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

Subset selection via continuous optimization with applications to network design

Radislav Vaisman

Received: date / Accepted: date

Abstract Choosing a subset of representative items from a set of alternatives is an important problem in many scientific fields such as environmental science and statistics. For most practical problems, however, a computationally efficient solution method is not known to exist. While this problem has attracted a significant amount of attention, the majority of specifically designed algorithms do not scale well with respect to the problem size or do not provide a usable open-source package. In this study, we show that any global continuous optimization technique can be used for solving the representative subset selection problem. The latter is achieved by designing a simple transformation which embeds the problem's discrete space into a larger continuous space. The proposed methodology is applied to design problems in environmental and statistical domains. We evaluate the proposed method using several open-source global optimization packages, and show that this technique compares favorably with existing direct methods.

Keywords Ozone · Monitoring network design · D-optimal experimental design · global optimization · space embedding

1 Introduction

Networks are pervasive in modern society. Environmental and distribution networks (waste-water, power grids, aviation, World Wide Web), Social networks (Facebook, YouTube, Twitter), and biological networks (neural, metabolic, protein-protein interaction) all play a key role in the functioning of our world. Monitoring such networks is now an important part of scientific endeavor. For example, an improper monitoring of waste-water or carbon dioxide emission can result in a serious damage to the ecosystem (Stokes and Horvath, 2010; Mi et al., 2017; Park et al., 2013; Chan and Yao, 2008; Ramanathan and Carmichael, 2008; Bruns et al., 1991;

Radislav Vaisman
School of Mathematics and Physics, The University of Queensland, Brisbane 4072, Australia
Tel.: +61 7 336 53264
E-mail: r.vaisman@uq.edu.au
ORCID Identifier: 0000-0001-9875-0616

Wiersma, 1984; von Brömssen et al., 2018). Likewise, a diffusion of false information from social media ("fake news"), can potentially have a devastating effect on national security and political stability (Allcott and Gentzkow, 2017; Silverman and Singer-Vine, 2016). In this study, we address the *monitoring network design* problem (Le and Zidek, 2006) by introducing a general technique that allows one to use *any* global continuous optimization method for solving the problem. The proposed method provides high-quality solutions to the monitoring network design problem with minimal development effort.

Regardless of the application domain, it is convenient to define a network as a graph with m vertices. These vertices can represent, for example, possible locations for a placement of wastewater monitoring sensor, web-servers, influential bloggers, or electrical re-transmission blocks. Due to the high operational cost associated with the size of modern networks, monitoring the entire system is generally unfeasible. Consequently, a natural approach is to analyze some representative part of the network in order to infer the entire network state. In this way, the optimal monitoring network design task can be viewed as an *optimal fixed-size subset selection* (OFSS) problem in which a subset of size $k \ll m$ vertices (from m initial vertices) has to be chosen such that a certain *utility function* is maximized. Under the monitoring network design setting, the utility function will generally measure the information gain obtained from a decision of choosing a particular subset of vertices.

Because of the importance of the OFSS problem from both a theoretical and practical point of view, it has attracted a significant amount of research attention (Wolters, 2015; Chao et al., 2015; Yu and Yuan, 1992). However, excluding some trivial cases, this problem belongs to the NP-hard complexity class (Natarajan, 1995; Geoffrey et al., 1997; Chun-Wa et al., 1995), necessitating approximate solution methods. Due to the hardness result, no approximation algorithm can guarantee the discovery of the optimal solution to the OFSS problem. From the practical point of view, however, it is beneficial to implement a number of different approximation methods. In this way, the designer can then solve the problem with several algorithms, and adopt the best solution found.

Despite the problem importance, there exists a deficit of freely available and accessible software that is specifically designed to handle the OFSS problem (Ramiro et al., 2010; Wolters, 2015). In fact, to the best of our knowledge, the only open-source package available is `kofnGA` (Wolters, 2015). This study aims to address the above gap by introducing a simple transformation, which embeds the (discrete) solution space of the OFSS problem into a larger continuous space, allowing the practitioner to use any freely available or proprietary continuous global optimization software to obtain a solution to the OFSS problem. Since the majority of global optimization packages (Mullen, 2014) use different heuristics, the proposed technique introduces a significant practical advantage, in the sense that one can take advantage of diverse algorithms with minimal development effort.

The rest of the study is organized as follows. In Section 2 we formulate the OFSS problem and show that both the optimal network monitoring design and the D-optimal experimental design problems fit into the OFSS framework. Then, we give a brief overview of the modern continuous global optimization research, and introduce a simple transformation from the continuous to the original discrete state space. In Section 3 we perform a detailed experimental study using several popular open-source global optimization packages. Finally, in Section 4 we summarize our findings and discuss possible directions for future research.

