Multi-objective constrained Bayesian optimization for structural design

(article starts on next page)

RESEARCH PAPER

Multi-objective constrained Bayesian optimization for structural design

Alexandre Mathern[1,2] · Olof Skogby Steinholtz[3,4] · Anders Sjöberg[3,4] · Magnus Önnheim[3,4] · Kristine Ek[2] · Rasmus Rempling[1] · Emil Gustavsson[3,4] · Mats Jirstrand[3,4]

Abstract

The planning and design of buildings and civil engineering concrete structures constitutes a complex problem subject to constraints, for instance, limit state constraints from design codes, evaluated by expensive computations such as finite element (FE) simulations. Traditionally, the focus has been on minimizing costs exclusively, while the current trend calls for good trade-offs of multiple criteria such as sustainability, buildability, and performance, which can typically be computed cheaply from the design parameters. Multi-objective methods can provide more relevant design strategies to find such trade-offs. However, the potential of multi-objective optimization methods remains unexploited in structural concrete design practice, as the expensiveness of structural design problems severely limits the scope of applicable algorithms. Bayesian optimization has emerged as an efficient approach to optimizing expensive functions, but it has not been, to the best of our knowledge, applied to constrained multi-objective optimization of structural concrete design problems. In this work, we develop a Bayesian optimization framework explicitly exploiting the features inherent to structural design problems, that is, expensive constraints and cheap objectives. The framework is evaluated on a generic case of structural design of a reinforced concrete (RC) beam, taking into account sustainability, buildability, and performance objectives, and is benchmarked against the well-known Non-dominated Sorting Genetic Algorithm II (NSGA-II) and a random search procedure. The results show that the Bayesian algorithm performs considerably better in terms of rate-of-improvement, final solution quality, and variance across repeated runs, which suggests it is well-suited for multi-objective constrained optimization problems in structural design.

1 Introduction

The construction, operation, and maintenance of civil engineering and building structures account for not only very large costs, but also major negative environmental and social impacts in terms of the tremendous material consumption, as well as health and safety issues often associated with construction activities.

Today, it becomes increasingly important to take various objectives into account to improve buildability and performance of structures and reduce their economic, environmental, and social impacts in a life cycle perspective. Nevertheless, in the early structural design process, when the possibility to influence the outcome is the largest, the traditional practice is to focus solely on the lowest initial costs, see, e.g., International Federation for Structural Concrete (2009), Mathern et al. (2013), and World Economic Forum (2016). Cost optimization is often conducted by means of employing a point-based incremental improvement strategy, relying on trial-and-error and the designer's intuition, see, e.g., Parrish (2009) and Maher (1987). This

Responsible Editor: Ji-Hong Zhu

✉ Alexandre Mathern
  alexandre.mathern@chalmers.se

[1] Department of Architecture and Civil Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

[2] NCC AB, SE-405 14 Gothenburg, Sweden

[3] Fraunhofer-Chalmers Centre, SE-412 88 Gothenburg, Sweden

[4] Fraunhofer Center for Machine Learning, SE-412 88 Gothenburg, Sweden

🌀 Springer

approach overlooks potentially better design solutions and neglects other relevant objectives than cost.

A better alternative is to take into account many different aspects simultaneously in the design process to find the most efficient design solution with regard to all relevant objectives simultaneously. Finding the most appropriate design for the set of objectives considered requires modern design methods that allow to explore many design choices and evaluate them against these objectives.

Multi-objective optimization concerns the problem of simultaneously maximizing the utility values of multiple, typically conflicting, objective functions (Tamaki et al. 1996). For engineering applications, this setting is generally more common than the single-objective setting as usually many objectives have to be balanced (Nakayama 2005), e.g., the cost and performance of a construction. Typically, the different objectives are conflicting, which is why no single solution exists and hence instead the challenge becomes to retrieve a set of solutions providing a human decision maker with a set of diverse objective trade-offs (Vlennet et al. 1996). Additionally, in real-world optimization applications, constraints are generally imposed on the candidate solutions, which may render individual solutions infeasible despite exhibiting favorable objective values (Deb et al. 2001). In structural design, such constraints take the form of requirements defined by structural norms and standards such as the Eurocodes (European Committee for Standardization 2005). Appropriate structural and load models are required to determine the design effects of actions and verify the associated design constraints in ultimate limit states (ULS) and serviceability limit states (SLS).

