Noname manuscript No.
(will be inserted by the editor)

Siting Renewable Power Generation Assets with Combinatorial Optimisation

Mathias Berger · David Radu · Antoine Dubois · Hrvoje Pandžić[†] · Yury Dvorkin[‡] · Quentin Louveaux · Damien Ernst

Received: date / Accepted: date

Abstract This paper studies the problem of siting renewable power generation assets using large amounts of climatological data while accounting for their spatiotemporal complementarity. The problem is cast as a combinatorial optimisation problem selecting a pre-specified number of sites so as to minimise the number of simultaneous low electricity production events that they experience relative to a pre-specified reference production level. It is shown that the resulting model is closely related to submodular optimisation and can be interpreted as generalising the well-known maximum coverage problem. Both deterministic and randomised algorithms are discussed, including greedy, local search and relaxation-based heuristics as well as combinations of these algorithms. The usefulness of the model and methods is illustrated by a realistic case study inspired by the problem of siting onshore wind power plants in Europe, resulting in instances featuring over ten thousand candidate locations and ten years of hourly-sampled meteorological data. The proposed solution methods are benchmarked against a state-of-the-art mixed-integer programming solver and several algorithms are found to consistently produce better solutions at a fraction of the computational cost. The physical nature of solutions provided by the model is also investigated, and all deployment patterns are found to be unable to supply a constant share of the electricity demand at all times. Finally, a cross-validation analysis shows that, except for an edge case, the model can successfully and reliably identify deployment patterns that perform well on previously unseen climatological data from historical data spanning a small number of weather years.

Mathias Berger
E-mail: mathias.berger@alumni.duke.edu
Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium
[†] Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia
[‡] Department of Electrical and Computer Engineering, NYU Tandon School of Engineering, New York, USA

Keywords coverage problems · submodular maximisation · combinatorial optimisation · renewable energy · asset siting · resource complementarity

1 Introduction

In recent years, the large-scale deployment of technologies harnessing renewable energy sources (RES) for electricity production has been a mainstay of climate and decarbonisation policies. However, widely-available RES (e.g., solar irradiance and wind) are inherently variable on time scales ranging from minutes to years [22], which greatly complicates power system operation and planning procedures. Several solutions have been proposed to alleviate this issue, such as the widespread deployment of energy storage systems in power systems [32,65] or the implementation of demand response programs [55]. Alternatively, since the distribution of RES is heterogeneous in space and time, it has been suggested that carefully selecting renewable power generation sites to exploit this diversity could at least partly alleviate variability issues and reduce the residual demand (i.e., the part of the electricity demand that cannot be supplied by renewable power plants) [48,33].

The concept of complementarity of renewable resources, which formalises this idea, has received much attention of late [41]. In particular, a number of methods have been applied to evaluate resource complementarity, ranging from correlation coefficients [47,56] and frequency-domain analyses [3,43] to principal component analysis [46,71] and custom scalar indicators [59,66]. In addition, a novel framework [8] that is particularly well-suited for exploiting the vast amounts of high-resolution climatological data unlocked by sophisticated reanalysis models [31,24] has recently been proposed to carry out highly granular asset siting analyses. Roughly speaking, in this framework, locations are considered complementary if they rarely experience periods of simultaneous low electricity production relative to a pre-specified reference production level (that may be proportional to the electricity demand), and the resulting complementarity criterion is directly amenable to optimisation. Hence, it can be used to provide a systematic and efficient way of selecting deployment patterns while accounting for their spatiotemporal complementarity.

Building upon these ideas, in this paper, the renewable power generation asset siting problem is cast as a combinatorial optimisation problem selecting a pre-specified number of sites so as to minimise the number of simultaneous low-production events that they experience relative to a pre-specified reference production level. Model properties are carefully analysed, and a formal connection with submodular optimisation and coverage problems is established. Both deterministic and randomised algorithms are discussed, including greedy, local search and relaxation-based heuristics. The usefulness of the model and methods is illustrated by a realistic case study inspired by the problem of siting onshore wind power plants in Europe, resulting in instances featuring over ten thousand candidate sites and ten years of hourly-sampled meteorological data. The proposed solution methods are benchmarked against a state-of-

the-art mixed-integer programming (MIP) solver, and the physical nature of solutions provided by the model is also analysed. Finally, a cross-validation analysis is performed in order to evaluate whether the model and algorithms can successfully identify deployment patterns that perform well on previously unseen climatological data from historical data spanning a small number of weather years.

This paper is structured as follows. Section 2 reviews the relevant literature. In Section 3, the framework used to evaluate spatiotemporal complementarity is presented, the optimisation model is introduced, its properties are analysed, and solution methods are discussed. Section 4 presents the experimental set-up, benchmarks solution methods on a realistic test case and provides a detailed account of model capabilities. Finally, Section 5 concludes the paper and discusses future work directions.

2 Related Works

This section reviews optimisation models used in the context of power system planning and renewable asset siting as well as relevant algorithmic results.

On the one hand, centralised planning and game-theoretical models that optimise purely economic criteria such as revenue or total system cost have been proposed. For instance, the strategic investment in wind power generation assets [6], the co-optimisation of power generation and transmission assets [50,37] as well as the co-optimisation of merchant storage and power transmission assets [21] have all been studied. In theory, such models are capable of evaluating the economic implications of strategic renewable power generation asset siting and resource complementarity for power systems. However, owing to computational limitations, these models usually have fairly low spatial and temporal resolutions, which makes it difficult to accurately capture correlations between variable renewable resources and properly site renewable power generation assets.

On the other hand, models such as the one presented in this paper put more emphasis on the representation of renewable resources at the expense of other modelling features such as network physics and constraints. These models typically have much higher spatial and temporal resolutions and use non-monetary objectives such as the residual demand. For example, Musselman et al. [51] proposed two mixed-integer linear programming models that seek to site wind power plants so as to balance two competing objectives. More precisely, the first model seeks to simultaneously minimise the average residual demand and the average step-wise power output variability, while the second formulation attempts to simultaneously minimise the average residual demand and the maximum increase in residual demand over all sets of successive time periods of pre-specified length. Greedy heuristics are proposed to solve the problems approximately. In the same vein, Wu et al. [69] proposed a linear optimisation model that sites and sizes wind power plants so as to minimise the maximum residual demand in different regions of Southern Africa.

It is unclear whether the proposed model has continuous or integer variables. Zappa et al. [70] took a similar approach to wind and solar photovoltaic power generation asset deployment in Europe and formulated a linear least-squares model minimising the average residual demand. Notably, a full matrix storing capacity factors appears in the constraints or in the objective in all of these models [51,69,70], which limits their scalability.

From an algorithmic standpoint, the combinatorial optimisation problem studied in this paper is closely related to submodular maximisation [53] and coverage problems [40,19,38,7]. A number of negative results are known for these two classes of problems. More specifically, the problem of maximising a monotone submodular set function subject to a cardinality constraint is known to be NP-hard [17]. Thus, unless P = NP, no exact polynomial-time algorithm exists. Alternatives include approximation algorithms, which typically have both polynomial time complexity and worst-case performance guarantees. However, Nemhauser and Wolsey [52] showed that approximation algorithms for monotone submodular maximisation over the uniform matroid (which generalises the cardinality constraint) may not achieve an approximation ratio better than $1 - 1/\mathrm{e}$ (where e is the base of the natural logarithm) with a polynomial number of queries to the objective function. In other words, in the value oracle model, the problem is also hard to approximate. Inapproximability results are also known for several coverage problems, including *maximum coverage* [38] (which, given a collection of subsets of a ground set, consists in selecting k subsets so as to maximise the cardinality of their union), *maximum ℓ-multi-coverage* [7] (which can be viewed as a relaxed version of maximum coverage where an element can be counted up to ℓ times and is also a particular case of submodular maximisation subject to a cardinality constraint), *set cover* [40,16] and *set multicover* [19,57,58] (which, given a collection of subsets of a ground set, consist in selecting the smallest number of subsets so as to guarantee that each element of the ground set is covered at least once or multiple times, respectively). More precisely, unless P = NP, it is known that no polynomial-time algorithm with an approximation ratio better than $1-1/\mathrm{e}$ exists for the maximum coverage problem [25]. In addition, under the unique games conjecture [42], the best approximation ratio that any polynomial-time algorithm can achieve for maximum ℓ-multi-coverage is $1 - \ell^\ell \mathrm{e}^{-\ell}/\ell!$ [7]. Feige [25] also showed that set cover cannot be approximated efficiently below a threshold of $(1 - o(1)) \ln n$ (where n is the number of elements to be covered), unless NP has slightly superpolynomial-time algorithms. Finally, set multicover is essentially as hard as set cover [57].

Three main classes of approximation algorithms have been studied for submodular maximisation and coverage problems, namely greedy [53,44,4,49], local search [26,27] and (multi-)linear relaxation-based [12,14] algorithms. Although several algorithms may have the same (tight) approximation ratio for a given problem, their computational complexity may vary widely. In particular, greedy algorithms often provide good approximation guarantees and have low computational complexity. The classic greedy algorithm [53] requires $O(nk)$ function evaluations to achieve a constant approximation ratio of $1-1/\mathrm{e}$ (both

for monotone submodular maximisation subject to a cardinality constraint [53] and maximum coverage [38]). In addition, the first linear-time algorithm for monotone submodular maximisation subject to a cardinality constraint is a variant of the classic greedy algorithm with randomised partial enumeration that was introduced by Mirzasoleiman et al. [49]. Its complexity is $O(n \log \frac{1}{\epsilon})$ and an approximation ratio of $1 - 1/\mathrm{e} - \epsilon$ is guaranteed in expectation for any $\epsilon > 0$. Greedy algorithms also achieve logarithmic approximation ratios for both set cover [40,16] and set multicover [19,57]. Interestingly, for maximum ℓ-multi-coverage, the greedy algorithm does not have a tight approximation ratio [7]. Instead, the best algorithm combines a linear programming relaxation with the pipage rounding scheme [2,7]. Finally, it is worth mentioning that sophisticated heuristics lacking theoretical guarantees have also been used to tackle coverage problems, such as the greedy randomized adaptive search procedure (GRASP) of Resende [63].

3 Methods

This section presents the framework used to quantify the spatiotemporal complementarity of candidate power generation sites, introduces the combinatorial optimisation problem used for siting renewable power generation assets and carefully analyses its properties. An integer programming formulation of this problem is also discussed along with several solution methods.

3.1 Evaluating Spatiotemporal Complementarity

Recall that in the framework of [8], locations are considered complementary if they rarely experience periods of simultaneous low electricity production (compared with a pre-specified reference production level). A set function formalising this intuition and assigning a complementarity score to any set of candidate sites is derived next.

Formally, a geographical region is represented by a finite set of locations $\mathcal{L}, |\mathcal{L}| = L$, and a time series $\mathbf{s}_l \in \mathbb{R}_+^T$ describing renewable resource data (e.g., wind speed, solar irradiation) over a set of time periods $\mathcal{T}$, $|\mathcal{T}| = T$, is assumed to be available at each location $l \in \mathcal{L}$. The instantaneous power output of a given location $l \in \mathcal{L}$ is estimated using a suitable transfer function $h_l : \mathbb{R}_+^T \to \mathbb{R}_+^T$ and is represented by a time series $\mathbf{u}_l = h_l(\mathbf{s}_l)$. This transfer function may be that of a single RES power generation technology (e.g., a wind turbine or a solar photovoltaic module) or that of an entire power station (e.g., a wind farm or a photovoltaic power station).

Furthermore, a set of time windows $\mathcal{W} \subseteq 2^{\mathcal{T}}$, $|\mathcal{W}| = W$, is constructed from the set of time periods $\mathcal{T}$. More precisely, a time window $\mathrm{w} \in \mathcal{W}$ can be interpreted as a subset $\mathcal{T}_\mathrm{w} \subseteq \mathcal{T}$ of δ successive time periods, and all time windows $\mathrm{w} \in \mathcal{W}$ have the same length δ. Note that successive time windows overlap and share exactly $\delta - 1$ time periods, while the union of all time windows covers the set of time periods.