2 Methodology

2.1 The OFSS problem

The formal definition of the OFSS problem is as follows. Let $\mathcal{L} = \{1, \ldots, m\}$ be the set of indices that corresponds, for example, to m available locations for a placement of wastewater monitoring sensor. Since we wish to select $k \leq m$ indices from $\mathcal{L}$, the set of all possible solutions of the k-sized OFSS problem is defined via

$$\mathscr{X} = \Big\{ \mathbf{x} \in \mathscr{P}\left(\{1, \ldots, m\}\right), \ |\mathbf{x}| = k \Big\}, \tag{1}$$

where $\mathscr{P}$ stands for the power-set, and $|\mathbf{x}|$ denotes the cardinality of $\mathbf{x}$. Recall that the selection should be performed subject to a maximization of certain utility criterion. Therefore, we define a *utility function* $U : \mathscr{X} \to \mathbb{R}$, which we seek to maximize. Finally, the OFSS problem can be formulated via:

$$\max_{\mathbf{x} \in \mathscr{X}} \ U(\mathbf{x}), \tag{2}$$

while the set of optimal indices is available from the solution of $\mathrm{argmax}_{\mathbf{x} \in \mathscr{X}} \ U(\mathbf{x})$. In this study, we consider two real-life applications that fall into the OFSS problem setting; the optimal monitoring network design, and the D-optimal experimental design.

Entropy-based environmental monitoring network design

Environmental monitoring networks design involves the selection of k representative locations out of $m \geq k$ possible locations in a region of interest. For example, Fig. 1 from (Wolters, 2015) shows $m = 100$ potential locations (left panel), and $k = 9$ actual locations (right panel) of ozone monitoring stations in the state of New York (Le and Zidek, 2006).

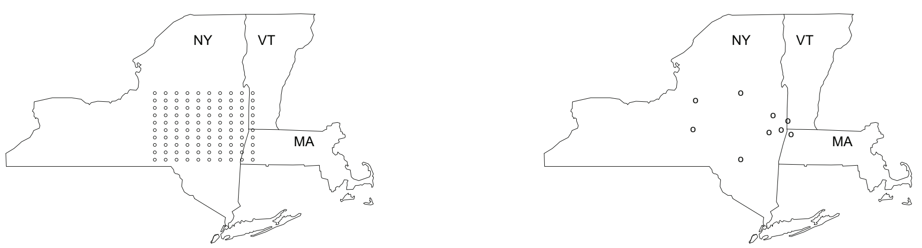


Fig. 1: Potential (left panel), and existing (right panel), ozone monitoring sites in the state of New York.

Since the cost of auditing of all possible locations can be prohibitively high, we generally wish to select $k \ll m$ representative sites for the placement of monitoring stations. The selection is performed with a view to maximize some utility function such as an obtained information gain from the selection of a particular set of k monitoring sites. Formally, the ozone concentration data that is collected from a certain site can be modeled via a stochastic time series. In addition, it is convenient to model the data collected from all sites as a random field $\left\{Z(l) : l \in D \subset \mathbb{R}^d\right\}$, where D generally represents a discretized subset of $\mathbb{R}^d$. Our task is to select k locations from a finite set D. The monitoring network design problem is usually associated with the two-dimensional or the three-dimensional space, that is, we generally take $d = 2$ or $d = 3$.

To put the monitoring networks design problem into the OFSS setting, we need to specify the utility function. Note that the m-dimensional vector of observations $\mathbf{Z}$, which is obtainable from the entire set of m sites, can be partitioned into two vectors $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$, where $\mathbf{Z}^{(1)}$ represents $m-k$ observations from the set of *unmeasured* sites and $\mathbf{Z}^{(2)}$ represents the remaining k observations from the set of *measured* sites. Let $\mathcal{D}(\mathbf{z})$ be the joint probability density of $\mathbf{Z}$. Then, the total uncertainty about $\mathbf{Z}$ is given by the entropy $H(\mathbf{Z}) = \mathbb{E}\left[-\ln \mathcal{D}(\mathbf{Z})\right] = \mathbb{E}\left[-\ln \mathcal{D}\left(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\right)\right]$ (Shannon, 1948). Under the above setting, only the $\mathbf{Z}^{(2)}$ vector is observable, so we are interested in minimizing the uncertainty about $\mathbf{Z}^{(1)}$ given $\mathbf{Z}^{(2)}$. That is, we seek to minimize the conditional entropy $H\left(\mathbf{Z}^{(1)} \mid \mathbf{Z}^{(2)}\right)$. From the chain rule of conditional entropy, namely, from

$$H\left(\mathbf{Z}^{(1)} \mid \mathbf{Z}^{(2)}\right) = H\left(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\right) - H\left(\mathbf{Z}^{(2)}\right),$$

combined with the fact that the total system entropy $H(\mathbf{Z}) = H\left(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\right)$ is fixed, we conclude that the minimization of $H\left(\mathbf{Z}^{(1)} \mid \mathbf{Z}^{(2)}\right)$ is equivalent to the maximization of $H\left(\mathbf{Z}^{(2)}\right)$. Namely, our task is to maximize the entropy of observations associated with the observed sites.

The definition of the corresponding utility function $U : \mathscr{X} \to \mathbb{R}$ is straightforward. Specifically, for $\mathbf{x} = \{x_1, \ldots, x_k\} \in \mathscr{X}$, let $\mathbf{Z}_{\mathbf{x}} = (Z_{x_1}, \ldots, Z_{x_k})$ be a k-sized vector of measurements which was extracted from $\mathbf{Z} = (Z_1, \ldots, Z_m)$ using the set of indices $\mathbf{x}$. Then, by defining $U(\mathbf{x}) \triangleq H(\mathbf{Z}_{\mathbf{x}})$, we establish the correspondence between the optimal network design and the OFSS problems.