Mathematically, we are interested in solving the problem:

$$\min_{\boldsymbol{x}} \ (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \tag{1a}$$

$$s.t. \ g_j(\boldsymbol{x}) \leq 0, \tag{1b}$$

$$\boldsymbol{x} \in \mathbb{R}^D, \tag{1c}$$

where f_i, $i = 1, \ldots, m$, are objective functions, g_j, $j = 1, \ldots, n$, are constraint functions, and the variable $\boldsymbol{x}$ is a D-dimensional real valued input. Given a set of multi-objective solutions retrieved by some kind of search procedure, the quality of individual solutions is often classified in terms of Pareto optimality. Again, assuming minimization of m objectives subject to n constraints according to (1), a feasible solution $\boldsymbol{x}^*$ is said to be Pareto optimal, or equivalently, non-dominated, if there does not exist another feasible $\boldsymbol{x}$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all i $\in \{1, 2, \ldots, m\}$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one j (Miettinen 2012).

Structural optimization problems for RC structures are characterized by their evaluation expensiveness. The verification of the constraints, in particular, is often associated with relatively expensive computations, involving structural analysis with 2D or 3D finite element (FE) models and possibly a large number of load combinations (e.g., in bridge design). In order to compute high-quality solutions in a feasible amount of time, the constraint evaluations require a high degree of efficiency in the optimization process; hence, the number of applicable algorithms is severely limited.

Bayesian optimization has emerged as a capable approach to optimizing expensive functions by iteratively constructing a probabilistic surrogate model of the underlying target function, and has had many previous successful applications (Snoek et al. 2012; Calandra et al. 2016; Imani and Ghoreishi 2020). An acquisition function is used to translate the surrogate model to a metric reflecting potential for improvement, which is in turn used to guide the optimizer toward a promising new query point. Bayesian methods are designed to be efficient in terms of function evaluations, but in turn require extensive computational resources on the modelling side, which of course needs to be balanced against the computational time of the underlying function. Bayesian optimization has also been extended to constrained problems (Bernardo et al. 2011; Snoek 2013; Gelbart et al. 2014), and to multi-objective settings (Zuluaga et al. 2013).

Structural optimization of RC structures has been examined previously. Various evolutionary algorithms have been studied (Jahjouh et al. 2013; Mergos and Mantoglou 2020), however without explicitly dealing with the expensiveness aspect of the problem. In order to deal with expensiveness, Penadés-Plà et al. (2019) suggest fitting a constraint-weighted surrogate model to prior evaluations sampled from a latin-hypercube, whereafter the surrogate is optimized in order to select the final design. In contrast to this approach, Bayesian optimization allows surrogate models to be constructed sequentially, utilizing information from prior evaluations at each iteration, while guiding the search procedure, which should decrease the sensitivity to poor initialization.

The aim of this paper is to study the applicability and efficiency of a state-of-the-art constrained Bayesian optimization algorithm on a generic structural design case. The benchmark problem considered consists of structurally optimizing a doubly reinforced concrete beam over a number of design parameters (i.e., dimensions, reinforcement layout, material type) with respect to objective functions covering economic, environmental, and social factors, buildability, and performance aspects of the design options, while fulfilling structural design constraints according to design codes. Two other optimization algorithms are also applied for comparison: the commonly applied NSGA-II (Deb et al. 2002), which is a genetic algorithm specifically targeting multi-objective problems, and a slightly modified random search algorithm.
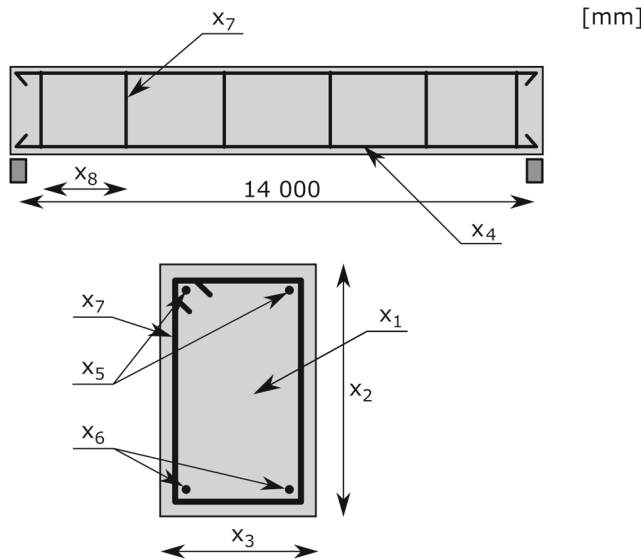
Fig. 1 Drawing of the RC beam considered as design case with indication of the index of the design parameters (see Table 1)

