



DOMINO: Data-driven Optimization of bi-level Mixed-Integer NOnlinear Problems

Burcu Beykal^{1,2} · Styliani Avraamidou^{1,2} · Ioannis P. E. Pistikopoulos^{1,2} ·
Melis Onel^{1,2} · Efstratios N. Pistikopoulos^{1,2} 

Received: 26 March 2019 / Accepted: 12 February 2020 / Published online: 18 February 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The Data-driven Optimization of bi-level Mixed-Integer NOnlinear problems (DOMINO) framework is presented for addressing the optimization of bi-level mixed-integer nonlinear programming problems. In this framework, bi-level optimization problems are approximated as single-level optimization problems by collecting samples of the upper-level objective and solving the lower-level problem to global optimality at those sampling points. This process is done through the integration of the DOMINO framework with a grey-box optimization solver to perform design of experiments on the upper-level objective, and to consecutively approximate and optimize bi-level mixed-integer nonlinear programming problems that are challenging to solve using exact methods. The performance of DOMINO is assessed through solving numerous bi-level benchmark problems, a land allocation problem in Food-Energy-Water Nexus, and through employing different data-driven optimization methodologies, including both local and global methods. Although this data-driven approach cannot provide a theoretical guarantee to global optimality, we present an algorithmic advancement that can guarantee feasibility to large-scale bi-level optimization problems when the lower-level problem is solved to global optimality at convergence.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10898-020-00890-3>) contains supplementary material, which is available to authorized users.

✉ Efstratios N. Pistikopoulos
stratos@tamu.edu

Burcu Beykal
burcubeykal@tamu.edu

Styliani Avraamidou
styliana@tamu.edu

Ioannis P. E. Pistikopoulos
y.pistikopoulos@gmail.com

Melis Onel
melis@tamu.edu

¹ Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX, USA

² Texas A&M Energy Institute, Texas A&M University, College Station, TX, USA

Keywords Data-driven modeling · Bi-level optimization · Global optimization · Grey-box optimization · Food-energy-water nexus

1 Introduction

Multi-level programming is a class of mathematical optimization with hierarchical structures, where one optimization problem is constrained by other optimization problems. It arises in the presence of multiple decision makers, where each of them is concerned with optimizing its own objective function. As a result, multi-level programming problems are encountered in many different application areas, including supply chain planning [4,36], scheduling [5,8,49], government policy decision [16], price setting problems [33,46] and economics [72], and other multi-stage decision making problems [6,50].

This manuscript presents a data-driven framework for the solution of bi-level mixed-integer nonlinear problems with the general mathematical form shown in Eq. 1. The considered class of problems contain two optimization levels with $F(\mathbf{x}, \mathbf{y})$ and $f(\mathbf{x}, \mathbf{y})$ representing the objective functions of the upper and lower-level problems, respectively. The upper-level problem (ULP) is constrained by the inequality $\mathbf{G}(\mathbf{x}, \mathbf{y})$, whereas the lower-level problem (LLP) is constrained both by the inequality $\mathbf{g}(\mathbf{x}, \mathbf{y})$ and the equality constraint $\mathbf{h}(\mathbf{y})$, where $\mathbf{y}$ is a vector of continuous and/or integer variables strictly controlled by the LLP, and $\mathbf{x}$ is a vector of continuous variables strictly controlled by the ULP. It is worth noting here that the developed framework cannot address bi-level problems with upper-level integer variables, although lower-level integer variables can appear in the ULP.

$$\begin{aligned} \min_{\mathbf{x}} \quad & F(\mathbf{x}, \mathbf{y}) \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0} \\ & \mathbf{y} \in \underset{\mathbf{y}}{\operatorname{argmin}}\{f(\mathbf{x}, \mathbf{y}) : \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq \mathbf{0}, \mathbf{h}(\mathbf{y}) = \mathbf{0}\} \\ & [x_1, \dots, x_n] \in \mathbb{R}^n \\ & [y_1, \dots, y_m] \in \mathbb{R}^m, [y_{m+1}, \dots, y_k] \in \mathbb{Z}^{k-m} \end{aligned} \quad (1)$$

This hierarchical structure can be viewed as a Stackelberg game [71,74] where the upper-level objective will lead and decide on the decision variables $\mathbf{x}$, and the lower-level decision maker will then follow the leader by reacting accordingly, choosing the optimal values for $\mathbf{y}$ to optimize its own objective function. Previously, the solutions of bi-level and multi-level programming problems have been studied extensively using branch and bound algorithms [15, 34,35,44,56] and multi-parametric optimization techniques [7,9–11,28,31,32,64]. Although the aforementioned studies represent important theoretical advances for retrieving either ϵ -optimal or exact solutions of bi-level and multi-level optimization problems, the primary goal of this work is to tackle problems where the deterministic solution strategies cannot be applied due to the highly nonlinear nonconvex nature of many two-level large-scale optimization problems (i.e. problems that contain high number of variables and/or constraints).

To this end, many studies have focused on implementing evolutionary algorithms (i.e., genetic and meta-heuristic algorithms) and trust-region approaches to solve problems with multiple nested layers as presented in the detailed review by Sinha et al. [73]. Although evolutionary algorithms are very-well established and can be applicable to bi-level optimization problems, these methodologies typically require a large number of function evaluations for convergence, which come with a significant computational burden. Furthermore, evolutionary algorithms are generally implemented to unconstrained or box-constrained problems which limit their applicability to many real-life, constrained optimization problems. Extensions of evolutionary algorithms are proposed in the literature for handling constraints using

aggregated approaches, through penalty functions [27,39] or Augmented Lagrangian techniques [70].

In fact, several novel genetic and evolutionary algorithms have been presented for the solution of integer linear bi-level problems [37,63] but both of these studies cannot guarantee global optimality or feasibility. Further advances to genetic algorithms have also been presented for the solution of mixed-integer nonlinear bi-level problems in the last decade [2,38]. However, the study by Hecheng and Yuping [38] is not applicable to bi-level programming problems with general nonlinear lower-level problems. In addition, similar to the integer linear algorithms, these nonlinear genetic algorithms [2,38] cannot also guarantee global optimality or feasibility. As an alternative approach to evolutionary algorithms, Sinha et al. [73] suggested building a local single-level approximation of the bi-level problem using Artificial Neural Networks (ANNs). The authors briefly discuss how local surrogate modeling efforts can be a useful tool for solving bi-level optimization problems. However, the challenges associated with training an ANN, such as the hyperparameter optimization, decisions on the architecture of the network, and the number of samples required for training are not addressed. Therefore, new algorithmic approaches are necessary for solving nonlinear nonconvex bi-level mixed-integer optimization problems with improved constraint handling capabilities and maximum computational efficiency.

Hence, in this work, a new data-driven optimization framework is proposed to alleviate the aforementioned challenges as well as to bridge the gaps in solving a special class of bi-level programming problems, as shown in Eq. 1. To this end, the Data-driven Optimization of bi-level Mixed-Integer NONlinear problems (DOMINO) algorithm is presented where this approach reformulates bi-level optimization problems into single-level approximations through collecting samples on the upper-level objective, while the lower-level is solved to global optimality at these sampling points. This data-driven approach enables the collected input-output information to be utilized by a grey-box optimization solver, where the upper-level objective is solved to optimality via a derivative-free optimization methodology. Through this work, the aim is to:

- Establish a powerful computational algorithm for solving large-scale bi-level mixed-integer nonlinear programming (B-MINLP) problems of the form provided in Eq. 1, which are difficult to solve using deterministic algorithms,
- Test the framework on an extensive list of bi-level optimization benchmark problems,
- Assess the performance of different grey-box solvers on the benchmark problems,
- Utilize the framework for the optimization of a large scale bi-level engineering problem.

This paper is structured as follows. In Sect. 2, a background on constrained grey-box optimization and the DOMINO framework are presented in detail. Furthermore, in Sect. 3, the results for an extensive set of benchmark studies are presented alongside the results of a large-scale case study of land allocation in Food-Energy-Water Nexus problem. Finally, the concluding remarks are provided in Sect. 4.

2 Grey-box optimization and the DOMINO framework

2.1 Grey-box optimization

Grey-box problems are typically encountered in systems that lack closed-form equations or in systems that are defined by high-fidelity models, which may contain a large set of partial differential equations. In such problems, the system of interest is usually subject to; (a)

noise, computational expense and highly nonconvex behavior; and (b) partial availability of the derivatives of the objective function and the constraints as a function of the decision variables. Hence, the direct use of deterministic global optimization methods is challenging and/or restrictive for grey-box systems where the problem characteristics are generally provided in the form of input-output data. A grey-box optimization problem is mathematically defined in Eq. 2, where the analytical form of the objective function $f(\mathbf{x})$ and the constraints $\mathbf{g}_{unknown}(\mathbf{x})$ as a function of the decision variables is unavailable, whereas the inequalities defined by $\mathbf{g}_{known}(\mathbf{x})$ may be present with an explicitly known mathematical form (i.e. known constraints).

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}_{unknown}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{g}_{known}(\mathbf{x}) \leq \mathbf{0} \\ & x_i \in [x_i^L, x_i^U] \quad i = 1, \dots, n \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \quad (2)$$

This class of problems is tackled using data-driven or derivative-free optimization (DFO) techniques where the derivative information of the original formulation is not utilized to get the optimal solution [25]. A typical DFO procedure starts with an initial design of experiments on the decision variables $\mathbf{x}$, which provides a set of pre-determined locations for evaluating the system and collecting the corresponding outputs (objective function value and constraint violations) from the simulated high-fidelity model. This input-output data will further be used by the data-driven optimizer to find the true optimum of the original model either through (a) a purely sample-based methodology; or (b) a hybrid methodology that integrates samples with simple, approximate models (surrogate models). Many algorithmic advances have been made in the last decade for data-driven grey-box optimization of both box-constrained problems [42,58] and general constrained problems [14,21,29] including, the ARGONAUT framework [18,20,23], the ALAMO framework [26,77] and the SO-MI algorithm [57]. Further details on DFO and other algorithmic advances in this field can be found in several recent and valuable review articles and surveys, including a review by Kolda et al. [45] on sample-based methods, by Rios and Sahinidis [67] on box-constrained DFO and comparison of software implementations, by Boukouvala et al. [22] on constrained DFO and by Bhosekar and Ierapetritou [19] and Vu et al. [76] on surrogate-based DFO.

Although grey-box optimization has been predominantly applied to single-level optimization problems, for both constrained and unconstrained problems, we have recently shown that these methods can be extended towards solving constrained multi-objective optimization problems [17]. In this work, we are extending our capabilities that had been established previously for the global optimization of nonlinear nonconvex constrained grey-box problems [18,20,23] towards solving bi-level optimization problems. Our data-driven methodology provides an efficient and consistent way for approximating constrained bi-level mixed-integer nonlinear nonconvex optimization problems, which are challenging to solve using deterministic techniques and/or evolutionary algorithms.

2.2 DOMINO framework

The Data-driven Optimization of bi-level Mixed-Integer NOnlinear problems (DOMINO) framework solves the constrained bi-level mixed-integer nonlinear nonconvex optimization problems following a similar procedure as a generic grey-box optimization algorithm, where the novelty of the work underlies in approximating the bi-level problem into a single-level

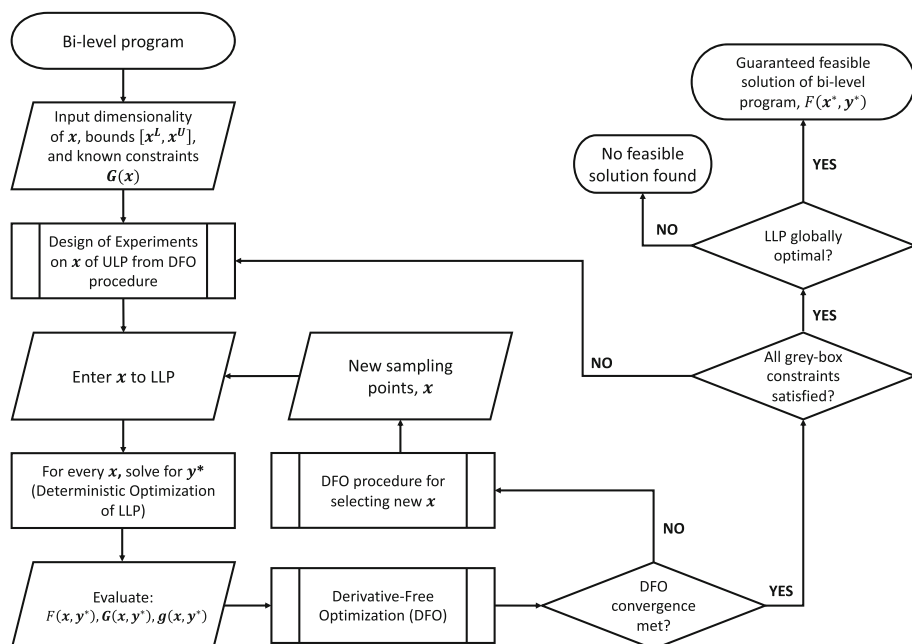


Fig. 1 Algorithmic flowchart of the DOMINO framework. DOMINO is integrated with a DFO algorithm and a deterministic global optimizer for solving bi-level programming problems. The LLP is solved to global optimality at each iteration for a given vector of upper-level decision variables, x (input data). The objective function and the constraint violations (output data) that contain at least one upper-level variable are enumerated using the optimal solution y^* and the corresponding input upper-level decision variables x . This input-output data is later passed to a DFO subroutine to retrieve a candidate solution of the bi-level programming problem

grey-box optimization problem. A general overview of the algorithm is provided in Fig. 1. Given a bi-level programming problem, the first step to DOMINO framework is to pass the dimensionality information of the ULP (i.e. number of upper-level decision variables, n , and their respective bounds) along with any known constraints (i.e. constraints that are explicitly and solely imposed on the upper-level decision variables) to the design of experiments, if the data-driven optimizer can explicitly handle this information. In the absence of such a capability, the known constraints are directly handled as grey-box constraints.