While the above setup is valid for a general joint probability density $\mathcal{D}(\mathbf{z})$, we concentrate on the important special case, for which $\mathcal{D}(\mathbf{z})$ is the multivariate Normal distribution; namely, $\mathbf{Z} \sim \mathsf{N}_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For this particular distribution, we have that $H(\mathbf{Z}) \propto \ln|\boldsymbol{\Sigma}|$, where $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix (Ahmed and Gokhale, 2006). Therefore, the maximization of $H\left(\mathbf{Z}^{(2)}\right)$ requires the maximization of the natural logarithm of a *principal sub-matrix* of $\boldsymbol{\Sigma}$, indexed by the observable sites index set $\mathbf{x}$. Such principal sub-matrix is denoted by $\boldsymbol{\Sigma}_{\mathbf{x}}$ and one can extract it from $\boldsymbol{\Sigma}$ by taking all rows and columns of $\boldsymbol{\Sigma}$ that are in $\mathbf{x}$. Finally, we define the utility function to be $U(\mathbf{x}) \triangleq \ln|\boldsymbol{\Sigma}_{\mathbf{x}}|$.

D-optimal experimental design

The D-optimal experimental design problem involves a discovery of the best subset of m possible experiments. Formally, let X be an $m \times m$ model matrix. The D-optimal design is one that maximizes the determinant of the principal sub-matrix, $X_{\mathbf{x}}^{\top} X_{\mathbf{x}}$, where $X_{\mathbf{x}}$ is a $k \times k$ matrix $(k \leq m)$ extracted from X by taking all rows and columns of X that are indexed by $\mathbf{x} \in \mathscr{X}$ (Goos and Bradley, 2011). The objective is to maximize the so-called D-optimality criterion. The latter is defined as a negative of natural logarithm of the experiment's *information matrix* $X_{\mathbf{x}}^{\top} X_{\mathbf{x}}$ (Roy, 2000). Therefore, by defining the utility function to be $U(\mathbf{x}) \triangleq \ln |X_{\mathbf{x}}^{\top} X_{\mathbf{x}}|$, we can easily see that the D-optimal experimental design problem corresponds to the OFSS setting (Wolters, 2015).

Our objective is to develop an efficient procedure for the maximization of (2). However, since this problem is known to be hard (Chun-Wa et al., 1995), we resort to approximate evolutionary strategies. A brief overview of these methods is detailed in Section 2.2. We refer to (Mullen, 2014) for a detailed discussion of global optimization techniques.

2.2 Global continuous optimization

Evolutionary strategies are stochastic optimization procedures that are motivated by a biological process of natural selection (Bäck, 1996). Specifically, these methods simulate the evolutionary mechanism by producing a sequence of population generations that (hopefully) improve the population's average fitness (utility) over time. A very general evolutionary framework, which is implemented in many optimization packages, is summarized in Fig. 2. In this study, we consider several state-of-the-art evolutionary algorithms, namely, Differential Evolution (DE) (Price et al., 2005), Particle Swarm Optimization (PSO) (Kennedy and Mendes, 2002), Genetic algorithm (GA) (Melanie, 1998), and Cross Entropy algorithm (CE) (Rubinstein and Kroese, 2017).

The major advantage of such nature-inspired methods, is that they can operate in both the discrete and the continuous state spaces. However, for non-standard discrete applications such as the OFSS problem, an evolutionary method will customarily require an adjustment which will generally involve a development of a specialized *next population creation* (see Fig. 2). Designing such operations can be a time-consuming task that may require a substantial development effort. For example, the `kofnGA` (Wolters, 2015)

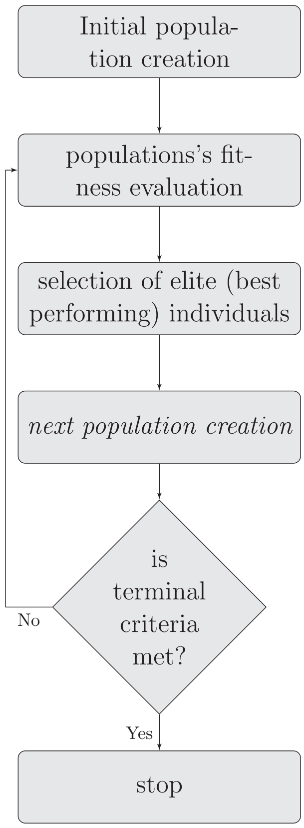


Fig. 2: A general evolutionary framework.

package, which is suitable for handling OFSS tasks, such as the optimal monitoring network design problems, is based on the discrete version of GA (Melanie, 1998).

Motivated by the fact that the next population creation in the continuous space is quite standard (Mullen, 2014), we explore a possibility of adopting the continuous optimization approach. In particular, we design a *transformation* that allows us to use an embedding of the original discrete space of the OFSS problem into a larger continuous space. The transformation is detailed in Section 2.3.

2.3 The transformation

In this section, we assume that one has an access to a global continuous optimization procedure (such as DE, PSO, GA, or CE), which is capable of handling problems of the type:

$$\begin{aligned} &\max_{\mathbf{y}\in\mathbb{R}^m} \; C(\mathbf{y}) \\ &\text{subject to: } \mathbf{y} \in \mathscr{Y}, \end{aligned} \tag{3}$$

where $C : \mathbb{R}^m \to \mathbb{R}$ is an objective function, and $\mathscr{Y} \subseteq \mathbb{R}^m$ is a compact convex set. The transformation from the continuous state space $\mathscr{Y}$ to the original discrete state space $\mathscr{X}$ (see (1)), and the corresponding calculation of the utility, is summarized in Algorithm 1.

