In the past few decades, public and private initiatives have been sponsoring research on smaller-scale and cleaner manufacturing processes. The F3 Factory Project was launched in 2009 to enhance the competitiveness of the European chemical industry by promoting modular continuous plants with small and medium scale production [2]. Likewise, the U.S. Advanced Manufacturing National Program Office (AMNPO) has brought together corporations, federal agencies, and universities to advance manufacturing technologies by investing in areas such as High Efficiency Modular Chemical Processes (HEMCP), Additive Manufacturing (3D printing), and Process Intensification [3].

Modular plants consist of manufacturing sites with their major equipment pieces in standardized modules instead of having customized site-specific design [4]. Their potential advantages include higher flexibility, faster time-to-market, and improved safety [5, 6, 7]. This concept is not new [8], but combined with distributed manufacturing and the recent advances in process intensification [9, 10], it can be a viable and beneficial alternative to traditional large-scale manufacturing.

The concept of Distributed Manufacturing - a geographically distributed network of facilities - has arisen as a promising option for supply-chain networks in which the transportation costs and infrastructure are the main bottlenecks (e.g., biomass [11, 12, 13], shale gas [14, 15], and electric power). However, despite the potential advantages of having distributed facilities, conventional large-scale centralized manufacturing can be more cost-effective due to economies of scale. Therefore, there is a need for a general optimization framework that can support the selection of centralized and distributed facilities taking into account the potential trade-offs [16].

We address the design and planning of manufacturing networks considering the selection and location of centralized and/or distributed facilities. This multi-period problem involves the selection of which facilities to build in each time period, their location in the continuous 2-dimensional space, and how to link them with suppliers and customers, in order to meet demand and to minimize costs. The problem is formulated as a version of the continuous facility location and allocation problem with limited capacity, also known as the Capacitated Multi-facility Weber Problem (CMWP) [17].

The original Weber problem was proposed by Alfred Weber [18], a pioneer of the modern location theories. In his original problem, he considered one facility to be located based on two suppliers and one customer, when these three points are not collinear [18, 19], and assuming Euclidean distances.

The capacitated version of the Weber problem (CMWP) was first proposed by Cooper [20], and assumes a maximum capacity for the facilities to be installed. This class of problems has been proved to be NP-hard even if all the fixed points are located on a straight line [21]. Copper proposes a rigorous solution method to the CMWP that relies on explicit enumeration of the extreme points of the transportation polytope, thus limiting its application to small problems. He also proposes a heuristic approach, known as the Alternating Transportation-Location (ATL) method, which alternates the solution of the transportation and allocation problems until convergence is achieved, although there is no guarantee of global optimality. The ATL heuristic is further developed in [22, 23].

Sherali and co-authors propose in 1977 a cutting plane algorithm for the rectilinear distance location-allocation problem [24], and in 1992 they introduce a branch-and-bound algorithm for the squared-Euclidean distance location-allocation problem [25]. They also propose in 2002 a branch-and-bound algorithm based on the partitioning of the allocation space that finitely converges to a global optimum within a given tolerance [26]. Chen, Pan, Ko [27] reformulate the CMWP as a sequence of nonlinear second-order cone problems, and apply the semi-smooth Newton method to solve it. Akyüz et al. [28] propose two branch-and-bound algorithms for solving exactly the multi-commodity CMWP: one based on partitioning the allocation space, and the other one considers partitioning of the location space. Besides exact methods, there are several heuristics developed for this class of problem [17, 29, 30, 31, 32, 33, 34].

In this paper, we extend the work by Lara et al. [1] (in which the single-period design problem for a general manufacturing network with multiple facility types is addressed) to solve the design and multi-period planning of centralized and distributed manufacturing networks. The model proposed in this paper is a multi-period nonlinear Generalized Disjunctive Programming (GDP), reformulated as a multi-period nonconvex Mixed-Integer Nonlinear Programming (MINLP). Due to the extra layer of complexity added by the multi-period formulation, we propose an accelerated version of the algorithm proposed by Lara et al. [1] to improve its computational performance and scalability. Accordingly, the contributions of this work are on the formulation (multi-period), application (centralized and distributed networks), and solution method (extension of the Bilevel Decomposition algorithm).

The remainder of the paper is organized as follows. We begin by presenting in Section 2 the problem statement. Section 3 includes the General Disjunctive Programming (GDP) formulation and its reformulation as a nonconvex MINLP. In Section 4 we propose an accelerated version of the global optimization algorithm by [1], which is guaranteed to have ϵ-convergence. The additional steps consist of a strategy for reducing the optimization search space by reducing sets of potential facilities and their two-dimensional feasible region, as well as providing an initial solution to the Master Problem. We illustrate the method for a test problem in the same section. In Section 5 we benchmark the performance of the accelerated algorithm against the original and the commercial global solvers available for the set of randomly generated instances from [1] extended to multi-period problems. Finally, in Section 6 we apply the formulation and solution strategy to a biomass supply chain case study, and in Section 7 we draw the conclusions.

2. Problem Statement

Given is a set of suppliers $i \in \mathcal{I}$, with their respective fixed location coordinates (X_i, Y_i), availability $AV_{i,t}$, and cost of material supply $CRM_{i,t}$ at each time period $t \in \mathcal{T}$. Given is also a set of customers $j \in \mathcal{J}$, with their respective fixed locations (X_j, Y_j), and demands $DM_{j,t}$ per time period t. Given are the fixed and variable investment costs ($FIC_{k,t}$ and $VIC_{k,t}$, respectively) and variable operating costs ($VOC_{k,t}$) of potential facilities $k \in \mathcal{K}$ with N different types (i.e. centralized and distributed $N = 2$), which are partitioned into subsets $\mathcal{K}_n \ \forall \ n \in \mathcal{N} = \{1, ..., N\}$ such that $\bigcup_{n \in \mathcal{N}} \mathcal{K}_n = \mathcal{K}$ and $\mathcal{K}_{n_l} \cap \mathcal{K}_{n_m} = \emptyset \ \forall n_l, n_m \in \mathcal{N}, l \neq m$. The corresponding maximum capacity, MC_k, and conversion to product flows, CV_k, of these potential facilities are also known. Given are also the transportation costs between suppliers and facilities, and facilities and markets ($FTC^{\mathrm{s}}_{i,k}, FTC^{\mathrm{c}}_{k,j}$: fixed costs; $VTC^{\mathrm{s}}_{i,k}, VTC^{\mathrm{c}}_{k,j}$: variable costs). The problem is to find the optimal network of facilities (number, types, location, when to build, and corresponding flows) that minimizes the total cost.

The variables in the problem are the coordinates of potential facilities, (x_k, y_k), the distances between supplier and facility, $d^{\mathrm{s}}_{i,k}$, and between facility and customer, $d^{\mathrm{c}}_{k,j}$, the flows between supplier and facility, $ff^{s}_{i,k,t}$, and between facility and customer, $ff^{\mathrm{c}}_{k,j,t}$, and the amount produced by each facility, $f_{k,t}$, in each time period t. There are also Boolean variables: $B_{k,t}$ (true if facility is built in time period t; false otherwise); $W_{k,t}$ (true if facility is in operation in time period t; false otherwise); $Z^{\mathrm{s}}_{i,k,t}$ (true if material supply is transported between supplier and facility during time period t; false otherwise); and $Z^{\mathrm{c}}_{k,j,t}$ (true if product is transported between facility and customer during time period t; false otherwise).

3. Model Formulation

3.1. Generalized Disjunctive Programming (GDP)

We first formulate the problem as Generalized Disjunctive Programming (GDP) to take advantage of the disjunctive structure of some of the decisions. Extending [1], the GDP formulation is given by Equations (1a)-(1t).

$$\min \ \Phi = \sum_{t \in \mathcal{T}} \frac{1}{(1+R)^t} \cdot \sum_{k \in \mathcal{K}} \left(inv_{k,t} + op_{k,t} + \sum_{i \in \mathcal{I}} cost^{\mathrm{s}}_{i,k,t} + \sum_{j \in \mathcal{J}} cost^{\mathrm{c}}_{k,j,t} \right) \tag{1a}$$

$$
\text{s.t.} \quad \begin{bmatrix} B_{k,t} \\ inv_{k,t} = FIC_{k,t} + VIC_{k,t} \cdot MC_k \end{bmatrix} \vee \begin{bmatrix} \neg B_{k,t} \\ inv_{k,t} = 0 \end{bmatrix} \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{1b}
$$

$$
\begin{bmatrix} W_{k,t} \\ op_{k,t} = VOC_{k,t} \cdot f_{k,t} \\ 0 \le f_{k,t} \le MC_k \end{bmatrix} \vee \begin{bmatrix} \neg W_{k,t} \\ op_{k,t} = 0 \\ f_{k,t} = 0 \end{bmatrix} \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{1c}
$$

$$
\begin{bmatrix} Z^{\text{s}}_{i,k,t} \\ cost^{\text{s}}_{i,k,t} = CS_{i,t} \cdot ff^{\text{s}}_{i,k,t} + FTC^{\text{s}}_{i,k} + VTC^{\text{s}}_{i,k} \cdot ff^{\text{s}}_{i,k,t} \cdot d^{\text{s}}_{i,k} \\ 0 \le ff^{\text{s}}_{i,k,t} \le \overline{FF^{\text{s}}_{i,k,t}} \end{bmatrix} \vee \begin{bmatrix} \neg Z^{\text{s}}_{i,k,t} \\ cost^{\text{s}}_{i,k,t} = 0 \\ ff^{\text{s}}_{i,k,t} = 0 \end{bmatrix} \quad \forall\, i \in \mathcal{I}, k \in \mathcal{K}, t \in \mathcal{T} \tag{1d}
$$

$$
\begin{bmatrix} Z^{\text{c}}_{k,j,t} \\ cost^{\text{c}}_{k,j,t} = FTC^{\text{c}}_{k,j} + VTC^{\text{c}}_{k,j} \cdot ff^{\text{c}}_{k,j,t} \cdot d^{\text{c}}_{k,j} \\ 0 \le ff^{\text{c}}_{k,j,t} \le \overline{FF^{\text{c}}_{k,j,t}} \end{bmatrix} \vee \begin{bmatrix} \neg Z^{\text{c}}_{k,j,t} \\ cost^{\text{c}}_{k,j,t} = 0 \\ ff^{\text{c}}_{k,j,t} = 0 \end{bmatrix} \quad \forall\, k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T} \tag{1e}
$$

$$
\begin{bmatrix} \bigvee_{t \in \mathcal{T}} B_{k,t} \\ 0 \le x_k \le \overline{X_k} \\ 0 \le y_k \le \overline{Y_k} \end{bmatrix} \vee \begin{bmatrix} \neg \bigvee_{t \in \mathcal{T}} B_{k,t} \\ x_k = 0 \\ y_k = 0 \end{bmatrix} \quad \forall\, k \in \mathcal{K} \tag{1f}
$$

$$
d^{\text{s}}_{i,k} \ge \sqrt{(X_i - x_k)^2 + (Y_i - y_k)^2} \quad \forall\, i \in \mathcal{I}, k \in \mathcal{K} \tag{1g}
$$

$$
d^{\text{c}}_{k,j} \ge \sqrt{(X_j - x_k)^2 + (Y_j - y_k)^2} \quad \forall\, k \in \mathcal{K}, j \in \mathcal{J} \tag{1h}
$$

$$
W_{k,t} \iff \bigvee_{i \in \mathcal{I}} Z^{\text{s}}_{i,k,t} \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{1i}
$$

$$
W_{k,t} \iff \bigvee_{j \in \mathcal{J}} Z^{\text{c}}_{k,j,t} \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{1j}
$$

$$
W_{k,t} \iff W_{k,t-1} \vee B_{k,t} \quad \forall\, k \in \mathcal{K} \tag{1k}
$$

$$
\sum_{k \in \mathcal{K}} f^{\text{s}}_{i,k,t} \le AV_{i,t} \quad \forall\, i \in \mathcal{I}, t \in \mathcal{T} \tag{1l}
$$

$$
\sum_{i \in \mathcal{I}} f^{\text{s}}_{i,k,t} \cdot CV_k = f_{k,t} \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{1m}
$$

$$
f_{k,t} = \sum_{j \in \mathcal{J}} f^{\text{c}}_{k,j,t} \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{1n}
$$

$$
\sum_{k \in \mathcal{K}} f^{\text{c}}_{k,j,t} = DM_{j,t} \quad \forall\, j \in \mathcal{J}, t \in \mathcal{T} \tag{1o}
$$

$$
w_k \ge w_{k+1} \quad \forall\, k \in \mathcal{K}_n,\ n \in \mathcal{N} \tag{1p}
$$

$$
x_k \ge x_{k+1} \quad \forall\, k \in \mathcal{K}_n,\ n \in \mathcal{N} \tag{1q}
$$

$$
D^{min} \le d^{\text{s}}_{i,k} \le D^{max} \quad \forall\, k \in \mathcal{K} \tag{1r}
$$

$$
D^{min} \le d^{\text{c}}_{k,j} \le D^{max} \quad \forall\, k \in \mathcal{K} \tag{1s}
$$

$$
B_{k,t}, W_{k,t}, Z^{\text{s}}_{i,k,t}, Z^{\text{c}}_{k,j,t} \in \{True, False\} \quad \forall\, i \in \mathcal{I}, k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}. \tag{1t}
$$

The objective function (1a) is the net present cost, which includes investment and operating costs for building and operating the facilities, and transportation cost from the suppliers and to the customers with an interest rate, R. This is different than the GDP proposed by [1] as it now includes a series of cash flows occurring at each time period, and the facility costs are divided into investment and operating costs.