The dimensionality information of the ULP is further processed by the data-driven optimizer to identify an initial starting point or an initial design of experiments at random. The choice of starting with a random initial point or a random design of experiments strictly depends on the type of grey-box solver that is incorporated in the framework. Typically, local black/grey-box solvers, such as a direct search algorithm [3], start with random single initial point whereas global approaches like ARGONAUT [20,23] create a random space-filling maximin Latin Hypercube Design within the provided bounds. Then, at each of these pre-determined candidate locations of x , the corresponding optimal value of the LLP, y^* , is determined using either a local solver such as CPLEX [40], or global MINLP solvers such as ANTIGONE [51–53] and BARON [75], depending on the problem type. CPLEX is implemented for linear (LP), mixed-integer linear (MILP), quadratically constrained (QCP), and mixed-integer quadratically constrained (MIQCP) programming problems, whereas BARON and ANTIGONE are implemented to general nonlinear (NLP) and mixed-integer nonlinear

(MINLP) programming problems at the lower-level. Thus, the LLP is solved deterministically to global optimality at each iteration at the given upper-level sampling points. Later, the optimal solution of the LLP, $\mathbf{y}^*$, and the pre-determined sampling points will be used to enumerate the upper-level objective, $F(\mathbf{x}, \mathbf{y}^*)$, and the constraint violations of both levels, $\mathbf{G}(\mathbf{x}, \mathbf{y}^*)$ and $\mathbf{g}(\mathbf{x}, \mathbf{y}^*)$. This input-output data will be further passed onto the derivative-free optimization stage to retrieve a candidate solution of the original bi-level programming problem once the DFO convergence criteria are met. If this returned solution violates any of the grey-box constraints, the algorithm is restarted to explore a feasible solution, starting with a new initial point/design. If all constraints are satisfied but the LLP is only locally optimal or feasible, then the algorithm will terminate without identifying a feasible solution to the bi-level programming problem. If the solution satisfies all grey-box constraints, and the LLP is globally optimal at the given solution, the solution is a guaranteed feasible point for the original bi-level programming problem.

DOMINO is a flexible algorithm where any type of data-driven optimizer (i.e. local versus global or sample-based versus model-based algorithms) and deterministic solver (i.e. CPLEX, ANTIGONE, BARON) can be incorporated depending on the problem definition. This flexibility allows DOMINO to benefit from the advantages of different approaches and does not impose a strict form on the single-level approximation of different bi-level optimization problems. The most important properties of the DOMINO framework are listed as remarks below.

Remark 1 The proposed framework is tailored to handle special classes of bi-level optimization problems that are given in the form of Eq. 1.

Remark 2 DOMINO cannot guarantee ϵ -global optimality to the upper-level objective. Although commercially available optimization solvers such as CPLEX, ANTIGONE [51–53], and BARON [75] are incorporated within the framework for the deterministic optimization of the LLP, the ULP is treated as a grey-box, where the explicit analytical formulation and the convexity of the problem is assumed to be unknown.

Remark 3 Feasibility of the bi-level programming problem is guaranteed at convergence if and only if a feasible solution for the ULP is identified by DOMINO and the lower-level converges to a globally optimal solution at the given upper-level solution. The feasibility guarantee is achieved by formulating all the upper-level variable-containing constraints, $\mathbf{G}(\mathbf{x}, \mathbf{y})$ and $\mathbf{g}(\mathbf{x}, \mathbf{y})$, as black/grey-box constraints where their respective violations are tracked throughout the DFO procedure. As the LLP is solved to global optimality deterministically at every iteration, the constraints with only lower-level variables (i.e. $\mathbf{h}(\mathbf{y}) = \mathbf{0}$), are satisfied for a feasible solution of an ULP. In addition, the lower-level feasibility is verified through an *a posteriori* analysis for the returned bi-level solution.

Remark 4 DOMINO framework can handle a wide range of dimensionality, including several hundred variables, and constraints in both upper and lower-level problems, and provide feasible near-optimal solutions to varying bi-level programming problem types.

Remark 5 When the optimal solution of the LLP is not unique for the vector of optimal upper-level variables, the decision maker can take a pessimistic decision, an optimistic decision or any decision in between. Although many other bi-level approaches can guarantee and characterize the solution type as pessimistic or optimistic, the proposed framework is not able to provide this characterization.

Table 1 Descriptions and the convergence criteria of data-driven algorithms tested in this study

Algorithm name	Description
NOMAD	Local optimization based on pattern method (search, poll and update). Convergence criteria: maximum number of samples reached, mesh size tolerance reached [48].
COBYLA	Constraint handling via progressive barrier approach. Local optimization using linear approximations for the objective and constraints by interpolation at the vertices of a simplex. Convergence criteria: maximum number of samples reached, minimum trust region radius is exceeded/reached, an optimization step causes a relative change in the decision variables less than the set tolerance [41,66].
ARGONAUT	Global optimization using surrogate model identification for the objective and constraints. Convergence criteria: maximum number of samples reached, no improvement of the incumbent solution over a consecutive set of iterations, all unknown functions are modeled with high accuracy (i.e. very low cross-validation mean squared error) and the incumbent solution is feasible [20].
ISRES	Global optimization via evolutionary method; couples mutation rule and differential variation. Constraint handling via stochastic ranking. Convergence criteria: maximum number of samples reached, an optimization step causes a relative change in the decision variables less than the set tolerance [41].

Remark 6 DOMINO does not impose any extra criterion for convergence or re-sampling. These decisions solely depend on the data-driven optimizer that is integrated within the DOMINO framework and vary from one data-driven methodology to another.

In our previous study [12], we have tested the basic idea of the data-driven approach using a single data-driven optimizer for solving a B-MINLP problem in Food-Energy-Water Nexus considerations. In this work, we are further demonstrating the properties of the framework that are listed here on an extended class of benchmark problems and improve the number of problems solved to global optimality. We extended the framework to include an array of data-driven optimizers, which are presented in the following section. In addition, we are providing the full formulation of the Food-Energy-Water Nexus case study, its reformulation to B-MILP problem using Big-M constraints, as well as its detailed computational study with DOMINO in Sect. 3.2.

3 Computational studies

The proposed data-driven methodology for solving bi-level optimization problems is tested on a challenging set of 100 test problems and a land allocation case study. In this work, we have identified 4 different constrained data-driven optimization strategies that can be implemented in the DOMINO framework: (1) Nonlinear Optimization by Mesh Adaptive Direct search (NOMAD) [47]; (2) Constrained Optimization BY Linear Approximations (COBYLA) [66]; (3) Algorithms for Global Optimization of coNstrained grey-box compUTational problems (ARGONAUT) [18,20,23]; and (4) Improved Stochastic Ranking Evolution Strategy (ISRES) [68]. The selection of these solvers is based on their ability to perform constrained optimization on black/grey-box problems as well as their difference in solution methodology, where both local (NOMAD and COBYLA) and global (ARGONAUT and ISRES) optimization strategies are investigated. Each algorithm is briefly described in Table 1. These DFO solvers

are available and/or implemented in R statistical software. ARGONAUT is implemented in R, the NLOpt implementation of ISRES and COBYLA [41] is available in “nloptr” package in R, and the NOMAD software is available at [1]. All the tested case studies are modeled in GAMS and interfaced through R, where the input-output data collection on each grey-box problem is performed via text files.

All benchmark problems and high-dimensional case studies are executed 10 times on a High-Performance Computing (HPC) machine at Texas A&M High-Performance Research Computing facility using Ada IBM/Lenovo Intel Xeon E5-2670 v2 (Ivy Bridge-EP) HPC Cluster operated with Linux (CentOS 6). COBYLA, ISRES and NOMAD algorithms are executed using 1 node (1 core per node with 64 GB RAM), whereas the ARGONAUT algorithm is executed as a parallel job, using 1 node (20 cores per node with 64 GB RAM) on the supercomputer. Furthermore, for a fair comparison of results, the starting points of COBYLA, ISRES, and NOMAD are randomly generated, as well as the starting initial design of experiments for ARGONAUT is randomly determined for each run. In addition, all data-driven solvers are tested and implemented at their default setting provided from [1,41], with the exception of ARGONAUT. By default, ARGONAUT sets the number of initial sampling points to $10k + 1$ for $k \leq 20$ and to 251 when $k > 20$, where k is the dimensionality of the problem (i.e. number of inputs). Since for $k \leq 2$, the number of initial samples is not sufficient to reveal the input-output relationship for both levels in a bi-level programming problem, the number of initial points to be collected is increased to $40k + 1$. For problems with dimensionality $2 < k \leq 20$ and $k > 20$, the default values are implemented.

3.1 Benchmark studies

The comprehensive test set from Mitsos and Barton [55] (Errata: from Paulavicius et al. [65]), as well as individual bi-level programming problems from Edmunds and Bard [30], Sahin and Ciric [69], Gümüç and Floudas [35], Colson [24], Mitsos [54], Kleniati and Adjiman [43], Woldemariam and Kassa [78], and Nie et al. [59] are used for assessing the performance of the DOMINO framework and for comparing the performance of different data-driven optimizers in finding the true global solution of the bi-level programming problems. In addition to this set, we have randomly generated 61 benchmark studies using the bi-level random problem generator in B-POP toolbox [7] and solved to global optimality, where the formulation of these are provided in the Supplementary Material. The selection of the benchmark problems aim to cover various different types of bi-level optimization problems with varying dimensionalities in both upper and lower-level problems. Especially for the problems generated by B-POP, we have limited ourselves to dimensionalities that this solver can handle, so as to establish a basis for comparison and to be able to assess the performance of DOMINO accurately throughout the benchmark problems.

The dimensionality of each problem and their corresponding properties are provided in Tables 2, 3 and 4, where n represents the number of upper-level continuous variables, m and $k - m$ represents the lower-level dimensionality (continuous and integer, respectively) and n_g^{grey} represents the number of grey-box constraints for each problem. The number of grey-box constraints shown here is the sum of the number of the upper-level constraints and the lower-level constraints that include at least one upper-level variable in its mathematical form. This criterion is imposed since the LLP is solved deterministically within the framework, where the optimal solution already satisfies the constraints with only lower-level decision variables. This allows us to eliminate redundant model building or point search in the optimization phase, which speeds up the computational time required for convergence

Table 2 Dimensionality of continuous bi-level linear benchmark problems tested with DOMINO

Problem ID [Source]	Label	Problem type (upper–lower)	n	m	$k - m$	n_g^{grey}	n_g^y
1 [69]	sc_1	LP-LP	1	2	0	3	0
2 [7]	LPLP1	LP-LP	2	2	0	2	0
3 [7]	LPLP2	LP-LP	2	2	0	5	2
4 [7]	LPLP3	LP-LP	5	5	0	2	0
5 [7]	LPLP4	LP-LP	10	10	0	4	0
6 [7]	LPLP5	LP-LP	20	500	0	350	0
7 [7]	LPLP6	LP-LP	20	20	0	4	0
8 [7]	LPLP7	LP-LP	20	30	0	5	0
9 [7]	LPLP8	LP-LP	20	50	0	7	0
10 [7]	LPLP9	LP-LP	20	80	0	7	0
11 [7]	LPLP10	LP-LP	40	150	0	10	0
12 [7]	LPLP11	LP-LP	50	200	0	20	0
13 [7]	LPLP12	LP-LP	80	90	0	3	0
14 [7]	LPLP13	LP-LP	200	200	0	200	0

Table 3 Dimensionality of continuous bi-level nonlinear benchmark problems tested with DOMINO

Problem ID [Source]	Label	Problem type (upper–lower)	n	m	$k - m$	n_g^{grey}	n_g^y
15 [55]	mb_1_1_06	LP-QP	1	1	0	0	0
16 [7]	LPQP1	LP-QP	30	60	0	10	0
17 [55]	mb_1_1_16	QP-QP	1	1	0	2	0
18 [78]	wk_2015_01	QP-QP	1	1	0	2	0
19 [35]	gf_4	QP-QP	1	1	0	3	0
20 [69]	sc_2	QP-QP	1	1	0	3	0
21 [35]	gf_2	NLP-QP	1	2	0	2	0
22 [55]	mb_2_3_02	NLP-QP	2	3	0	1*	2
23 [55]	mb_1_1_03	LP-NLP	1	1	0	0	0
24 [55]	mb_1_1_04	LP-NLP	1	1	0	0	0
25 [55]	mb_1_1_05	LP-NLP	1	1	0	0	0
26 [55]	mb_1_1_08	LP-NLP	1	1	0	0	0
27 [55]	mb_1_1_09	LP-NLP	1	1	0	0	0
28 [55]	mb_1_1_12	LP-NLP	1	1	0	0	0
29 [55]	mb_1_1_01	LP-NLP	1	1	0	0	2
30 [55]	mb_1_1_02	LP-NLP	1	1	0	1	0
31 [35]	gf_5	LP-NLP	1	2	0	1	1
32 [35]	gf_3	LP-NLP	2	3	0	2	1
33 [55]	mb_1_1_07	QP-NLP	1	1	0	0	0
34 [55]	mb_1_1_10	QP-NLP	1	1	0	0	0
35 [55]	mb_1_1_11	QP-NLP	1	1	0	0	0
36 [55]	mb_1_1_13	QP-NLP	1	1	0	0	0
37 [55]	mb_1_1_14	QP-NLP	1	1	0	0	0

Table 3 continued

Problem ID [Source]	Label	Problem type (upper–lower)	n	m	$k - m$	n_g^{grey}	n_g^y
38 [55]	mb_1_1_17	QP-NLP	1	1	0	0	0
39 [55]	mb_1_1_15	QP-NLP	1	1	0	1	0
40 [35]	gf_1	QP-NLP	1	1	0	2	0
41 [24]	c_2002_01	NLP-NLP	1	1	0	2	0
42 [24]	c_2002_03	NLP-NLP	1	1	0	2	0
43 [24]	c_2002_05	NLP-NLP	1	2	0	2	0
44 [59]	nwj_2017_02	NLP-NLP	2	3	0	1	2
45 [55]	mb_2_3_01	NLP-NLP	2	3	0	3	2
46 [78]	wk_2015_04	NLP-NLP	2	4	0	4	0
47 [78]	wk_2015_06	NLP-NLP	4	4	0	4	0
48 [43]	ka_2014_02	NLP-NLP	5	5	0	4	0
49 [55]	mb_5_5_01	NLP-NLP	5	5	0	4	2
50 [55]	mb_5_5_02	NLP-NLP	5	5	0	4	2

* This constraint is handled as “known” in ARGONAUT runs and as a grey-box constraint for other solvers

for all data-driven algorithms. In addition, we perform an *a posteriori* analysis on the LLP to ensure feasibility of the unmodeled constraints at convergence. The number of constraints with only the lower-level decision variables, hence not presented as grey-box constraints, are also provided in Tables 2, 3 and 4 under n_g^y .