Algorithm 1: The transformation

Input: A vector of real numbers $\mathbf{y} \in \mathscr{Y} \subseteq \mathbb{R}^m$, $m, k \in \mathbb{N}$ such that $(k \leq m)$, and a utility function $U : \mathscr{X} \to \mathbb{R}$ from (2).
Output: The utility of $\mathbf{x} \in \mathscr{X}$, where $\mathbf{x}$ corresponds to the continuous input $\mathbf{y} \in \mathscr{Y}$.

```
1 x′ ≜ (x_1, ..., x_m) ← argsort(y)
2 x ← {x_1, ..., x_k}
3 return U(x)
```

In Line 1 of Algorithm 1, an *ascending* sorting procedure (Cormen et al., 2001) is used to determine the internal ordering of indices of $\mathbf{y}$. Note that $\mathbf{x}'$ is a permutation of $(1, \ldots, m)$. Then, in line 2, we fix $\mathbf{x}$ to be a set that contains the k-sized prefix of $\mathbf{x}'$; note that $\mathbf{x} \in \mathscr{X}$. Finally, in Line 3 we use the previously obtained $\mathbf{x}$ to calculate the utility in the discrete space.

It is not very hard to see that there exists $\mathbf{y}^* \in \mathscr{Y}$ that corresponds (via the transformation in Algorithm 1), to the optimal (discrete) solution $\mathbf{x}^* \in \mathscr{X}$, such that $U(\mathbf{x}^*) \geq U(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{X}$. To see this, suppose without the loss of generality that $\mathbf{x}^*$ contains the first k sites, that is, $\mathbf{x}^* = \{1, \ldots, k\}$. Then, for example, $\mathbf{y}^* = (\underbrace{0, \ldots, 0}_{k}, \underbrace{1, \ldots, 1}_{n-k})$ corresponds to $\mathbf{x}^* = \{1, \ldots, k\}$ via Algorithm 1.

The proposed transformation opens the way for addressing the OFSS problem via any global continuous optimization method. To see this, note that Algorithm 1 implements the function $C(\mathbf{y})$ in (3). In Section 3 we perform a comprehensive experimental study to examine the performance of the proposed transformation.

3 Results and discussion

We investigate the accuracy of the proposed method when applied to several representative examples. The rest of the section is structured as follows.

1. The first example is a custom-made environmental monitoring design task for with we known the optimal solution. Specifically, we examine two regular lattices of dimensions 5×5 and 9×9. To ensure symmetry, we choose $k = 9$ and $k = 25$ (see Fig. 3), for the first and the second lattice, respectively. This example enables the precise bench-marking of the proposed algorithm accuracy.
2. Our second problem is a real-life example in which we consider the ozone concentrations in the state of New York. The example involves an analysis of 100 potential locations for a placement of 15 monitoring stations.
3. Next, we examine a larger monitoring network design instance with 1000 possible site placements among which we seek to determine the optimal location of 10, 15, and 20 monitoring stations. This example is motivated by the fact that nowadays, a monitoring network designer encounters typical networks with several hundreds of potential locations.
4. To demonstrate the generality of the proposed technique, we conclude the numerical study with an example from a quite different domain. Specifically, we consider the optimal design of statistical experiments and in particular the D-optimal design problem (Goos and Bradley, 2011). The application of the method to the D-optimal design problem is feasible, since the latter can be modeled via the OFSS setting.

The experimental setup

All tests were implemented using the open-source `R` statistical software package (version 3.5.2). The software is available from the author's web-page along with all examples from this section. The script was executed on 64 bit Windows 10 desktop machine with Intel Core i7-3770 quad-core 3.4GHz processor and 16GB of RAM.

With a view to provide a fair comparison between different global optimization algorithms, we use a set of freely available `R` packages. Specifically, we work with the following libraries: `kofnGA` version 1.3, `DEoptim` version 2.2.4, `pso` version 1.0.3, `GA` version 3.1.1, and `CEoptim` version 1.2. In addition, we ran a number of preliminary benchmarks (not reported here), to determine a reasonably robust parameter setting for the optimization packages under consideration. While the optimal parameter determination is subject to a further research, our objective was to allow an approximately equal running (CPU) time for all methods involved.

Finally, note that we seek to *maximize* the entropy of the measurements obtained from the set of observable sites. In this section, we rather minimize the corresponding negative entropy. The latter is motivated by a purely technical consideration, since the majority of global optimization packages minimize the objective function. Specifically, instead of (3), we work with:

$$\min_{\mathbf{y} \in \mathbb{R}^m} \; -C(\mathbf{y}), \quad \text{subject to: } \mathbf{y} \in \mathscr{Y}.$$

Unless stated otherwise, each algorithm (kofnGA, DE, PSO, GA, and CE), was executed for 20 times. All CPU times are reported in seconds. To ensure reproducibility, we use the same random seed (of "12345") for all experiments reported in this section.