Disjunction (1b) determines whether facility k is built at time t ($B_{k,t}$), and disjunction (1c) determines whether facility k is in operation at time t ($W_{k,t}$). These disjunctions indirectly address the choice between centralized and distributed facilities as each of the potential facilities have a specified type (i.e. distributed or centralized) and their characteristics and costs are drawn from their type. This differs from the formulation by [1] where multiple types are allowed instead of only two.

Disjunctions (1d) and (1e) decide if there is material flow between the transportation links $\{i, k\}$ and $\{k, j\}$ at each time period t, which is determined by the corresponding Boolean variables ($Z^{\mathrm{s}}_{i,k,t}$, $Z^{\mathrm{c}}_{k,j,t}$). The last disjunction, (1f), specifies that if a facility k is built at any point within the planning horizon, its coordinates should be within the appropriate bounds. However, if this facility is not built, then its coordinates should be set to $(0, 0)$, to avoid degeneracy in the solution. These five proposed disjunctions, (1b)-(1f), are similar to the disjunctions in the GDP model by [1], but have the additional flexibility of allowing different allocations by time-period, specifying in which time period a facility is built, and only accounting for operating costs in the time periods the facility is in operation.

Constraints (1g) and (1h) represent the Euclidean distances between suppliers and facilities, and facilities and customers, which is the same distance representation used by [1]. The logic relations in (1i), (1j) and (1i) establish the existence of links depending on the choice of the facilities and vice-versa, and specify that a facility k can only operate ($W_{k,t}$) if it has been built before ($B_{k,t}$). Constraints (1l)-(1o) define the mass balances, as well as the availability and demands, same as in [1].

We assume that the facilities of the same type have the same costs and characteristics associated with them, i.e., $FIC_{k,t}$, $VIC_{k,t}$, $VOC_{k,t}$, $FTC_{i,k,t}$, $VTC_{i,k,t}$, $FTC_{k,j,t}$, $VTC_{k,j,t}$, MC_k, and CV_k are the same $\forall\ k \in \mathcal{K}_n$. Analogously to [1], we have constraints (1p)-(1q) to break the symmetry in the facility selection within the same type and avoid degeneracy in the solution. These constraints enforce that for facilities k of the same type, i.e., $k \in \mathcal{K}_n,\ n \in \mathcal{N}$, the model will chose first to build the ones with the lower indices, and those will be located in lower x_k coordinate. Finally, constraints (1r)-(1s) determine the bounds for the distances, D^{min} and D^{max}, and (1t) defined the Boolean variables.

The GDP model (1) is nonconvex due to the bilinear terms ($ff \cdot d$) in the transportation cost, as can be seen in disjunctions (1d)-(1e). There is a large body of literature on relaxations and reformulations of bilinear terms, most of them deriving from the McCormick envelope [35]: e.g. [36, 37, 38, 39, 40]. The presence of bilinear terms, which can give rise to local minima, is the main motivation behind choosing a GDP formulation. By having the bilinear terms as part of the disjunctions, they are calculated only for the selected connections within an iterative procedure. Therefore, for a fixed choice of the Boolean variables, the GDP leads to a reduction in the number of bilinear terms and generates a more favorable structure that can be exploited in a decomposition scheme. Additionally, equations (1g)- (1h) are nonlinear convex constraints since they correspond to Euclidean norms [41].

3.2. Mixed-integer nonlinear Programming (MINLP) model

The GDP can be transformed into an MINLP using the hull reformulation, which yields the tightest relaxation for each disjunction [42]. Since the disaggregated variables can be reformulated back to the original variables, the resulting MINLP is given by Equations (2a)-(2x). Again, the main difference between the MINLP reformulation presented in [1] and the following MINLP is the added flexibility of allowing multi-period operating and allocation decisions, as well as accounting for operating costs by time-period.

$$\min\ \Phi = \sum_{t\in\mathcal{T}} \frac{1}{(1+R)^t} \cdot \sum_{k\in\mathcal{K}} \left(inv_{k,t} + op_{k,t} + \sum_{i\in\mathcal{I}} cost^{\mathrm{s}}_{i,k,t} + \sum_{j\in\mathcal{J}} cost^{\mathrm{c}}_{k,j,t} \right) \tag{2a}$$

$$\text{s.t}\quad inv_{k,t} = (FIC_{k,t} + VIC_{k,t} \cdot MC_k) \cdot b_{k,t} \qquad \forall\ k\in\mathcal{K}, t\in\mathcal{T} \tag{2b}$$

$$op_{k,t} = VOC_{k,t} \cdot f_{k,t} \qquad \forall\ k\in\mathcal{K}, t\in\mathcal{T} \tag{2c}$$

$$cost^{\mathrm{s}}_{i,k,t} = CS_{i,t} \cdot f^{\mathrm{s}}_{i,k,t} + FTC^{\mathrm{s}}_{i,k} \cdot z^{\mathrm{s}}_{i,k,t} + VTC^{\mathrm{s}}_{i,k} \cdot ff^{\mathrm{s}}_{i,k,t} \cdot d^{\mathrm{s}}_{i,k} \qquad \forall\ i\in\mathcal{I}, k\in\mathcal{K}, t\in\mathcal{T} \tag{2d}$$

$$cost^{\mathrm{c}}_{k,j,t} = FTC^{\mathrm{c}}_{k,j} \cdot z^{\mathrm{c}}_{k,j,t} + VTC^{\mathrm{c}}_{k,j} \cdot ff^{\mathrm{c}}_{k,j,t} \cdot d^{\mathrm{c}}_{k,j} \qquad \forall\ k\in\mathcal{K}, j\in\mathcal{J}, t\in\mathcal{T} \tag{2e}$$

$$d^{\mathrm{s}}_{i,k} \geq \sqrt{(X_i - x_k)^2 + (Y_i - y_k)^2} \qquad \forall\ i\in\mathcal{I}, k\in\mathcal{K} \tag{2f}$$

$$d^{\mathrm{c}}_{k,j} \geq \sqrt{(X_j - x_k)^2 + (Y_j - y_k)^2} \qquad \forall\ k\in\mathcal{K}, j\in\mathcal{K} \tag{2g}$$

$$\sum_{k\in\mathcal{K}} ff^{\mathrm{s}}_{i,k,t} \leq AV_{i,t} \qquad \forall\ i\in\mathcal{I}, t\in\mathcal{T} \tag{2h}$$

$$\sum_{i\in\mathcal{I}} ff^{\mathrm{s}}_{i,k,t} \cdot CV_k = f_{k,t} \qquad \forall\ k\in\mathcal{K}, t\in\mathcal{T} \tag{2i}$$

$$f_{k,t} = \sum_{j\in\mathcal{J}} ff^{\mathrm{c}}_{k,j,t} \qquad \forall\ k\in\mathcal{K}, t\in\mathcal{T} \tag{2j}$$

$$\sum_{k\in\mathcal{K}} ff^{\mathrm{c}}_{k,j,t} = DM_{j,t} \qquad \forall\ j\in\mathcal{J}, t\in\mathcal{T} \tag{2k}$$

$$w_{k,t} = \sum_{i\in\mathcal{I}} z^{\mathrm{s}}_{i,k,t} \qquad \forall\ k\in\mathcal{K}, t\in\mathcal{T} \tag{2l}$$

$$w_{k,t} = \sum_{j\in\mathcal{J}} z^{\mathrm{c}}_{k,j,t} \qquad \forall\ k\in\mathcal{K}, t\in\mathcal{T} \tag{2m}$$

$$w_{k,t} = w_{k,t-1} + b_{k,t} \qquad \forall\ k \in \mathcal{K}, t \in \mathcal{T} \tag{2n}$$

$$0 \leq f_{k,t} \leq MC_k \cdot w_{k,t} \qquad \forall\ k \in \mathcal{K}, t \in \mathcal{T} \tag{2o}$$

$$0 \leq ff^{\text{s}}_{i,k,t} \leq \overline{FF^{\text{s}}_{i,k,t}} \cdot z^{\text{s}}_{i,k,t} \qquad \forall\ i \in \mathcal{I}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2p}$$

$$0 \leq ff^{\text{c}}_{k,j,t} \leq \overline{FF^{\text{c}}_{k,j,t}} \cdot z^{\text{c}}_{k,j,t} \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T} \tag{2q}$$

$$0 \leq x_k \leq \overline{X_k} \cdot \sum_{t \in \mathcal{T}} b_{k,t} \qquad \forall\ k \in \mathcal{K} \tag{2r}$$

$$0 \leq y_k \leq \overline{Y_k} \cdot \sum_{t \in \mathcal{T}} b_{k,t} \qquad \forall\ k \in \mathcal{K} \tag{2s}$$

$$w_{k,t} \geq w_{k+1,t} \qquad \forall\ k \in \mathcal{K}_n,\ n \in \mathcal{N}, t \in \mathcal{T} \tag{2t}$$

$$x_k \geq x_{k+1} \qquad \forall\ k \in \mathcal{K}_n,\ n \in \mathcal{N} \tag{2u}$$

$$D^{min} \leq d^{\text{s}}_{i,k} \leq D^{max} \qquad \forall\ i \in \mathcal{I}, k \in \mathcal{K} \tag{2v}$$

$$D^{min} \leq d^{\text{c}}_{k,j} \leq D^{max} \qquad \forall\ k \in \mathcal{K}, j \in \mathcal{J} \tag{2w}$$

$$b_{k,t}, w_{k,t}, z^{\text{s}}_{i,k,t}, z^{\text{c}}_{k,j,t} \in \{0,1\} \qquad \forall\ i \in \mathcal{I}, k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}. \tag{2x}$$

The MINLP model (2) can be more concisely represented by (3).

$$\Phi = \quad \min \quad g(f, ff, z) + d^{\intercal} C ff \tag{3a}$$

$$\text{s.t.} \quad d_{l,k} \geq \sqrt{(X_l - x_k)^2 + (Y_l - y_k)^2} \qquad \forall l \in \mathcal{I} \cup \mathcal{J},\ k \in \mathcal{K} \tag{3b}$$

$$f, ff, z, d, x, y \in \Omega, \tag{3c}$$

where ff is the vector of all flows between suppliers and facilities, $ff^{s}_{i,k,t}$, and between facilities and customers, $ff^{\text{c}}_{k,j,t}$; f is the vector of all facilities' productions at each time period, $f_{k,t}$; z is the vector of all discrete decision variables ($b_{k,t}$, $w_{k,t}$, $z^{\text{s}}_{i,k,t}$, and $z^{\text{c}}_{k,j,t}$); and $g(f, ff, z)$ is the cost function associated with these decision variables. Additionally, d is the vector of distances, C is the matrix of variable transportation costs ($VTC^{\text{s}}_{i,k}$ and $VTC^{\text{c}}_{k,j}$), and ($d^{\intercal} C ff$) is the bilinear term associated with the variable transportation cost. Constraint (3b) represents both constraints (1g) and (1h), and the feasible region Ω is given by (2h)-(2x).

4. Accelerated Bilevel Decomposition Algorithm

As shown by [1], global optimization solvers do not perform well for mid to large instances of the single-period version of this problem. Thus, it is expected that with the added complexity of having multi-period decisions their performance will degrade even further. Lara et al. [1] propose a Bilevel Decomposition algorithm that consists of decomposing the problem into a master problem and a subproblem, for which ϵ-convergence can be proved. The master problem is based on a relaxation of the nonconvex MINLP, which yields an MILP that predicts the selection of facilities and their links to suppliers and customers, as well as a lower bound on the cost of the original problem. The subproblem corresponds to a nonconvex NLP of reduced dimensionality that results from fixing the binary variables in the MINLP problem, according to the binary variables predicted in the MILP master problem.

In this paper, we propose an accelerated version of the algorithm proposed by [1] that keeps its rigor (i.e., its ϵ-convergence), but has some additional steps to improve its performance to allow the solution of large-scale multi-period instances of this problem within a reasonable amount of time. The additional steps consist of an attempt of reducing the optimization search space such that it is easier for the Bilevel Decomposition to find good bounds and the optimal solution. These steps consist of: i) possibly reducing the set of potential facilities by performing branch-and-bound on the facilities that were not selected; ii) potentially reducing the feasible two-dimensional space by performing a branch-and-bound on the partitions that did not have any facility being built on; iii) giving an initial feasible solution to the Master Problem based on the solution of the previous iteration.

The main steps in the Accelerated Bilevel Decomposition are shown in Figure 1.