2 Method

2.1 Design case definition

The structural design case considered in this work is the one of a RC beam, as illustrated in Fig. 1. The beam is simply supported with an effective span length (L_{span}) of 14 m. It is designed for two uniformly distributed loads: a characteristic permanent load $g_k = 12\,\text{kN/m} + g_{sw}$, where g_{sw} corresponds to the self-weight of the beam per unit length (assuming densities of $24\,\text{kN/m}^3$ and $78.5\,\text{kN/m}^3$ for concrete and steel, respectively), and a characteristic variable load $q_k = 18\,\text{kN/m}$. This design case was chosen

for its simplicity and because the structural design process for a RC beam constitutes the basis of many more complex common structural engineering cases (e.g., design of parts of buildings' structural systems, of RC beam bridges, of RC wailing beams for support of sheet pile walls). The design parameters considered and their possible values, detailed in Table 1, generate a design space consisting of more than 186 million possible configurations.

2.2 Structural design

The structural analysis was conducted in accordance with Eurocode 2 (European Committee for Standardization 2004) considering bending and shear in ULS and deflection control in SLS. Bending is checked taking into account the parabola-rectangle diagram for concrete under compression and the design stress-strain diagram with an inclined top branch for reinforcing steel under tension and compression. The resulting design constraints are described in Table 2. Crack control and anchorage of longitudinal reinforcement are not included in this study.

2.3 Evaluation of sustainability, performance, and buildability

The objective functions taken into account in the evaluation of a design choice are described in Table 3. These were carefully chosen to address the aim of this work, while being relevant to building or civil engineering design cases regardless of their complexity. All objective functions are to be minimized in the optimization process. Unit values used in the evaluation of the different objectives are described in Appendix 1.

The first objective function, $f_1(x)$, corresponds to the total production cost, C, which is estimated as the sum of the cost of materials, C_m, and the cost of labor, C_w, for the

Table 1 Set of design parameters and possible values

	Design parameters	Index	Values
Concrete	Strength class	x_1	15 values: 14 classes from European Committee for Standardization (2004) and C100/115
Beam	Height (mm)	x_2	21 values in [500, 1500] with step of 50
	Width (mm)	x_3	10 values in [150, 600] with step of 50
Longitudinal bars	Diameter (mm)	x_4	6 values: 12, 14, 16, 20, 25 and 32
	No. of bars (top)	x_5	13 values in [2, 14] with step of 1
	No. of bars (bottom)	x_6	19 values in [2, 20] with step of 1
Stirrups	Diameter (mm)	x_7	5 values: 6, 8, 10, 12 and 14
	Spacing (mm)	x_8	8 values in [100, 450] with step of 50

Table 2 Set of design constraints

	Design constraints	Index	Definition
Geometry and buildability	Placing of longitudinal bars	g_1	Number of bar layers $- 3 \leq 0$, to ensure that bars fit (in max. 3 layers)
			according to detailing rules from European Committee for Standardization (2004)
Ultimate limit state (ULS)	Bending resistance	g_2	$M_{Ed} - M_{Rd} \leq 0$
	Shear resistance	g_3	$V_{Ed} - V_{Rd} \leq 0$
Serviceability limit state (SLS)	Deflection control	g_4	$d_{max} - L_{span}/250 \leq 0$

related construction works:

$$f_1(\boldsymbol{x}) := C(\boldsymbol{x}) = C_{\mathrm{m}}(\boldsymbol{x}) + C_{\mathrm{w}}(\boldsymbol{x}). \tag{2}$$