3.1 Custom grid with known solution

In order to benchmark the proposed method, we consider a custom 2D-grid (regular lattice) of dimensions 5×5 and 9×9. Note that these examples represent random fields with $m = 25$ and $m = 81$ possible locations, respectively. In these benchmarks, we assume that the data is distributed according to the multivariate Normal distribution, namely $\mathbf{Z} \sim \mathsf{N}_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and that the covariance matrix $\boldsymbol{\Sigma}$ is known. Specifically, for each component $\Sigma_{i,j}$ for $1 \leq i, j \leq m$ in $\boldsymbol{\Sigma}$, we set $\Sigma_{i,j} = 10 \exp\{-0.1 \, \mathrm{dist}(i, j)\}$, where $\mathrm{dist}(i, j)$ is the Euclidean distance between locations i and j in the lattice. We set $k = 9$ and $k = 25$ for the 5×5 and the 9×9 grids, respectively. The corresponding optimal locations of the monitoring stations are shown in Fig. 3.

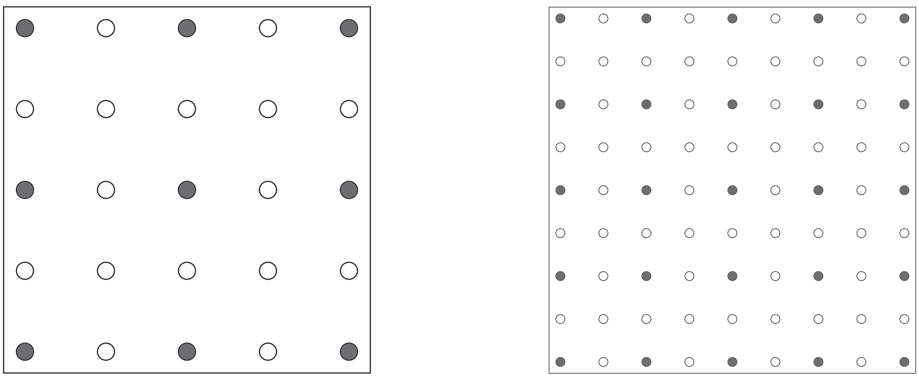
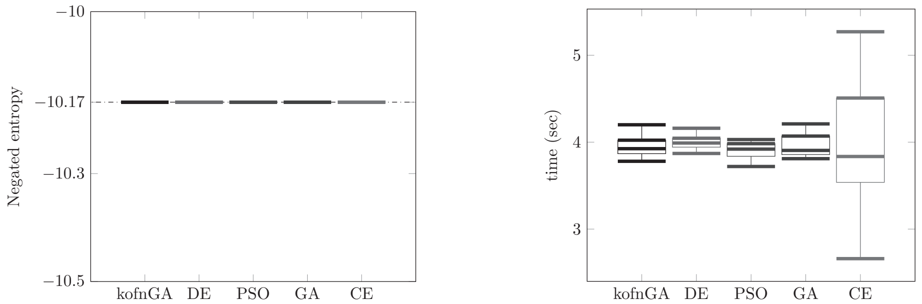


Fig. 3: Optimal locations for the 5×5 $k = 9$ (left panel), and 9×9 $k = 25$ (right panel), latices.

The 5×5 lattice

We start with a small 5×5 lattice for which we wish to select $k = 9$ observable sites from $m = 25$ possible locations. In this particular setting, a full-enumeration procedure requires to consider $\binom{25}{9} = 2042975$ combinations. The optimal objective function value found is -10.167694 and the obtained optimization results are shown in Fig. 4. As we are dealing with a small toy example, it is not very surprising to see that all methods deliver the optimal solution.



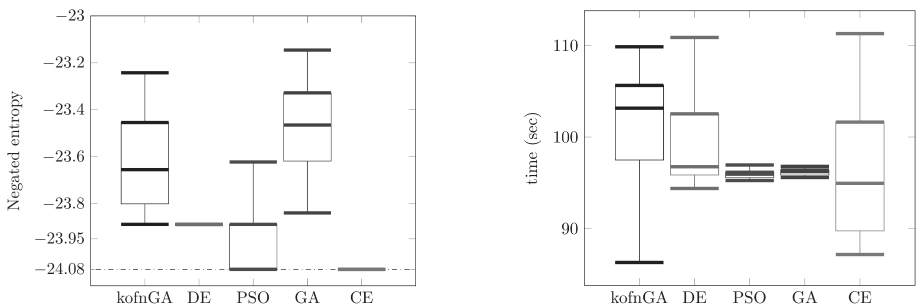
(a) The negative entropy as a function of algorithm.

(b) CPU time as a function of algorithm.

Fig. 4: The 5 × 5 lattice optimization results.

The 9 × 9 *lattice*

Next, we continue with a bigger 9×9 lattice for which we set $m = 81$ and $k = 25$. In this situation, a full-enumeration procedure needs to consider $\binom{81}{25} \approx 5.256 \times 10^{20}$ combinations. The optimal objective function value found is −24.08018. Fig. 5 summarizes the obtained optimization results. For this particular problem, CE always found the optimal solution, and the PSO was next to the best. Similar to the 5 × 5 lattice example, the CE running time introduced a higher variance.



(a) The negative entropy as a function of algorithm, (lower is better).

(b) CPU time as a function of algorithm.

Fig. 5: The 9 × 9 lattice results obtained with kofnGA, DE, PSO, GA, and CE.

3.2 Environmental monitoring

In this section, we consider a real-life network monitoring design problem from (Le and Zidek, 2006). Since the authors examine 100 possible locations for placing 15 monitoring stations, the full-enumeration procedure requires the consideration of $\binom{100}{15} \approx 2.533 \times 10^{17}$ combinations.