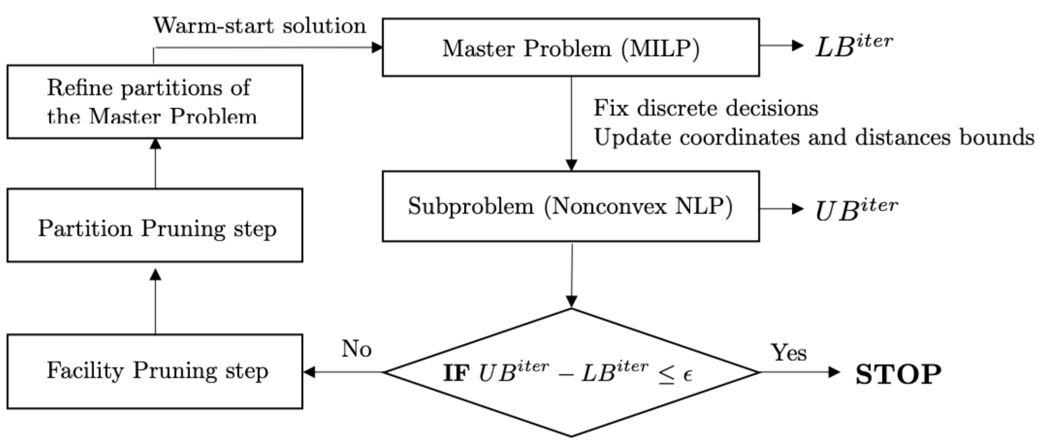

Figure 1: Accelerated Bilevel Decomposition concise representation

We start by explaining the basic steps of the original algorithm by Lara et al. [1] applied to the current MINLP formulation (2), and then cover the proposed additional steps to improve its performance.

4.1. Master Problem

The nonlinearity and nonconvexity of the formulation come from the fact that the location of the potential facilities is a decision variable. The master problem takes advantage of this property and partitions the space into uniform rectangular sub-regions. By having a grid to represent the feasible area, we can pre-compute the minimum distance between the fixed points (suppliers and customers) and use them as parameters in the model [1]. The minimum distances between the fixed points and the sub-regions p, $\widehat{D}_{i,p}$ and $\widehat{D}_{j,p}$, are computed as follows:

$$dx_{i,p} = \max\{|X_i - x_p| - \Delta x/2, 0\} \qquad \forall\, i \in \mathcal{I}, p \in \mathcal{P} \tag{4a}$$

$$dy_{i,p} = \max\{|Y_i - y_p| - \Delta y/2, 0\} \qquad \forall\, i \in \mathcal{I}, p \in \mathcal{P} \tag{4b}$$

$$dx_{j,p} = \max\{|X_j - x_p| - \Delta x/2, 0\} \qquad \forall\, j \in \mathcal{J}, p \in \mathcal{P} \tag{4c}$$

$$dy_{j,p} = \max\{|Y_j - y_p| - \Delta y/2, 0\} \qquad \forall\, j \in \mathcal{J}, p \in \mathcal{P} \tag{4d}$$

$$\widehat{D}_{i,p} = \max\{\sqrt{dx_{i,p}^2 + dy_{i,p}^2}, D^{min}\} \qquad \forall\, i \in \mathcal{I}, p \in \mathcal{P} \tag{4e}$$

$$\widehat{D}_{j,p} = \max\{\sqrt{dx_{j,p}^2 + dy_{j,p}^2}, D^{min}\} \qquad \forall\, j \in \mathcal{J}, p \in \mathcal{P}, \tag{4f}$$

where (x_p, y_p) are the coordinates of the mid-point of each sub-region p; Δx and Δy are the length of sub-region p in the x and y directions, respectively; D^{min} is the lower bound for the distances, not allowing the model to choose to build a facility k on top of a fixed point from a supplier or a customer [1].

By using the minimum distance parameters, the MINLP formulation (2) can be reformulated as a mixed-integer linear programming (MILP) model (5), which yields a lower bound to the solution of the original models (1) and (2), as proved in Proposition 1 of [1].

$$\min\ \Phi = \sum_{t \in \mathcal{T}} \frac{1}{(1+R)^t} \cdot \sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}} \left(inv_{k,p,t} + op_{k,p,t} + \sum_{i \in \mathcal{I}} cost^{\mathrm{s}}_{i,k,p,t} + \sum_{j \in \mathcal{J}} cost^{\mathrm{c}}_{k,j,p,t} \right) \tag{5a}$$

$$\text{s.t} \quad inv_{k,p,t} = (FIC_{k,t} + VIC_{k,t} \cdot MC_k) \cdot b_{k,p,t} \qquad \forall\, k \in \mathcal{K}, p \in \mathcal{P}, t \in \mathcal{T} \tag{5b}$$

$$op_{k,p,t} = VOC_{k,t} \cdot f_{k,p,t} \qquad \forall\, k \in \mathcal{K}, p \in \mathcal{P}, t \in \mathcal{T} \tag{5c}$$

$$cost^{\mathrm{s}}_{i,k,p,t} = CS_{i,t} \cdot ff^{\mathrm{s}}_{i,k,p,t} + FTC^{\mathrm{s}}_{i,k} \cdot z^{\mathrm{s}}_{i,k,p,t} + VTC^{\mathrm{s}}_{i,k} \cdot \hat{D}^{\mathrm{s}}_{i,p} \cdot ff^{\mathrm{s}}_{i,k,p,t} \qquad \forall\, i \in \mathcal{I}, k \in \mathcal{K}, p \in \mathcal{P}, t \in \mathcal{T} \tag{5d}$$

$$cost^{\mathrm{c}}_{k,j,p,t} = FTC^{\mathrm{c}}_{k,j} \cdot z^{\mathrm{c}}_{k,j,p,t} + VTC^{\mathrm{c}}_{k,j} \cdot \hat{D}^{\mathrm{c}}_{j,p} \cdot ff^{\mathrm{c}}_{k,j,p,t} \qquad \forall\, k \in \mathcal{K}, j \in \mathcal{J}, p \in \mathcal{P}, t \in \mathcal{T} \tag{5e}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}} ff^{\mathrm{s}}_{i,k,p,t} \le AV_{i,t} \qquad \forall\, i \in \mathcal{I}, t \in \mathcal{T} \tag{5f}$$

$$\sum_{i \in \mathcal{I}} ff^{\mathrm{s}}_{i,k,p,t} \cdot CV_k = f_{k,p,t} \qquad \forall\, k \in \mathcal{K}, p \in \mathcal{P}, t \in \mathcal{T} \tag{5g}$$

$$f_{k,p,t} = \sum_{j \in \mathcal{J}} ff^{\mathrm{c}}_{k,j,p,t} \qquad \forall\, k \in \mathcal{K}, p \in \mathcal{P}, t \in \mathcal{T} \tag{5h}$$

$$\sum_{k\in\mathcal{K}}\sum_{p\in\mathcal{P}} ff^{\mathrm{c}}_{k,j,p,t} = DM_{j,t} \qquad \forall\, j\in\mathcal{J}, t\in\mathcal{T} \tag{5i}$$

$$w_{k,p,t} = \sum_{i\in\mathcal{I}} z^{\mathrm{s}}_{i,k,p,t} \qquad \forall\, k\in\mathcal{K}, p\in\mathcal{P}, t\in\mathcal{T} \tag{5j}$$

$$w_{k,p,t} = \sum_{j\in\mathcal{J}} z^{\mathrm{c}}_{k,j,p,t} \qquad \forall\, k\in\mathcal{K}, p\in\mathcal{P}, t\in\mathcal{T} \tag{5k}$$

$$w_{k,p,t} = w_{k,p,t-1} + b_{k,p,t} \qquad \forall\, k\in\mathcal{K}, p\in\mathcal{P}, t\in\mathcal{T} \tag{5l}$$

$$\sum_{p\in\mathcal{P}} w_{k,p,t} \leq 1 \qquad \forall\, k\in\mathcal{K}, t\in\mathcal{T} \tag{5m}$$

$$\sum_{p\in\mathcal{P}} b_{k,p,t} \leq 1 \qquad \forall\, k\in\mathcal{K}, t\in\mathcal{T} \tag{5n}$$

$$\sum_{p\in\mathcal{P}} z^{\mathrm{s}}_{i,k,p,t} \leq 1 \qquad \forall\, k\in\mathcal{K}, t\in\mathcal{T} \tag{5o}$$

$$\sum_{p\in\mathcal{P}} z^{\mathrm{c}}_{k,j,p,t} \leq 1 \qquad \forall\, k\in\mathcal{K}, t\in\mathcal{T} \tag{5p}$$

$$0 \leq f_{k,p,t} \leq MC_k \cdot w_{k,p,t} \qquad \forall\, k\in\mathcal{K}, p\in\mathcal{P}, t\in\mathcal{T} \tag{5q}$$

$$0 \leq ff^{\mathrm{s}}_{i,k,p,t} \leq \overline{FF^{\mathrm{s}}_{i,k,t}} \cdot z^{\mathrm{s}}_{i,k,p,t} \qquad \forall\, i\in\mathcal{I}, k\in\mathcal{K}, p\in\mathcal{P}, t\in\mathcal{T} \tag{5r}$$

$$0 \leq ff^{\mathrm{c}}_{k,j,p,t} \leq \overline{FF^{\mathrm{c}}_{k,j}} \cdot z^{\mathrm{c}}_{k,j,p,t} \qquad \forall\, k\in\mathcal{K}, j\in\mathcal{J}, p\in\mathcal{P}, t\in\mathcal{T} \tag{5s}$$

$$\sum_{p} w_{k,p,t} \geq \sum_{p\in\mathcal{P}} w_{k+1,p,t} \qquad \forall\, k\in\mathcal{K}_n,\ n\in\mathcal{N}, t\in\mathcal{T} \tag{5t}$$

$$\sum_{p'\leq p} w_{k',p',t} \geq \sum_{p\in\mathcal{P}} w_{k,p,t} \qquad \forall\, k' < k,\ k,k'\in\mathcal{K}_n,\ n\in\mathcal{N}, t\in\mathcal{T} \tag{5u}$$

$$b_{k,p,t}, w_{k,p,t}, z^{\mathrm{s}}_{i,k,p,t}, z^{\mathrm{c}}_{k,j,p,t} \in \{0,1\} \qquad \forall\, i\in\mathcal{I}, k\in\mathcal{K}, j\in\mathcal{J}, p\in\mathcal{P}, t\in\mathcal{T}. \tag{5v}$$

Following the same notation as in (3), the MILP master problem (5) can be concisely represented by (6).

$$\Phi^{LB} = \quad \min \quad g(f, ff, z) + D^{\intercal} C ff \tag{6a}$$

$$\text{s.t.} \qquad f, ff, z, d, x, y \in \Omega', \tag{6b}$$

where D is the vector of minimum distance parameters, $\widehat{D}_{i,p}$ and $\widehat{D}_{j,p}$, and Ω' represents the feasible region described by constraints (5f)-(5v).

4.2. Subproblem

After solving the master problem (5), the subproblem consists of solving (2) for the fixed decisions of which facilities k to build and operate at each time period t, $\widehat{b}_{k,t}$ and $\widehat{w}_{k,t}$, respectively, and how to allocate their material supply $\widehat{z}^{\mathrm{s}}_{i,k,t}$ and products $\widehat{z}^{\mathrm{c}}_{k,j,t}$ as selected in the MILP (5).

Besides fixing the discrete decisions, we also update the bounds of the facilities coordinates such that their location (x_k, y_k) has to lie within the bounds of the sub-region p chosen in the Master Problem; i.e., for a p such that $\sum_{t\in\mathcal{T}} b_{k,p,t} = 1$ in the solution of Problem (5) we have that

$\underline{X}'_{k,p} \leq x_k \leq \overline{X}'_{k,p}$ and $\underline{Y}'_{k,p} \leq y_k \leq \overline{Y}'_{k,p}$, where:

$$
\begin{aligned}
&\underline{X}'_{k,p} = x_p - \Delta x/2 && \forall\ k \in \mathcal{K} && \text{(7a)}\\
&\overline{X}'_{k,p} = x_p + \Delta x/2 && \forall\ k \in \mathcal{K} && \text{(7b)}\\
&\underline{Y}'_{k,p} = y_p - \Delta y/2 && \forall\ k \in \mathcal{K} && \text{(7c)}\\
&\overline{Y}'_{k,p} = y_p + \Delta y/2 && \forall\ k \in \mathcal{K}. && \text{(7d)}
\end{aligned}
$$

This assumption greatly impacts tractability because the bounds for $d_{i,k}$ and $d_{k,j}$, which are part of the bilinear terms, become tighter, i.e., $\underline{D}'_{i,k,p} \leq d_{i,k} \leq \overline{D}'_{i,k,p}$ and $\underline{D}'_{k,j,p} \leq d_{k,j} \leq \overline{D}'_{k,j,p}$, where:

$$
\begin{aligned}
&\underline{D}'_{i,k,p} = \widehat{D}_{i,p} && \forall\ i \in \mathcal{I},\ k \in \mathcal{K} && \text{(8a)}\\
&\overline{D}'_{i,k,p} = \widehat{D}_{i,p} + \sqrt{\Delta x^2 + \Delta y^2} && \forall\ i \in \mathcal{I},\ k \in \mathcal{K} && \text{(8b)}\\
&\underline{D}'_{k,j,p} = \widehat{D}_{j,p} && \forall\ j \in \mathcal{J},\ k \in \mathcal{K} && \text{(8c)}\\
&\overline{D}'_{k,j,p} = \widehat{D}_{j,p} + \sqrt{\Delta x^2 + \Delta y^2} && \forall\ j \in \mathcal{J},\ k \in \mathcal{K}. && \text{(8d)}
\end{aligned}
$$

Accordingly, the McCormick convex envelopes [35] also become tighter, strengthening the lower bounds in the global optimization search of this NLP.