The performance of each solver within DOMINO is assessed based on its efficiency and consistency in identifying the true global optimum of the benchmark studies over multiple repetitive runs. The accuracy and the consistency of each algorithm is evaluated by calculating the normalized mean absolute error ($\% \text{MAE} = 100 \cdot |(F_{best} - F_{global})/F_{global}|$) of the best found solution with respect to the true global optimum and the standard deviation of this error over 10 runs, respectively. In the benchmark problems with $F_{global} = 0$, the percent absolute error ($\% \text{MAE} = 100 \cdot |F_{best} - F_{global}|$) is calculated. It is important to note that 100% MAE is assigned for runs that returned an infeasible solution (constraint violation $\geq 10^{-6}$ and/or lower-level is not globally optimal (lower-level absolute optimality gap > 0 for LP, QP, MILP, MIQP-type lower-level problems and lower-level absolute optimality gap $\geq 10^{-6}$ for NLP and INLP-type lower-level problems) and their respective standard deviation of error is not calculated. Furthermore, the efficiency of the framework is evaluated based on the average elapsed time it takes for each solver to converge and based on the total number of function evaluations (i.e. samples) collected at convergence. The results for continuous linear, continuous nonlinear, mixed-integer linear and mixed-integer nonlinear bi-level programming problems are discussed in Sects. 3.1.1, 3.1.2 and 3.1.3, respectively.

3.1.1 Results for bi-level linear programming problems

The results of the bi-level linear benchmark problems are reported in Table 5. The overall performance of all grey-box solvers, tested as a part of the DOMINO framework, indicate that they return consistent feasible solutions with low errors to the bi-level linear programming (B-LP) problems. Specifically, we observe that NOMAD, as a local sample-based grey-box optimization solver, outperforms the rest of the solvers in B-LP problems. Only in the benchmark problem with the highest number of upper-level variables, NOMAD returns an

Table 4 Dimensionality of bi-level mixed-integer benchmark problems tested with DOMINO

Problem ID [Source]	Label	Problem type (upper–lower)	n	m	$k - m$	n_g^{grey}	n_g^y
51 [54]	am_1_0_0_1_01	LP-ILP	1	0	1	0	0
52 [7]	LPMILP1	LP-MILP	10	10	10	4	0
53 [7]	LPMILP2	LP-MILP	10	10	10	4	0
54 [7]	LPMILP3	LP-MILP	20	20	10	2	0
55 [7]	LPMILP4	LP-MILP	30	30	30	4	0
56 [7]	QPMILP1	QP-MILP	5	5	5	4	1
57 [7]	QPMILP2	QP-MILP	10	5	5	5	0
58 [7]	QPMILP3	QP-MILP	10	10	6	3	0
59 [7]	QPMILP4	QP-MILP	20	10	5	2	3
60 [7]	QPMILP5	QP-MILP	22	12	7	5	0
61 [7]	QPMILP6	QP-MILP	25	20	15	3	0
62 [7]	QPMILP7	QP-MILP	25	25	10	6	0
63 [7]	QPMILP8	QP-MILP	30	120	120	120	0
64 [7]	QPMILP9	QP-MILP	30	200	200	250	0
65 [7]	NLPMILP1	NLP-MILP	5	8	6	9	1
66 [7]	NLPMILP2	NLP-MILP	10	10	10	10	0
67 [7]	NLPMILP3	NLP-MILP	15	15	15	14	1
68 [7]	NLPMILP4	NLP-MILP	20	20	20	20	0
69 [7]	NLPMILP5	NLP-MILP	25	30	30	30	0
70 [7]	NLPMILP6	NLP-MILP	25	50	50	50	0
71 [7]	NLPMILP7	NLP-MILP	30	70	70	70	0
72 [7]	NLPMILP8	NLP-MILP	30	100	100	100	0
73 [7]	NLPMILP9	NLP-MILP	30	200	200	200	0
74 [7]	LPMIQP1	LP-MIQP	7	7	6	1	0
75 [7]	LPMIQP2	LP-MIQP	7	7	6	1	0
76 [7]	LPMIQP3	LP-MIQP	10	7	6	1	0
77 [7]	LPMIQP4	LP-MIQP	10	7	6	1	0
78 [7]	LPMIQP5	LP-MIQP	10	10	6	1	0
79 [7]	LPMIQP6	LP-MIQP	10	13	6	1	0
80 [7]	LPMIQP7	LP-MIQP	10	13	6	1	0
81 [7]	LPMIQP8	LP-MIQP	12	13	6	1	0
82 [30]	eb_1	QP-IQP	1	0	1	3	0
83 [7]	QPMIQP1	QP-MIQP	5	20	10	1	0
84 [7]	QPMIQP2	QP-MIQP	6	5	2	3	0
85 [7]	QPMIQP3	QP-MIQP	6	5	3	4	0
86 [7]	QPMIQP4	QP-MIQP	6	5	5	4	0
87 [7]	QPMIQP5	QP-MIQP	10	3	3	3	0
88 [7]	QPMIQP6	QP-MIQP	10	30	7	1	0
89 [7]	QPMIQP7	QP-MIQP	10	40	7	1	0
90 [7]	NLPMIQP1	NLP-MIQP	5	5	2	0	3
91 [7]	NLPMIQP2	NLP-MIQP	7	5	3	3	0
92 [7]	NLPMIQP3	NLP-MIQP	9	6	3	2	0

Table 4 continued

Problem ID [Source]	Label	Problem type (upper-lower)	n	m	$k - m$	n_g^{grey}	n_g^y
93 [7]	NLPMIQP4	NLP-MIQP	11	7	5	2	0
94 [7]	NLPMIQP5	NLP-MIQP	12	10	10	1	0
95 [7]	NLPMIQP6	NLP-MIQP	12	11	10	0	1
96 [7]	NLPMIQP7	NLP-MIQP	12	11	5	1	0
97 [7]	NLPMIQP8	NLP-MIQP	12	12	6	1	0
98 [7]	NLPMIQP9	NLP-MIQP	13	9	8	1	0
99 [7]	NLPMIQP10	NLP-MIQP	15	15	4	1	0
100 [69]	sc_3	NLP-INLP	2	0	2	0	1

Table 5 Average % MAE and average standard deviation of % MAE for the bi-level linear programming problems. No infeasibility is reported by any of the grey-box solvers for this set of bi-level linear programming problems

Problem ID	Average % MAE				Average Standard Deviation of % MAE			
	NOMAD	COBYLA	ARGONAUT	ISRES	NOMAD	COBYLA	ARGONAUT	ISRES
1	0.0000	16.1538	0.0007	0.0001	0.0000	26.0102	0.0007	0.0001
2	0.0000	0.0000	0.0000	0.0011	0.0000	0.0000	0.0000	0.0032
3	0.0000	8.0448	0.0000	0.0001	0.0000	10.3858	0.0000	0.0001
4	0.0000	0.0388	0.0000	0.0000	0.0000	0.1225	0.0000	0.0000
5	0.1044	6.4958	11.2746	0.0000	0.0960	8.8862	6.9450	0.0000
6	0.0000	4.6180	16.1528	0.0287	0.0000	9.3016	14.1927	0.0119
7	0.2804	6.4321	1.3349	0.1767	0.1462	4.6339	0.6668	0.0427
8	0.0000	0.0000	0.0000	0.1018	0.0000	0.0000	0.0000	0.0336
9	0.0000	0.0000	0.0000	0.0830	0.0000	0.0000	0.0000	0.0107
10	0.0000	0.0000	0.0000	0.1493	0.0000	0.0000	0.0000	0.0283
11	0.0000	0.0000	0.0000	1.4393	0.0000	0.0000	0.0000	0.1732
12	0.0000	0.0000	0.0000	1.8667	0.0000	0.0000	0.0000	0.1101
13	0.0001	0.0584	0.0664	30.2788	0.0004	0.1838	0.0975	1.3798
14	6.9641	0.0000	0.0779	57.6624	1.4234	0.0000	0.2463	1.0838

objective value with more than 5% average MAE. A similar trend is also observed in the ISRES algorithm, where at higher upper-level dimensionality benchmarks (i.e. 80 and 200 upper-level variables) the algorithm converges with high % MAE. One possible reason for this behavior in sample-based methodologies is reported in Fig. 2b, where both NOMAD and ISRES algorithms converge and return the incumbent solution after hitting the maximum number of samples allowed (i.e. 10^5 samples) in all computational studies. Hence, by allowing these algorithms to collect more samples at high-dimensional B-LP problems, it is possible to get more consistent solutions with lower errors. Specifically, in problem 5 (“LPLP4”), we observe that the ISRES algorithm hits the maximum number of samples even though a solution with 0.0000 average % MAE and 0.0000 average standard deviation of % MAE is found. This is due to the fact that the tolerance set for the criterion that defines the convergence with respect to the relative change in the decision variables is not met. Another optimization step taken by ISRES will result in a relative change in the decision variables that is greater than

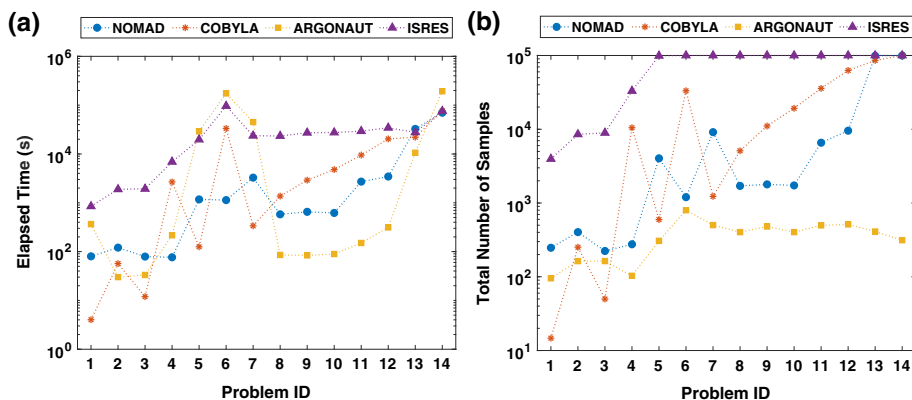


Fig. 2 **a** Average elapsed time for solving bi-level linear programming problems; **b** Average total number of samples collected by each solver in bi-level linear programming problems

10^{-6} . Hence for this specific case, ISRES algorithm terminates by reaching the maximum number of samples allowed. On the other hand, we observe that model-based algorithms, such as COBYLA and ARGONAUT, can provide consistent near-optimal solutions to these high-dimensional B-LPs. However in certain benchmark problems, these methodologies may return solutions with higher % MAE, where also a higher variability is observed among 10 repetitive runs of these test problems.

In addition to the solution accuracy of each grey-box solver tested as a part of the DOMINO framework, we compare the computational performance of each methodology with respect to the total elapsed time for convergence and the average number of samples collected at convergence (Fig. 2). The overall computational performance of all solvers, shown in Fig. 2a, b, indicates that the computational requirements for the DOMINO increases as the ULP dimensionality increases. This is an expected result since the computational efficiency of all grey-box solvers will highly depend on the number of decision variables and the grey-box constraints handled by these algorithms. Although the overall trend shows an increase in computational expense with increasing upper-level dimensionality, Fig. 2a shows that the total elapsed time for DOMINO is comparable when using NOMAD, COBYLA or ARGONAUT algorithms as the preferred grey-box solvers within the framework. On the contrary, the elapsed computational time for the ISRES algorithm is at least an order of magnitude higher for most of the B-LP benchmark problems when compared to other solvers. This is mainly because the solution strategy of the ISRES algorithm dictates significantly higher number of samples for convergence for all B-LP problems, where this in return increases the computational requirements for DOMINO, as shown in Fig. 2b. It is also important to note that the computational time for solving the LLP in B-LP benchmark problems is minimal. On average, the amount of time required to solve the LLP took 0.013–0.065 seconds per sample. For example, for the ARGONAUT algorithm, we observe that the total sampling time (i.e. total time spent to solve the LLP for a given B-LP benchmark problem) accounted for less than 9% of the total elapsed time spent for convergence. For this grey-box solver, the parameter estimation and the surrogate model optimization stages accounted for at least 59% of the total elapsed time, showing that the grey-box optimization stage was computationally much more expensive than solving the LLP at different sampling points.

The overall results demonstrate that NOMAD, as a sample-based local grey-box solver, is more favorable to be incorporated in the DOMINO framework for solving B-LP problems.

NOMAD is shown to achieve highly consistent solution accuracy with good computational efficiency compared to other methodologies. In spite of that, it is important to note that the incumbent solution obtained at convergence from all algorithms in the DOMINO framework are guaranteed feasible solutions to the B-LP problems, as all constraints, including the optimality of the LLP, are satisfied.

3.1.2 Results for continuous nonlinear bi-level programming problems

In addition to the B-LPs, we have also extensively tested the DOMINO framework with continuous bi-level nonlinear programming (B-NLP) problems. The results of this computational study are provided in Table 6. The overall results show that in B-NLP problems, the global methodologies outperform local solution strategies. Global grey-box solvers, namely ARGONAUT and ISRES, solve more benchmark problems with lower % MAE and with lower standard deviations of this error. ISRES solves 30 benchmark problems with less than 5% MAE and ARGONAUT solves 28 in the same error range out of the 36 benchmark problems tested. This number drops to 23 and 14 for NOMAD and COBYLA, respectively. Especially, the deteriorating performance of COBYLA is somewhat expected since this algorithm uses linear approximations for the objective function and constraints. In many of these B-NLP case studies, the linear approximations constructed by COBYLA are not sufficient to capture the nonlinear relationship in the input-output data. Hence, DOMINO is more prone to converging to suboptimal solutions in B-NLP benchmark problems when COBYLA is preferred over other solvers.

Furthermore, Table 6 provides a more detailed overview on DOMINO's accuracy and consistency in solving many challenging B-NLP problems. In the LP-QP test problems, we observe that for problem 16 ("LPQP1") NOMAD, COBYLA and ARGONAUT converge consistently to the true global solution over multiple repetitive runs, whereas ISRES converges to a near-optimal solution with less than 5% MAE. For benchmark 15 ("mb_1_1_06"), we observe that DOMINO returns feasible solutions with high % MAE regardless of the grey-box solver of choice. The underlying reason for this inferior performance by DOMINO is due to the fact that the problem is degenerate. The optimal solution to the bi-level problem exists at $x = 0$, where all points for $y \in [-1, 1]$ are trivially optimal [55]. However, for $-1 \leq x < 0$ the unique global solution exists at $y = -1$ and for $0 < x \leq 1$ the unique global minimum is at $y = x^2$. Hence, the data-driven algorithms tend to go to either unique optimal solution at the lower-level ($y = -1$ or $y = x^2$) due to the deterministic optimization step taken by the DOMINO at provided sampling points for x . As a result, higher deviations are observed in DOMINO solutions compared to the true global solution. It is also important to note that for this class of bi-level benchmark problems, all grey-box solvers provide guaranteed feasible solutions as the LLP returns the global optimum and a feasible solution to the grey-box problem is identified at convergence (Remark 3).