The electricity production level $p_{\mathrm{w}l}$ of each candidate site $l \in \mathcal{L}$ is evaluated over the duration of each time window $\mathrm{w} \in \mathcal{W}$ using a prescribed measure $q_l : \mathbb{R}_+^T \times \mathcal{W} \to \mathbb{R}_+$, such that $p_{\mathrm{w}l} = q_l(\mathbf{u}_l, \mathrm{w})$. This measure may for instance compute the average production level over each window $\mathrm{w} \in \mathcal{W}$. This would essentially be equivalent to applying a moving average-based filter to the original power production signal and result in a smoothed power output signal. The degree of smoothing would be controlled by δ, which makes it possible to study resource complementarity on different time scales. A local, time-dependent reference production level $\alpha_{\mathrm{w}l} \in \mathbb{R}_+$ is also specified at each candidate site $l \in \mathcal{L}$, and may for instance be proportional to the electricity demand. A location $l \in \mathcal{L}$ is considered productive enough over window $\mathrm{w} \in \mathcal{W}$ if $p_{\mathrm{w}l} \geq \alpha_{\mathrm{w}l}$, and location l is then said to *cover* window w.

In order to formalise the intuitive definition of resource complementarity introduced earlier, a *coverage parameter* $c \in \mathbb{N}$ is specified, such that for any subset of candidate locations $L \subseteq \mathcal{L}$, a window $\mathrm{w} \in \mathcal{W}$ is said to be *c-covered* if at least c locations produce enough electricity over its duration. Let N_{w}^L denote the number of candidate sites $l \in L$ covering window $\mathrm{w} \in \mathcal{W}$, and let $f_c : 2^{\mathcal{L}} \to \mathbb{N}$ be a nonnegative set function associating its number of c-covered time windows to any subset of locations $L \subseteq \mathcal{L}$,

$$f_c(L) = |\{\mathrm{w} \in \mathcal{W} | N_{\mathrm{w}}^L \geq c\}|. \tag{1}$$

Computing this function takes $O(|L|W)$ elementary (arithmetic and comparison) operations. In addition, dividing the number of c-covered windows $f_c(L)$ by the total number of time windows W shows that this function can be interpreted as quantifying the empirical probability of having sufficient levels of electricity production across at least c locations simultaneously. A low value of $f_c(L)$ therefore implies that simultaneous low electricity production events occur often, which indicates poor complementarity between locations. In the context of renewable power plant siting, the coverage parameter c can be interpreted as distributing power generation across sites while imposing an implicit joint electricity production requirement.

3.2 Combinatorial Optimisation Model

The combinatorial optimisation problem proposed for siting renewable power generation assets selects a pre-defined number of sites $k \in \mathbb{N}$ so as to maximise the number of c-covered windows throughout the horizon considered, which can be expressed as follows

$$\begin{aligned} \max_{L \subseteq \mathcal{L}} \quad & f_c(L) \\ \text{s.t. } & |L| = k. \end{aligned} \tag{2}$$

It is worth analysing the properties of f_c more closely in order to gain a better understanding of the problem at hand. To this end, the concepts of monotone and submodular set functions are formally introduced next.

Definition 1 (Monotonicity) Let Z be a finite ground set and let $g : 2^Z \to \mathbb{N}$ be a nonnegative set function. Then, g is said to be monotone (non-decreasing) if and only if $g(X) \leq g(Y)$ for every $X \subseteq Y \subseteq Z$.

Definition 2 (Submodularity) Let Z be a finite ground set and let $g : 2^Z \to \mathbb{N}$ be a nonnegative set function. Then, g is said to be submodular if and only if $g(X) + g(Y) \geq g(X \cup Y) + g(X \cap Y),\ \forall X, Y \subseteq Z$.

Putting these definitions to use yields the following results.

Proposition 1 *Let $\mathcal{L}$ be a finite set of locations and let $f_c : 2^{\mathcal{L}} \to \mathbb{N}$ be the set function defined in Eq.* (1)*. Then, f_c is monotone non-decreasing.*

Proof Let us assume that $\exists L' \subsetneq L \subseteq \mathcal{L}$ such that $f_c(L') > f_c(L)$. This implies that there exists at least one window $\mathrm{w} \in \mathcal{W}$ that is c-covered by L' but not by L. However, since L' is a proper subset of L, the number of locations in L that cover any window must be at least equal to that of L', which leads to a contradiction. □

Proposition 2 *(Diminishing Returns) Let Z be a finite ground set and let $g : 2^Z \to \mathbb{N}$ be a nonnegative set function. Then, g is submodular if and only if $g(X \cup \{z\}) - g(X) \geq g(Y \cup \{z\}) - g(Y),\ \forall X \subseteq Y \subseteq Z$ and $z \in Z \setminus Y$ [53].*

Proposition 3 *Let $\mathcal{L}$ be a finite set of locations and let $f_c : 2^{\mathcal{L}} \to \mathbb{N}$ be the set function defined in Eq.* (1)*. Then, f_c is submodular if $c = 1$. However, in general, f_c is not submodular if $c > 1$.*

Proof Let $L_1 \subseteq \mathcal{L}$ and $L_2 \subseteq \mathcal{L}$ be two arbitrary subsets of locations. We seek to show that f_c satisfies Definition 2 if $c = 1$. Let $\mathrm{N} : 2^{\mathcal{L}} \times \mathcal{W} \to \mathbb{N}$ be a function that, to any subset $L \subseteq \mathcal{L}$ and window $\mathrm{w} \in \mathcal{W}$, associates the number of locations in L that cover w. It is clear that the following identities hold, $\forall \mathrm{w} \in \mathcal{W}$,

$$\begin{aligned}
\mathrm{N}(L_1, \mathrm{w}) &= \mathrm{N}(L_1 \cap L_2, \mathrm{w}) + \mathrm{N}(L_1 \setminus L_2, \mathrm{w}), \\
\mathrm{N}(L_2, \mathrm{w}) &= \mathrm{N}(L_1 \cap L_2, \mathrm{w}) + \mathrm{N}(L_2 \setminus L_1, \mathrm{w}), \\
\mathrm{N}(L_1 \cup L_2, \mathrm{w}) &= \mathrm{N}(L_1 \cap L_2, \mathrm{w}) + \mathrm{N}(L_1 \setminus L_2, \mathrm{w}) + \mathrm{N}(L_2 \setminus L_1, \mathrm{w}).
\end{aligned}$$

In addition, let $g : \mathbb{N} \to \{0, 1\}$ be a function such that

$$g(k) = \begin{cases} 1 & \text{if } k \geq 1 \\ 0 & \text{otherwise} \end{cases}.$$

Note that for $c = 1$, f_c can be simply expressed as a sum of compositions of g and N, hence

$$f_c(L) = \sum_{\mathrm{w} \in \mathcal{W}} g\big(\mathrm{N}(L, \mathrm{w})\big),\ \forall L \subseteq \mathcal{L}.$$

Substituting this expression in Definition 2 leads to

$$\sum_{\mathrm{w}\in\mathcal{W}} g\big(\mathrm{N}(L_1,\mathrm{w})\big) + \sum_{\mathrm{w}\in\mathcal{W}} g\big(\mathrm{N}(L_2,\mathrm{w})\big) \geq \sum_{\mathrm{w}\in\mathcal{W}} g\big(\mathrm{N}(L_1\cup L_2,\mathrm{w})\big) + \sum_{\mathrm{w}\in\mathcal{W}} g\big(\mathrm{N}(L_1\cap L_2,\mathrm{w})\big).$$

Thus, showing that the composition of g and N satisfies Definition 2 for any window $\mathrm{w}\in\mathcal{W}$ suffices for our purpose. For the sake of clarity, in the following, $\mathrm{N}(L,\mathrm{w})$ will be written as $\mathrm{N}(L)$. Now, invoking the identities introduced above yields

$$g\big(\mathrm{N}(L_1\cap L_2)+\mathrm{N}(L_1\setminus L_2)\big)+g\big(\mathrm{N}(L_1\cap L_2)+\mathrm{N}(L_2\setminus L_1)\big) \geq g\big(\mathrm{N}(L_1\cap L_2)+\mathrm{N}(L_1\setminus L_2)+\mathrm{N}(L_2\setminus L_1)\big)+g\big(\mathrm{N}(L_1\cap L_2)\big).$$

A number of cases can be considered. If $\mathrm{N}(L_1\cap L_2)\geq 1$, the inequality is always tight. By contrast, if $\mathrm{N}(L_1\cap L_2)=0$, the second term on the right-hand side will be null. In this case, the inequality will be tight if $\mathrm{N}(L_1\setminus L_2)=0$ or $\mathrm{N}(L_2\setminus L_1)=0$, while the inequality will be strict if $\mathrm{N}(L_1\setminus L_2)\geq 1$ and $\mathrm{N}(L_2\setminus L_1)\geq 1$. Since the inequality is satisfied in all cases, f_c is submodular if $c=1$.

In order to show that f_c is not submodular if $c>1$, let us assume that it is not uniformly equal to 0 over $2^{\mathcal{L}}$. Then, $\exists L\subseteq\mathcal{L}$ that has a strictly positive value $f_c(L)>0$. Note that $|L|\geq c$, and it therefore has nontrivial proper subsets. In particular, $\exists L'\subsetneq L$ nontrivial and $\exists l\in L, l\notin L'$, such that $f_c(L')=0$ and $f_c(L'\cup\{l\})>0$. In addition, $\exists L''\subsetneq L'$ (possibly empty) such that $f_c(L'')=0$ and $f_c(L''\cup\{l\})=0$. Taking $X=L''$, $Y=L'$, $z=l$ and substituting them in the inequality of Proposition 2 shows that the diminishing returns property of submodular functions is violated. □

Based on the algorithmic literature surveyed in Section 2, these results imply that problem (2) is both NP-hard and hard to approximate. An integer programming formulation of problem (2) is discussed next.

3.3 Integer Programming Formulation

Let $D_{\mathrm{w}l}\in\{0,1\}$ denote the entry of a binary matrix indicating whether location $l\in\mathcal{L}$ covers window $\mathrm{w}\in\mathcal{W}$, such that $D_{\mathrm{w}l}=1$ if location l covers window w and $D_{\mathrm{w}l}=0$ otherwise. Let $x_l\in\{0,1\}$ be a binary variable indicating whether location $l\in\mathcal{L}$ is selected for deployment, such that $x_l=1$ if location l is selected for deployment and $x_l=0$ otherwise. Recall that a window $\mathrm{w}\in\mathcal{W}$ is said to be c-covered if $\sum_{l=1}^{L} D_{\mathrm{w}l}x_l\geq c$. Let $y_{\mathrm{w}}\in\{0,1\}$ be a binary variable indicating whether window $\mathrm{w}\in\mathcal{W}$ is c-covered, such that $y_{\mathrm{w}}=1$ if window w is c-covered and $y_{\mathrm{w}}=0$ otherwise.

Then, problem (2) admits the following integer programming formulation

$$\max \quad \sum_{\mathrm{w}=1}^{W} y_{\mathrm{w}} \tag{3}$$

$$\text{s.t.} \sum_{l=1}^{L} D_{\mathrm{w}l} x_l \geq c y_{\mathrm{w}}, \ \mathrm{w} = 1, \ldots, W, \tag{4}$$

$$\sum_{l=1}^{L} x_l = k, \tag{5}$$

$$x_l \in \{0, 1\}, \ l = 1, \ldots, L, \tag{6}$$

$$y_{\mathrm{w}} \in \{0, 1\}, \ \mathrm{w} = 1, \ldots, W. \tag{7}$$

The objective function (3) simply computes the number of c-covered time windows over the time horizon of interest. Constraints (4) indicate whether windows are c-covered by the subset of locations selected for deployment. The cardinality constraint (5) ensures that exactly k locations are selected for deployment, while constraints (6) and (7) express the binary nature of location selection and time window coverage, respectively.