Following the same notation as in (3) and (6), the NLP subproblem can be concisely represented by (9).

$$
\begin{aligned}
\Phi^{UB} = \quad & \min && g(f, ff, \hat{z}) + d^{\intercal} C ff && && \text{(9a)}\\
& \text{s.t.} && d_{l,k} \geq \sqrt{(X_l - x_k)^2 + (Y_l - y_k)^2} && \forall l \in \mathcal{I} \cup \mathcal{J},\ k \in \mathcal{K} && \text{(9b)}\\
& && f, ff, d, x, y \in \Omega'' && && \text{(9c)}
\end{aligned}
$$

where $\hat{z}$ represents the discrete decisions obtained in the solution of the Master Problem (6) and fixed for this Subproblem, and Ω'' represents the feasible region Ω with the updated bounds for the distances and (x, y) coordinates, $d_{i,k}$, $d_{k,j}$, and (x_k, y_k), respectively.

The subproblem (9) is a reduced nonconvex NLP. Since it comprises the original problem (3) for a set of fixed discrete decisions and tighter bounds for the distances and (x,y) coordinates, it yields a feasible Upper Bound (UB) to the total cost, $\Phi^{UB} \geq \Phi$.

4.3. Facility Pruning

There are instances, especially the ones that favor centralized networks, in which having a large set of potential distributed facilities adds unnecessary burden to their solution. With this in mind, we propose an additional step to the original Bilevel Decomposition [1] based on the branch-and-bound algorithm. After solving the Master Problem and the Subproblem, this step consists of solving the MILP (5) with the additional constraint:

$$\sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} b_{k',p,t} \geq 1 \tag{10}$$

for each facility k' that was not selected to be built by the Master Problem (5). Constraint (10) enforces facility k' to be built in one of the partition during the considered planning horizon. $\Phi^{LB,k'}$ is the optimal solution of the MILP (5) with constraint (10) for facility k'. Based on this additional constraint, we have the following Proposition 1.

Proposition 1. *If the result of the MILP (5) plus the additional constraint (10), $\Phi^{LB,k'}$, is greater than the upper bound obtained by the NLP subproblem, it means that building this facility k' will never be optimal hence it can be excluded from the set of potential facilities.*

Proof. From Proposition 1 of [1] we know that the optimal value of the MILP (5), Φ^{LB}, is an underestimator of the optimal value of MINLP (2), Φ. Hence, the optimal value of the MILP (5) with the additional constraint (10), $\Phi^{LB,k'}$, underestimates the optimal value of MINLP (2) with this additional requirement of forcing facility k' to be built within the planning horizon, $\Phi^{k'}$.

Moreover, from Theorem 1 of [1] we have that the optimal value of the NLP subproblem, Φ^{UB} is an incumbent (i.e. feasible solution) of the MINLP (2), such that $\Phi^{LB} \leq \Phi \leq \Phi^{UB}$. Therefore, if $\Phi^{LB,k'} > \Phi^{UB}$ and $\Phi^{LB,k'} \leq \Phi^{k'}$, then $\Phi^{k'} > \Phi^{UB}$ and, consequently, building this facility k' will never be optimal. Accordingly, facility k' can be pruned from the set of potential facilities. $\square$

Since all facilities of the same type have exactly the same characteristics and data and the symmetry breaking constraint (2t) forces lower-index facilities of the same type to be build first, then if facility k' is pruned, it means that all facilities k'' such that $k'' > k'$ should also be pruned (i.e., excluded from the set of potential facilities).

This step can be computationally expensive; therefore we only perform it in the first iteration of the algorithm, and also set a maximum solution time for the solution of each $\Phi^{LB,k'}$.

4.4. Partition Pruning

The idea of the Partition Pruning step is very similar to the Facility Pruning. It consists of running a set of MILPs (5) with the additional constraint:

$$\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} b_{k,p',t} \geq 1 \tag{11}$$

for each partition p' that did not have any facility k being built on by the Master Problem (5). This constraint enforces that at least one facility k is built on this partition p' during the planning horizon. $\Phi^{LB,p'}$ is the optimal solution of the MILP (5) with constraint (11) for partition p'.

Following the same idea as before, we can establish the following Proposition 2.

Proposition 2. *If the result of the MILP (5) with the additional constraint (11), $\Phi^{LB,p'}$, is greater than the upper bound obtained by the NLP subproblem, it means that building on this partition p' will never be optimal and this partition and its further refinements can be excluded from the set of potential partitions.*

Proof. This proof is very similar to the proof of Proposition 1. Following the same logic as before we have that the optimal value of the MILP (5) with the additional constraint (11), $\Phi^{LB,p'}$, underestimates the optimal value of MINLP (2) with this additional requirement of forcing at least one facility to be built on partition p', $\Phi^{p'}$. Therefore, knowing that $\Phi^{LB} \leq \Phi \leq \Phi^{UB}$, if $\Phi^{LB,k'} > \Phi^{UB}$ and $\Phi^{LB,k'} \leq \Phi^{k'}$, we can conclude that $\Phi^{k'} > \Phi^{UB}$ and, consequently, building a facility on partition p' will never be optimal. Accordingly, partition p' and its further refinements can be pruned from the set of potential facilities. □

This step can also be computationally expensive; therefore we only perform it in the first two iterations of the algorithm, and also set a maximum solution time for the solution of each $\Phi^{LB,p'}$.

Additional to the Partition Pruning step, we automatically prune the partitions that have their minimum distance to the fixed points, $\hat{D}_{i,p}$ and $\hat{D}_{j,p}$, plus the diagonal size of the partition $\sqrt{\Delta x^2 + \Delta y^2}$ to be less than the allowed minimum distance D^{min}, which means that we prune the partitions in which the maximum distance between them and a fixed point is less than minimum distance allowed, which would violate the distance bound constraints in the original MINLP (2v) and (2w).

4.5. Warm-start MILP solutions

The solution of the MILP (5) is the main bottleneck to the solution of the Bilevel Decomposition algorithm because as the number of partitions increases, it greatly impacts the size of the model and, consequently, its solution time. In order to mitigate this issue, we warm-start the MILP solutions by providing to the solver a good feasible solution.

This initial feasible solution is directly obtained from the solution of the Master Problem and Subproblem in the previous iteration, not requiring to solve any additional MILP primal heuristic [43]. This feasible solution consists of building the facilities selected on the previous Master Problem (at the same time period as before), and choosing for their location the partition corresponding to the (x_k, y_k) coordinates given by the previous NLP Subproblem. In case the NLP Subproblem builds the facility on the boundary of the partition chosen by the MILP Master Problem, we select for the warm-start solution the adjacent partition that shares this boundary. This feasible solution is provided to the MILP solver (e.g. Gurobi and CPLEX) through the *initialize* option in Pyomo.

This step is not necessary, as we did not encounter any case in which the MILP solver could not find a feasible solution without the warm-start. Also, it does not reduce the computational time required by the MILP solver to solve the LP relaxation. However, it does provide a good incumbent solution that can help the convergence of the Branch-and-Bound algorithm, and expedite the overall convergence of the Accelerated Bilevel Decomposition, as can be seen in results in sections 4.8 and 5.

4.6. Accelerated Algorithm

As discussed earlier in this section, the Accelerated Bilevel Decomposition Algorithm consists of iteratively solving the MILP master problem and the NLP subproblem with additional steps to help convergence: Facility Pruning, Partition Pruning, and Warm-start of the Master Problem. The proposed algorithm is shown in Figure 2.

As proved by Theorem 1 in [1], the proposed Bilevel Decomposition algorithm in Figure 2 converges to the global optimum in a finite number of steps within an ϵ-tolerance.

4.7. Relation between space discretization and optimality tolerance ϵ

The lower bound of the algorithm is tightly related to how refined the discretization of space is in the current iteration, as the lower bound comes from the solution of the MILP (5) in which

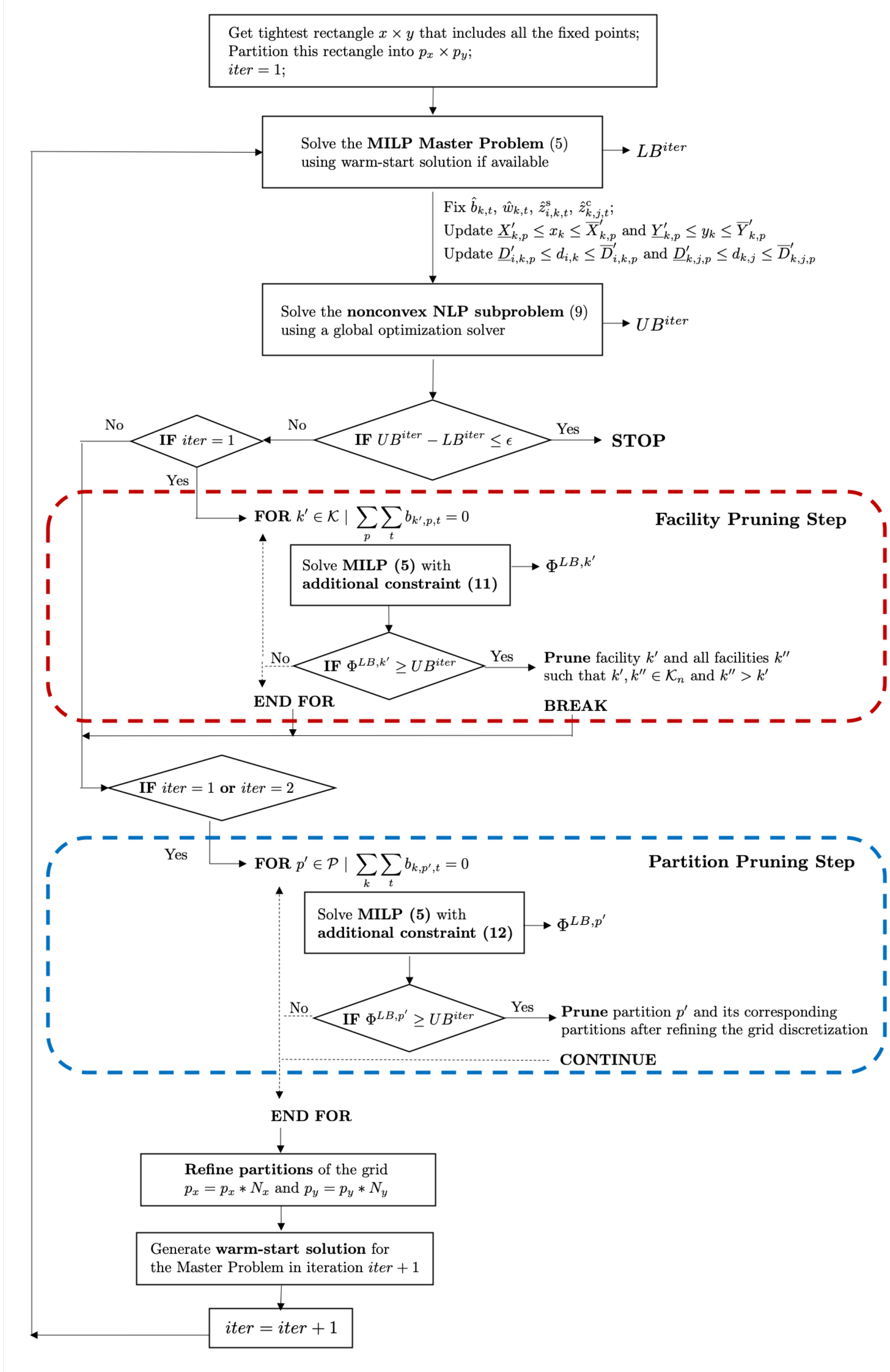

Figure 2: Accelerated Bilevel Decomposition Representation

the distance variable is underestimated as the minimum distance between the fixed points and each partition on the grid. Therefore, Proposition 3 finds an upper bound to the dimensions of the partitions in the grid, Δ^*, such that if $\Delta x \leq \Delta^*$ and $\Delta y \leq \Delta^*$ the algorithm will converge in one iteration.