In the QP-QP problem set, the results indicate that global solvers can provide consistent near-optimal solutions to these benchmark problems. Especially, ISRES algorithm consistently converges to the true optimal solution in 3 out of 4 QP-QP benchmark problems. However, local methodologies (NOMAD and COBYLA) converge to suboptimal solutions with high variability. Moreover, it is important to note that NOMAD's standard deviation of the % MAE for problem 17 ("mb_1_1_16") is not reported since this algorithm has returned an infeasible solution in 1 of the 10 random runs. In this case, the lower-level optimality is satisfied, however, one of the grey-box constraints is violated. In addition, it is important to highlight that a better solution for the problem 18 ("wk_2015_01") is identified by the DOMINO framework. Different decision variables are identified at the LLP with an improved

Table 6 Average % MAE and average standard deviation of % MAE for continuous nonlinear bi-level benchmark problems

Problem ID	Average % MAE				Average Standard Deviation of % MAE			
	NOMAD	COBYLA	ARGONAUT	ISRES	NOMAD	COBYLA	ARGONAUT	ISRES
LP-QP								
15	90.0046	70.0139	100.0000	30.0175	31.6081	48.2822	0.0000	48.2929
16	0.0000	0.0000	0.0000	4.1468	0.0000	0.0000	0.0000	0.6435
QP-QP								
17	89.6785	56.8518	3.3533	0.0314	–	101.4962	3.0133	0.0007
18 [†]	0.0000	0.0000	0.8891	0.0001	0.0000	0.0000	1.2213	0.0001
19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	78.4000	0.0077	0.0000	0.0000	28.6713	0.0161	0.0000
NLP-QP								
21	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000	0.0005
22	17.2004	59.1898	15.8680	6.3927	13.2570	32.6424	9.8335	3.0217
LP-NLP								
23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	FS*	FS*	FS*	FS*	32.9404	32.2749	0.0000	0.1002
25	0.0141	0.0252	0.5802	0.0141	0.0000	0.0352	0.4652	0.0000
26	0.0000	10.0001	0.0000	0.0001	0.0000	31.6227	0.0000	0.0002
27	0.0000	5.0002	0.0000	0.0002	0.0000	15.8114	0.0000	0.0004
28	31.4656	FS*	0.1973	0.0293	47.3844	121.6939	0.3698	0.0000
29	1.0000	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000
30	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001
31	0.0009	10.0007	0.8179	0.0007	0.0000	–	1.4561	0.0003
32	14.5205	29.9388	12.8843	0.0000	–	–	5.8265	0.0000
QP-NLP								
33	0.0000	90.0002	1.4847	0.0000	0.0000	144.9136	1.1691	0.0001
34	0.0000	0.0000	0.0035	0.0000	0.0000	0.0000	0.0069	0.0000
35	40.0000	20.0000	0.0000	0.0024	51.6398	42.1637	0.0000	0.0068
36	54.0004	FS*	2.6282	0.0005	88.4684	140.8542	2.1959	0.0003
37	0.0024	0.0024	0.0165	0.0024	0.0000	0.0000	0.0214	0.0000
38	83.3109	83.3109	83.3109	83.3109	0.0000	0.0000	0.0000	0.0000
39	0.0024	0.0024	0.0326	0.0024	0.0000	0.0000	0.0625	0.0000
40	1.1953	1.4353	0.0001	0.0000	1.2599	1.2353	0.0002	0.0000
NLP-NLP								
41	1.1490	1.1490	1.1490	1.1490	0.0000	0.0000	0.0000	0.0000
42	0.0000	20.0000	0.0000	0.0007	0.0000	–	0.0000	0.0008
43	0.0084	10.9125	0.0867	0.0084	0.0000	9.3847	0.1319	0.0000
44	9.9140	79.9494	5.8744	0.7041	19.5323	–	4.5579	0.0774
45	27.3509	37.5217	0.1481	0.0004	35.3098	40.7806	0.4682	0.0006
46 [‡]	56.6928	62.3959	FS*	64.7918	39.1217	–	–	34.1483
47 [§]	FS*	FS*	FS*	FS*	–	–	–	–

Table 6 continued

Problem ID	Average % MAE				Average Standard Deviation of % MAE			
	NOMAD	COBYLA	ARGONAUT	ISRES	NOMAD	COBYLA	ARGONAUT	ISRES
48	0.0000	16.0616	0.0054	2.8125	0.0000	–	0.0172	4.5286
49	0.0025	40.0776	4.0394	0.0025	0.0000	43.6221	2.0734	0.0000
50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Number of infeasible solutions reported out of 10 runs: by NOMAD for problem 17 (“mb_1_1_16”) is 1, for problem 32 (“gf_3”) is 1, for problem 47 (“wk_2015_06”) is 4; by COBYLA for problem 31 (“gf_5”) is 1, for problem 32 (“gf_3”) is 1, for problem 42 (“c_2002_03”) is 2, for problem 44 (“nwj_2017_02”) is 1, for problem 46 (“wk_2015_04”) is 3, for problem 47 (“wk_2015_06”) is 9, problem 48 (“ka_2014_02”) is 1; by ARGONAUT for problem 46 (“wk_2015_04”) is 1, for problem 47 (“wk_2015_06”) is 1; by ISRES for problem 47 (“wk_2015_06”) is 8

*Feasible solution with more than 100% MAE on average is returned at convergence

† % MAE calculated with respect to the best solution found by DOMINO ($F_{best} = 99.9955$)

‡ % MAE calculated with respect to the best solution found by DOMINO ($F_{best} = 0$)

§ % MAE calculated with respect to the best solution found by DOMINO ($F_{best} = 4.5078 \cdot 10^{-6}$)

objective function value compared to the ones reported by Woldemariam and Kassa [78]. Thus, the solution reported by this study [78] does not meet the optimality condition of the lower-level where the overall solution becomes infeasible for this B-NLP problem. The best found solution by DOMINO is reported in the Appendix.

In the NLP-QP problem set, a similar trend is observed where global solvers outperform the local grey-box solution strategies. For problem 22 (“mb_2_3_02”), the global optimization step taken at the lower-level returned the optimal solution to all repetitive runs of the 4 grey-box solvers tested as a part of the DOMINO framework. However, due to the nonconvexity at the upper-level, we observe that the local solvers converge to suboptimal solutions and yield higher % MAE values with higher deviations. Hence, we observe that the global exploration of candidate sampling points by ARGONAUT and ISRES leads to improved solution accuracy in this challenging B-NLP problem.

Similarly, in the LP-NLP problem set, the overall performance of ARGONAUT and ISRES show that these solvers are more favorable to be incorporated into the DOMINO framework for solving B-NLP problems, as they provide highly consistent and accurate solutions to these case studies. In several benchmark problems, we observe that NOMAD and COBYLA return highly variable solutions with a high % MAE. Especially for problems 31 (“gf_5”) and 32 (“gf_3”), COBYLA returns 1 infeasible solution out of 10 repetitive runs of these bi-level problems. In case of the NOMAD algorithm, an infeasible solution is returned for problem 32 (“gf_3”). In addition, it is important to note that, for problem 24 (“mb_1_1_04”) all grey-box solvers provide feasible solutions with more than 100% MAE with respect to the true global solution. In this case, we observe that the upper-level objective consists of the lower-level variable, y , and the inner objective is parametrized in x . As a result, our data-driven approach can detect the unique global minimum for the inner objective, which is $y^* = 0.5$ for $x > 0$ and $y^* = 1$ for $x < 0$. However, none of the data-driven solvers can pinpoint the unique optimal solution of this bi-level problem at $x = 0$ where any $y \in [-0.8, 1]$ is trivially optimal. The main reason behind this issue is that the LLP is degenerate and the piecewise nature of the input-output data hinders the information collected at the sampling stage. Even though various points are sampled, with different x values, the corresponding upper-level objective is either 0.5 or 1. As a result, the solvers terminate the optimization procedure after several consecutive iterations, since there is no improvement to the best found objective

as new sampling points are added. Hence, DOMINO fails to pinpoint the unique optimal solution to this benchmark problem.

Furthermore, in the QP-NLP problem set, the global grey-box solvers continue to provide optimal or near-optimal solutions consistently to many B-NLPs of this type. However, in problem 38 (“mb_1_1_17”), we recognize that all solvers consistently converge to the same suboptimal solution. The main reason for this is that the LLP has two global minima with the objective function value of zero and $y = 1 + 0.1x \pm 0.5\sqrt{2} + 2x$. By default, the negative counterpart is used for computing y , whereas the optimal solution reported in Mitsos and Barton [55] uses the positive counterpart for the inner problem. Hence, all the grey-box solvers converge to the same suboptimal solution and the results reported in Table 6 reflect the errors based on the negative counterpart of y . However, if we strictly constrain y to the positive counterpart, then all the grey-box solvers will identify a near-optimal solution with 0.0161 average % MAE and 0.0000 average standard deviation of % MAE. This observation is also consistent with Remark 5, where DOMINO cannot characterize pessimistic, optimistic and other types of decisions in the presence of multiple optima at the lower-level.

Finally for the NLP-NLP type bi-level problems, we observe that global solvers return consistent feasible near-optimal solutions whereas the local solvers are prone to converging to suboptimal solutions in a portion these nonconvex B-NLPs. This difference is also supported by the standard deviation values of the % MAE provided in Table 6, where high values of the deviation indicates that in a portion of the repeated test runs, these local solvers can find a feasible near-optimal solution, whereas in the rest they converge to feasible suboptimal solutions that are distant to the true global solution. However, it is important to state that, COBYLA struggles to find feasible solutions in 50% of the NLP-NLP type benchmark problems. As this algorithm uses linear approximations, using the COBYLA algorithm within the DOMINO framework is not favorable for solving nonconvex nonlinear bi-level programming problems. We also observe that ARGONAUT returns an infeasible result for problems 46 (“wk_2015_04”) and 47 (“wk_2015_06”), whereas NOMAD and ISRES return infeasible solutions to problem 47 (“wk_2015_06”). Both of these case studies are particularly challenging since they contain the absolute value function, where the derivative of the objective/constraints is discontinuous. Nonetheless, it is important to note that, for both of these benchmark problems, out of 10 random runs for each solver, a better objective function value is found than the solution reported in Woldemariam and Kassa [78]. This is possible since the lower-level optimality in this study [78] was not satisfied at the provided optimal solution, hence making the reported solution an infeasible point for both of these bi-level programming problems. The best found solutions by DOMINO for these benchmark problems are reported in the Appendix.

Computational performance of DOMINO is also provided in Fig. 3. As expected, the elapsed time for local solvers is significantly less than the global ones (Fig. 3a). Specifically, ISRES stands out as the most computationally demanding methodology both in the time required to retrieve the optimal solution as well as the total number of samples required for convergence, where in many instances it hits the maximum number of function evaluations (10^5 samples) allowed for the algorithm, as shown in Fig. 3b. This occurrence is due to the evolutionary nature of this method, as ISRES requires too many samples for convergence, even for the lower dimensional and relatively simpler benchmark problems. This is followed by the ARGONAUT algorithm where in certain benchmark problems the time required for convergence is higher, where in others the overall performance is comparable to local methodologies. The computation time required to solve the continuous nonlinear lower-level problems is minimal similar to the B-LP benchmark problems with the exception of problem 47 (“wk_2015_06”). On average, the computational expense for solving the lower-

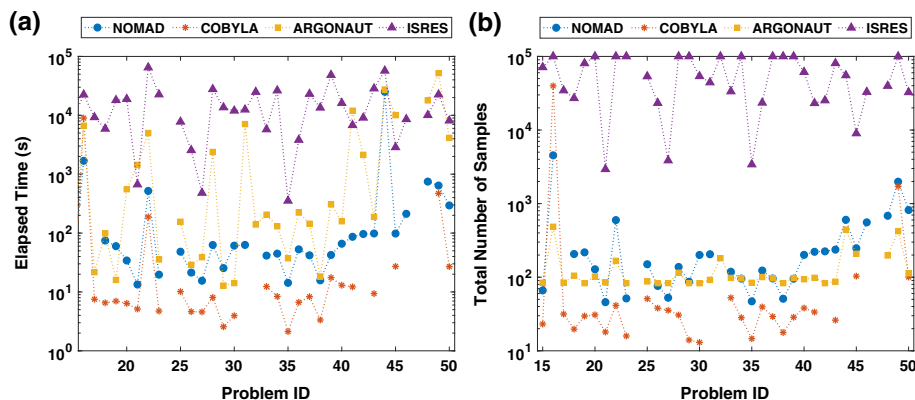


Fig. 3 **a** Average elapsed time for solving continuous bi-level nonlinear programming problems; **b** Average total number of samples collected by each solver in continuous bi-level nonlinear programming problems

level varies between 0.0171–5.5514 seconds and the overall contribution of sampling to the total elapsed time varies between 0.03–18.9%. Specifically, in problem 47 (“wk_2015_06”), the average computational time required to solve the LLP is 88.789 seconds with an overall contribution of 50.9% in total elapsed time. As this problem is more challenging to optimize due to the discontinuous derivatives at the lower-level, we observe a higher contribution from the sampling phase to the overall DFO procedure than the grey-box optimization phase. On the contrary, for the other B-NLP problems, the grey-box optimization phase (i.e. surrogate model building and its respective optimization) is the most computationally demanding step in ARGONAUT’s solutions. As for the sampling requirements, ARGONAUT collects fewer samples than the ISRES algorithm, since ARGONAUT is a model-based grey-box solver. The overall results show that COBYLA is the most computationally efficient methodology; however, this solver was unable to provide consistent feasible solutions to several B-NLP benchmark problems. Although the ARGONAUT and ISRES are computationally more expensive to execute, it is possible to retrieve optimal or near-optimal solutions more consistently through using these global data-driven solvers in DOMINO for B-NLP problems.

3.1.3 Results for bi-level mixed-integer programming problems

The results for the bi-level mixed-integer programming problems are summarized in Table 7. For this class of problems, we observe that sample-based grey-box solvers outperform model-based methodologies. DOMINO can identify optimal or near-optimal solutions consistently to various types of bi-level mixed-integer programming problems when using NOMAD as the grey-box solver of choice. NOMAD almost perfectly returns solutions with low errors where only in 1 benchmark problem this algorithm returns a suboptimal feasible solution. Likewise, the ISRES algorithm is very successful in finding near-optimal solutions, but struggles in finding near-optimal solutions in higher dimensional benchmark problems. It is also important to highlight that NOMAD, ARGONAUT and ISRES identify feasible solutions in all of the bi-level mixed-integer programming problems tested. However, COBYLA fails to identify a feasible solution in 1 of the 10 repetitive runs of benchmark 57 (“QPMILP2”).