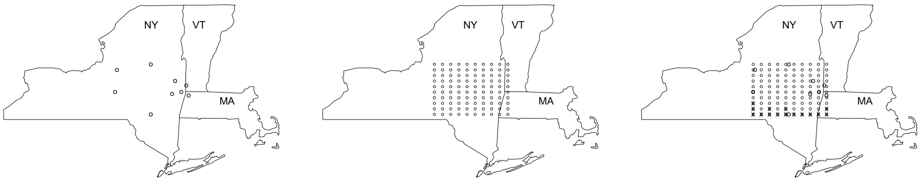
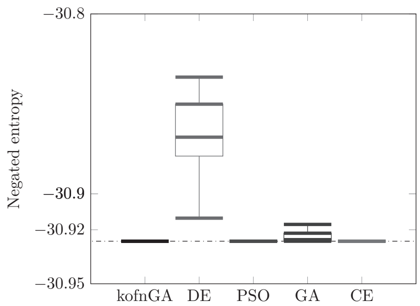


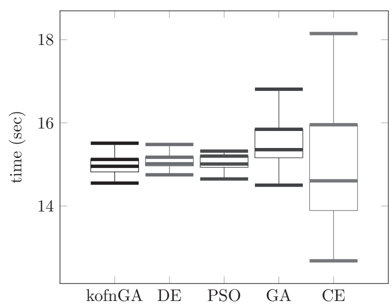
Fig. 6: Existing sites (left panel), potential placements (middle panel), and a combination of existing ($\circ$) and optimal new ($\otimes$) site placement (right panel).

The left and the middle panel of Fig. 6 show the 9 existing and the 100 potential site locations, respectively. The Fig. 6 right panel shows the nine original ($\circ$) locations and the new placement of optimized locations ($\otimes$) of 15 monitoring stations.

Fig. 7 summarizes the optimization results obtained with different algorithms. We can see that for this particular problem, kofnGA, PSO, GA, and CE, managed to obtain the -30.9263 value for the objective function — the best solution known so far (Wolters, 2015).



(a) The negative entropy as a function of algorithm, (lower is better).

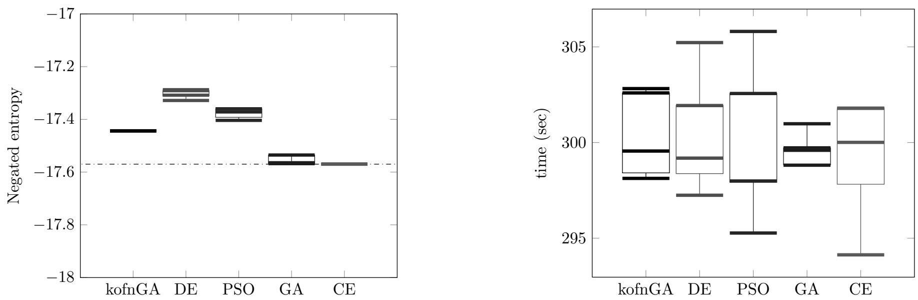


(b) CPU time as a function of algorithm.

Fig. 7: The $m = 100$ $k = 15$ environmental monitoring design problem optimization results.

3.3 Custom covariance matrix

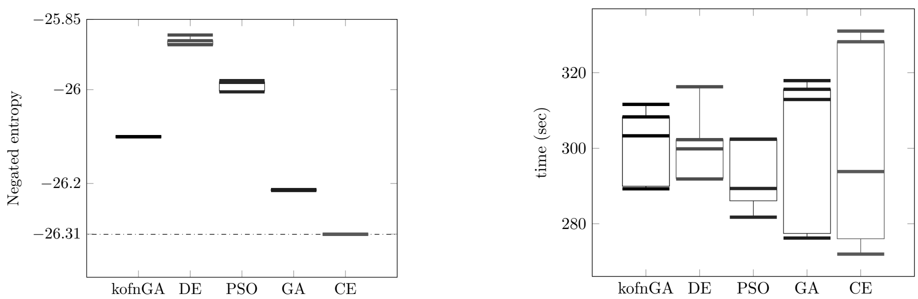
To test the performance of the proposed technique on a larger network design problem, we consider a 1000×1000 custom covariance matrix ($m = 1000$), with $k = 10$, $k = 15$, and $k = 20$. As mentioned in the beginning of this section, we choose to examine a 1000 sites example, since many practical monitoring design tasks generally contain several hundreds of potential locations for the placement of monitoring stations. For this particular problem, we are not aware of any optimal solution, since the full-enumeration procedure requires a consideration of a large number of $\binom{1000}{10} \approx 2.634 \times 10^{23}$, $\binom{1000}{15} \approx 6.881 \times 10^{32}$, and $\binom{1000}{20} \approx 3.395 \times 10^{41}$ combinations, for $k = 10$, $k = 15$, and $k = 20$ cases, respectively. Therefore, we simply compare all algorithms and deliver the best solution found. Fig. 8, Fig. 9, and Fig. 10, summarize the obtained results. Among all algorithms, CE performed in the best fashion. Namely, the CE method manages to find the smallest values of $-17.5699, -26.3084$, and -35.0301 of the objective function, for $k = 10$, $k = 15$, and $k = 20$, respectively.