Proposition 3. *By starting the Bilevel Decomposition algorithm with a specific $p_x^* \times p_y^*$ partitioning of the space such that $\Delta x \leq \Delta^*$ and $\Delta y \leq \Delta^*$, the algorithm converges within ϵ-tolerance in a single iteration.*

Proof. This proposition is true if by starting the Bilevel Decomposition algorithm with a partitioning of the space such that $\Delta x \leq \Delta^*$ and $\Delta y \leq \Delta^*$, the Master Problem in $iter = 1$ yields an upper bound, Φ^{UB}, the Subproblem in $iter = 1$ yields a lower bound, Φ^{LB}, and both satisfy the optimality tolerance $\Phi^{UB} - \Phi^{LB} \leq \epsilon$. Thus, from (6) and (9) we have that:

$$\Phi^{UB} - \Phi^{LB} \leq \epsilon \tag{12a}$$

$$g(f^*, ff^*, \hat{z}) + d^{*\intercal} C ff^* - g(\hat{f}, \widehat{ff}, \hat{z}) - D^\intercal C \widehat{ff} \leq \epsilon \tag{12b}$$

where the superscript * denotes the optimal solution of the variables in the NLP (9), and the accent $\hat{}$ denotes the optimal solution of the variables in the MILP (6).

From Proposition 1 by Lara et al. [1] we know that the difference between the MINLP (2) and the MILP (5) is the underestimation of transportation costs by the latter. Additionally, from Proposition 2 of the same paper, we have that for an infinite number of partitions the MILP (5) becomes an exact infinite dimensional representation of the MINLP (2) and both (2) and (5) have the same optimal solution $\Phi^* = \hat{\Phi}$.

Since the difference in the optimal value of the MINLP and its MILP underestimation is only due to the underestimation of the bilinear term, we can fix the optimal solution for the continuous variables f and ff to be the same between in MILP and the NLP, i.e., $f^* = \hat{f}$ and $ff^* = \widehat{ff}$, and this would give as a feasible solution $\Phi^{feas} \geq \Phi^{UB}$. Thus, if the optimality tolerance is satisfied by Φ^{feas}, it is also satisfied by Φ^{UB}. Therefore, for the sake of simplicity, we can omit the superscripts and write that

$$d^\intercal C ff - D^\intercal C ff \leq \epsilon \tag{12c}$$

$$\sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t} \cdot (d_{l,k} - D_{l,k}) \leq \epsilon \tag{12d}$$

Since $d_{l,m} - D_{l,m} \leq \sqrt{(\Delta x^*)^2 + (\Delta y^*)^2}$, if

$$\sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t} \cdot \sqrt{(\Delta x^*)^2 + (\Delta y^*)^2} \ \leq \epsilon$$

is satisfied, then (12d) will consequently be satisfied.

Therefore, we can write that

$$\sqrt{(\Delta x^*)^2 + (\Delta y^*)^2} \cdot \sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t} \ \leq \epsilon \tag{12e}$$

Considering an upper bound on the term multiplying the $\sqrt{(\Delta x^*)^2 + (\Delta y^*)^2}$, we denote it with the $\overline{}$ accent. If the following condition is satisfied,

$$\sqrt{(\Delta x^*)^2 + (\Delta y^*)^2} \cdot \overline{\sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t}} \ \leq \epsilon \tag{12f}$$

then we can ensure that (12d) is satisfied.

Going back to the original (not concise) representation:

$$\sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t} = \sum_{t \in \mathcal{T}} \frac{1}{(1+R)^t} \sum_{k \in \mathcal{K}} \left(\sum_{i \in \mathcal{I}} VTC^{\mathrm{s}}_{i,k,t} \cdot ff^{\mathrm{s}}_{i,k,t} + \sum_{j \in \mathcal{J}} VTC^{\mathrm{c}}_{k,j,t} \cdot ff^{\mathrm{c}}_{k,j,t} \right) \tag{12g}$$

We can then take the maximum of the variable transportation costs over the facilities $k \in \mathcal{K}$ such that $\overline{VTC^{\mathrm{s}}}_{i,t} = \max_{k \in \mathcal{K}} VTC^{\mathrm{s}}_{i,k,t}$, and $\overline{VTC^{\mathrm{c}}}_{j,t} = \max_{k \in \mathcal{K}} VTC^{\mathrm{c}}_{k,j,t}$, and substitute these parameters back into (12g):

$$\sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t} \leq \sum_{t \in \mathcal{T}} \frac{1}{(1+R)^t} \left(\sum_{i \in \mathcal{I}} \overline{VTC^{\mathrm{s}}}_{i,t} \sum_{k \in \mathcal{K}} ff^{\mathrm{s}}_{i,k,t} + \sum_{j \in \mathcal{J}} \overline{VTC^{\mathrm{c}}}_{j,t} \sum_{k \in \mathcal{K}} ff^{\mathrm{c}}_{k,j,t} \right) \tag{12h}$$

Combining (12h) with constraint (2k) we can rewrite (12h) as follows

$$\sum_{l \in \mathcal{I} \cup \mathcal{J}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} C_{l,k,t} \cdot ff_{l,k,t} \leq \sum_{t \in \mathcal{T}} \frac{1}{(1+R)^t} \left(\sum_{i \in \mathcal{I}} \overline{VTC^{\mathrm{s}}}_{i,t} \sum_{k \in \mathcal{K}} ff^{\mathrm{s}}_{i,k,t} + \sum_{j \in \mathcal{J}} \overline{VTC^{\mathrm{c}}}_{j,t} DM_{j,t} \right) \tag{12i}$$

and the only remaining variable is $ff^{\mathrm{s}}_{i,k,t}$.

Now, we can also take maximum value of the variable transportation cost over suppliers i such that $\overline{\overline{VTC^{\mathrm{s}}}}_t = \max_{i \in \mathcal{I}} \overline{VTC^{\mathrm{s}}}_{i,t}$. Using this new parameter combined with constraints (2i) and (2k), and knowing that CV_k represents the conversion of facility k thus it is a fraction number between

[0, 1] considering its minimum $\underline{CV} = \min_{k\in\mathcal{K}} CV_k$, we can rewrite (12i) as follows

$$\sum_{l\in\mathcal{I}\cup\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} C_{l,k,t}\cdot ff_{l,k,t} \leq \sum_{t\in\mathcal{T}}\frac{1}{(1+R)^t}\left(\overline{\overline{VTC^{\mathrm{s}}}}_t\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} ff^{\mathrm{s}}_{i,k,t} + \sum_{j\in\mathcal{J}}\overline{VTC^{\mathrm{c}}}_{j,t}DM_{j,t}\right) \tag{12j}$$

$$\leq \sum_{t\in\mathcal{T}}\frac{1}{(1+R)^t}\left(\overline{\overline{VTC^{\mathrm{s}}}}_t\sum_{k\in\mathcal{K}}\frac{f_{k,t}}{CV_k} + \sum_{j\in\mathcal{J}}\overline{VTC^{\mathrm{c}}}_{j,t}DM_{j,t}\right) \tag{12k}$$

$$\leq \sum_{t\in\mathcal{T}}\frac{1}{(1+R)^t}\left(\overline{\overline{VTC^{\mathrm{s}}}}_t\frac{\sum_{k\in\mathcal{K}} f_{k,t}}{\min_{k\in\mathcal{K}} CV_k} + \sum_{j\in\mathcal{J}}\overline{VTC^{\mathrm{c}}}_{j,t}DM_{j,t}\right) \tag{12l}$$

$$\leq \sum_{t\in\mathcal{T}}\frac{1}{(1+R)^t}\left(\frac{\overline{\overline{VTC^{\mathrm{s}}}}_t}{\underline{CV}}\sum_{j\in\mathcal{J}} DM_{j,t} + \sum_{j\in\mathcal{J}}\overline{VTC^{\mathrm{c}}}_{j,t}DM_{j,t}\right) \tag{12m}$$

With this result, we can go back to (12f) and rewrite it as:

$$\sqrt{(\Delta x^*)^2 + (\Delta y^*)^2} \leq \frac{\epsilon}{\sum_{t\in\mathcal{T}}\frac{1}{(1+R)^t}\left(\frac{\overline{\overline{VTC^{\mathrm{s}}}}_t}{\underline{CV}}\sum_{j\in\mathcal{J}} DM_{j,t} + \sum_{j\in\mathcal{J}}\overline{VTC^{\mathrm{c}}}_{j,t}DM_{j,t}\right)} \tag{12n}$$

The last step can be applied since the costs and flows are positive, not affecting the sign of the inequality. Therefore, for $\Delta^* = \max(\Delta x^*,\ \Delta y^*)$, we can write

$$\Delta^* \leq \frac{\epsilon}{\sqrt{2}\sum_{t\in\mathcal{T}}\frac{1}{(1+R)^t}\left(\frac{\overline{\overline{VTC^{\mathrm{s}}}}_t}{\underline{CV}}\sum_{j\in\mathcal{J}} DM_{j,t} + \sum_{j\in\mathcal{J}}\overline{VTC^{\mathrm{c}}}_{j,t}DM_{j,t}\right)} \tag{12o}$$

Hence, if the user selects a partitioning of the space $p^*_x \times p^*_y$ such that $\Delta x \leq \Delta^*$ and $\Delta y \leq \Delta^*$ and Δ^*, and Δ^* is bounded by above as in (12o), then the Bilevel Decomposition algorithm converges in the first step. This means that the solution of the Master Problem in $iter = 1$ and the Subproblem in $iter = 1$ yield bounds that satisfy the optimality tolerance $\Phi^{UB} - \Phi^{LB} \leq \epsilon$. □

4.8. Illustrative Example

We illustrate how the algorithm works by solving a test-case using Network 2 from [1], with facility types 1 and 2 (centralized and distributed, respectively), 5 time-periods, 10% increase in demand by time period, and interest factor $R = 0.01$, and optimality tolerance of $\epsilon = 1\%$. We start $iter = 1$ with a $p_x = 2$ and $p_y = 2$ partition of the space, as shown in Fig. 3.

By solving the Master Problem (5) for this grid, we get a $LB^1 = 144,712$, and a solution that builds one centralized facility (type 1), $k = cf_1$, on partition $p = 2$ at time period $t = 1$ and keeps it operating throughout the planning horizon. We then solve Subproblem (9) for these fixed discrete decisions and obtain a solution that builds facility $k = cf_1$ on coordinate $(46.51, 70.76)$,

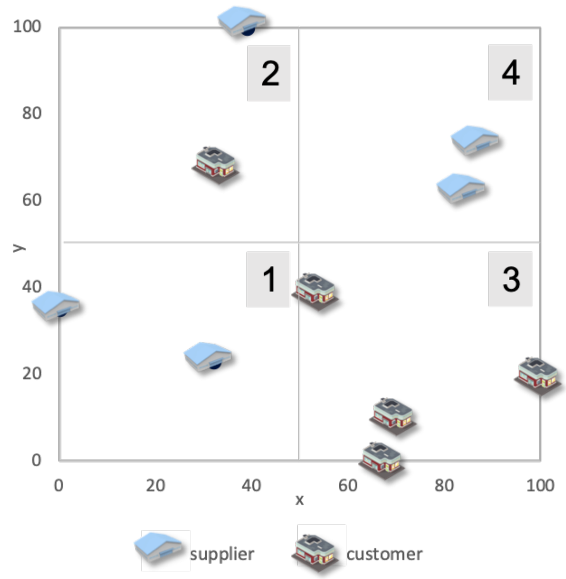

Figure 3: Illustrative problem: iteration 1 ($p_x = 2$, $p_y = 2$)

yielding a feasible upper bound of $UB^1 = 149,236$, and an optimality gap of 3%. This gap is higher than the optimality tolerance, hence we proceed with the algorithm.

Since this is the first iteration, we perform the Facility Pruning step. We start by the second centralized facility $k = cf_2$ which was not built by the Master Problem. By solving the MILP (5) with the additional constraint (10), which enforces that $k = cf_2$ is built, we get $\Phi^{LB,cf_2} = 150,135$ which is higher than the current UB^1, thus we can prune $k = cf_2$ and know that the optimal solution does not have more than one centralized facility. We then continue to solve the Facility Pruning step for the distributed facilities. We start by solving the MILP with the additional constraint (10) for $k = df_1$ and get $\Phi^{LB,df_1} = 144,751$, which is lower than the current UB^1, thus cannot be pruned. We continue doing the same for $k = df_2$ and get $\Phi^{LB,df_2} = 146,267$, which is still lower than the UB^1. We then perform the same step for $k = df_3$ and get $\Phi^{LB,df_3} = 150,168$, which is higher than UB^1, thus we can prune $k = df_3$ and all the remaining distributed facilities, and know that the optimal solution does not have more than two distributed facilities.

The next step is to solve the Partition pruning. We solve the MILP (5) with the additional constraint (11) for $p = \{1, 3, 4\}$, and the results are shown in Table 1. Since none of the $\Phi^{LB,p}$ were higher than $UB^1 = 149,236$, we cannot prune any partition in this iteration. We proceed then to $iter = 2$, with $p_x = 4$ and $p_y = 4$ partition of the space, as represented in Fig. 4, keeping the updated set of potential facilities after pruning.