Furthermore, the computational performance of DOMINO in solving bi-level mixed-integer programming problems is summarized in Fig. 4. Figure 4a shows that ISRES requires an order of magnitude higher time for convergence compared to other algorithms, and

Table 7 Average % MAE and average standard deviation of % MAE for bi-level mixed-integer benchmark problems. Infeasible solutions reported: by COBYLA for problem 57 (“QPMILP2”) in 1 out of 10 runs

Problem ID	Average % MAE				Average Standard Deviation of % MAE			
	NOMAD	COBYLA	ARGONAUT	ISRES	NOMAD	COBYLA	ARGONAUT	ISRES
LP-MILP								
51	0.0000	0.0000	0.0000	0.0009	0.0000	0.0000	0.0000	0.0018
52	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
53	0.3050	0.0000	0.8756	0.0000	0.8052	0.0000	1.3089	0.0000
54	0.0000	1.7135	8.3347	0.2276	0.0000	3.7096	5.2589	0.0846
55	0.0000	0.0004	0.0000	2.8790	0.0000	0.0011	0.0000	0.5561
QP-MILP								
56	0.0000	0.0028	0.0365	0.0000	0.0000	0.0088	0.1151	0.0000
57	0.0074	28.4286	2.3668	0.0002	0.0042	–	1.7631	0.0002
58	0.0000	18.9344	0.0000	0.0000	0.0000	57.5193	0.0000	0.0000
59	0.0000	FS*	FS*	FS*	0.0000	$> 10^5$	791.0469	$> 10^3$
60	0.0000	FS*	7.9741	2.5208	0.0000	220.4544	2.5821	1.3129
61	0.0000	FS*	55.6621	50.5639	0.0000	$> 10^3$	49.1259	27.5322
62	0.0000	2.8949	0.4577	0.5772	0.0000	5.3286	0.2089	0.1691
63	0.0000	FS*	FS*	36.5575	0.0000	364.1700	188.6630	5.9444
64	0.0000	26.7727	FS*	8.9426	0.0000	23.3885	116.9957	2.1176
NLP-MILP								
65	0.0000	27.5060	0.7382	0.0000	0.0000	44.6080	1.8132	0.0000
66	0.4039	0.4038	4.6050	0.4038	0.0000	0.0001	7.3684	0.0001
67	0.0000	1.2185	1.8888	0.0087	0.0000	3.8531	1.6111	0.0037
68	0.0026	6.9802	23.7610	0.5531	0.0049	10.8381	22.8133	0.1229
69	0.0000	23.3259	5.6201	0.5171	0.0001	32.5728	7.8602	0.2278
70	0.0039	0.2861	2.2180	0.8578	0.0079	0.9044	1.8165	0.1276
71	0.0006	0.0115	0.7633	1.1059	0.0016	0.0358	0.6795	0.1253
72	0.0023	1.7054	3.9910	1.1129	0.0074	3.2354	4.6783	0.1573
73	0.0030	1.3933	1.7064	1.1861	0.0068	1.3013	1.8636	0.1718
LP-MIQP								
74	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
76	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
77	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
78	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
79	0.1192	0.0000	0.0000	0.0000	0.3770	0.0000	0.0000	0.0000
80	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
81	0.0000	0.0000	0.0000	0.0012	0.0000	0.0000	0.0000	0.0008
QP-MIQP								
82	37.5000	25.0001	2.3111	0.0003	60.3807	52.7046	1.7433	0.0003
83	0.0000	3.7386	0.0000	0.0000	0.0000	11.6915	0.0000	0.0000
84	0.0000	13.5220	0.6207	0.0000	0.0000	36.4485	1.5634	0.0000

Table 7 continued

Problem ID	Average % MAE				Average Standard Deviation of % MAE			
	NOMAD	COBYLA	ARGONAUT	ISRES	NOMAD	COBYLA	ARGONAUT	ISRES
85	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
86	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
87	0.0000	6.9291	3.6101	0.0000	0.0000	21.9079	7.8285	0.0000
88	0.0000	FS*	0.3254	0.0000	0.0000	$> 10^4$	0.5022	0.0000
89	0.0000	FS*	0.0015	0.0000	0.0000	945.6580	0.0047	0.0000
NLP-MIQP								
90	0.0000	0.0000	0.2280	0.0000	0.0000	0.0000	0.2448	0.0000
91	0.0000	0.0055	0.0000	0.0000	0.0000	0.0056	0.0001	0.0000
92	0.0006	0.0024	0.6045	0.0004	0.0018	0.0021	0.8118	0.0007
93	0.1603	9.7658	0.8294	0.0774	0.3379	30.3412	0.7244	0.2442
94	0.0000	0.0076	28.2385	0.0004	0.0000	0.0184	16.3438	0.0006
95	0.0002	0.0126	0.4367	0.0007	0.0006	0.0393	0.3314	0.0009
96	0.0000	0.0000	0.0014	0.0004	0.0001	0.0000	0.0041	0.0004
97	0.0050	0.0022	1.5956	0.0072	0.0087	0.0046	4.7409	0.0053
98	0.0044	0.0131	1.0333	0.0074	0.0135	0.0180	1.5260	0.0142
99	0.0002	0.9624	2.1801	0.0071	0.0007	3.0432	2.2069	0.0022
NLP-INLP								
100	2.5628	32.4959	0.0000	0.0000	8.1043	52.6317	0.0000	0.0000

*Feasible solution with more than 100% MAE on average is returned at convergence

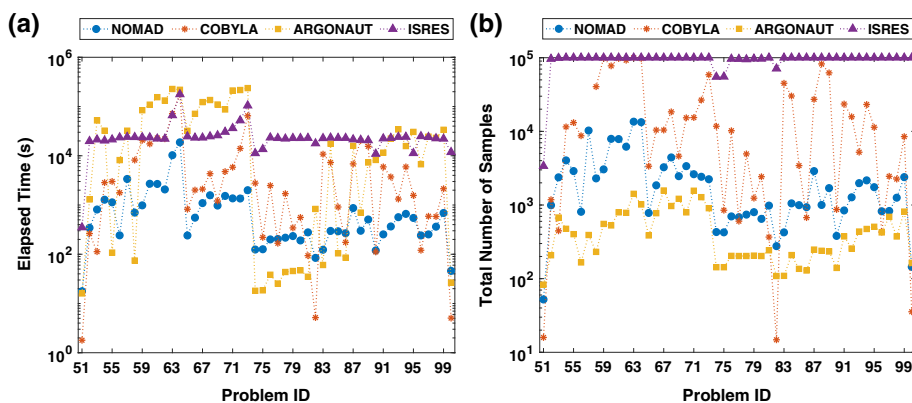


Fig. 4 **a** Average elapsed time for solving bi-level mixed-integer programming problems; **b** Average total number of samples collected by each solver in bi-level mixed-integer programming problems

converges prematurely by hitting the maximum number of samples allowed in almost all tested case studies (Fig. 4b). Moreover, it is important to note that for many of the bi-level mixed-integer benchmark problems both model-based methodologies (COBYLA and ARGONAUT) are recorded to have higher computational expense. Like in the other classes of bi-level programming problems, we observe that the computation time to deterministically solve the LLP is small, between 0.016–0.067 seconds on average per sample. The overall

contribution of solving the LLP deterministically to the total elapsed computation time was at most 15%, where the rest of the computational expense was sourced majorly from the grey-box optimization phase in the ARGONAUT results. Overall, NOMAD is computationally efficient both in terms of the computational time required for convergence as well as in terms of the total number of samples collected throughout the data-driven optimization step. Although in Fig. 4b, ARGONAUT is shown to be the most sample efficient algorithm, the errors reported in Table 7 indicate that ARGONAUT converges to a suboptimal feasible solution in high-dimensional problems, hindering the overall performance of DOMINO in finding the globally optimal solution to bi-level mixed-integer programming problems. The overall results show that NOMAD is more favorable to be incorporated in the DOMINO framework for solving bi-level mixed-integer programming problems. In the following section, the DOMINO framework is tested on a larger bi-level MINLP case study, which considers a land allocation problem under Food-Energy-Water Nexus considerations.

3.2 Land allocation problem in food-energy-water nexus

The sustainable development of an agricultural farming area is of critical importance for maintaining the interconnected elements, namely food, energy and water, that depend on the same land resources. Hence, the actions taken towards allocating land resources will essentially affect food production, which requires energy, in the form of fertilizers, and water for irrigation. On the other hand, clean water production requires energy (i.e. operating a filtration system) and energy can be produced through agriculture as biofuels. This interconnected relationship between these key resources is referred to as the Food-Energy-Water Nexus (FEW-N) and has recently gained a lot of attention for land use optimization in areas with resource scarcity [60–62].

While the government regulators would like to minimize the stress on the nexus in the long-term, many companies allocating and processing the land are concerned with short-term goals, such as maximizing profit. Thus, a formidable challenge exists in the optimization of the land allocation problem, where multiple stakeholders, each concerned with optimizing their own objective functions, are acting upon the optimal decision-making process. We have previously developed a hierarchical FEW-N approach to tackle this issue and to facilitate decision making under competition for these key resources while promoting the sustainable development of the land [12]. In this Section, we will be addressing the data-driven optimization of the land allocation problem through the DOMINO framework.

The land allocation case study consists of two players: the government regulators and the agricultural developer. The goal of the agricultural developer is to maximize its profit whereas the government that regulates this piece of land aims to minimize the stress on the FEW-N, by offering subsidies to the agricultural producer or land developer. Hence, this can be viewed as a Stackelberg game where the government regulators will lead, making the first move by assigning the subsidies, whereas the agricultural producer will follow the leader by reacting accordingly, taking optimal actions towards maximizing its own profit. This leads to the following hierarchical optimization problem [12],

$$\begin{aligned}
 & \min && \text{Stress on FEW Nexus} \\
 & \text{s.t.} && \text{Government's Budget} \\
 & \max && \text{Developer's Profit} \\
 & \text{s.t.} && \text{Land Properties} \\
 & && \text{Land Process Models}
 \end{aligned} \tag{3}$$

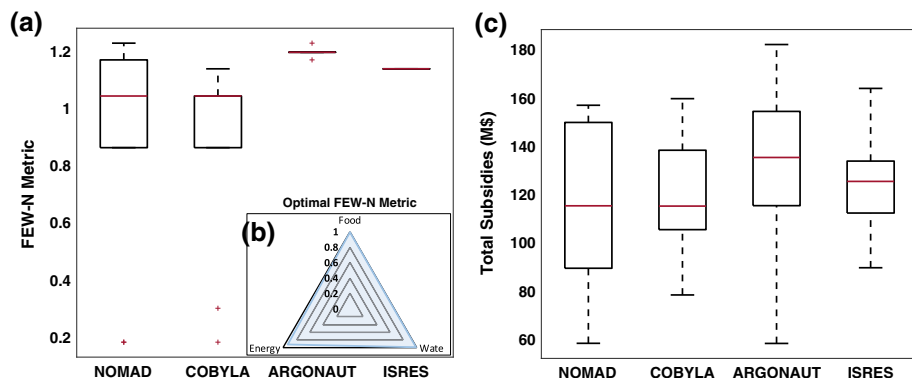


Fig. 5 **a** Optimal FEW-N metric returned by DOMINO when coupled with local and global grey-box solvers; **b** Optimal nexus solution represented as the area of a triangle (Best solution found by ARGONAUT and NOMAD algorithms in DOMINO, $f_{best} = 1.2258$); **c** Boxplot of total amount of subsidies offered by the government for the solution of FEW-N land allocation problem over 10 runs

where the agricultural developer will invest on a piece of land to maximize its profit through a careful consideration of land properties, subsidies offered by the government and land process models at the lower-level. On the other hand, at the upper-level, the government agency that regulates this land will focus on sustainable development through minimizing the FEW-N stress, with respect to their allowed budget.

The detailed land allocation model (please see Appendix for the model equations) is developed in GAMS and the lower-level problem is an MILP problem with 1,721 equations, 216 discrete variables and 772 continuous variables. The upper-level is an NLP problem consisting of 5 continuous variables with 165 grey-box constraints from the Big-M formulation (Eqs. 22, 24). This large-scale bi-level NLP-MILP optimization problem is solved using the DOMINO framework and the performance of the 4 data-driven solvers are compared in the following section.

3.2.1 Computational results of the FEW-N case study

The results of the hierarchical land allocation problem are summarized in Figs. 5 and 6. The boxplot results in Fig. 5a show that the DOMINO framework, when coupled with a global solver, consistently returns the same objective value over multiple repetitive runs, whereas some variability is observed in the solutions returned when the framework is coupled with local data-driven solvers. This result clearly indicates that the hierarchical FEW-N land allocation problem is nonconvex and global optimization is necessary to find a superior solution. The maximum value for the FEW-N metric for this case study is identified by two algorithms, namely NOMAD and ARGONAUT. In addition, Fig. 5b, c shows the globally optimal FEW-N metric found by the DOMINO framework and the distribution of the total amount of subsidies offered by the government for each solver over 10 runs, respectively. The radar plot in Fig. 5b shows that the globally optimal solution can capture the food and water dimensions of the nexus almost perfectly (99.5% in food and 99% in water) with a small trade-off in the energy dimension (93%).