3.4 Connection with Coverage Problems

Besides the connection with submodular optimisation, which is made apparent in Section 3.2, the problem considered in this paper is closely related to coverage problems already studied in the literature. The nature of these connections is investigated more thoroughly in this subsection.

First, it is clear from its integer programming formulation that the problem at hand can be interpreted as generalising the maximum coverage problem [38], which is retrieved for $c = 1$. More specifically, it generalises maximum coverage in the same way that set multicover [19] generalises set cover [40]. Indeed, in set cover, each element of the ground set must be covered at least once, whereas each element must be covered multiple times in set multicover.

Barman et al. [7] studied a different coverage problem that can also be construed as generalising maximum coverage, although in a less direct way. Their problem is as follows. Given a collection of subsets of a ground set, the goal is to select k subsets so as to maximise the number of times each element of the ground set is covered, but the number of times an element is covered can count towards the objective only up to a certain (integer) threshold $\ell > 0$. In other words, if an element is covered by more than ℓ subsets, its contribution to the objective will only be ℓ. By contrast, in this paper, an element will contribute to the objective if and only if it is covered by at least c subsets. This difference has far-reaching algorithmic implications, as the submodularity of the objective function considered in [7] is preserved for any $\ell > 1$, while submodularity breaks down as soon as $c > 1$ in problem (2). Interestingly, the problem considered in this paper and the one studied in [7] share the

same linear programming relaxation (up to a positive constant multiplying the objective). More formally, let $F_c : [0,1]^L \to \mathbb{R}_+$ be such that

$$F_c(x) = \frac{1}{c} \sum_{\mathrm{w}=1}^{W} \min \Big\{ c, \sum_{l=1}^{L} D_{\mathrm{w}l} x_l \Big\}.$$

Then, it is easy to see that the linear programming relaxation of problem (3) is equivalent to

$$\begin{aligned} \max \quad & F_c(x) \\ \text{s.t.} \quad & \sum_{l=1}^{L} x_l = k \\ & x_l \in [0,1], \ l = 1, \ldots, L, \end{aligned} \tag{8}$$

where F_c is essentially a scaled version of the objective considered in [7]. It is also worth noting that F_c only coincides with f_c on integral vectors (any of which can be interpreted as the characteristic vector of a set $L \subseteq \mathcal{L}$) if $c = 1$, and the quality of the approximation of f_c offered by F_c thus deteriorates as the value of the coverage parameter increases.

Finally, a connection with a coverage problem on hypergraphs can also be pointed out. More precisely, since there is a one-to-one correspondence between binary matrices and undirected hypergraphs [28] (i.e., any binary matrix can be interpreted as the incidence matrix of an undirected hypergraph), the present problem would be equivalent to the *densest k-subhypergraph* problem [15] on c-uniform hypergraphs if $\sum_{l=1}^{L} D_{\mathrm{w}l} = c$ for all $\mathrm{w} \in \mathcal{W}$ (i.e., if each row of D summed to c). Such a situation seems very unlikely to occur in practice, however.

In summary, even though problem (2) is closely related to several coverage problems studied in the literature, to the authors' best knowledge, it has not been considered as such elsewhere.

3.5 Solution Methods

This section discusses a number of solution methods, including state-of-the-art mixed-integer programming solvers, a mixed-integer relaxation of problem (3), two randomised greedy algorithms, a randomised local search algorithm, and algorithms combining some of the aforementioned methods.

3.5.1 Mixed-Integer Programming Solvers

State-of-the-art mixed-integer programming solvers such as Gurobi [35], CPLEX [18] and SCIP [29] typically implement a branch-and-bound (B&B) algorithm augmented with a variety of heuristics, powerful pre-solve capabilities and constraint programming techniques [1]. The core algorithm usually relies on a

linear programming engine to solve a sequence of relaxations, which are tightened using a variety of cuts, while bounding schemes are used to efficiently explore the search space. Such solvers have now reached a level of maturity enabling them to tackle and solve large-scale MIP problems encountered in industrial applications [30], in spite of their unfavourable computational complexity. These methods can be used to directly solve the integer programming formulation (3) or solve a relaxed version of it. In particular, the latter approach can usually be combined with rounding techniques or other algorithms to yield efficient heuristics, as discussed in Sections 3.5.2, 3.5.6 and 4.2.

3.5.2 Mixed-Integer Relaxation

This algorithm directly solves a mixed-integer relaxation (MIR) of problem (3) using a mixed-integer programming solver. The mixed-integer relaxation is formed by relaxing the integrality constraint (7) of time window variables. The key advantage of this approach lies in the fact that siting variables remain integer in the solution and it is therefore not necessary to apply a rounding scheme in order to recover a feasible solution. However, the worst-case time complexity of the algorithm may not be polynomial. From a practical standpoint, this can be mitigated by imposing lax requirements on the algorithm used to solve the mixed-integer relaxation (e.g., stopping as soon as a moderate optimality gap is achieved). Finally, note that this algorithm is essentially equivalent to solving problem (8) with an integrality requirement added for siting variables. Hence, one can intuitively expect the performance of this algorithm to decrease as the value of the coverage parameter increases.

3.5.3 Randomised Greedy Algorithm

Greedy algorithms form a class of purely combinatorial algorithms that are conceptually simple, usually offer good worst-case performance guarantees and have low computational complexity. This is especially true for the problem of maximising a submodular set function subject to a cardinality constraint [53] and maximum coverage [38], for which the classic greedy algorithm achieves an optimal approximation ratio with only $O(Lk)$ queries to the objective.

In the context of problem (2), the classic greedy algorithm would start from an empty set and, in each iteration, enumerate all unselected locations and add a location l to the incumbent solution $L \subseteq \mathcal{L}$ so as to maximise the objective $f_c(L \cup \{l\})$. This procedure would be repeated until k locations have been added. However, if $c > 1$, the classic greedy algorithm will struggle to find good locations in the first $c-1$ iterations since $f_c(L) = 0$ if $|L| < c$. Hence, the classic greedy algorithm will myopically select locations in the first $c - 1$ iterations, which most certainly has a detrimental impact on its performance. To remedy this, an auxiliary objective function $\hat{f}_c : 2^{\mathcal{L}} \times \mathbb{N} \to \mathbb{N}$ is introduced, which is such that

$$\hat{f}_c(L, i) = \begin{cases} f_{i+1}(L) & \text{if } i < c \\ f_c(L) & \text{otherwise} \end{cases} . \tag{9}$$

This auxiliary function can be interpreted as implementing a moving threshold that increases with the number of selected locations in order to drive the selection of good locations by the greedy algorithm in the first $c-1$ iterations. The auxiliary objective indeed reduces to the original objective function once $c-1$ locations have been selected.

In practice, many problems display some level of symmetry and it is fairly common to identify several locations that yield the same (maximal) increase in objective value in a given iteration. A number of tie-breaking mechanisms can be envisaged. A popular deterministic mechanism consists in breaking ties lexicographically. Nevertheless, computational experiments have shown that randomising the tie-breaking process can be beneficial for problems related to problem (2) [34]. Such a strategy is therefore adopted in this paper as well. Algorithm 1 displays the resulting randomised greedy (RG) algorithm.

Algorithm 1 Randomised Greedy Algorithm

Input $\mathcal{L},\ k,\ \hat{f}_c$
  $L \leftarrow \emptyset$
  $i \leftarrow |L|$
  while $|L| < k$ **do**
    $L^\star \leftarrow \arg\max_{l \in \mathcal{L} \setminus L} \hat{f}_c(L \cup \{l\}, i)$
    $l \leftarrow$ one location sampled from $L^\star$ uniformly at random
    $L \leftarrow L \cup \{l\}$
    $i \leftarrow |L|$
  end while
Output $L, \hat{f}_c(L, k)$

3.5.4 Randomised Greedy Algorithm with Partial Enumeration

The randomised greedy algorithm discussed in Section 3.5.3 has complexity $O(kL)$ as a result of the fact that all unselected locations are enumerated in each iteration. A straightforward way of speeding it up consists in using partial enumeration instead. In addition, the partial enumeration process can be randomised. This idea was already exploited by Mirzasoleiman et al. [49] to design the first linear-time (randomised) algorithm achieving an (almost-optimal) approximation ratio of $1 - 1/\mathrm{e} - \epsilon$ (guaranteed in expectation). In their algorithm, the approximation ratio to be achieved dictated the size of the subset sampled and enumerated in each iteration through ϵ.

Since no such result is currently available for problem (2), in this paper, the number of locations sampled and enumerated in each iteration is set by a fixed parameter $p \in [0, 1]$ that represents a fraction of the total number of candidate locations $|\mathcal{L}|$. Algorithm 2 displays the resulting randomised greedy algorithm with partial enumeration (RGP).

Algorithm 2 Randomised Greedy Algorithm with Partial Enumeration

Input $\mathcal{L},\ k,\ p,\ \hat{f}_c$
  $L \leftarrow \emptyset$
  $i \leftarrow |L|$
  $s \leftarrow \lceil p|\mathcal{L}| \rfloor$
  while $|L| < k$ **do**
    $R \leftarrow s$ locations sampled from $\mathcal{L} \setminus L$ uniformly at random
    $L^\star \leftarrow \arg\max_{l \in R} \hat{f}_c(L \cup \{l\}, i)$
    $l \leftarrow$ one location sampled from $L^\star$ uniformly at random
    $L \leftarrow L \cup \{l\}$
    $i \leftarrow |L|$
  end while
Output $L, \hat{f}_c(L, k)$

3.5.5 Simulated Annealing Local Search

Basic local search heuristics are known to sometimes have both undesirable worst-case time complexity and weaker worst-case performance guarantees than greedy algorithms for problems closely related to problem (2). For instance, Nemhauser et al. [53] show that for monotone submodular maximisation subject to a cardinality constraint, a classic deterministic exchange heuristic can take a number of iterations that is exponential in the cardinality of the solution before finishing. Although local search heuristics that terminate in a polynomial number of iterations and have good worst-case performance guarantees exist [26], their computational complexity remains too high to be practical. On the other hand, simple randomised local search heuristics such as the simulated annealing (SA) algorithm [10] have met with considerable success in practice. In this paper, a local search heuristic that performs a pre-specified number of iterations and is inspired by the simulated annealing algorithm is employed.

Starting from a solution $L_0 \subseteq \mathcal{L}, |L_0| = k$, the local search heuristic performs a fixed number of iterations $I \in \mathbb{N}$ in the hope of improving the initial solution. More specifically, in each iteration, a fixed number $N \in \mathbb{N}$ of neighbouring solutions is drawn uniformly at random from the neighbourhood of the incumbent solution $L \subseteq \mathcal{L}, |L| = k$. This neighbourhood is formed by solutions that share exactly $k - r$ locations with the incumbent solution. Hence, a neighbouring solution $\hat{L}$ can be constructed from the incumbent solution by selecting r different locations from L, r different locations from $\mathcal{L} \setminus L$ and swapping them. Each of the N neighbouring solutions is tested against the incumbent solution and stored in a temporary candidate solution $\tilde{L}$ if it is found to outperform previously-explored neighbouring solutions. Their performance is evaluated via the difference $\tilde{\Delta}$ between the objectives achieved by the neighbouring and incumbent solutions. Once N neighbouring solutions have been explored, the candidate solution corresponds to a neighbouring solution that maximises $\tilde{\Delta}$ among all sampled solutions. Note that $\tilde{\Delta}$ may be negative (i.e., if the algorithm does not manage to improve on the incumbent). If $\tilde{\Delta} > 0$, the candidate solution becomes the new incumbent solution. By contrast, if $\tilde{\Delta} < 0$,

whether the candidate solution becomes the new incumbent solution depends on the outcome b of a random variable drawn from a Bernoulli distribution with parameter p. This parameter depends on both $\tilde{\Delta}$ and the so-called *annealing temperature* $T(i)$. Roughly speaking, the annealing temperature controls the extent to which the search space is explored in an attempt to find better solutions and exit local optima. The temperature is specified by a *temperature schedule* that provides a temperature $T(i)$ for each iteration i. This procedure is repeated until the maximum number of iterations I is reached. Algorithm 3 summarises these ideas.