(a) The negative entropy as a function of algorithm, (lower is better).

(b) CPU time as a function of algorithm.

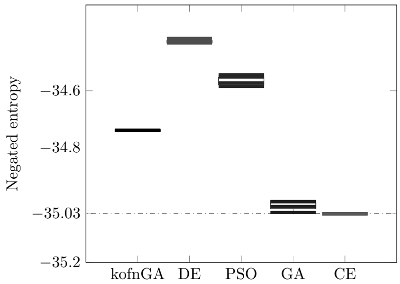
Fig. 8: The $m = 1000$ $k = 10$ custom monitoring design problem optimization results.



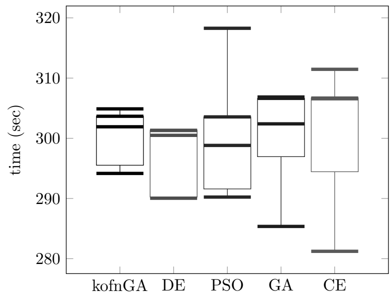
(a) The negative entropy as a function of algorithm, (lower is better).

(b) CPU time as a function of algorithm.

Fig. 9: The $m = 1000$ $k = 15$ custom monitoring design problem optimization results.



(a) The negative entropy as a function of algorithm, (lower is better).



(b) CPU time as a function of algorithm.

Fig. 10: The $m = 1000$ $k = 20$ custom monitoring design problem optimization results.

Remark 1 Since only the CE algorithm obtained the best known objective function values for all cases, we allowed all algorithms to run for an additional time. Specifically, we increased the CPU time limit by a factor of 10 for each global optimization method. Each algorithm was executed for 5 times. As a consequence, both CE and GA, managed to achieve the (possibly) optimal results for all $k \in \{10, 15, 20\}$. Table 1 summarizes the best results obtained by all optimization methods when using the the additional computational time.

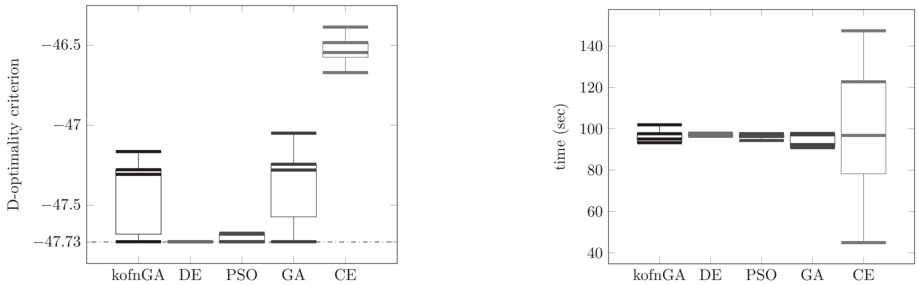
Table 1: The minimal negative entropy achieved by global optimization algorithms using the increased computational time limit.

algorithm	$k = 10$	$k = 15$	$k = 20$
kofnGA	−17.4434	−26.1002	−34.7382
DE	−17.4304	−26.1307	−34.6078
PSO	−17.3985	−26.0879	−34.5897
GA	**−17.5699**	**−26.3084**	**−35.0301**
CE	**−17.5699**	**−26.3084**	**−35.0301**

By analyzing the results from Sections 3.1, 3.2, and 3.3, one can arrive to a wrong conclusion about the CE algorithm superiority as compared to other methods. To see that there is no single best algorithm, we will examine a problem from a different domain, the *D-optimal experimental design*.

3.4 D-optimal experimental design

In this section, we consider an instance of the D-optimal design problem called the *robustness experiment* (Goos and Bradley, 2011). Following (Wolters, 2015), we view this task as an OFSS problem with $m = 81$ and $k = 24$. Consequently, a full-enumeration procedure requires the evaluation of $\binom{81}{24} \approx 2.305 \times 10^{20}$ combinations. For this problem, Wolters (Wolters, 2015) manages to obtain the (probably) optimal result (of -47.728 on the negative log scale) in four out of 20 runs of the problem using the kofnGA method. We repeat the experiment using DE, PSO, GA, and CE algorithms. Similar to previous experiments, we set the parameters such that the running times of all algorithms are comparable. Fig. 11 summarizes the obtained results.



(a) The D-optimality as a function of algorithm, (lower is better).

(b) CPU time as a function algorithm.

Fig. 11: The $m = 81$ $k = 24$ D-optimal design problem optimization results.

Table 2: Frequency of obtaining the best-known solution.

algorithm	success rate
kofnGA	15%
DE	100%
PSO	70%
GA	5%
CE	0%

Table 2 shows the frequencies of obtaining the best-known solution using different algorithms. It is interesting to note that the CE method always failed to achieve the best known objective function value. Nevertheless, the DE algorithm always managed to obtain the optimal result, and the PSO method performed reasonably well, too. The kofnGA and GA methods achieved a relatively law success rate of 15% and 5%, respectively.

4 Conclusion

In this study we showed that *any* global continuous optimization method is applicable for solving the optimal fixed-size subset selection problem. We presented a suitable embedding of the original discrete state space into the continuous state space, and introduced a transformation that allows to employ continuous optimization procedures for the original discrete problem solution. Our numerical study indicates that the achieved performance compares favorably with the existing open-source specifically designed software. We showed that by utilizing the proposed method, one can run several algorithms in parallel and report the best solution obtained. Moreover, this procedure can be carried out with a minimal development effort. Unsurprisingly, our numerical study indicates that there is no singe best method available, and in practice, we advise to run several global optimization algorithms side-by-side.