In addition, the boxplot in Fig. 5c shows that all solvers are subject to some variability in finding the optimal set of decisions for the government regulators' objective. More specifically, the variability within the results of two global solvers, which returned consistent

(a) Optimal Land Allocation for Autumn, Spring and Summer Seasons

1. Wind & Fruits	2. Wind & Fruits	3. Wind & Fruits	4. Wind & Fruits
5. Wind	6. Wind	7. Fruits	8. Fruits

Optimal Land Allocation for Winter

1. Wind & Fruits	2. Wind & Fruits	3. Wind & Fruits	4. Wind & Fruits
5. Wind	6. Wind	7. Veggie	8. Fruits

(b) Optimal Land Allocation for Autumn, Spring and Summer Seasons

1. Wind & Fruits	2. Wind & Fruits	3. Wind & Fruits	4. Wind & Fruits
5. Wind	6. Wind	7. Fruits	8. Fruits

Optimal Land Allocation for Winter

1. Wind & Fruits	2. Wind & Fruits	3. Wind & Veggie	4. Wind & Fruits
5. Wind	6. Wind	7. Fruits	8. Fruits

Fig. 6 **a** Optimal land allocation returned by ARGONAUT; **b** Optimal land allocation returned by NOMAD. Both solutions are equally optimal with the FEW-N metric value of 1.2258

objective function values as shown in Fig. 5a, is a clear indication of the multiplicity of solutions that exists in the problem. For the same optimal FEW-N metric value ($f_{best} = 1.2258$), NOMAD allocates a total of \$58.1M with a breakdown of \$0M for livestock grazing and solar energy, \$7.6M for wind energy, \$37.8M and \$12.7M for fruit and vegetable production, respectively. On the other hand, for the same optimal FEW-N metric value, ARGONAUT allocates a total of \$115.2M with a breakdown of \$0M for livestock grazing and solar energy, \$15.2M for wind energy and \$50M for both fruit and vegetable production. A clear difference between the solutions provided by these two algorithms is more apparent at the lower-level objective function value, where the solution provided by NOMAD enables the agricultural developer to have \$3.47B profit, whereas this number increases by \$500M to \$3.97B profit with the ARGONAUT solution. This difference in profit values is captured in the optimal allocation results that are provided in Fig. 6, where the allocation patterns for the same nexus solution differ as the subsidies offered by the government is lowered. Figures 6a, b show that the optimal allocation pattern for the land is exactly the same for the spring, summer and autumn seasons for both NOMAD and ARGONAUT, where a mix of wind energy and fruit production is preferred on the land. However, in winter, the optimal allocation for plot 7 changes to vegetable production for the ARGONAUT solution, while others remain the same. In the case of the NOMAD solution, the allocation pattern for plot 3 in winter changes from wind energy and fruit production to wind energy and vegetable production. Overall, both configurations are equally optimal and are sufficient to minimize the nexus stress, where the government will decide whether to subsidize the agricultural processes with a higher or a lower amount depending on their available budget and preferences.

The computational performance of each solver within the DOMINO framework for the FEW-N case study is also compared (Table 8). The average elapsed time and the average number of samples collected at convergence indicates that COBYLA is computationally very efficient. However, COBYLA was unable to locate the best solution found by NOMAD and ARGONAUT algorithms for the FEW-N problem, which is undesirable to a decision maker. NOMAD stands out as a grey-box solver of choice for this problem as this is the second most efficient algorithm that was able to locate the global solution. As mentioned earlier in this section, the optimal solution provided by NOMAD is more favorable for the government regulator as the total amount of subsidy offered is minimal. On the other hand, the solution

Table 8 Computational performance of DOMINO with different grey-box solvers for the land allocation problem. The results are averaged over 10 runs

Solver	Average elapsed time (s)	Average total number of Samples
NOMAD	138.6	283.9
COBYLA	23.6	67.1
ARGONAUT	1.2×10^4	247.4
ISRES	3.3×10^4	10^5

offered by ARGONAUT is equally optimal with respect to the NOMAD solution, and favors the agricultural developer at the lower-level as this solution provides an additional \$500M in their profit. However, ARGONAUT being a global model-based grey-box solver makes it more computationally demanding for this problem with respect to the elapsed time for convergence, since ARGONAUT explicitly constructs individual surrogate formulations for the 165 unknown grey-box constraints in this case study. In terms of sampling requirements, as shown in Table 8, we observe that NOMAD and ARGONAUT are comparable as they both collect about equal number of samples on average over 10 repetitive runs. Finally, as observed in the results of many benchmark problems that are provided in Sect. 3.1, ISRES reaches the maximum number of samples allowed for the algorithm in all repetitive runs, which also leads to a more demanding computational time for the execution of this algorithm. Overall, the results of the benchmark studies and the large-scale land allocation problem demonstrate that the DOMINO framework serves as an effective methodology for solving many large-scale bi-level MINLPs.

4 Conclusion

In this work, the DOMINO framework is presented as an algorithmic advancement for solving bi-level mixed-integer nonlinear programming (B-MINLP) problems with guaranteed feasibility when the lower-level problem is solved to global optimality at convergence. A novel data-driven approach is followed to approximate a bi-level optimization problem into a single-level problem, where the upper-level decision variables are used to simulate the optimality of the lower-level problem. The resulting input-output data is further sent to a data-driven optimizer to retrieve the optimal solution to the bi-level problem, where the DOMINO framework is flexible to house any type of data-driven/grey-box optimizer. The accuracy, consistency and the computational performance of DOMINO is extensively investigated on a large set of benchmark problems consisting of bi-level linear, continuous nonlinear and mixed-integer programming problems. In addition, the effect of the data-driven solver on DOMINO's performance is investigated by incorporating a local sample-based, local model-based, global sample-based, and global model-based methodologies. Furthermore, the performance of the DOMINO framework is tested on a large-scale bi-level mixed-integer nonlinear case study in Food-Energy-Water Nexus (FEW-N). The results of the benchmark studies show that the DOMINO framework can identify the true global solution or a near-optimal solution for an extensive set of challenging bi-level optimization problems. Moreover, the results of the FEW-N case study demonstrate that DOMINO can handle large-scale bi-level mixed-integer nonlinear programming problems and provide superior feasible solutions consistently over multiple repetitive runs. Hence, DOMINO serves as a powerful computa-

tional algorithm for solving large-scale B-MINLPs which are traditionally difficult to solve using exact techniques. In the future, the algorithm is going to be compared with other data-driven bi-level optimization methodologies available in the literature. Furthermore, a rigorous upper-level equality constraint handling scheme will be investigated for bi-level optimization with DOMINO.

Acknowledgements The authors would like to acknowledge the funding and support provided by the U.S. National Institutes of Health Superfund Research Program (NIH P42-ES027704), the National Science Foundation projects INFEWS (1739977) and PAROC (CBET-1705423), the U.S. Department of Energy project RAPID SYNOPSIS (DE-EE0007888-09-03), the Texas A&M University Superfund Research Center and the Texas A&M Energy Institute. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High Performance Research Computing. The manuscript contents are solely the responsibility of the grantee and do not necessarily represent the official views of the NIH. Further, NIH does not endorse the purchase of any commercial products or services mentioned in the publication.

Appendix

Best found solutions for benchmark problems 18, 46 and 47

Problem 18 (“wk_2015_01”):

$$x^* = 9.999776, y^* = 9.9998, f_{best} = 4.5443471 \cdot 10^{-7}, F_{best} = 99.9955201008.$$

Lower Level Relative Gap: 0 (Retrieved from CPLEX version 12.8.0.0)

Problem 46 (“wk_2015_04”):

$$x_1^* = 0, x_2^* = 0, y_1^* = 0, y_2^* = 0, y_3^* = 0, y_4^* = 0, f_{best} = 0, F_{best} = 0.$$

Lower Level Relative Gap: $1 \cdot 10^{-9}$ (Retrieved from ANTIGONE version 1.1)

Problem 47 (“wk_2015_06”):

$$x_1^* = 0.000984369218350, x_2^* = -0.001021751016379, x_3^* = 1.663984077237546, \\ x_4^* = -0.076938496530056, y_1^* = -1.0187598163, y_2^* = 1.0574476104, y_3^* = -0.0004531744, y_4^* = 0, f_{best} = -5, F_{best} = 0.0000045078.$$

Lower Level Relative Gap: $1.76 \cdot 10^{-7}$ (Retrieved from BARON version 18.11.12)

Notation for the food-energy-water nexus case study

<i>e</i>	efficiency
<i>energy</i>	energy
<i>max</i>	maximum
<i>min</i>	minimum
<i>profit</i>	profit
<i>total</i>	total
<i>trans</i>	transportation
<i>H₂O</i>	water

List of land processes considered in the food-energy-water nexus case study

Energy Land Processes

1. Solar Energy
2. Wind Energy

Agricultural Processes

3. Fruit Production
4. Vegetable Production
5. Livestock Grazing

Agricultural developer's problem

The chosen land allocation problem considers a piece of land which will be processed by an agricultural developer over 4 seasons in a climate similar to that of Texas, U.S. and is divided into 8 equal (1 km^2) plots. The nomenclature for this problem is provided in Table 9. On each piece of land, a subset of agricultural and energy land processes can occur, where fruit production, vegetable production, and livestock grazing are representatives of agricultural processes defined by the subset T_A , whereas solar energy and wind energy are representatives of energy land processes, defined by the subset T_E . Two important properties regarding these subsets are given in Eqs. 4 and 5.

$$T_A \cup T_E = T_L \quad (4)$$

$$T_A \cap T_E = \emptyset \quad (5)$$

The agricultural producer will be subject to various constraints regarding the properties of the land, the properties of the agricultural and energy production processes while making an optimal decision towards its own objective. First, the land characteristics will affect the selection of any process that can occur in each land plot. If good soil is not available in a plot section, agricultural processes are restricted to not to take place in that land section for all seasons. If the adequate sun is not available in a plot section, solar energy will not be implemented in that land section for all seasons. Finally, if a plot section does not have access to the adequate amount of wind, wind energy production will not be implemented in that land section for all seasons. These characteristics are summarized in Table 10. Based on this information, constraints regarding water transportation can be defined for the problem such as water must be transported to the land if there is no water on a plot and an agricultural process is selected to occur on that plot:

$$y_{i,j,k}^{trans,H_2O} \leq y_{i,j,k} + y_j^{H_2O} \quad \forall i \in T_A, j, k \quad (6)$$

No water will be transported, if water is already available on the plot:

$$y_{i,j,k}^{trans,H_2O} \leq 1 - y_j^{H_2O} \quad \forall i \in T_A, j, k \quad (7)$$

No water should be transported, if there is no water on the plot and no agricultural process is selected to occur on that plot:

$$y_{i,j,k}^{trans,H_2O} \geq y_{i,j,k} - y_j^{H_2O} \quad \forall i \in T_A, j, k \quad (8)$$

In addition to the land properties, there are other constraints that further influence the selection of land processes and restrict the feasible space for this case study. The constraints regarding the selection of land processes is imposed such that at least one land process must be allocated on each plot.

$$\sum_{i \in I} y_{i,j,k} \geq 1 \quad \forall j, k \quad (9)$$

Table 9 Nomenclature for the Food-Energy-Water Nexus case study

Type	Name	Description
Indices	$i \in \{1, 2, \dots, I\}$	Land processes ($card(i) = 5$)
	$j \in \{1, 2, \dots, J\}$	Land plot square number ($card(j) = 8$)
	$k \in \{1, 2, \dots, K\}$	Seasons in a Texas-type climate ($card(k) = 4$)
Sets	T_L	Land use types
	$T_A \subset T_L$	Agriculture land use type ($card(T_A) = 3$)
	$T_E \subset T_L$	Energy land use type ($card(T_E) = 2$)
Binary Variables	$y_{i,j,k}$	Activates the i^{th} process that occurs on the j^{th} plot in the k^{th} season
	$y_j^{H_2O}$	Activates water availability on the j^{th} plot
	$y_{i,j,k}^{trans,H_2O}$	Activates water transportation that is required for the i^{th} process on the j^{th} plot in the k^{th} season, where $i \in T_A$
Parameters	$P_{i,k}^e$	Efficiency multiplier of the i^{th} land process in the k^{th} season
	$P_{i,k}^{profit}$	Profit multiplier of the i^{th} land process for the k^{th} season, where $i \in T_E$
	$D_k^{H_2O}$	Multiplier of minimum water required for the k^{th} season
	C_k^{trans,H_2O}	Water transportation cost multiplier for the k^{th} season
	$L_i^{H_2O}$	Lower bound on water transportation and consumption for the i^{th} land process in kg, where $i \in T_A$
	$U_i^{H_2O}$	Upper bound on water transportation and consumption for the i^{th} land process in kg, where $i \in T_A$
	L_i^{energy}	Lower bound on energy consumption for the i^{th} land process in kWh, where $i \in T_A$
	U_i^{energy}	Upper bound on energy consumption for the i^{th} land process in kWh, where $i \in T_A$
	M_i^{energy}	Metric ton of yield per kWh energy consumed for the i^{th} land process, where $i \in T_A$
	$M_i^{H_2O}$	Metric ton of yield per kg of water consumption in the i^{th} land process, where $i \in T_A$
	M_i^{profit}	Profit made from the i^{th} land process per unit energy produced in k\$/kWh when $i \in T_E$ and profit made from i^{th} land process per unit yield obtained in k\$/ton when $i \in T_A$
	B_i	Government budget allocated for supporting the i^{th} land process type in k\$
	BM	Big-M parameter
Continuous Variables	$EP_{i,j,k}$	Energy produced by the i^{th} land process type on the j^{th} plot during the k^{th} season in kWh, where $i \in T_E$
	$EC_{i,j,k}$	Energy consumed by the i^{th} land process type on the j^{th} plot during the k^{th} season in kWh, where $i \in T_A$
	$W_{i,j,k}$	Water consumed from an existing source by the i^{th} land process type on the j^{th} plot during the k^{th} season in kg, where $i \in T_A$
	$W_{i,j,k}^{trans}$	Water consumed from a transported source by the i^{th} land process type on the j^{th} plot during the k^{th} season in kg, where $i \in T_A$

Table 9 continued

Type	Name	Description
	$Y_{i,j,k}$	Yield produced by the i^{th} land process type on the j^{th} plot during the k^{th} season in metric tonnes, where $i \in T_A$
	$G_{i,j,k}^{profit}$	Profit gained by the i^{th} land process type on the j^{th} plot during the k^{th} season in k\$
	S_i	Subsidies offered by the government for using the i^{th} land process
	$\hat{S}_{i,j,k}$	Variable introduced in the Big-M formulation for replacing the bilinear term $S_i \cdot y_{i,j,k}$
	E^{total}	Total energy gained from the land in kWh
	Y^{total}	Total yield gained from the land in metric tonnes
	W^{total}	Total water consumed on the land in kg
	$G^{profit,total}$	Total profit gained from the land in k\$

Table 10 Land properties for the case study. These limit the processes that can occur on each plot over 4 seasons, defined by the binary variable $y_{i,j,k}$. The water availability is defined by the binary variable $y_j^{H_2O}$. 1 indicates existence and 0 indicates absence of that property

Land properties for all seasons ($\forall k$)	Land plot number (j)							
	1	2	3	4	5	6	7	8
Good Soil ($y_{i,j,k} \forall i \in T_A$)	1	1	1	1	0	0	1	1
Adequate Sun ($y_{1,j,k}$)	0	0	1	1	1	1	1	1
Adequate Wind ($y_{2,j,k}$)	1	1	1	1	1	1	0	0
Water Available ($y_j^{H_2O}$)	0	0	0	0	1	1	0	1

Furthermore, it is not practical to have solar panels and agricultural production on the same plot. Thus, at most one out of solar energy, fruit, vegetables and livestock can be allocated in one plot:

$$\sum_{i \neq 2, i \in T_L} y_{i,j,k} \leq 1 \quad \forall j, k \quad (10)$$

Wind energy will occupy minimal space on the land plot, compared to solar energy production systems, hence both wind energy and either fruit or vegetable production can be allocated on the same plot:

$$\sum_{i=2}^4 y_{i,j,k} \leq 2 \quad \forall j, k \quad (11)$$

Moreover, only one energy process is allowed on a plot:

$$\sum_{i \in T_E} y_{i,j,k} \leq 1 \quad \forall j, k \quad (12)$$

If an energy process is selected in a plot, the type of energy production will stay the same throughout the year, since it is too expensive to move equipment over seasons:

$$y_{i,j,k+1} \geq y_{i,j,k} \quad \forall i \in T_E, j, k \leq \text{card}(k) - 1 \quad (13)$$

Second, the seasonal differences must be considered, as these can impact the energy demand, water transportation cost, water availability for irrigation and efficiency of energy production processes. For example, in seasons with rainfall, such as winter, spring and fall, the transportation cost for water will be less and less water will be required for irrigation. On the other hand, the solar systems will have lower efficiency due to the reduced amount of sunshine throughout these seasons. A similar analysis is also done for the summer, where there is going to be greater demand for energy and water, and higher transportation costs for water will be in effect. However, the solar systems will have greater efficiency since there will be plenty of sunshine during summer. Hence, both spatial and time scenarios are considered and their respective parameters are included in the model equations (for the parameters please see Tables 11–14).