Based on the solution of the Master Problem and Subproblem for iteration 1, and the mapping

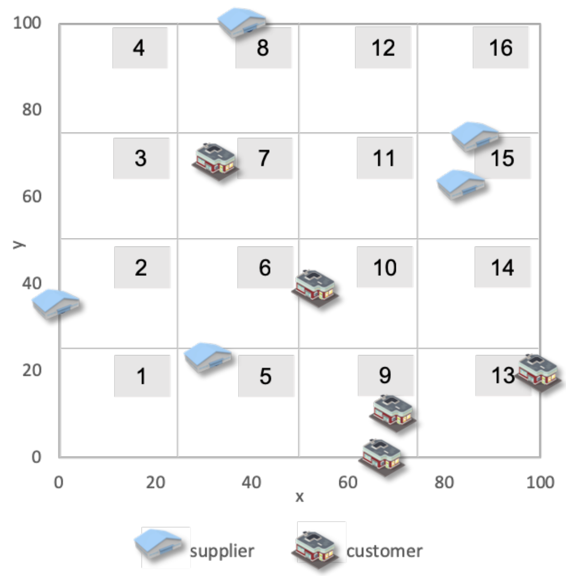

Figure 4: Illustrative problem: iteration 2 ($p_x = 4$, $p_y = 4$)

between partitions in iterations 1 and 2, we warm-start the Master Problem MILP (5) with an initial feasible solution of building $k = cf_1$ on partition $p = 7$. The solution yields $LB^2 = 146,482$, and a solution that builds one centralized facility (type 1), $k = cf_1$, on partition $p = 11$ at time period $t = 1$ and keeps it operating throughout the planning horizon. We then solve Subproblem (9) for these fixed discrete decisions and get a solution that builds facility $k = cf_1$ on coordinate $(50.00, 69.97)$, yielding a feasible that is higher than the previous upper bound, so we keep $UB^2 = 149,236$. The optimality gap is now 2%, which is still higher than the optimality tolerance of 1%.

The following step is to solve the Partition pruning for the current grid. Since the Subproblem builds facility $k = cf_1$ on the boundary between partitions $p = 7$ and $p = 11$, we consider both of them as active and exclude them of the list of partitions to perform the Partition Pruning step. We solve the MILP (5) with the additional constraint (11) for $p = \{1, \ldots, 16\} \setminus \{7, 11\}$ and the respective results are shown in Table 1. Based on the results we can prune the current partitions $p = \{13, 14, 16\}$ and their further refinements.

We proceed to $iter = 3$, with $p_x = 8$ and $p_y = 8$ partition of the space, as represented in Fig. 5. All partitions marked with a stripped pattern were pruned in the previous iteration.

Using the solution of the Master Problem and Subproblem for iteration 2, and the mapping between partitions in iterations 2 and 3, we warm-start the Master Problem MILP (5) with an initial feasible solution of building $k = cf_1$ on partition $p = 30$. The solution yields $LB^3 = 147,805$, and a solution that builds one centralized facility (type 1), $k = cf_1$, on partition $p = 38$ at time

Table 1: Partition Pruning step results (numbers in bold correspond to partitions that were pruned in the respective iteration)

	Iteration 1	Iteration 2
	$UB^1 = 149,236$	$UB^2 = 149,236$
Partition p	$\Phi^{LB,p}$	$\Phi^{LB,p}$
1	146,845	148,146
2	-	147,964
3	144,828	148,146
4	144,828	148,892
5		148,093
6		147,452
7		-
8		146,781
9		149,364
10		147,828
11		-
12		146,994
13		**150,531**
14		**149,646**
15		148,688
16		**149,691**

period $t = 1$ and keeps it operating throughout the planning horizon. We then solve Subproblem (9) for these fixed discrete decisions and obtain a solution that builds facility $k = cf_1$ on coordinate $(50.00, 69.97)$, yielding a feasible that is higher than the previous upper bound, so we keep $UB^2 = 149,236$. The optimality gap is now 0.96%, which is lower than the optimality tolerance of 1%, therefore the algorithm has converged. The lower bound, upper bound and optimality gap at each iteration are reported in Table 2.

As one can see, the lower bound gradually tightens up as the number of iterations $iter$, and consequently the number of partitions increase. The optimal network is shown in Figure 6. It takes 192 seconds to solve this instance on a macOS 2.3 GHz Intel Core i5, using Gurobi 8.0.1

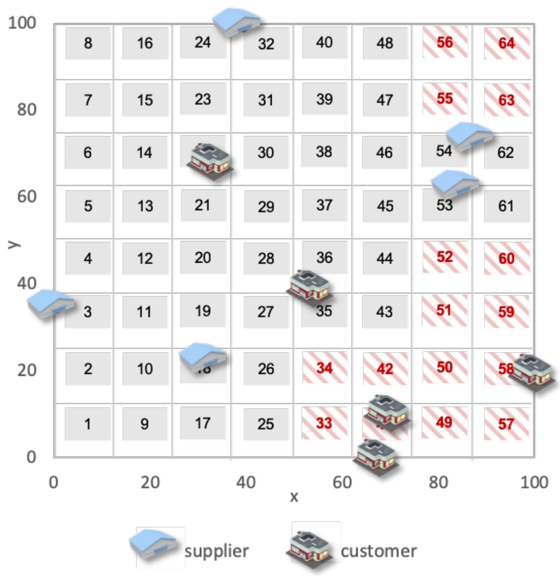

Figure 5: Illustrative problem: iteration 3 ($p_x = 8$, $p_y = 8$)

Table 2: Illustrative test problem results

iter	Lower Bound	Upper Bound	Gap
1	144,712	149,236	3%
2	146,482	149,236	2%
3	147,805	149,236	1%

to solve the MILPs (optimality tolerance of 0.01% for each MILP) and BARON 18.5.8 to solve the nonconvex NLP (time limit of 30 seconds per NLP). For the Facility Pruning and Partition Pruning steps, we limit the solution time of the MILPs to 10 seconds.

To evaluate the impact of each of the proposed steps, we solve the same instance using the Accelerated Bilevel decomposition: (i) without the Facility Pruning step, which takes 2770 seconds; (ii) without the Partition Pruning Step, which takes 211 seconds; and (iii) without the Warm-start step, which takes 196 seconds. This shows the proposed additional steps have an additive effect of the performance of the algorithm, and that the Facility pruning is the step with the greatest impact in the performance for this instance.

It takes 2,778 seconds to solve this same instance with the previous Bilevel Decomposition proposed by [1] using the same p_x, p_y, n_x and n_y. Additionally, BARON 18.5.8 takes 1,835 seconds to solve the original nonconvex MINLP (2) for this instance, while SCIP 5.0 and ANTIGONE 1.1 cannot solve it in 3,600 seconds (remaining optimality gaps of 2% and 4%, respectively).

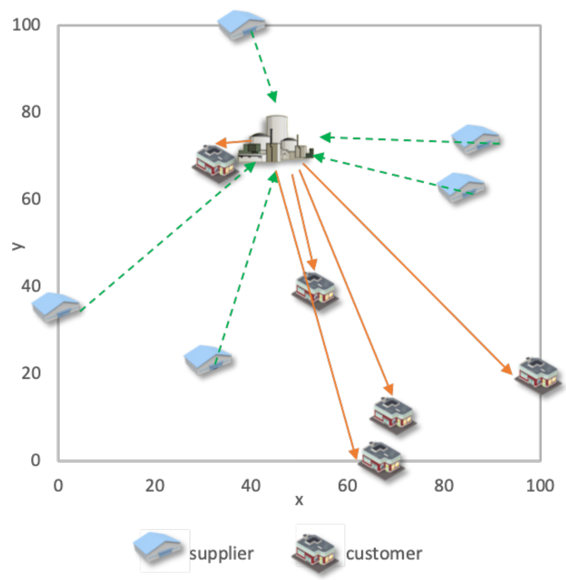

Figure 6: Illustrative problem optimal network

By using Proposition 3, we get that if we start with $p_x, p_y \geq 20$ we have guaranteed convergence within 1% in the first iteration. This is considerably more refined than the $p_x, p_y = 8$ needed for the algorithm to converge, showing that even though Proposition 3 provides a valid bound, it is loose for this case, thus using it may add an unnecessary burden to the solution of the algorithm.

5. Computational results

In order to compare the performance of our proposed accelerated algorithm with the original algorithm and the currently available general purpose global optimization solvers, we 10 test cases from [1]. The network varies in size as follows.

- Network 1: 2 suppliers $\times$ 2 consumers;
- Network 2: 5 suppliers $\times$ 5 consumers;
- Network 3: 10 suppliers $\times$ 10 consumers;
- Network 4: 20 suppliers $\times$ 20 consumers;
- Network 5: 40 suppliers $\times$ 40 consumers;

The 5 network structures are represented in Figures 7-11.

For each of the network options, we use as centralized facilities the Type 1 facilities from Lara et al. [1] (up to 2 large-scale facilities); and as distributed facilities, we first use Type 2 (up to 10

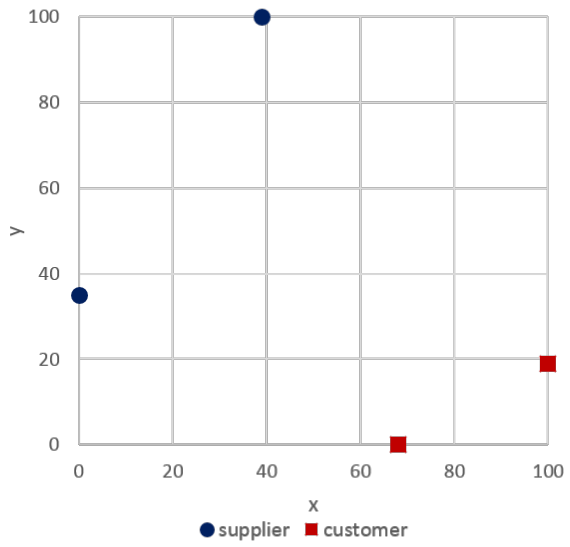

Figure 7: Network 1

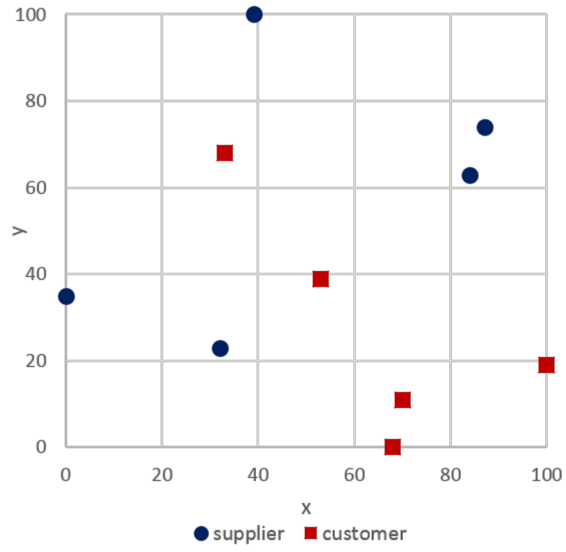

Figure 8: Network 2

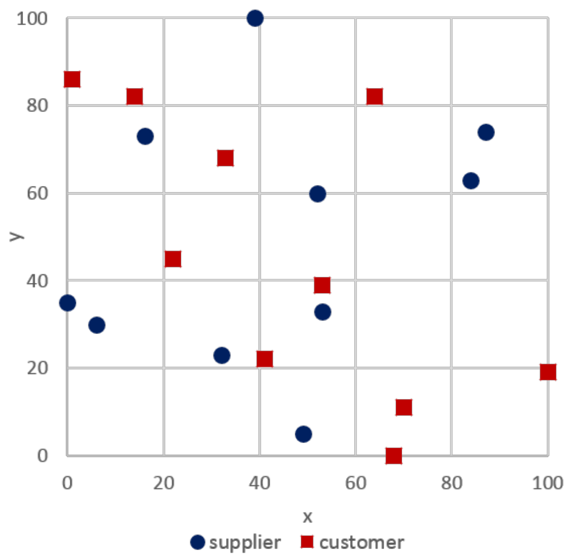

Figure 9: Network 3

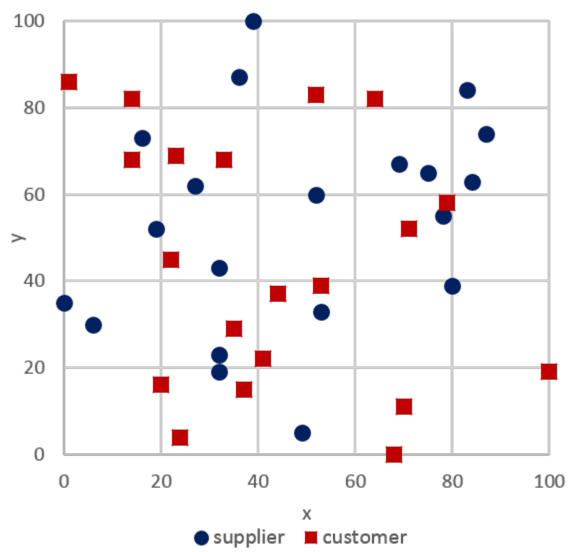

Figure 10: Network 4

mid-scale facilities) and then Type 3 (up to 20 small-scale facilities). Therefore, for each of the network structures, the problem was solved for 12 and 22, respectively.