Algorithm 3 Simulated Annealing Local Search Algorithm

Input $\mathcal{L},\ L_0,\ I,\ N,\ r,\ T,\ f_c$
$L \leftarrow L_0$
$i \leftarrow 0$
while $i < I$ **do**
  $\tilde{\Delta} \leftarrow -\infty$
  $n \leftarrow 0$
  while $n < N$ **do**
    $S_+ \leftarrow r$ locations sampled from $\mathcal{L} \setminus L$ uniformly at random
    $S_- \leftarrow r$ locations sampled from L uniformly at random
    $\hat{L} \leftarrow (L \setminus S_-) \cup S_+$
    $\hat{\Delta} \leftarrow f_c(\hat{L}) - f_c(L)$
    if $\hat{\Delta} > \tilde{\Delta}$ **then**
      $\tilde{L} \leftarrow \hat{L}$
      $\tilde{\Delta} \leftarrow \hat{\Delta}$
    end if
    $n \leftarrow n + 1$
  end while
  if $\tilde{\Delta} > 0$ **then**
    $L \leftarrow \tilde{L}$
  else
    $p \leftarrow \exp(\tilde{\Delta}/T(i))$
    draw b from Bernoulli distribution with parameter p
    if $b = 1$ **then**
      $L \leftarrow \tilde{L}$
    end if
  end if
  $i \leftarrow i + 1$
end while
Output $L, f_c(L)$

3.5.6 Combinations of Algorithms

The first algorithm combines the mixed-integer relaxation of problem (3) discussed in Section 3.5.2 and the simulated annealing local search algorithm described in Section 3.5.5. The algorithm simply initialises the local search algorithm with the solution of the former algorithm, and is therefore called MIRSA.

The second algorithm combines the randomised greedy algorithm with partial enumeration discussed in Section 3.5.4 and the simulated annealing local search algorithm described in Section 3.5.5. More precisely, the algorithm runs RGP n times, takes the best solution and initialises the local search algorithm with it. This algorithm will be referred to as RGPSA.

3.6 Implementation

The branch-and-bound/cut algorithm used in this paper is the one provided by Gurobi 9.1 [35], which is a (commercial) state-of-the-art mixed-integer programming solver. All other algorithms discussed in Section 3.5 were implemented in the Julia programming language [11]. They are readily available as stand-alone scripts [9] and have also been integrated in a Python 3.7 framework that can be used for detailed asset siting analyses and facilitates data pre- and post-processing [61]. In addition, the mixed-integer programming relaxation of problem (3) was formulated in the Julia-based algebraic modelling language JuMP [20] and solved with Gurobi 9.1. All experiments were performed on a workstation running under CentOs, with an 18-core Intel Xeon Gold 6140 CPU clocking at 2.3 GHz and 256 GB RAM.

4 Numerical Experiments

In this section, the experimental set-up is described, the solution methods presented in Section 3.5 are benchmarked, the physical nature of solutions provided by the model is analysed, and a cross-validation analysis is carried out in order to evaluate the extent to which the model and algorithms can identify deployment patterns that perform well on unseen climatological data from a subset of weather years.

4.1 Experimental Set-Up

The proposed RES siting method is illustrated by a case study focusing on the deployment of onshore wind power generation assets in Europe. The resulting problem instances, which can be constructed using scripts and data provided in the following repositories [61,60], are described next.

Ten years of hourly-sampled wind data (from 2011 to 2020 included) with a spatial resolution of 0.25° in both coordinate directions are retrieved from the ERA5 reanalysis database [24]. It is assumed that onshore wind power plants can be deployed at every ERA5 onshore grid point, resulting in $L = 10138$ candidate sites. A wind speed time series $\mathbf{s}_l \in \mathbb{R}_+^T$ spanning $T = 87648$ time periods is associated to each site, and is converted into a capacity factor time series $\bar{\mathbf{s}}_l \in [0,1]^T$ using a pre-specified transfer function. More precisely, different transfer functions corresponding to different wind turbine models (the Vestas *V110*, Enercon *E103*, Vestas *V90* and Enercon *E126*) are considered in

this paper. One of these transfer functions is assigned to each candidate site based on its wind regime (taking its IEC class [39] into account) and passed through an additional filter modelling the local wind smoothing effect that results from the deployment of several wind turbines in the same area [54,36]. Candidate sites also have an effective technical potential $\kappa \in \mathbb{R}_+$ corresponding to the maximum power generation capacity that may be deployed there. This potential is assumed to be uniform across all sites and can be computed as

$$\kappa = \rho_p \times A_{cell} \times \sigma,$$

where ρ_p represents the power density of the chosen power generation technology (in $\mathrm{MW/km^2}$), A_{cell} denotes the surface area of the ERA5 grid cell (in $\mathrm{km^2}$) and σ is the dimensionless cell surface utilisation factor (since only a share of the cell surface area can be exploited for power generation due to competing land uses). In this paper, an onshore wind power density ρ_p of $3\,\mathrm{MW/km^2}$, a reference cell surface area A_{cell} of $521\,\mathrm{km^2}$ and a surface utilisation factor σ of 0.3 are assumed, resulting in an effective technical potential of around 0.47 GW per candidate site. The power output time series $\mathbf{u}_l \in \mathbb{R}_+^T$ of each candidate site is then computed by multiplying each entry of its capacity factor time series $\bar{\mathbf{s}}_l$ by its effective technical potential κ.

The number of sites k to be selected for deployment is derived from a recent study estimating that at least 263 GW of onshore wind power generation capacity should be available in Europe by 2030 in order to supply approximately 20% of the total European electricity demand [67]. Mapping this capacity requirement to a number of sites is achieved via the following formula,

$$k = \lceil C/\kappa \rceil, \tag{10}$$

where C denotes the required capacity and $\lceil \cdot \rceil$ represents the ceiling function. Hence, in total, $k = 560$ sites are required to deploy 263 GW of onshore wind power plants.

A time window length of $\delta = 1$ hour is considered, resulting in $W = 87648$ windows. Since $\delta = 1$, each time window w corresponds to a time period t, thus $p_{\mathrm{w}l} = u_{tl}$ (i.e., the power output is not smoothed). In addition, the reference production level $\alpha_{\mathrm{w}l} \in \mathbb{R}_+$ used for each candidate site is assumed to be time-dependent only (i.e., it is uniform across all locations). More precisely, $\alpha_{\mathrm{w}l}$ is assumed to be proportional to the aggregate hourly European electricity demand $\lambda_t \in \mathbb{R}_+$, which can be retrieved from the public ENTSO-E database [23] for the 2011-2020 period. Hence, $\alpha_{\mathrm{w}l} = \zeta\lambda_t$, where $\zeta = \kappa/\lambda_M \approx 10^{-3}$ is the ratio between the effective technical potential κ and the peak electricity demand $\lambda_M \approx 525\,\mathrm{GW}$. In other words, location $l \in \mathcal{L}$ covers window $\mathrm{w} \in \mathcal{W}$ if it produces more than 0.1% of the aggregate electricity demand over this time window, which is consistent with the number of candidate sites to be deployed, the share of the annual electricity demand to be supplied by onshore wind power plants and average capacity factors of typical European onshore wind power plants (around 25% [68]).

In the next subsections, different values of the coverage parameter $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$ are considered. The rationale behind the selection of these values is twofold. First, testing a broad range of c values makes it possible to assess the degree of spatiotemporal complementarity that European onshore wind sites exhibit and the extent to which this complementarity can be leveraged to effectively supply a fraction of the electricity demand as often as possible. For example, if $c = 224$, the model will identify the set of onshore wind power plants that produce more than 0.1% of the aggregate electricity demand on an individual basis and jointly supply more than (roughly) 20% of it as often as possible. The model will also compute the fraction of the time that they can do so. Second, as discussed in Section 3.4, the quality of the linear and mixed-integer linear programming relaxations of problem (3) is expected to decrease as the value of c increases. In other words, the value of the coverage parameter also has algorithmic implications that should be evaluated empirically.

4.2 Benchmarking

In this subsection, the algorithms described in Section 3.5 are benchmarked against a state-of-the-art mixed-integer programming solver for different values of the coverage parameter $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$. In the following experiments, mixed-integer linear relaxations of problem (3) are solved with Gurobi to an optimality gap of 1%. Moreover, p is set equal to 0.05 in all algorithms relying on RGP. The parameters of the simulated annealing local search heuristic are set to $I = 2000$, $N = 500$, $r = 1$, in all algorithms that rely on it, and an exponentially-decreasing temperature schedule $T(i) = 100 \times \exp(-10 \times i/I)$ is used. Finally, for RGPSA, $n = 10$.

First, the branch-and-bound-based (B&B) MIP solver is run with a time limit of 12 hours (43200 seconds). The optimality gaps achieved by the solver for different values of the coverage parameter are displayed in Table 1. The solver manages to solve the $c = 1$ instance to optimality in 112 seconds, but the time limit is reached for all other instances. In addition, the quality of the optimality gaps produced by the solver decreases substantially as the value of c increases, to the point of quickly becoming meaningless. This observation confirms the intuition that problem (3) has a weak linear programming relaxation for high values of c and suggests that the solver does not manage to improve it with cutting planes. This weak linear programming relaxation also limits the ability of the solver to effectively prune the enumeration tree, reduce the size of the search space and systematically identify good feasible solutions. Notably, for high values of c, virtually all good feasible solutions are produced by heuristics embedded within the solver. For the sake of completeness, run times are also reported for various algorithms in Table 2. All heuristics run in a few hundred seconds, which represents a small fraction of the time taken by the MIP solver.

c	1	28	56	112	224	336	448	560
Optimality Gap [%]	0.0	0.8	3.8	15.1	63.5	146.0	280.0	5533.0

Table 1: Optimality gaps returned by MIP solver for $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$ ($L = 10138$, $W = 87648$ and $k = 560$). All runs hit the time limit of 43200 seconds, except for $c = 1$, which was completed in 112 seconds.

c	B&B	MIR	RG	RGP	SA	MIRSA	RGPSA
1	112.0	111.9	878.6	47.8	191.1	303.0	669.1
28	43200.0	146.6	881.3	47.8	192.3	338.9	670.3
56	43200.0	185.4	885.4	46.9	199.3	384.7	668.3
112	43200.0	205.6	888.4	45.5	179.9	385.5	634.9
224	43200.0	179.6	910.8	48.2	184.2	363.8	666.2
336	43200.0	151.8	862.2	46.0	193.0	344.8	653.0
448	43200.0	140.7	915.5	48.4	180.9	321.6	664.9
560	43200.0	171.5	936.8	47.0	182.6	354.1	652.6

Table 2: Typical run times of different algorithms (in seconds) for the instance considered ($L = 10138$, $W = 87648$ and $k = 560$). For randomised algorithms, run times are averaged over twenty runs.

Since the output of randomised algorithms is inherently stochastic, running them several times is necessary to provide an accurate account of their performance. In this paper, the number of times each algorithm was run was determined based on some preliminary testing, and usually depended on the variance of its output. More specifically, RG was run a hundred times, RGP was run three hundred times, and each algorithm involving SA was run fifty times. A random search (RS) algorithm was also run and explored one million candidate solutions. Figure 1 displays the outcomes of these experiments, where the objective returned by each algorithm is divided by the objective of the solution provided by the MIP solver for each value of $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$.