While this study introduce a number of obvious benefits, it also suggests several promising directions for future research. First, we explored only a very limited number of continuous global optimization methods, namely, the Differential Evolution, the Particle Swarm Optimization, the Genetic Algorithm, and the Cross Entropy method. Our numerical study, and in particular, the D-optimal design example from the statistical domain, shows that it is important to explore additional continuous optimization techniques. The second promising direction is to explore different transformations from the continuous to the discrete space. A successful transformation should be constructed with a view to a satisfaction of smoothness condition of the objective function. A carefully designed embedding can assist a global optimization algorithm to improve both the speed of convergence and the solution obtained. Finally, it is important to note that the the Cross Entropy method deserves an additional attention as compared to other methods, since it can operate in stochastic environments. That is, it can be potentially used for an on-line monitoring network designs, and thus improve our capabilities of decision making under uncertainty.

Declaration of interest statement

The author declares that there are no conflicts of interest in the preparation of this manuscript.

Acknowledgments

I am thoroughly grateful to the anonymous reviewers and the editor for their valuable and constructive remarks and suggestions. This work was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under CE140100049 grant number.

References

Ahmed NA, Gokhale DV (2006) Entropy expressions and their estimators for multivariate distributions. IEEE Trans Inf Theor 35(3):688–692

Allcott H, Gentzkow M (2017) Social media and fake news in the 2016 election. Journal of Economic Perspectives

Bäck T (1996) Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, Inc., New York, NY, USA

von Brömssen C, Fölster J, Futter M, McEwan K (2018) Statistical models for evaluating suspected artefacts in long-term environmental monitoring data. Environmental Monitoring and Assessment 190(9):558

Bruns DA, Wiersma GB, Jr EJR (1991) Ecosystem monitoring at global baseline sites. Environmental Monitoring and Assessment 17(1):3–31

Chan CK, Yao X (2008) Air pollution in mega cities in China. Atmospheric Environment 42(1):1–42

Chao Q, Yu Y, Zhou Z (2015) Subset Selection by Pareto Optimization. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in Neural Information Processing Systems 28, Curran Associates, Inc., pp 1774–1782

Chun-Wa K, Jon L, Queyranne M (1995) An exact algorithm for maximum entropy sampling. Oper Res 43(4):684–691

Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to Algorithms, Second Edition. The MIT Press and McGraw-Hill Book Company, Cambridge, Massachusetts London, England

Geoffrey D, Stepahie M, Marco A (1997) Adaptive greedy approximations. Constructive Approximation 13(1):57–98

Goos P, Bradley J (2011) Optimal Design of Experiments: A Case Study Approach. John Wiley & Sons, West Sussex

Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), vol 2, pp 1671–1676

Le ND, Zidek JV (2006) Statistical Analysis of Environmental Space-Time Processes. Springer-Verlag, New York

Melanie M (1998) An Introduction to Genetic Algorithms. The MIT Press, Cambridge, MA, USA

Mi Z, Meng J, Guan D, Shan Y, Liu Z, Wang Y, Feng K, Wei Y (2017) Pattern changes in determinants of Chinese emissions. Environmental Research Letters 12(7):074003

Mullen K (2014) Continuous global optimization in R. Journal of Statistical Software, Articles 60(6):1–45

Natarajan B (1995) Sparse approximate solutions to linear systems. SIAM J Comput 24(2):227–234

Park SS, Jeong JU, Schauer JJ (2013) Sources and Their Contribution of Particulate Water-Soluble Organic Carbon Observed During One Year at a Traffic-Dominated Site. Atmospheric Environment 77:348–357

Price K, Storn RM, Lampinen JA (2005) Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series). Springer-Verlag, Berlin, Heidelberg

Ramanathan V, Carmichael G (2008) Global and regional climate changes due to black carbon. Nature Geoscience 1:221–227

Ramiro R, Ferreira M, Schmidt AM (2010) Stochastic search algorithms for optimal design of monitoring networks. Environmetrics 21(1):102–112

Roy FB (2000) Physics from Fisher information: A unification. American Journal of Physics 68(11):1064–1065

Rubinstein RY, Kroese DP (2017) Simulation and the Monte Carlo Method, 3rd edn. John Wiley & Sons, New York

Shannon CE (1948) A mathematical theory of communication. The Bell System Technical Journal 27(3):379–423

Silverman C, Singer-Vine J (2016) Most Americans Who See Fake News Believe It, New Survey Says. BuzzFeed News

Stokes JR, Horvath A (2010) Supply-chain environmental effects of wastewater utilities. Environmental Research Letters 5(1):014015

Wiersma G (1984) Integrated global background monitoring network. In: Symposium on research and monitoring in circumpolar biosphere reserves, Alberta, Canada, 27 Aug 1984, United States

Wolters M (2015) A genetic algorithm for selection of fixed-size subsets with application to design problems. Journal of Statistical Software, Code Snippets 68(1):1–18

Yu B, Yuan B (1992) A dynamic selection algorithm for globally optimal subsets. Engineering Applications of Artificial Intelligence 5(5):457–462