We assume that all instances are solved for 5 time periods, and the product demand and availability of raw material have a 10% increase per time-period.

Each test case is solved using the Accelerated Bilevel Decomposition (Fig 2), the original Bilevel Decomposition [1], and by general purpose global optimization solvers, BARON, ANTIGONE and SCIP. We set the optimality tolerance to 2% and the maximum total CPU time to 1 hour.

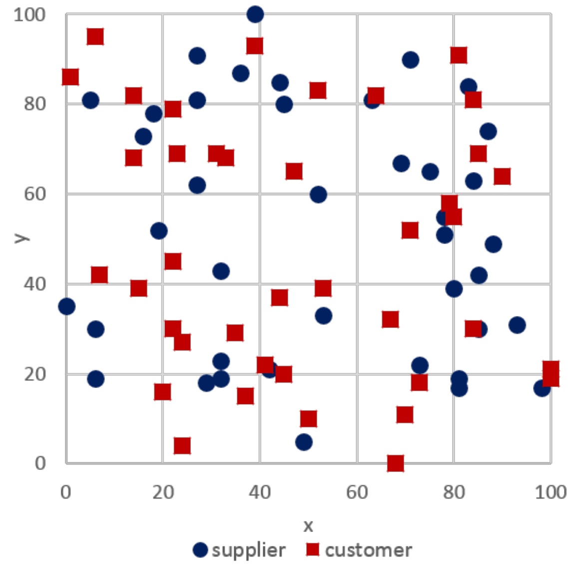

Figure 11: Network 5

Regarding the algorithm, it is required that the Master Problem is solved to 0.5% optimality gap (and we use the lower bound of the MILP as the lower bound in the algorithm), and it is allowed a maximum CPU time of 30 seconds for the solution of each NLP Subproblem. We start the algorithm with a 2×2 partitioning of the space and at each iteration this partitioning is doubled, i.e $N_x, N_y = 2$.

Our computational tests were performed on a MacBook Pro laptop with a 2.3 GHz Intel Core i5, with 8GB of RAM, running on MacOS Mojave. We implemented the monolithic formulation and the global optimization algorithm in Python/Pyomo [44], solving the MILPs using Gurobi version 8.0.1 [45], the NLPs using BARON version 16.3.4 [46], and the MINLPs using BARON version 18.5.8 [46], ANTIGONE 1.1 [47], and SCIP 5.0 [48]. Source code reproducing our results is on Github [49].

The case-studies are named such that the first 2 letters represent the network (i.e., N1, N2, N3, N4, and N5, represent Network 1, 2, 3, 4, and 5, respectively), and the last 2 letters represent the facility types considered (i.e., T1T2 and T1T3 represent types 1 and 2, and types 1 and 3, respectively). The size of monolithic MINLP formulation (2) for each of the test cases is shown in Table 3.

The performance curves for the Accelerated Bilevel Decomposition, the original Bilevel Decomposition from [1] and each of the global solvers are shown in Figure 12.

The results show that the Accelerated Bilevel Decomposition algorithm was able to find the

Table 3: Monolithic MINLP formulation size

	Binary Variables	Continuous Variables	Constraints
N1-T1T2	360	393	1,265
N2-T1T2	720	825	2,087
N3-T1T2	1,320	1,545	3,457
N4-T1T2	2,520	2,985	6,197
N5-T1T2	4,920	5,865	11,677
N1-T1T3	660	703	2,925
N2-T1T3	1,320	1,495	4,407
N3-T1T3	2,420	2,815	6,877
N4-T1T3	4,620	5,455	11,817
N5-T1T3	9,020	10,735	21,697

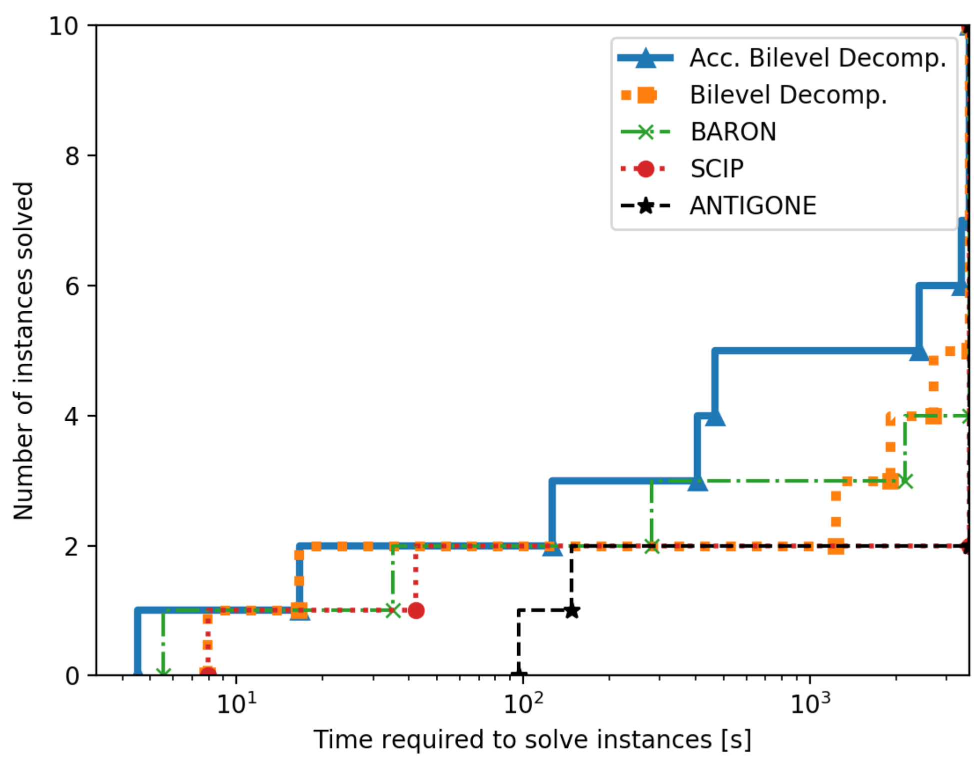

Figure 12: Performance curves comparing the Accelerated Bilevel Decomposition algorithm, with its original version and the commercial global optimization solvers.

optimal solution within 2% optimality tolerance in 70% of the case studies, and performed better (i.e. found the optimal faster) than the other options in all of them. It can be noticed that there was a noticeable improvement in performance between the original Bilevel Decomposition and Our Accelerated version of it, being able to solve 7 out of 10 instances instead of 5 out of 10. The global optimization solver that had the best performance for this problem and these instances was BARON. SCIP and ANTIGONE had a similar performance, only being able to solve 2 out of the 10 instances.

To evaluate the impact of each of the proposed steps, we solve these same 10 instances using the Accelerated Bilevel decomposition: (i) without the Facility Pruning step, (ii) without the Partition Pruning Step, and (iii) without the Warm-start step. The performance curves comparing these options against the proposed Accelerated Bilevel decomposition are shown in Figure 13.

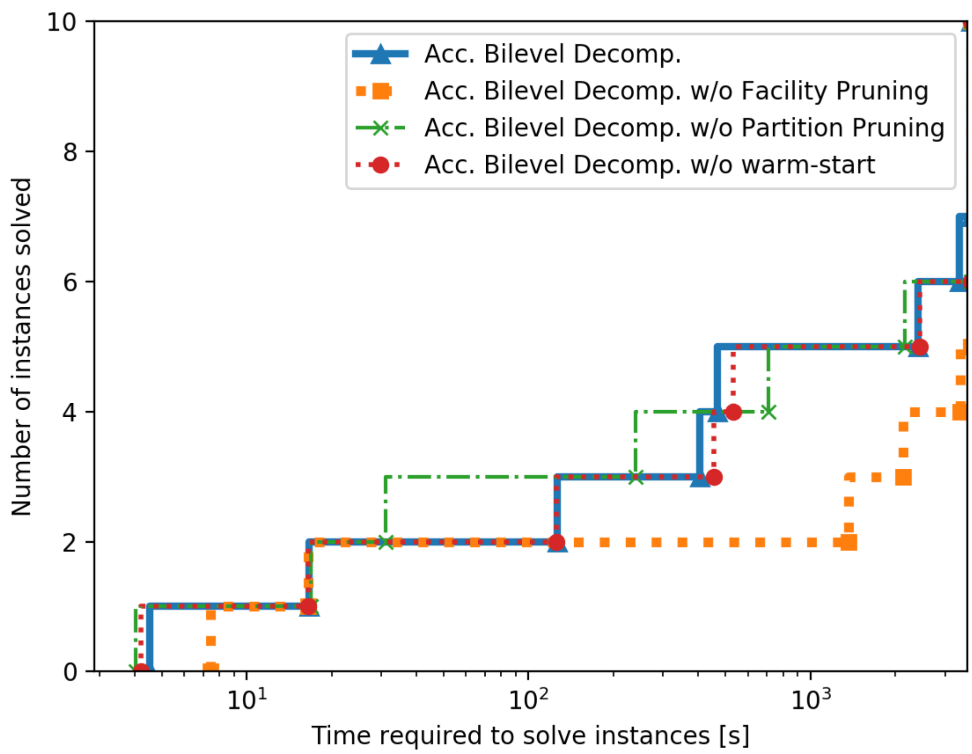

Figure 13: Performance curves comparing the Accelerated Bilevel Decomposition algorithm, with versions without Facility Step, Partition Pruning Step and Warm-start.

The results show that for the smaller instances the absence of each additional step did not have a great impact on performance. However, for larger instances each additional step was necessary to allow the solution of 7 instances. The algorithm without the Partition Pruning and without the Warm-Start could only solve 6 instances within 1 hour, and the algorithm without the Facility Pruning could only solve 5 instances within 1 hour, which shows that this is the step with the greatest impact in the performance. It is interesting to note that for smaller instances not having the Partition Pruning Step reduces the solution time, which makes sense since this can be a time consuming step that hurts the performance of easy instances.

6. Biomass supply chain case study

We present a bioethanol case study, adapted from the literature [16, 4], to illustrate a real-world application for the proposed model and solution strategy. Given are 10 switchgrass suppliers and 10 ethanol markets with locations that are represented in Figure 14. There are 12 potential facilities to be built, of which 10 are distributed ($MC = 40.4$ MGal/year) and 2 are centralized facilities ($MC = 404$ MGal/year). All of the facilities have a conversion of $CV_k = 26\%$. Each market has a demand of 40 MGal of ethanol in the first year, with a 10% increase in demand each of the following years. Each supplier has 500 kilotonnes/year of switchgrass available, with a cost of \$30/ton, \$35/ton, \$33/ton, \$32/ton, \$37/ton, \$40/ton, \$34/ton, \$35/ton, \$31/ton and \$39/ton for suppliers 1 to 10, respectively. The fixed transportation costs ($FTC_{i,k,t}$, $FTC_{k,j,t}$) are \$10,000/year for all the possible links, and the variable transportation costs are \$2/ton-mile for the switchgrass ($VTC_{i,k,t}$) and \$0.40E-3/gal-mile for the ethanol ($VTC_{k,j,t}$). We solve this problem for a 5-year planning horizon.

The resulting model has 3,457 constraints, 1,545 continuous variables, and 1,320 binary variables. Starting with $p_x, p_y = 5$ and $N_x, N_y = 2$ it takes 3 iterations and 6 hours to solve it with the Accelerated Bilevel Decomposition within 2% optimality gap, with an optimal value of \$2.178 billion. We attempted to solve this same instance with BARON (the commercial global solver that has the best performance in the computational experiments in Section 5), but it only achieved 68% optimality gap when it reached the maximum solution time of 10 hours, highlighting again the need for a specialized algorithm such as the proposed Accelerated Bilevel Decomposition to be able to solve real-world applications of this problem.

The optimal network for the biomass supply chain problem is shown in Figure 15 (without the allocation links since it changes according to the time period). All the 10 distributed facilities were built in year 1, and one centralized facility was built in year 2. It is interesting to notice that

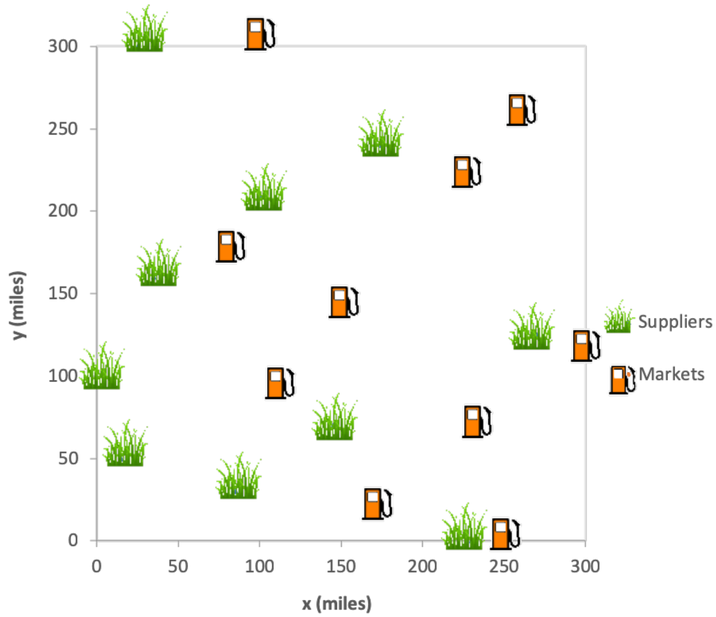

Figure 14: Network structure of the biomass supply chain [16]

in some cases the optimization decides to build 2 distributed modular plants right next to each other instead of replacing them with a larger-scale centralized plant.