For $c = 1$, all algorithms except the random search manage to find an optimal solution. The solution found by RS is fairly close to being optimal, which suggests that this instance is relatively easy to solve. For $c = 28$ and $c = 56$, all algorithms except RS virtually achieve the same objective value. Although these instances could not be solved to optimality by the MIP solver before hitting the time limit, the optimality gaps (0.8% and 3.8%, respectively) imply that the solutions produced by all algorithms are provably near-optimal (especially for $c = 28$). For $c = 112$, all algorithms except RGP and RS are again virtually tied. The output of RGP displays some variance, although the quality of the median solution (indicated by a dark blue dot in Figure 1) that it returns is fairly close to that of other algorithms. For $c = 224$, the three algorithms that rely on the local search heuristic slightly outperform all other algorithms. Note that the local search heuristic initialised with a random solution achieves objective values comparable to those achieved by both MIRSA

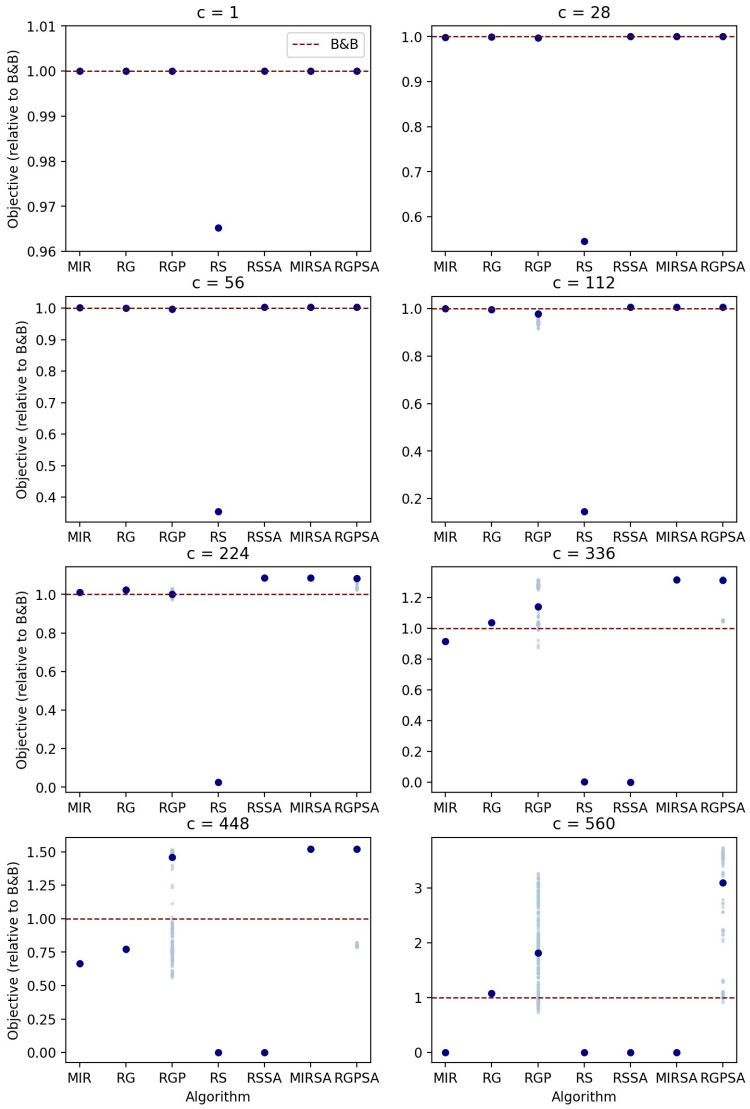

Fig. 1: Benchmarking of branch-and-bound-based (B&B) MIP solver, mixed-integer relaxation (MIR), randomised greedy (RG), randomised greedy with partial enumeration (RGP), random search (RS), simulated annealing initialised from a random solution (RSSA), mixed-integer relaxation combined with simulated annealing local search (MIRSA), randomised greedy with partial enumeration combined with simulated annealing local search (RGPSA) for $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$.

and RGPSA, which suggests that starting from a good solution is not particularly advantageous for this value of c. For $c = 336$, a starkly different picture emerges. In particular, SA alone achieves an objective of zero (which is as good as the random search), which rules out the possibility of it offering worst-case performance guarantees for the problem at hand. On the other hand, MIRSA and RGPSA find the best solutions, while RG outperforms B&B and displays little variance. By contrast, the median RGP solution also outperforms B&B, but its output range is quite broad. More precisely, the worst solutions returned by RGP are not as good as the solution produced by B&B but the best RGP solutions are comparable to the solutions of MIRSA and RGPSA. This suggests that randomising the partial enumeration procedure can help escape local optima in which RG gets stuck. For $c = 448$, MIRSA and RGPSA still provide the best solutions, while the median RGP solution is almost as good, achieving objectives that are almost twice as high as those returned by RG and MIR. Nevertheless, a great deal of variance can be seen in the output of RGP, which appears to have a bimodal (or multimodal) output distribution. Finally, for $c = 560$, only greedy algorithms achieve a nonzero objective. This suggests that neither MIR nor MIRSA can offer worst-case performance guarantees for the problem at hand (and for all values of c). RG outperforms B&B, while the median RGP and RGPSA solutions significantly outperform B&B. However, the output of RGP still suffers from a high level of variance, which can also be observed for RGPSA.

Overall, at least three heuristics match or outperform the MIP solver across all values of the coverage parameter c. Greedy algorithms perform particularly well in most cases and are the only algorithms that achieve nonzero objective values for all values of c. In particular, RGP is particularly effective but also suffers from high levels of variance in its output. Hence, this suggests that this algorithm should be run several times in order to obtain a good solution. MIRSA is one of the best algorithms in most cases, and its output displays virtually no variance, which is particularly appealing.

Figure 2 shows the local search trajectories of two of the best-performing algorithms, namely MIRSA (in red) and RGPSA (in blue). For MIRSA, the role played by the local search algorithm in improving solution quality increases as c increases. This observation is consistent with the intuition that the quality of solutions produced by solving the mixed-integer relaxation of problem (3) decreases as c increases. For RGPSA, on the other hand, the converse can be seen in Figure 2. The local search algorithm manages to improve solution quality on virtually all runs when c is smaller than or equal to 224. When c is greater than 224, however, the benefits of using the local search algorithm are less obvious. For most trajectories, it fails to significantly improve on the initial solution provided by the greedy algorithm. This observation also suggests that the greedy stage of RGPSA manages to find local optima from which it is difficult to escape.

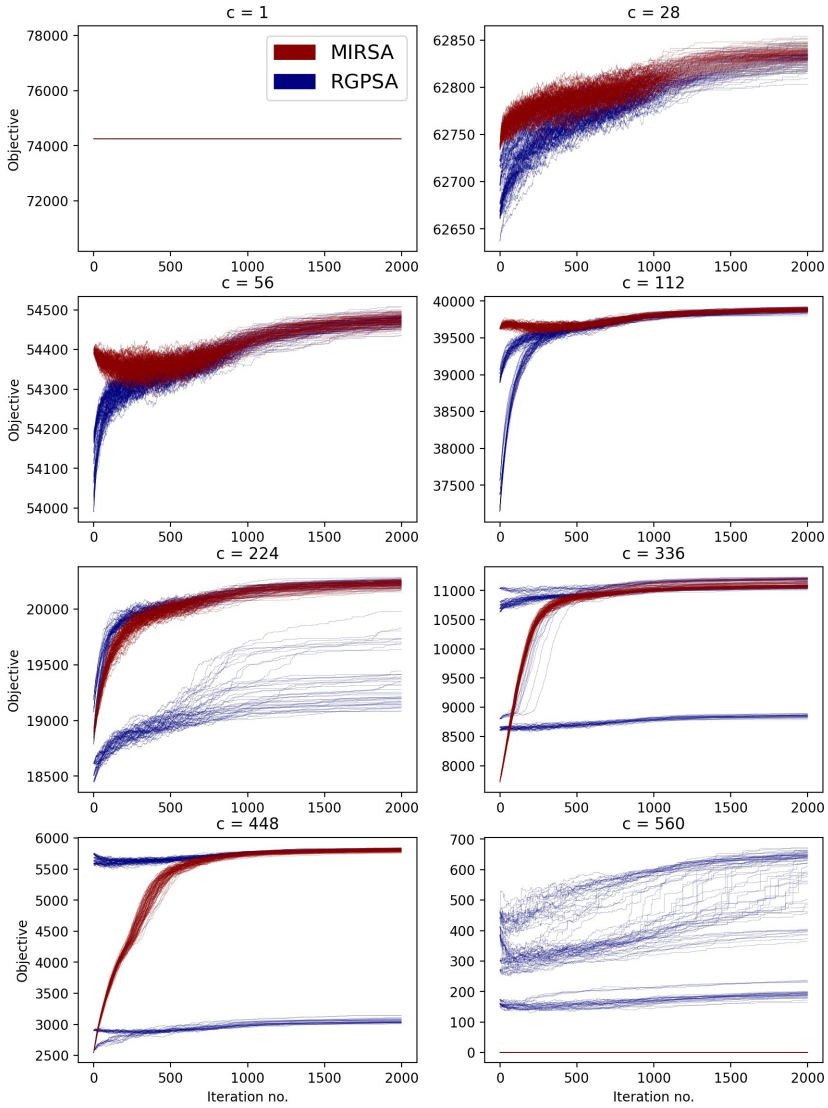

Fig. 2: Local search trajectories of MIRSA (red) and RGPSA (blue) for $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$.

4.3 Physical Interpretation

In this subsection, the physical nature of solutions provided by the model is discussed. To this end, the sets of locations identified by the MIRSA algorithm for $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$ are carefully analysed and compared with the set of locations that have the highest capacity factors (i.e., which

maximise the annual electricity output). The latter deployment strategy, which will be referred to as the PROD scheme, is fairly common in the wind power industry, as it typically allows wind developers and producers to maximise revenue.

Figure 3 shows the deployment patterns that both siting strategies yield, while Table 3 displays the number of time windows covered by the sets of locations identified by both schemes for each value of c.

c	1	28	56	112	224	336	448	560
MIRSA	74253	62846	54501	39916	20276	11228	5841	0
PROD	73521	59346	50608	37572	19882	9043	3149	0

Table 3: Number of time windows covered by the sets of locations identified by MIRSA and the PROD scheme, respectively.

It can be seen in Figure 3 that most PROD sites are located around the North Sea and Baltic Sea basins, in Ireland, the UK and on the southern coast of the English Channel. A small number of sites are also selected in clusters in the north of Finland and Norway. The average capacity factor of onshore wind power plants selected by the PROD scheme is 43.9%.

For $c = 1$, the onshore wind sites selected by MIRSA are scattered fairly uniformly across Europe. As discussed in Section 4.2, this instance is the only one for which a provably-optimal solution could be found, and an objective value of 74253 is achieved by the associated set of locations. This implies that roughly 85% of time windows can be covered by at least one location throughout the time horizon considered. Put differently, none of the 560 selected sites manages to supply more than 0.1% of the aggregate electricity demand 15% of the time. This result also suggests that the wind does not always blow hard somewhere in Europe, and it therefore seems unlikely that onshore wind power plants alone will be capable of supplying a constant share of the electricity demand at all times, even when strategically siting them at the level of the continent.