The land processes will be quantified on the amount of energy produced or agricultural yield, if an energy or an agricultural process is selected, respectively. It is important to note that, if an energy process is selected for a given plot in a given season, a fixed amount of energy can be produced from these technologies:

$$\begin{aligned} EP_{1,j,k} &= P_{1,k}^e \cdot 50 \cdot y_{1,j,k} & \forall j, k \\ EP_{2,j,k} &= P_{2,k}^e \cdot 1000 \cdot y_{2,j,k} & \forall j, k \end{aligned} \quad (14)$$

Likewise, the yield for agricultural processes can be calculated as a function of water and energy consumption. The parameter $P_{i,k}^e$ is used to take in consideration the changes in efficiency of land processes over different seasons.

$$Y_{i,j,k} = P_{i,k}^e (M_i^{\text{energy}} \cdot EC_{i,j,k} + M_i^{H_2O} \cdot W_{i,j,k}) \quad \forall i \in T_A, j, k \quad (15)$$

The amount of energy consumption and water consumption (from an already existing source) by agricultural processes, which are used to calculate the yield in Eq. 15, are bounded. Note that the lower bound on the water consumption depends on seasonal effects (dry seasons versus seasons with rainfall), hence multiplied by its respective parameter, $D_k^{H_2O}$.

$$\begin{aligned} L_i^{\text{energy}} \cdot y_{i,j,k} &\leq EC_{i,j,k} \leq U_i^{\text{energy}} \cdot y_{i,j,k} & \forall i \in T_A, j, k \\ D_k^{H_2O} \cdot L_i^{H_2O} \cdot y_{i,j,k} &\leq W_{i,j,k} \leq U_i^{H_2O} \cdot y_{i,j,k} & \forall i \in T_A, j, k \end{aligned} \quad (16)$$

In addition to the box-constraints, it is important to supply adequate amount of water to each plot in each season for the agricultural land processes. Thus, the amount of water consumption (source-based and transportation-based) is set to be at least 200 times greater than the energy consumption in each plot and in each season:

$$\sum_{i \in T_A} W_{i,j,k} + D_k^{H_2O} \cdot \sum_{i \in T_A} W_{i,j,k}^{\text{trans}} \geq 200 \cdot \sum_{i \in T_A} EC_{i,j,k} \quad \forall j, k \quad (17)$$

The amount of water transported for agricultural processes is also bounded and affected by the seasonal differences:

$$D_k^{H_2O} \cdot L_i^{H_2O} \cdot y_{i,j,k}^{\text{trans}, H_2O} \leq W_{i,j,k}^{\text{trans}} \leq U_i^{H_2O} \cdot y_{i,j,k}^{\text{trans}, H_2O} \quad \forall i \in T_A, j, k \quad (18)$$

As described previously in Sect. 3.2, the objective of the agricultural developer is to maximize its profit. The profit calculation for all land processes includes the money made from energy production and the yield from the agricultural processes, if an energy or an agricultural process is selected, respectively. For energy producing land processes profit is given as:

$$G_{i,j,k}^{\text{profit}} = M_i^{\text{profit}} \cdot P_{i,k}^{\text{profit}} \cdot EP_{i,j,k} + \hat{S}_{i,j,k} \quad \forall i \in T_E, j, k \quad (19)$$

For agricultural processes, the profit is given as:

$$G_{i,j,k}^{profit} = M_i^{profit} \cdot Y_{i,j,k} + \hat{S}_{i,j,k} \quad \forall i \in T_A, j, k \quad (20)$$

The profit calculations also considers the relevant subsidies ($\hat{S}_{i,j,k}$) offered by the government agencies for developing different processes on the land, where these subsidies should only be considered in the profit when their respective land process is activated.

$$\hat{S}_{i,j,k} = S_i \cdot y_{i,j,k} \quad \forall i, j, k \quad (21)$$

To avoid this bilinear term that appears in the profit equation, the variable $\hat{S}_{i,j,k}$ and its Big-M formulation is introduced in Eqs. 21–24, where BM is the Big-M parameter.

$$S_i \leq BM \cdot \sum_j \sum_k y_{i,j,k} \quad \forall i \quad (22)$$

$$\hat{S}_{i,j,k} \leq BM \cdot y_{i,j,k} \quad \forall i, j, k \quad (23)$$

$$\hat{S}_{i,j,k} \leq S_i \quad \forall i, j, k \quad (24)$$

Moreover, the agricultural developer is interested in maximizing the total profit, which is a function of the total energy production, total yield from agricultural production and total water consumption. The total energy, E^{total} , is defined as the difference between total energy produced from energy land processes and total energy consumed by the agricultural processes in all plots throughout the 4 seasons.

$$E^{total} = \sum_{i \in T_E} \sum_j \sum_k E P_{i,j,k} - \sum_{i \in T_A} \sum_j \sum_k E C_{i,j,k} \quad (25)$$

Similarly, the total yield, Y^{total} , is the summation of yield of all agricultural processes over all plots and 4 seasons.

$$Y^{total} = \sum_{i \in T_A} \sum_j \sum_k Y_{i,j,k} \quad (26)$$

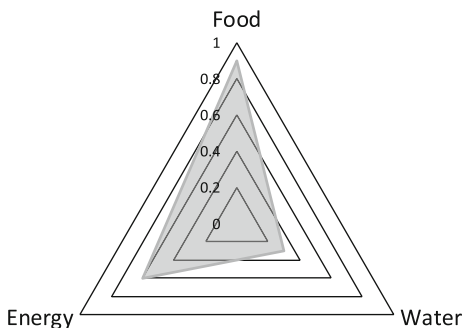
The total water consumption, W^{total} , includes both the amount of water consumed from a natural source (i.e. water already existing as in the land properties, given in Table 10) and from a transported source. The transported total water also considers seasonal demand, defined by the parameter $D_k^{H_2O}$.

$$W^{total} = \sum_{i \in T_A} \sum_j \sum_k W_{i,j,k} + \sum_k D_k^{H_2O} \sum_{i \in T_A} \sum_j W_{i,j,k}^{trans} \quad (27)$$

The total profit, $G^{profit,total}$, is calculated by subtracting the total water transportation cost throughout all plots, all seasons and all agricultural land processes from the cumulative profit from all land processes. The cost of water transportation is assumed to be \$10/kg of water. In addition, the cost of transportation is impacted by seasonal differences, as explained previously, hence the formulation includes the $C_k^{H_2O,trans}$ parameter to account for such effects. The objective function of the LLP is given as:

$$G^{profit,total} = \sum_i \sum_j \sum_k G_{i,j,k}^{profit} - 0.01 \cdot \sum_k C_k^{H_2O,trans} \sum_{i \in T_A} \sum_j W_{i,j,k}^{trans} \quad (28)$$

Fig. 7 FEW-N metric represented as the area of a triangle. Shaded area demonstrates an example solution to FEW-N



Finally, the continuous variables defined in Eqs. 25–28 are bounded and their respective values are obtained through minimizing and maximizing each variable as the sole objective to the land allocation problem.

$$\begin{aligned}
 0 &\leq W^{total} \leq 2.46 \cdot 10^9 \\
 0 &\leq Y^{total} \leq 13860 \\
 0 &\leq E^{total} \leq 21945 \\
 G^{profit,total} &\geq 0
 \end{aligned} \tag{29}$$

The variables defined in Eqs. 25–28 as well as their respective bounds, provided in Eq. 29, are used to enumerate the upper-level objective function of the government regulators. The ULP is discussed in detail in the following section.

Government regulators' problem

As shown in Eq. 3, the objective of the government regulators is to minimize the nexus stress. However, the mathematical quantification of the nexus, which will take in consideration of the trade-offs between food, energy and water, has not yet been fully established. Recently, Avraamidou et al. [13] has introduced a methodology to develop a FEW-N metric, which brings relevant decision elements and their respective quantification together through r^{th} order averaging. In this work, we adopt this idea through a similar methodology where a single geometric metric, i.e. the area of a triangle, is used to represent the FEW-N metric as the government regulators' objective. An illustration of the FEW-N metric is provided in Fig. 7.

In Fig. 7, the corners of the triangle represent the scaled quantities of each FEW-N element, where their respective values lie between 0 and 1. In this case, a value of 1 represents the best possible scenario and 0 represents the worst. The objective of the government regulators is to maximize the best possible scenario for each element, namely minimizing the total water consumed and maximizing the total energy and food produced, which essentially translates into maximizing the area of the triangle. The explicit formulation of this objective is provided in Eq. 30.

$$\begin{aligned}
 FEW_{metric} = & \left[\frac{E^{total} - E^{min}}{E^{max} - E^{min}} \cdot \left(1 - \frac{W^{total} - W^{min}}{W^{max} - W^{min}} \right) + \frac{E^{total} - E^{min}}{E^{max} - E^{min}} \cdot \frac{Y^{total} - Y^{min}}{Y^{max} - Y^{min}} \right. \\
 & \left. + \left(1 - \frac{W^{total} - W^{min}}{W^{max} - W^{min}} \right) \cdot \frac{Y^{total} - Y^{min}}{Y^{max} - Y^{min}} \right] \cdot \frac{\sin 120^\circ}{2}
 \end{aligned} \tag{30}$$

Table 11 Parameter values for $P_{i,k}^e$

i	k			
	1 (Autumn)	2 (Winter)	3 (Spring)	4 (Summer)
1 (solar energy)	0.85	0.70	0.90	1.00
2 (wind energy)	0.90	1.00	0.90	0.80
3 (fruit production)	1.00	0.85	1.00	1.00
4 (vegetable production)	1.00	0.85	1.00	1.00
5 (livestock grazing)	1.00	0.85	1.00	1.00

Table 12 Parameter values for $P_{i,k}^{profit}$

i	k			
	1 (Autumn)	2 (Winter)	3 (Spring)	4 (Summer)
1 (solar energy)	1.00	1.20	1.00	1.20
2 (wind energy)	1.00	1.20	1.00	1.20

Note that E^{total} , Y^{total} , and W^{total} is obtained through solving the agricultural producer's problem, explicitly defined in Eqs. 25–27, respectively.

In this case study, the government is offering subsidies (S_i) to the land developers for each nexus element, as much as their budget (B_i) allows.

$$0 \leq S_i \leq B_i \quad \forall i \quad (31)$$

These subsidies further motivate the land owner to properly allocate and utilize the land to maximize their own profit (Eqs. 19–20). The upper bound on the total governmental budget is set to be \$250M where this is allocated equally among all land processes. Essentially, the goal of the government agency is to decide on the amount of subsidies to be offered to the agricultural producer in such a way that the objective function defined in Eq. 30 is maximized.

Parameters

Parameter values are tabulated in Tables 11–14, where 4 seasons (autumn, winter, spring, and summer) are considered for the FEW-N case study with production starting in autumn and ending after summer. These parameters are used as multipliers to capture seasonal differences among technological efficiencies, water demand and transportation costs. The efficiency of the solar energy production process is lower in autumn and winter whereas it is higher in the summer. Likewise, the efficiency of agricultural processes is lower in winter as shown in Table 11.

The profit from energy production during winter and summer should be higher since there would be higher demand for energy in very cold and hot weathers. Hence, higher multipliers are assigned for both energy production land processes, which are summarized in Table 12.

Table 13 summarizes the multipliers for the minimum amount of water required as well as the cost of transporting water over 4 seasons. Both the required amount of water and the cost of transportation is expected to be higher in summertime due to elevated temperatures

Table 13 Parameter values for $D_k^{H_2O}$ and C_k^{trans,H_2O}

k	1 (Autumn)	2 (Winter)	3 (Spring)	4 (Summer)
$D_k^{H_2O}$	1.00	0.70	1.00	1.20
$C_k^{H_2O,trans}$	1.00	1.00	1.00	1.30

Table 14 Parameter values for $L_i^{H_2O}$, $U_i^{H_2O}$, L_i^{energy} , U_i^{energy} , M_i^{energy} , $M_i^{H_2O}$ and M_i^{profit}

i	1 (solar energy)	2 (wind energy)	3 (fruit production)	4 (vegetable production)	5 (livestock grazing)
$L_i^{H_2O}$	–	–	100	100	10^4
$U_i^{H_2O}$	–	–	10^6	10^6	10^8
L_i^{energy}	–	–	5	5	5
U_i^{energy}	–	–	50	50	100
M_i^{energy}	–	–	10	10	1
$M_i^{H_2O}$	–	–	10–4	15^{-4}	40^{-4}
M_i^{profit}	100	100	2	1.3	5

and higher demand for water in agricultural production. Finally, Table 14 summarizes other parameters used in the FEW-N case study.