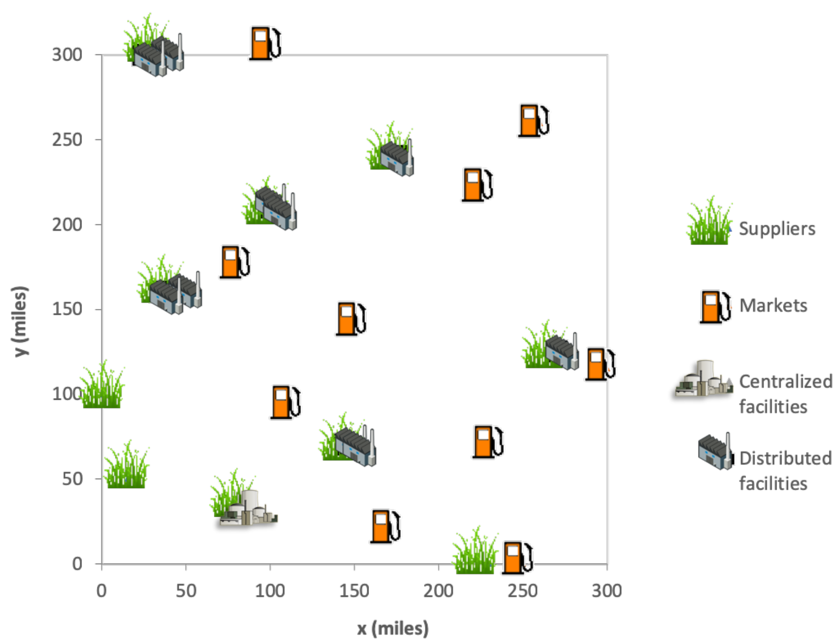

Figure 15: Optimal network for the biomass supply chain

7. Conclusions

This paper has highlighted the need for a general model to optimize the design and planning of Distributed and/or Centralized manufacturing networks. We propose a GDP formulation to solve this problem, which belongs to the class of Capacitated Multi-facility Weber Problem.

We show that with the added complexity of having multi-period decisions the original Bilevel Decomposition proposed by [1] and the available global optimization solvers (BARON, ANTIGONE and SCIP) do not perform well, taking a long time to find feasible solutions and an acceptable optimality gap. Therefore, we propose an accelerated version of the Bilevel Decomposition with additional steps: Facility Pruning, Partition Pruning and Warm-start of the Master Problem. The additional steps do not compromise the rigorousness of the algorithm, which still has ϵ-convergence as proven in [1]. We discuss theoretical properties of the algorithm and find an upper bound to the space discretization such that if the space is partitioned in any finer grid, the algorithm is guaranteed to converge in a single iteration.

Additionally, we perform computational experiments for the multi-period version of the random instances from [1], and show that the proposed Accelerated Bilevel Decomposition outperforms the original Bilevel Decomposition proposed by [1] and the available global optimization solvers (BARON, ANTIGONE and SCIP) in all the instances. Finally, we illustrate the applicability of the model and algorithm by solving a biomass supply chain problem from the literature.

References

[1] C. L. Lara, F. Trespalacios, I. E. Grossmann, Global optimization algorithm for capacitated multi-facility continuous location-allocation problems, Journal of Global Optimization 71 (2018) 871–889.

[2] T. Bieringer, S. Buchholz, N. Kockmann, Future production concepts in the chemical industry: Modular - small-scale - continuous, Chemical Engineering & Technology 36 (2013) 900–910.

[3] D. Ozokwelu, High Efficiency Modular Chemical Processes (HEMCP), Technical Report, U.S. Department of Energy, 2014.

[4] Q. Chen, I. E. Grossmann, Effective GDP optimization models for modular process synthesis, Submitted for publication (2019).

[5] S. Roy, Consider modular plant design, Chemical Engineering Progress (2017) 28–31.

[6] M. Baldea, T. F. Edgar, B. L. Stanley, A. A. Kiss, Modular manufacturing processes: Status, challenges, and opportunities, AIChE Journal 63 (2017) 4262–4272.

[7] M. Eilermann, C. Post, H. Radatz, C. Bramsiepe, G. Schembecker, A general approach to module-based plant design, Chemical Engineering Research and Design 137 (2018) 125 – 140.

[8] G. G. Rogers, L. Bottaci, Modular production systems: a new manufacturing paradigm, Journal of Intelligent Manufacturing 8 (1997) 147–156.

[9] J. A. Moulijn, A. Stankiewicz, J. Grievink, A. Grak, Process intensification and process systems engineering: A friendly symbiosis, Computers & Chemical Engineering 32 (2008) 3 – 11. Process Systems Engineering: Contributions on the State-of-the-Art.

[10] M. Baldea, T. F. Edgar, B. L. Stanley, A. A. Kiss, Modularization in chemical processing, Chemical Engineering Progress 114 (2018) 46–54.

[11] F. You, B. Wang, Life cycle optimization of biomass-to-liquid supply chains with distributedcentralized processing networks, Industrial & Engineering Chemistry Research 50 (2011) 10102–10127.

[12] R. T. Ng, D. Kurniawan, H. Wang, B. Mariska, W. Wu, C. T. Maravelias, Integrated framework for designing spatially explicit biofuel supply chains, Applied Energy 216 (2018) 116 – 131.

[13] R. Mohammad, D. Hartley, D. N. Thompson, Investigating the distributed and centralized preprocessing depot in the supply chain network design of a biorefinery designed for biochemical conversion, 2018 AIChE Annual Meeting Proceedings (2018).

[14] S. H. Tan, P. I. Barton, Optimal dynamic allocation of mobile plants to monetize associated or stranded natural gas, part i: Bakken shale play case study, Energy 93 (2015) 1581 – 1594.

[15] M. Yang, F. You, Modular methanol manufacturing from shale gas: Techno-economic and environmental analyses of conventional large-scale production versus small-scale distributed, modular processing, AIChE Journal 64 (2018) 495–510.

[16] C. L. Lara, I. E. Grossmann, Global optimization for a continuous location-allocation model for centralized and distributed manufacturing, in: Z. Kravanja, M. Bogataj (Eds.), 26th European Symposium on Computer Aided Process Engineering, volume 38 of *Computer Aided Chemical Engineering*, Elsevier, 2016, pp. 1009 – 1014.

[17] J. Brimberg, P. Hansen, N. Mladonovic, S. Salhi, A survey of solution methods for the continuous location allocation problem, International Journal of Operational Research 5 (2008) 1–12.

[18] A. Weber, C. J. Friedrich, Theory of the Location of Industries, University of Chicago Press, Chicago, IL, 1929.

[19] H. W. Khun, R. E. Kuenne, An efficient algorithm for the commercial solution of the generallized Weber problem in spatial economics, Journal of Regional Science 4 (1962) 21–33.

[20] L. Cooper, The transportation-location problem, Operations Research 20 (1972) 94–108.

[21] H. D. Sherali, F. L. Nordai, NP-Hard, capacitated, balanced p-median problems on a chain graph with a continuum of link demands, Mathematics of Operations Research 13 (1988) 32–49.

[22] L. Cooper, The fixed charge problemI: A new heuristic method, Computers and Mathematics with Applications 1 (1975) 89 – 95.

[23] L. Cooper, An efficient heuristic algorithm for the transportation-location problem, Journal of Regional Science 16 (1976) 309.

[24] A. D. Sherali, C. M. Shetty, The rectilinear distance location-allocation problem, A I I E Transactions 9 (1977) 136–143.

[25] H. D. Sherali, C. H. Tuncbilek, A squared-euclidean distance location-allocation problem, Naval Research Logistics (NRL) 39 (1992) 447–469.

[26] H. D. Sherali, I. Al-Loughani, S. Subramanian, Global optimization procedures for the capacitated euclidean and ℓ_p distance multifacility location-allocation problems, Operations Research 50 (2002) 433–448.

[27] J. S. Chen, S. Pan, C. H. Ko, A continuation approach for the capacitated multi-facility Weber problem based on nonlinear SOCP reformulation, Journal of Global Optimization 50 (2011) 713–728.

[28] M. H. Akyüz, T. Öncan, İ. K. Altınel, Branch and bound algorithms for solving the multi-commodity capacitated multi-facility weber problem, Annals of Operations Research (2018).

[29] N. Aras, . K. Altnel, M. Orbay, New heuristic methods for the capacitated multi-facility Weber problem, Naval Research Logistics (NRL) 54 (2007) 21–32.

[30] M. H. Akyüz, T. Öncan, İ. K. Altınel, Solving the multi-commodity capacitated multi-facility Weber problem using lagrangean relaxation and a subgradient-like algorithm, Journal of the Operational Research Society 63 (2012) 771–789.

[31] M. Luis, S. Salhi, G. Nagy, A constructive method and a guided hybrid GRASP for the capacitated multi-source Weber problem in the presence of fixed cost, Journal of Algorithms and Computational Technology 9 (2015) 215–232.

[32] M. Luis, M. F. Ramli, A. Lin, A greedy heuristic algorithm for solving the capacitated planar multi-facility location-allocation problem, AIP Conference Proceedings 1782 (2016) 040010.

[33] M. H. Akyz, Discretization based heuristics for the capacitated multi-facility weber problem with convex polyhedral barriers, An International Journal of Optimization and Control: Theories & Applications (IJOCTA) 8 (2017) 26–42.

[34] C. A. Irawan, M. Luis, S. Salhi, A. Imran, The incorporation of fixed cost and multilevel capacities into the discrete and continuous single source capacitated facility location problem, Annals of Operations Research (2018).

[35] G. P. McCormick, Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems, Mathematical Programming 10 (1976) 147–175.

[36] M. L. Bergamini, P. Aguirre, I. Grossmann, Logic-based outer approximation for globally optimal synthesis of process networks, Computers & Chemical Engineering 29 (2005) 1914 – 1933.

[37] C. E. Gounaris, R. Misener, C. A. Floudas, Computational comparison of piecewiselinear relaxations for pooling problems, Industrial & Engineering Chemistry Research 48 (2009) 5742–5766.

[38] J. P. Vielma, G. L. Nemhauser, Modeling disjunctive constraints with a logarithmic number of binary variables and constraints, Mathematical Programming 128 (2011) 49–72.

[39] R. Misener, J. P. Thompson, C. A. Floudas, Apogee: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes, Computers & Chemical Engineering 35 (2011) 876 – 892. Selected Papers from ESCAPE-20 (European Symposium of Computer Aided Process Engineering - 20), 6-9 June 2010, Ischia, Italy.

[40] S. Kolodziej, P. M. Castro, I. E. Grossmann, Global optimization of bilinear programs with a multiparametric disaggregation technique, Journal of Global Optimization 57 (2013) 1039–1063.

[41] I. M. Al-Loughani, Algorithmic Approaches for Solving the Euclidean Distance Location and Location-Allocation Problems, Ph.D. thesis, Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, 1997.

[42] F. Trespalacios, I. E. Grossmann, Review of mixed-integer nonlinear and generalized disjunctive programming methods, Chemie-Ingenieur-Technik 86 (2014) 991–1012.

[43] M. Fischetti, A. Lodi, Heuristics in Mixed Integer Programming, American Cancer Society.

[44] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, J. D. Siirola, Pyomo Optimization Modeling in Python, volume 67 of *Springer Optimization and Its Applications*, Springer International Publishing, Cham, 2017.

[45] L. Gurobi Optimization, Gurobi optimizer reference manual, 2018.

[46] M. Tawarmalani, N. V. Sahinidis, A polyhedral branch-and-cut approach to global optimization, Mathematical Programming 103 (2005) 225–249.

[47] R. Misener, C. A. Floudas, Antigone: Algorithms for continuous / integer global optimization of nonlinear equations, J. of Global Optimization 59 (2014) 503–526.

[48] A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, F. Serrano, Y. Shinano, J. M. Viernickel, S. Vigerske, D. Weninger, J. T. Witt, J. Witzig, The SCIP Optimization Suite 5.0, Technical Report, Optimization Online, 2017.

[49] C. L. Lara, D. E. Bernal, C. Li, Global optimization algorithm to solve multi-period version of the continuous location-allocation problem (ContFacLocation), `https://github.com/cristianallara/ContFacLocation`, 2019.