For $c \in \{28, 56, 112\}$, the deployment patterns selected by MIRSA are much less scattered than the one observed for $c = 1$. Instead, the model selects clusters of sites in different European regions, such as southern France, northern Spain or eastern Estonia. All of these areas are known to have very-high-quality wind resources [5], which is consistent with the average capacity factor values of 41.5%, 42.4% and 43.1% computed for these deployment patterns. In addition, the geographical distance between these clusters suggests that they experience distinct wind regimes influenced by local physical features that the model manages to identify and exploit. For example, the two clusters identified in southern France correspond to areas where the Mistral or Tramontane blow [62]. As can be seen from Table 3, the objective values achieved by the model steadily decrease as the value of c increases, which

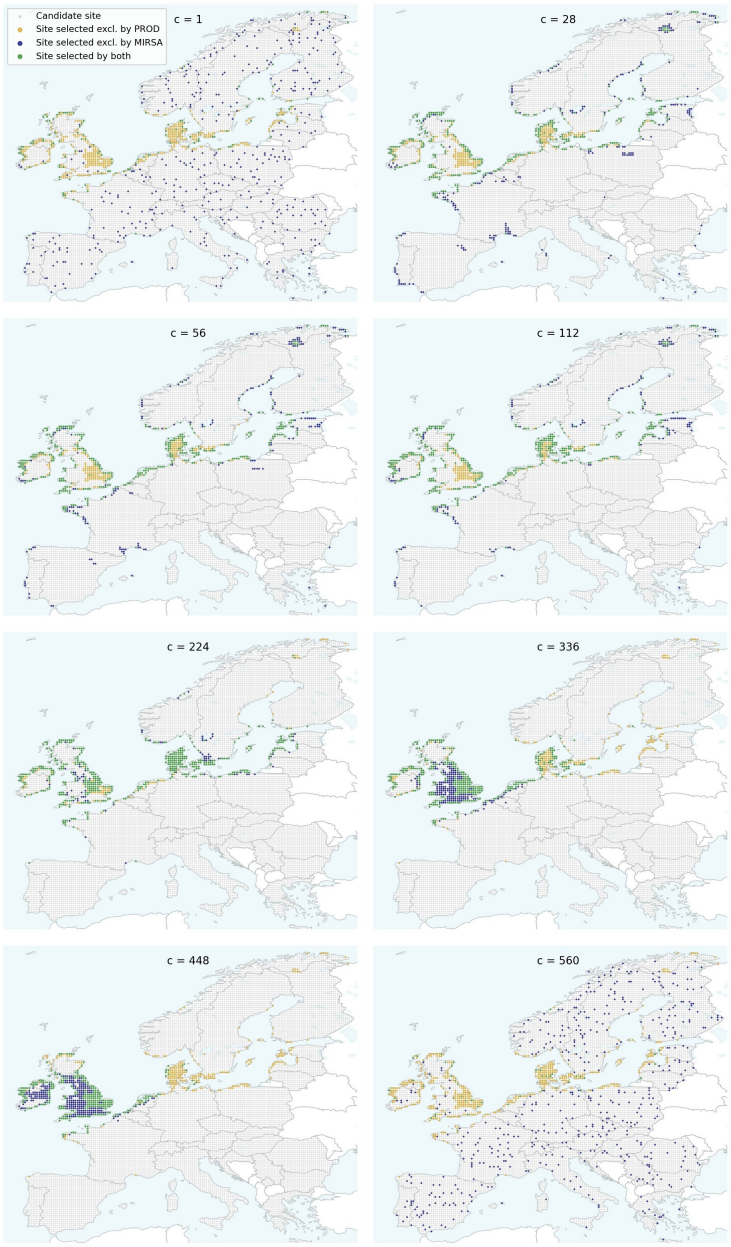

Fig. 3: Comparison between deployment patterns produced by MIRSA for $c \in \{1, 28, 56, 112, 224, 336, 448, 560\}$ (blue) and the deployment pattern maximising the annual electricity output (PROD, gold). Locations common to both schemes are shown in green.

implies that higher levels of joint electricity production can be sustained less frequently.

For $c = 224$, the clusters of sites identified for $c \in \{28, 56, 112\}$ are no longer visible, and most sites are located in a broad region including the North Sea and Baltic Sea basins. This set of locations also shares 480 sites with the PROD scheme, the most of any set of locations selected by MIRSA, and achieves an average capacity factor of 43.6%. The objective value of 39916 implies that the deployment pattern identified by MIRSA can supply at least 20% of the European electricity demand (roughly) 45.5% of the time.

For $c \in \{336, 448\}$, most sites are shifted towards the British Isles and Ireland, resulting in a large cluster of neighbouring locations. This somewhat counter-intuitive outcome can be explained as follows. Put simply, the diversity and quality of European wind regimes is not rich enough to sustain the joint production requirements imposed by these high values of c (which correspond to supplying roughly 30% and 40% of the aggregate electricity demand, respectively). Hence, the model identifies sets of locations that are very productive on average (with average capacity factors around 42.7% and 41.7%, respectively) and located close to each other for them to produce electricity at the same time.

For $c = 560$, MIRSA does not manage to identify a solution whose objective is nonzero and the resulting set of locations is therefore no better than a set of locations that would be selected uniformly at random, which explains the fairly uniform distribution of sites across Europe and the much lower average capacity factor (around 23.8%).

Finally, it is worth inspecting the residual demand time series associated with each deployment pattern. Figure 4 shows the cumulative distribution functions of residual demand times series obtained by subtracting the aggregate electricity production time series of each deployment pattern from a downscaled electricity demand time series. More precisely, each entry of the original aggregate electricity demand time series was multiplied by 0.2 in order to obtain a time series representing the share of the electricity demand that should ideally be supplied by wind power plants at all times. A positive value therefore implies a production shortage while a negative value corresponds to a production surplus. The fact that a high percentage of residual demand realisations shown in Figure 4 are positive for all deployment patterns confirms the earlier claim that onshore wind power plants will not be capable of supplying a constant share of the electricity demand at all times.

4.4 Cross-Validation

In this subsection, a cross-validation analysis is carried out in order to evaluate the extent to which the model and algorithms can successfully identify deployment patterns that perform well on previously unseen climatological data from historical data spanning a small number of weather years.

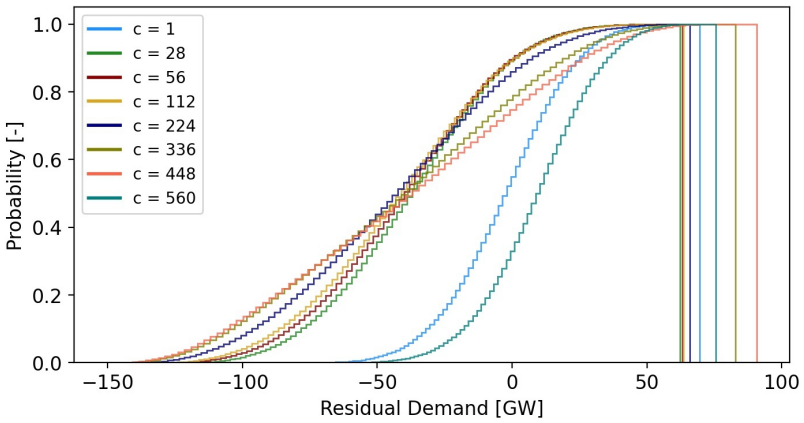

Fig. 4: Cumulative distribution functions of residual demand time series constructed by subtracting aggregate electricity production time series from a downscaled aggregate electricity demand time series.

In its simplest form, cross-validation works as follows. The data at hand is first split into training and testing sets (that are usually disjoint or have very little overlap). A model is then trained using data from the training set, and its performance is evaluated on the testing set. For example, in a supervised learning context where labelled data is readily available, the testing phase is rather straightforward and, roughly speaking, consists in measuring how far off the outcomes predicted by the model are from actual realisations in the testing data. The basic idea is that a good model should perform well on both training and testing sets (a situation where a model performs extremely well on training data and poorly on testing data is usually called *overfitting*, which is particularly undesirable).

In this case, the training and testing sets can be viewed as sets of weather years for which renewable resource and electricity load data are available. Since ten years of meteorological data are available, the training and testing sets could for instance contain five years each, with no year in common. In the training phase, problem (2) would be solved using one of the algorithms discussed in Section 3.5 in order to identify a set of promising locations and their objective would be computed using data from the testing set. However, no labelled data is available for this problem and a set of locations that is optimal for the testing data is not known *a priori*. Hence, in order to evaluate how good the locations identified using the training data are, one of the algorithms of Section 3.5 (preferably the same as the one used in the training stage) must be used to solve the problem for the testing data as well. Then, the difference between the testing objectives computed for locations identified in the training and testing stages can be used to measure the performance of the model. This procedure is essentially equivalent to training a second model on the testing set and comparing the performance of both models on the

testing set. Let $f^\star_{\text{train}}$ denote the objective value achieved on the testing set by locations identified using data from the training set, and let $f^\star_{\text{test}}$ denote the objective value achieved on the testing set by locations identified using data from the testing set. The performance of the model is evaluated using the relative error $e \in \mathbb{R}$,

$$e = \frac{f^\star_{\text{test}} - f^\star_{\text{train}}}{f^\star_{\text{test}}}. \tag{11}$$

Note that a positive relative error indicates that the model trained on the testing set outperforms the model trained on the training set when evaluating their performance on the testing set, while the opposite holds true for a negative relative error. Ideally, the model and algorithms would consistently achieve values of the relative error that are close to zero.

The values of the coverage parameter considered here are the same as the ones studied in Sections 4.2 and 4.3, and the analysis is divided into two parts. In the first part, *balanced* training and testing sets including five weather years each are used. In the second part, *unbalanced* training and testing sets that include seven and three weather years, respectively, are employed. Reducing the size of the testing set increases the risk of overfitting for locations identified using the testing data, and provides clues as to the robustness of the proposed model. Then, in both set-ups, one cross-validation experiment consists in sampling a pre-specified number of weather years without repetition from the pool of ten years for which data is available in order to construct the training set, while the remaining years are used to build the testing set.

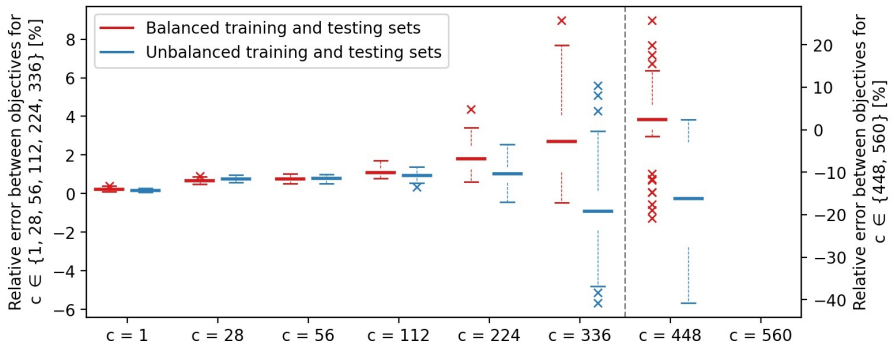

Fig. 5: Results of cross-validation analysis carried out with MIRSA. In the experiment with balanced sets (red), both training and testing sets include five years of data. In the unbalanced set-up (blue), the training and testing sets include seven and three weather years, respectively. Fifty independent experiments were carried out for each value of c and set-up. No box plot is shown for $c = 560$ since MIRSA fails to identify any solution whose objective value is nonzero.

Figure 5 shows the outcomes of the cross-validation experiments performed with MIRSA. In each experiment, MIRSA was run once for training and once for testing. Fifty independent experiments were carried out for each set-up. The relative error range clearly increases as c increases in both the balanced and unbalanced set-ups. In the balanced set-up, all realisations of the relative error have positive values smaller than 5%, while the median relative error is positive and under 2%, as long as c is smaller than or equal to 224. This indicates that little to no overfitting takes place for these values of c. In addition, the fact that the relative error is positive implies that the locations selected based on the testing set slightly outperform the ones identified using the training data. For values of c greater than 224, although the median relative error remains positive, some realisations of the relative error have negative values, which implies that the locations identified using the training data outperform the ones selected based on the testing data. This counter-intuitive observation can be partly explained by the fact that the algorithms used to identify deployment patterns are approximate in nature and can therefore produce solutions that are heavily suboptimal. In the unbalanced set-up, the outcomes of the cross-validation experiments are broadly comparable to the ones observed in the balanced set-up for values of c smaller than or equal to 224, although a few realisations of the relative error already have negative values for $c = 224$. For values of c greater than 224, however, most realisations of the relative error are consistently negative.