References

1. Abramson, M.A., Audet, C., Couture, G., Dennis, Jr. J.E., Le Digabel, S., Tribes, C.: The NOMAD project. <https://www.gerad.ca/nomad/> (2015). Accessed 16 Jan 2018
2. Arroyo, J.M., Fernández, F.J.: A genetic algorithm approach for the analysis of electric grid interdiction with line switching. In: 2009 15th International Conference on Intelligent System Applications to Power Systems, pp 1–6. IEEE (2009)
3. Audet, C., Dennis Jr., J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**(1), 188–217 (2006)
4. Avraamidou, S., Pistikopoulos, E.N.: A multiparametric mixed-integer bi-level optimization strategy for supply chain planning under demand uncertainty. IFAC PapersOnLine **50**(1), 10178–10183 (2017)
5. Avraamidou, S., Pistikopoulos, E.N.: A novel algorithm for the global solution of mixed-integer bi-level multi-follower problems and its application to planning scheduling integration. In: 2018 European Control Conference (ECC), pp 1056–1061 (2018)
6. Avraamidou, S., Pistikopoulos, E.N.: Adjustable robust optimization through multi-parametric programming. Optim. Lett. (2019). <https://doi.org/10.1007/s11590-019-01438-5>
7. Avraamidou, S., Pistikopoulos, E.N.: B-POP: bi-level parametric optimization toolbox. Comput. Chem. Eng. **122**, 193–202 (2019)
8. Avraamidou, S., Pistikopoulos, E.N.: A bi-level formulation and solution method for the integration of process design and scheduling. In: Muñoz S.G., Laird C.D., Realf M.J. (eds.) Proceedings of the 9th International Conference on Foundations of Computer-Aided Process Design, Computer Aided Chemical Engineering, vol. 47, pp. 17–22. Elsevier (2019)
9. Avraamidou, S., Pistikopoulos, E.N.: A global optimization algorithm for the solution of tri-level mixed-integer quadratic programming problems. In: Le Thi, H.A., Le, H.M., Pham Dinh, T. (eds.) WCGO 2019: Optimization of Complex Systems: Theory, Models, Algorithms and Applications, pp. 579–588. Springer, Cham (2019d)

10. Avraamidou, S., Pistikopoulos, E.N.: Multi-parametric global optimization approach for tri-level mixed-integer linear optimization problems. *J. Glob. Optim.* **74**(3), 443–465 (2019e)
11. Avraamidou, S., Pistikopoulos, E.N.: A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. *Comput. Chem. Eng.* **125**, 98–113 (2019f)
12. Avraamidou, S., Beykal, B., Pistikopoulos, I.P.E., Pistikopoulos, E.N.: A hierarchical food-energy-water nexus (FEW-N) decision-making approach for land use optimization. In: Eden M.R., Ierapetritou M.G., Towler G.P. (eds.) 13th International Symposium on Process Systems Engineering (PSE 2018), Computer Aided Chemical Engineering, vol. 44, pp. 1885–1890. Elsevier (2018)
13. Avraamidou, S., Milhorn, A., Sarwar, O., Pistikopoulos, E.N.: Towards a quantitative food-energy-water nexus metric to facilitate decision making in process systems: A case study on a dairy production plant. In: Friedl A., Klemes J.J., Radl S., Varbanov P.S., Wallek T. (eds.) 28th European Symposium on Computer Aided Process Engineering, Computer Aided Chemical Engineering, vol. 43, pp. 391–396. Elsevier (2018)
14. Bajaj, I., Iyer, S.S., Hasan, M.F.: A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point. *Comput. Chem. Eng.* **116**, 306–321 (2018)
15. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. *SIAM J. Sci. Stat. Comput.* **11**(2), 281–292 (1990)
16. Bard, J.F., Plummer, J., Sourie, J.C.: Determining tax credits for converting nonfood crops to biofuels: An application of bilevel programming. In: Migdalas, A., Pardalos, P.M., Värbrand, P. (eds.) *Multilevel Optimization: Algorithms and Applications*, pp. 23–50. Springer, Boston, MA (1998)
17. Beykal, B., Boukouvala, F., Floudas, C.A., Pistikopoulos, E.N.: Optimal design of energy systems using constrained grey-box multi-objective optimization. *Comput. Chem. Eng.* **116**, 488–502 (2018a)
18. Beykal, B., Boukouvala, F., Floudas, C.A., Sorek, N., Zalavadia, H., Gildin, E.: Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations. *Comput. Chem. Eng.* **114**, 99–110 (2018b)
19. Bhosekar, A., Ierapetritou, M.: Advances in surrogate based modeling, feasibility analysis, and optimization: a review. *Comput. Chem. Eng.* **108**, 250–267 (2018)
20. Boukouvala, F., Floudas, C.A.: ARGONAUT: algorithms for global optimization of constrained grey-box computational problems. *Optim. Lett.* **11**(5), 895–913 (2017)
21. Boukouvala, F., Ierapetritou, M.G.: Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function. *AIChE J.* **60**(7), 2462–2474 (2014)
22. Boukouvala, F., Misener, R., Floudas, C.A.: Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. *Eur. J. Oper. Res.* **252**, 701–727 (2016)
23. Boukouvala, F., Hasan, M.M.F., Floudas, C.A.: Global optimization of general constrained grey-box models: new method and its application to constrained pdes for pressure swing adsorption. *J. Glob. Optim.* **67**(1–2), 3–42 (2017)
24. Colson, B.: BIPA(bilevel programming with approximation methods)(software guide and test problems). Cahiers du GERAD <https://www.gerad.ca/en/papers/G-2002-37/view> (2002). Accessed 16 Jan 2018
25. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. Society for Industrial and Applied Mathematics, Philadelphia (2009)
26. Cozad, A., Sahinidis, N.V., Miller, D.C.: Learning surrogate models for simulation-based optimization. *AIChE J.* **60**(6), 2211–2227 (2014)
27. Deb, K.: An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **186**(2–4), 311–338 (2000)
28. Domínguez, L.F., Pistikopoulos, E.N.: Multiparametric programming based algorithms for pure integer and mixed-integer bilevel programming problems. *Comput. Chem. Eng.* **34**(12), 2097–2106 (2010)
29. Eason, J.P., Biegler, L.T.: A trust region filter method for glass box/black box optimization. *AIChE J.* **62**(9), 3124–3136 (2016)
30. Edmunds, T.A., Bard, J.F.: An algorithm for the mixed-integer nonlinear bilevel programming problem. *Ann. Oper. Res.* **34**(1), 149–162 (1992)
31. Faisca, N.P., Dua, V., Rustem, B., Saraiva, P.M., Pistikopoulos, E.N.: Parametric global optimisation for bilevel programming. *J. Glob. Optim.* **38**(4), 609–623 (2007)
32. Faisca, N.P., Saraiva, P.M., Rustem, B., Pistikopoulos, E.N.: A multi-parametric programming approach for multilevel hierarchical and decentralised optimisation problems. *Comput. Manag. Sci.* **6**(4), 377–397 (2009)
33. Fampa, M., Barroso, L.A., Candal, D., Simonetti, L.: Bilevel optimization applied to strategic pricing in competitive electricity markets. *Comput. Optim. Appl.* **39**(2), 121–142 (2008)
34. Garcia-Herreros, P., Zhang, L., Misra, P., Arslan, E., Mehta, S., Grossmann, I.E.: Mixed-integer bilevel optimization for capacity planning with rational markets. *Comput. Chem. Eng.* **86**, 33–47 (2016)

35. Gümiş, Z., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. *J. Glob. Optim.* **20**(1), 1–31 (2001)
36. Gupta, A., Maranas, C.D.: A two-stage modeling and solution framework for multisite midterm planning under demand uncertainty. *Ind. Eng. Chem. Res.* **39**(10), 3799–3813 (2000)
37. Handoko, S.D., Chuin, L.H., Gupta, A., Soon, O.Y., Kim, H.C., Siew, T.P.: Solving multi-vehicle profitable tour problem via knowledge adoption in evolutionary bi-level programming. In: 2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings, pp. 2713–2720 (2015)
38. Hecheng, L., Yuping, W.: Exponential distribution-based genetic algorithm for solving mixed-integer bilevel programming problems. *J. Syst. Eng. Electron.* **19**(6), 1157–1164 (2008)
39. Homaifar, A., Qi, C.X., Lai, S.H.: Constrained optimization via genetic algorithms. *Simulation* **62**(4), 242–253 (1994)
40. ILOG, I.B.M.: IBM ILOG CPLEX Optimization Studio Getting Started with CPLEX (2017)
41. Johnson, S.G.: The NLOpt nonlinear-optimization package. <http://ab-initio.mit.edu/nlopt> (2014). Accessed 16 Jan 2018
42. Kieslich, C.A., Boukouvala, F., Floudas, C.A.: Optimization of black-box problems using smolyak grids and polynomial approximations. *J. Glob. Optim.* **71**(4), 845–869 (2018)
43. Kleniati, P.M., Adjiman, C.S.: Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part ii: convergence analysis and numerical results. *J. Glob. Optim.* **60**(3), 459–481 (2014)
44. Kleniati, P.M., Adjiman, C.S.: A generalization of the branch-and-sandwich algorithm: from continuous to mixed-integer nonlinear bilevel problems. *Comput. Chem. Eng.* **72**, 373–386 (2015)
45. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev.* **45**, 385–482 (2003)
46. Labbé, M., Violin, A.: Bilevel programming and price setting problems. *Ann. Oper. Res.* **240**(1), 141–169 (2016)
47. Le Digabel, S.: Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. *ACM Trans. Math. Softw. (TOMS)* **37**(4), 44 (2011)
48. Le Digabel, S., Tribes, C., Montplaisir, V.R., Audet, C.: NOMAD user guide version 3.9.1. https://www.gerad.ca/nomad/Downloads/user_guide.pdf (2019). Accessed 14 July 2019
49. Li, Z., Ierapetritou, M.: Integrated production planning and scheduling using a decomposition framework. *Chem. Eng. Sci.* **64**, 3585–3597 (2009)
50. Lu, J., Han, J., Hu, Y., Zhang, G.: Multilevel decision-making. *Inf. Sci.* **346**(C), 463–487 (2016)
51. Misener, R., Floudas, C.A.: Global optimization of mixed-integer models with quadratic and signomial functions: a review. *Appl. Comput. Math.* **11**(3), 317–336 (2012)
52. Misener, R., Floudas, C.A.: GloMIQO: global mixed-integer quadratic optimizer. *J. Glob. Optim.* **57**(1), 3–50 (2013)
53. Misener, R., Floudas, C.A.: ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. *J. Glob. Optim.* **59**(2–3), 503–526 (2014)
54. Mitsos, A.: Global solution of nonlinear mixed-integer bilevel programs. *J. Glob. Optim.* **47**(4), 557–582 (2010)
55. Mitsos, A., Barton, P.I.: A test set for bilevel programs. https://www.researchgate.net/publication/228455291_A_test_set_for_bilevel_programs (2007). Accessed 16 Jan 2018
56. Mitsos, A., Lemonidis, P., Barton, P.I.: Global solution of bilevel programs with a nonconvex inner program. *J. Glob. Optim.* **42**(4), 475–513 (2008)
57. Müller, J., Shoemaker, C.A., Piché, R.: SO-MI: a surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. *Comput. Oper. Res.* **40**(5), 1383–1400 (2013)
58. Newby, E., Ali, M.M.: A trust-region-based derivative free algorithm for mixed integer programming. *Comput. Optim. Appl.* **60**(1), 199–229 (2015)
59. Nie, J., Wang, L., Ye, J.J.: Bilevel polynomial programs and semidefinite relaxation methods. *SIAM J. Optim.* **27**(3), 1728–1757 (2017)
60. Nie, Y., Avraamidou, S., Li, J., Xiao, X., Pistikopoulos, E.N.: Land use modeling and optimization based on food-energy-water nexus: a case study on crop-livestock systems. In: Eden M.R., Ierapetritou M.G., Towler G.P. (eds.) 13th International Symposium on Process Systems Engineering (PSE 2018), Computer Aided Chemical Engineering, vol. 44, pp. 1939–1944. Elsevier (2018)
61. Nie, Y., Avraamidou, S., Xiao, X., N.P.E., Li, J.: Two-stage land use optimization for a food-energy-water nexus study: a case study in Texas edwards region. In: Muñoz S.G., Laird C.D., Realff M.J. (eds.) Proceedings of the 9th International Conference on Foundations of Computer-Aided Process Design, Computer Aided Chemical Engineering, vol. 47, pp. 205–210. Elsevier (2019)

62. Nie, Y., Avraamidou, S., Xiao, X., Pistikopoulos, E.N., Li, J., Zeng, Y., Song, F., Yu, J., Zhu, M.: A food-energy-water nexus approach for land use optimization. *Sci. Total Environ.* **659**, 7–19 (2019b)
63. Nishizaki, I., Sakawa, M.: Computational methods through genetic algorithms for obtaining stackelberg solutions to two-level integer programming problems. *Cybern. Syst.* **36**(6), 565–579 (2005)
64. Oberdieck, R., Diangelakis, N.A., Avraamidou, S., Pistikopoulos, E.N.: On unbounded and binary parameters in multi-parametric programming: applications to mixed-integer bilevel optimization and duality theory. *J. Glob. Optim.* **69**(3), 587–606 (2017)
65. Paulavicius, R., Kleniati, P.M., Adjiman, C.S.: A library of nonconvex bilevel test problems with the corresponding ampl input files (version v1.0). [Data set] (2016)
66. Powell, M.J.: A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Hennart, J.P., Gomez, S. (eds.) *Advances in Optimization and Numerical Analysis*, pp. 51–67. Springer, Berlin (1994)
67. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Glob. Optim.* **56**, 1247–1293 (2013)
68. Runarsson, T.P., Yao, X.: Search biases in constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **35**(2), 233–243 (2005)
69. Sahin, K.H., Ciric, A.R.: A dual temperature simulated annealing approach for solving bilevel programming problems. *Comput. Chem. Eng.* **23**(1), 11–25 (1998)
70. Sedlaczek, K., Eberhard, P.: Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. *Struct. Multidisc. Optim.* **32**(4), 277–286 (2006)
71. Simaan, M., Cruz, J.B.: On the stackelberg strategy in nonzero-sum games. *J. Optim. Theory Appl.* **11**(5), 533–555 (1973)
72. Sinha, A., Malo, P., Frantsev, A., Deb, K.: Multi-objective stackelberg game between a regulating authority and a mining company: a case study in environmental economics. In: 2013 IEEE Congress on Evolutionary Computation, pp. 478–485. IEEE (2013)
73. Sinha, A., Malo, P., Deb, K.: A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Trans. Evol. Comput.* **22**(2), 276–295 (2017)
74. Stackelberg, H.: *Theory of the Market Economy*. Oxford University Press, Oxford (1952)
75. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. *Math. Program* **103**, 225–249 (2005)
76. Vu, K.K., D'Ambrosio, C., Hamadi, Y., Liberti, L.: Surrogate-based methods for black-box optimization. *Int. Trans. Oper. Res.* **24**(3), 393–424 (2017)
77. Wilson, Z.T., Sahinidis, N.V.: The ALAMO approach to machine learning. *Comput. Chem. Eng.* **106**, 785–795 (2017)
78. Woldemariam, A.T., Kassa, S.M.: Systematic evolutionary algorithm for general multilevel stackelberg problems with bounded decision variables (SEAMSP). *Ann. Oper. Res.* **229**(1), 771–790 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.