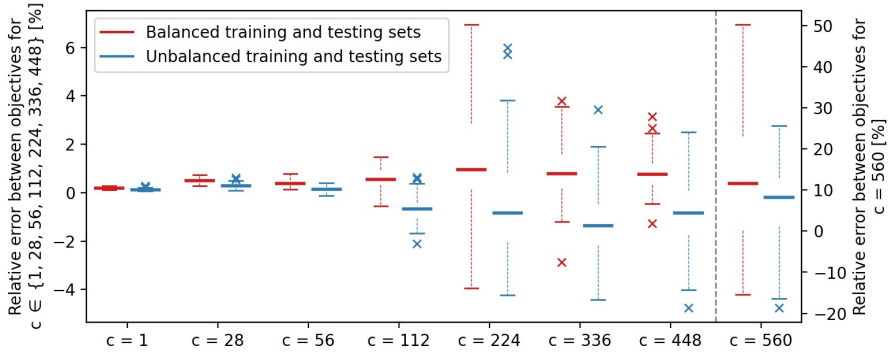

Fig. 6: Results of cross-validation analysis performed with a variant of RGP. In the experiment with balanced sets (red), both training and testing sets include five years of meteorological data. In the unbalanced set-up (blue), the training and testing sets include seven and three weather years, respectively. Fifty independent experiments were carried out for each value of c.

Figure 6 shows the outcomes of the cross-validation experiments carried out with a variant of RGP. More precisely, in each experiment, RGP was run twenty five times and the solution with the highest objective was retained. This

procedure was performed for both training and testing, and fifty independent experiments were carried out for each set-up. In the balanced set-up, except for $c = 560$, the median relative error remains positive and close to 1%. In addition, the relative error range remains between -2% and 4% for values of $c \notin \{224, 560\}$. For $c = 224$, the relative error range is much broader, and it blows up for $c = 560$. In the unbalanced set-up, the median relative error stays positive for $c \in \{1, 28, 56\}$ and becomes negative for $c \in \{112, 224, 336, 448\}$ but stays between -1% and 1%. The relative error range is also broader for $c \in \{224, 336, 448\}$, although most realisations are between -4% and 4%. In both set-ups, the median relative error is around 10% for $c = 560$ and the relative error range covers several dozens of percentage points.

Overall, these results suggest that at least one algorithm can successfully and reliably identify patterns in the climatological data and select locations that perform well on previously unseen data using both balanced and unbalanced training and testing sets containing a few different weather years, as long as the value of the coverage parameter is smaller than 560.

5 Conclusion

The problem of siting renewable power generation assets while accounting for their spatiotemporal complementarity is studied in this paper. The problem is cast as a combinatorial optimisation problem selecting a pre-specified number of sites so as to minimise the number of simultaneous low electricity production events that they experience relative to a pre-specified reference production level. It is shown that the resulting model is closely related to submodular optimisation and generalises the well-known maximum coverage problem.

Realistic problem instances inspired by the problem of siting onshore wind power plants in Europe are constructed and used to benchmark randomised greedy, local search and relaxation-based algorithms against a state-of-the-art MIP solver for different values of the so-called coverage parameter. At least three heuristics are found to consistently provide solutions that are at least as good or strictly better than the ones returned by the MIP solver at a fraction of the computational cost across all values of the coverage parameter. In particular, randomised greedy algorithms perform particularly well and are the only algorithms that achieve nonzero objective values across all instances considered.

The physical nature of solutions provided by the model is also analysed and the deployment patterns are shown to change substantially based on the implicit joint electricity production requirement imposed by the coverage parameter. Results also suggest that the wind does not always blow hard somewhere in Europe, and it therefore seems unlikely that onshore wind power plants will be capable of supplying a constant share of the electricity demand at all times, even when strategically siting them on a continental scale.

Finally, the results of the cross-validation analysis indicate that the model can successfully leverage historical data spanning a small number of weather

years in order to reliably identify deployment patterns that perform well on previously unseen climatological data, except for the edge case where the coverage parameter is set equal to the number of locations to be selected.

Several research directions would be worth pursuing in future work. From an algorithmic perspective, analysing the greedy algorithms discussed in Sections 3.5.3 and 3.5.4 more closely for $c > 1$ would be useful to understand whether they can indeed provide worst-case performance guarantees. Going beyond traditional worst-case analyses (e.g., by performing a parameterised performance guarantee analysis [64]) would also bring additional insight into the performance of the proposed algorithms. From a modelling perspective, if the cost of siting renewable power generation assets was known for all locations, a budget constraint could replace the cardinality constraint in order to introduce basic economic considerations into siting decisions. The algorithmic implications of this update should be carefully considered, however, as such problems are known to be particularly challenging [13]. Finally, using the present model as a pre-processing step to select locations used in integrated power system expansion planning assessments [45] that have much lower spatial and temporal resolutions and include network models would provide insight into the impact that renewable asset siting decisions and resource complementarity may have on power system design and economics.

Acknowledgements Mathias Berger and David Radu would like to gratefully acknowledge the support of the Belgian government through its Energy Transition Fund and the REMI project. Antoine Dubois would like to acknowledge the support of a FRIA fellowship of the FRS-FNRS. The authors would like to thank Raphael Fonteneau for valuable discussions and the anonymous reviewers whose comments helped improve the clarity and quality of this manuscript.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Achterberg, T.: SCIP: Solving Constraint Integer Programs. Mathematical Programming Computation **1**(1), 1–41 (2009). DOI https://doi.org/10.1007/s12532-008-0001-1
2. Ageev, A.A., Sviridenko, M.I.: Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee. Journal of Combinatorial Optimization **6**, 307–308 (2004). DOI https://doi.org/10.1023/B:JOCO.0000038913.96607.c2
3. Apt, J.: The Spectrum of Power from Wind Turbines. Journal of Power Sources **169**(2), 369 – 374 (2007). DOI https://doi.org/10.1016/j.jpowsour.2007.02.077
4. Badanidiyuru, A., Vondrák, J.: Fast Algorithms for Maximizing Submodular Functions. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14, pp. 1497–1514. SIAM (2014). DOI https://doi.org/10.5555/2634074.2634184
5. Badger, J., Bauwens, I., Casso, P., Davis, N., Hahmann, A., Bo Krohn Hansen, S., Ohrbeck Hansen, B., Heathfield, D., Knight, J.O., Lacave, O., Lizcano, G., Bosch i Mas, A., Gylling Mortensen, N., Olsen, B.T., Onninen, M., Potter van Loon, A., Volker, P.: Global Wind Atlas 3.0 (2021). URL `https://globalwindatlas.info/`

6. Baringo, L., Conejo, A.J.: Strategic Wind Power Investment. IEEE Transactions on Power Systems **29**(3), 1250–1260 (2014). DOI https://doi.org/10.1109/TPWRS.2013.2292859
7. Barman, S., Fawzi, O., Ghoshal, S., Gurpinar, E.: Tight Approximation Bounds for Maximum Multi-Coverage. Mathematical Programming **21** (2021). DOI https://doi.org/10.1007/s10107-021-01677-4
8. Berger, M., D.Radu, Fonteneau, R., Henry, R., Glavic, M., Fettweis, X., Du, M.L., Panciatici, P., Balea, L., Ernst, D.: Critical Time Windows for Renewable Resource Complementarity Assessment. Energy **198** (2020). DOI https://doi.org/10.1016/j.energy.2020.117308
9. Berger, M., Radu, D.: Algorithms for Max k-Multicover (2021). URL `https://gitlab.uliege.be/smart_grids/public/maxk_multicover`
10. Bertsimas, D., Tsitsiklis, J.: Simulated Annealing. Statist. Sci. **8**(1), 10–15 (1993). DOI https://doi.org/10.1214/ss/1177011077
11. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A Fresh Approach to Numerical Computing. SIAM review **59**(1), 65–98 (2017). DOI https://doi.org/10.1137/141000671
12. Calinescu, G., Chekuri, C., Pál, M., Vondrák, J.: Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. SIAM Journal on Computing **40**(6), 1740–1766 (2011). DOI https://doi.org/10.1137/080733991
13. Chekuri, C., Kumar, A.: Maximum Coverage Problem with Group Budget Constraints and Applications. In: K. Jansen, S. Khanna, J.D.P. Rolim, D. Ron (eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pp. 72–83. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). DOI https://doi.org/10.1007/s10878-016-0102-0
14. Chekuri, C., Vondrák, J., Zenklusen, R.: Submodular Function Maximization via the Multilinear Relaxation and Contention Resolution Schemes. SIAM Journal on Computing **43**(6), 1831–1879 (2014). DOI https://doi.org/10.1137/110839655
15. Chlamtác, E., Dinitz, M., Konrad, C., Kortsarz, G., Rabanca, G.: The Densest k-Subhypergraph Problem. SIAM Journal on Discrete Mathematics **32**(2), 1458–1477 (2018). DOI https://doi.org/10.1137/16M1096402
16. Chvatal, V.: A Greedy Heuristic for the Set-Covering Problem. Mathematics of Operations Research **4**(3), 233–235 (1979). DOI https://doi.org/10.1287/moor.4.3.233
17. Cornuejols, G., Fisher, M.L., Nemhauser, G.L.: Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. Management Science **23**(8), 789–810 (1977). DOI https://doi.org/10.1287/mnsc.23.8.789
18. CPLEX, I.: IBM CPLEX Optimizer Reference Manual (2021). URL `https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-users-manual`
19. Dobson, G.: Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data. Mathematics of Operations Research **7**(4), 515–531 (1982). DOI https://doi.org/10.1287/moor.7.4.515
20. Dunning, I., Huchette, J., Lubin, M.: JuMP: A Modeling Language for Mathematical Optimization. SIAM Review **59**(2), 295–320 (2017). DOI https://doi.org/10.1137/15M1020575
21. Dvorkin, Y., Fernandez-Blanco, R., Wang, Y., Xu, B., Kirschen, D.S., Pandzic, H., Watson, J.P., Silva-Monroy, C.A.: Co-Planning of Investments in Transmission and Merchant Energy Storage. IEEE Transactions on Power Systems **33**(1), 245–256 (2018). DOI https://doi.org/10.1109/TPWRS.2017.2705187
22. Engeland, K., Borga, M., Creutin, J.D., François, B., Ramos, M.H., Vidal, J.P.: Space-Time Variability of Climate Variables and Intermittent Renewable Electricity Production - A Review. Renewable and Sustainable Energy Reviews **79**, 600 – 617 (2017). DOI https://doi.org/10.1016/j.rser.2017.05.046
23. ENTSO-E: Power Statistics (2021). URL `https://www.entsoe.eu/data/power-stats`
24. European Centre for Medium-Range Weather Forecasts (ECMWF): ERA5 Hourly Data on Single Levels from 1979 to Present (2021). URL `https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era5`
25. Feige, U.: A Threshold of ln n for Approximating Set Cover. J. ACM **45**(4), 634–652 (1998). DOI https://doi.org/10.1145/285055.285059

26. Filmus, Y., Ward, J.: The Power of Local Search: Maximum Coverage over a Matroid. In: T.W. Christoph Dürr (ed.) STACS'12 (29th Symposium on Theoretical Aspects of Computer Science), vol. 14, pp. 601–612. LIPIcs (2012). DOI https://doi.org/10.4230/LIPIcs.STACS.2012.i
27. Filmus, Y., Ward, J.: Monotone Submodular Maximization over a Matroid via Non-Oblivious Local Search. SIAM Journal on Computing **43**(2), 514–542 (2014). DOI https://doi.org/10.1137/130920277
28. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed Hypergraphs and Applications. Discrete applied mathematics **42**(2-3), 177–201 (1993). DOI https://doi.org/10.1016/0166-218X(93)90045-P
29. Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Hojny, C., Koch, T., Le Bodic, P., Maher, S.J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., Pfetsch, M.E., Schlösser, F., Serrano, F., Shinano, Y., Tawfik, C., Vigerske, S., Wegscheider, F., Weninger, D., Witzig, J.: The SCIP Optimization Suite 7.0. ZIB-Report 20-10, Zuse Institute Berlin (2020)
30. Gamrath, G., Gleixner, A., Koch, T., Miltenberger, M., Kniasew, D., Schloegel, D., Martin, A., Weninger, D.: Tackling Industrial-Scale Supply Chain Problems by Mixed-Integer Programming. Journal of Computational Mathematics **37**(6), 866–888 (2019). DOI https://doi.org/10.4208/jcm.1905-m2019-0055
31. Gelaro, R., McCarty, W., Suárez, M.J., Todling, R., Molod, A., Takacs, L., Randles, C.A., Darmenov, A., Bosilovich, M.G., Reichle, R., et al.: The Modern-Era Retrospective Analysis for Research and Applications, Version 2 (MERRA-2). Journal of Climate **30**(14), 5419–5454 (2017). DOI https://doi.org/10.1175/JCLI-D-16-0758.1
32. Geth, F., Brijs, T., Kathan, J., Driesen, J., Belmans, R.: An Overview of Large-Scale Stationary Electricity Storage Plants in Europe: Current Status and New Developments. Renewable and Sustainable Energy Reviews **52**, 1212–1227 (2015). DOI https://doi.org/10.1016/j.rser.2015.07.145
33. Giebel, G.: On the Benefits of Distributed Generation of Wind Energy in Europe. PhD Thesis, University of Oldenburg, Germany (2001)
34. Grossman, T., Wool, A.: Computational Experience with Approximation Algorithms for the Set Covering Problem. European Journal of Operational Research **101**(1), 81–92 (1997). DOI https://doi.org/10.1016/S0377-2217(96)00161-0
35. Gurobi: Gurobi Optimizer Reference Manual (2021). URL `http://www.gurobi.com`
36. Haas, S., Schachler, B., Krien, U.: Windpowerlib - A Python Library to Model Wind Power Plants (v0.2.0) (2019). URL `https://windpowerlib.readthedocs.io/en/stable/index.html`
37. Hobbs, B.F., Xu, Q., Ho, J., Donohoo, P., Kasina, S., Ouyang, J., Park, S.W., Eto, J., Satyal, V.: Adaptive Transmission Planning: Implementing a New Paradigm for Managing Economic Risks in Grid Expansion. IEEE Power and Energy Magazine **14**(4), 30–40 (2016). DOI https://doi.org/10.1109/MPE.2016.2547280
38. Hochbaum, D.S., Pathria, A.: Analysis of the Greedy Approach in Problems of Maximum k-Coverage. Naval Research Logistics **45**(6), 615–627 (1998). DOI https://doi.org/10.1002/(SICI)1520-6750(199809)45:6⟨615::AID-NAV5⟩3.0.CO;2-5
39. International Electrotechnical Commission: IEC 61400-1:2019: Wind Energy Generation Systems - Part 1: Design Requirements (2019). URL `https://webstore.iec.ch/publication/26423`
40. Johnson, D.S.: Approximation Algorithms for Combinatorial Problems. Journal of Computer and System Sciences **9**(3), 256–278 (1974). DOI https://doi.org/10.1016/S0022-0000(74)80044-9
41. Jurasz, J., Canales, F., Kies, A., Guezgouz, M., Beluco, A.: A Review on the Complementarity of Renewable Energy Sources: Concept, Metrics, Application and Future Research Directions. Solar Energy **195**, 703–724 (2020). DOI https://doi.org/10.1016/j.solener.2019.11.087
42. Khot, S.: On the Power of Unique 2-Prover 1-Round Games. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, STOC '02, p. 767–775. Association for Computing Machinery, New York, NY, USA (2002). DOI https://doi.org/10.1145/509907.510017

43. Klima, K., Apt, J.: Geographic Smoothing of Solar PV: Results from Gujarat. Environmental Research Letters **10**(10), 104001 (2015). DOI https://doi.org/10.1088/1748-9326/10/10/104001
44. Krause, A., Singh, A., Guestrin, C.: Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. Journal of Machine Learning Research **9**(Feb), 235–284 (2008). URL `https://jmlr.org/papers/v9/krause08a.html`
45. Krishnan, V., Ho, J., Hobbs, B.F., Liu, A.L., McCalley, J.D., Shahidehpour, M., Zheng, Q.P.: Co-optimization of Electricity Transmission and Generation Resources for Planning and Policy Analysis: Review of Concepts and Modeling Approaches. Energy Systems **7**(2), 297–332 (2016). DOI https://doi.org/10.1007/s12667-015-0158-4
46. Li, W., Stadler, S., Ramakumar, R.: Modeling and Assessment of Wind and Insolation Resources with a Focus on Their Complementary Nature: A Case Study of Oklahoma. Annals of the Association of American Geographers **101**(4), 717–729 (2011). DOI https://doi.org/10.1080/00045608.2011.567926
47. Martin, C.M.S., Lundquist, J.K., Handschy, M.A.: Variability of Interconnected Wind Plants: Correlation Length and its Dependence on Variability Time Scale. Environmental Research Letters **10**(4) (2015). DOI https://doi.org/10.1088/1748-9326/10/4/044004
48. Milligan, M.R., Artig, R.: Choosing Wind Power Plant Locations and Sizes Based on Electric Reliability Measures Using Multiple-Year Wind Speed Measurements. Tech. Rep. CP-500-26724, NREL (1999). URL `https://www.osti.gov/biblio/750939`
49. Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrák, J., Krause, A.: Lazier than Lazy Greedy. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, pp. 1812–1818. AAAI Press (2015). DOI https://doi.org/10.5555/2886521.2886572
50. Munoz, F.D., Hobbs, B.F., Ho, J.L., Kasina, S.: An Engineering-Economic Approach to Transmission Planning Under Market and Regulatory Uncertainties: WECC Case Study. IEEE Transactions on Power Systems **29**(1), 307–317 (2014). DOI https://doi.org/10.1109/TPWRS.2013.2279654
51. Musselman, A., Thomas, V.M., Boland, N., Nazzal, D.: Optimizing Wind Farm Siting to Reduce Power System Impacts of Wind Variability. Wind Energy **22**(7), 894–907 (2019). DOI https://doi.org/10.1002/we.2328
52. Nemhauser, G.L., Wolsey, L.A.: Best Algorithms for Approximating the Maximum of a Submodular Set Function. Mathematics of Operations Research **3**(3), 177–188 (1978). DOI https://doi.org/10.1287/moor.3.3.177
53. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An Analysis of Approximations for Maximizing Submodular Set Functions-I. Mathematical Programming **14**, 265–294 (1978). DOI https://doi.org/10.1007/BF01588971
54. Norgård, P., Holttinen, H.: A Multi-Turbine Power Curve. In: NWPC '04: Nordic Wind Power Conference, no. 32R in Chalmers Tekniska Högskola, Institutionen för Elkraftteknik. Examensarbete (2004)
55. O'Connell, N., Pinson, P., Madsen, H., O'Malley, M.: Benefits and Challenges of Electrical Demand Response: A Critical Review. Renewable and Sustainable Energy Reviews **39**, 686–699 (2014). DOI https://doi.org/10.1016/j.rser.2014.07.098
56. Olauson, J., Bergkvist, M.: Correlation between Wind Power Generation in the European Countries. Energy **114**, 663 – 670 (2016). DOI https://doi.org/10.1016/j.energy.2016.08.036
57. Peleg, D., Schechtman, G., Wool, A.: Approximating Bounded 0-1 Integer Linear Programs. In: The 2nd Israel Symposium on Theory and Computing Systems, pp. 69–70. IEEE (1993). DOI https://doi.org/10.1109/ISTCS.1993.253482
58. Peleg, D., Schechtman, G., Wool, A.: Randomized Approximation of Bounded Multicovering Problems. Algorithmica **18**(1), 44–66 (1997). DOI https://doi.org/10.1007/BF02523687
59. Prasad, A.A., Taylor, R.A., Kay, M.: Assessment of Solar and Wind Resource Synergy in Australia. Applied Energy **190**, 354 – 367 (2017). DOI https://doi.org/10.1016/j.apenergy.2016.12.135
60. Radu, D., Berger, M.: Data for "Siting Renewable Power Generation Assets using Combinatorial Optimisation" (2021). URL `https://dox.uliege.be/index.php/s/jNoMUGUHi1QhsAf`

61. Radu, D., Berger, M.: resite_ip: A Framework for RES Siting Leveraging Resource Complementarity (2021). URL https://gitlab.uliege.be/smart_grids/public/resite_ip/tree/feature-ol
62. Radu, D., Berger, M., Fonteneau, R., Hardy, S., Fettweis, X., Du, M.L., Panciatici, P., Balea, L., Ernst, D.: Complementarity Assessment of South Greenland Katabatic Flows and West Europe Wind Regimes. Energy **175**, 393 – 401 (2019). DOI https://doi.org/10.1016/j.energy.2019.03.048
63. Resende, M.G.: Computing Approximate Solutions of the Maximum Covering Problem with GRASP. Journal of Heuristics **4**(2), 161–177 (1998). DOI https://doi.org/10.1023/A:1009677613792
64. Roughgarden, T. (ed.): Beyond the Worst-Case Analysis of Algorithms. Cambridge University Press, Cambridge (2021). DOI https://doi.org/10.1017/9781108637435
65. Stenclik, D., Denholm, P., Chalamala, B.: Maintaining Balance: The Increasing Role of Energy Storage for Renewable Integration. IEEE Power and Energy Magazine **15**(6), 31–39 (2017). DOI https://doi.org/10.1109/MPE.2017.2729098
66. Sterl, S., Liersch, S., Koch, H., van Lipzig, N.P.M., Thiery, W.: A New Approach for Assessing Synergies of Solar and Wind Power: Implications for West Africa. Environmental Research Letters **13**(9) (2018). DOI https://doi.org/10.1088/1748-9326/aad8f6
67. WindEurope: Wind Energy and On-Site Energy Storage - Exploring Market Opportunities (2017). URL https://windeurope.org/wp-content/uploads/files/about-wind/reports/Wind-energy-in-Europe-Scenarios-for-2030.pdf
68. WindEurope: Wind Energy in Europe in 2019 (2020). URL https://windeurope.org/wp-content/uploads/files/about-wind/statistics/WindEurope-Annual-Statistics-2019.pdf
69. Wu, G., Deshmukh, R., Ndhlukulac, K., Radojicic, T., Reilly-Moman, J., Phadke, A., Kammen, D., Callaway, D.: Strategic Siting and Regional Grid Interconnections Key to Low-Carbon Futures in African Countries. Proceedings of the National Academy of Sciences **114**, 3004 – 3012 (2017). DOI https://doi.org/10.1073/pnas.1611845114
70. Zappa, W., van den Broek, M.: Analysing the Potential of Integrating Wind and Solar Power in Europe using Spatial Optimisation under Various Scenarios. Renewable and Sustainable Energy Reviews **94**, 1192 – 1216 (2018). DOI https://doi.org/10.1016/j.rser.2018.05.071
71. Zhang, H., Cao, Y., Zhang, Y., Terzija, V.: Quantitative Synergy Assessment of Regional Wind-Solar Energy Resources based on MERRA Reanalysis Data. Applied Energy **216**, 172 – 182 (2018). DOI https://doi.org/10.1016/j.apenergy.2018.02.094