The material cost, C_{m}, is equal to the unit price, $p_{m,i}$, for each material i (i.e., concrete, steel, and form) times the respective material quantity, $Q_{m,i}$. That is:

$$C_{\mathrm{m}}(\boldsymbol{x}) = \sum_i p_{\mathrm{m,i}} \cdot Q_{\mathrm{m,i}}(\boldsymbol{x}), \tag{3}$$

and the cost of construction works, C_{w}, is equal to the unit labor cost, p_{w}, times the time required for construction work, T_{w}:

$$C_{\mathrm{w}}(\boldsymbol{x}) = p_{\mathrm{w}} \cdot T_{\mathrm{w}}(\boldsymbol{x}). \tag{4}$$

The time of construction works, T_{w}, which also corresponds to the third objective function, $f_3(\boldsymbol{x})$, is calculated according to (5). It is the sum of the unit time for the construction activity associated with each material times the respective material quantity:

$$f_3(\boldsymbol{x}) := T_{\mathrm{w}}(\boldsymbol{x}) = \sum_i t_{\mathrm{m,i}} \cdot Q_{\mathrm{m,i}}(\boldsymbol{x}), \tag{5}$$

where $t_{\mathrm{m,i}}$ refers to the unit construction time for the activities related to each material.

In addition to the construction time, another objective function related to buildability is considered: the height of the beam, $f_4(\boldsymbol{x})$. This choice is also motivated by the interest, in this work, of using an objective function directly equal to one of the design parameters and strongly conflicting with some others.

The second objective function, $f_2(\boldsymbol{x})$, corresponds to the aggregated life cycle analysis (LCA) result for the 16 impact categories using the normalization and weighting factors defined in the European Product Environmental Footprint (Sala et al. 2017, 2018) considering the life cycle stages A1–A3 (material production) and A4 (transportation to the site) (European Committee for Standardization 2019). Impact categories in both the environmental dimension (e.g., climate change, ozone depletion, acidification) and the social dimension (e.g., water use, human health) are covered.

Finally, the fifth objective function, $f_5(\boldsymbol{x})$, is equal to the maximum deflection of the beam at mid-span, d_{max}.

Although the maximum deflection is limited by design codes, which results in a design constraint, it is interesting to include it as an objective as well, as there may be reasons, when it comes to performance or safety, to optimize design options beyond the design requirements set in the codes.

2.4 Formalizing the problem

In this paper, we consider the problem of structurally optimizing a RC beam over the 8 design parameters $(x_1, \ldots, x_8)$ specified in Table 1 with respect to the 5 objective functions $(f_1, \ldots, f_5)$ described in Table 3. Furthermore, each design candidate is obliged to pass three (g_1, g_2, g_3) of the four constraints in Table 2 during the structural analysis procedure described in Section 2.2. The remaining constraint g_4, directly corresponding to objective f_5, is left out for implementation reasons.

As the design parameters are either discrete or categorical (but of ordinal type), each parameter value is chosen to be represented by a non-negative integer index value of its discrete domain. The allowed input space in (1c) hence becomes $\boldsymbol{x} \in \mathcal{B}^8 \subset \mathbb{Z}_{\geq 0}^8$, where $\mathcal{B}^8$ is an 8-dimensional integer-valued hypercube, with the length of each side given by the number of allowed values per design parameter in Table 1.

For the objectives, all functions are to be minimized and no modifications of the problem specified in (1a) are needed, using the case of $m = 5$.

Table 3 Set of objective functions considered

	Objective functions	Index
Economic	Production cost	f_1
Environmental and social	Aggregated LCA result	f_2
Buildability	Construction time	f_3
	Height of the beam	f_4
Performance	Mid-span deflection of the beam	f_5

As previously declared, the three constraint functions g_1, g_2, and g_3 in Table 2 are considered. Furthermore, the routine for evaluating the beam constraints imposes a conditional structure on the evaluations of the constraint functions where g_2 and g_3 may only be evaluated if g_1 is satisfied.

With these modifications, the problem in (1) is restated as:

$$\min_{\boldsymbol{x}} \ (f_1(\boldsymbol{x}), \ldots, f_5(\boldsymbol{x})) \tag{6a}$$

$$s.t. \ \ g_j(\boldsymbol{x}) \leq 0, \quad j = 1, \ldots, 3, \tag{6b}$$

$$\boldsymbol{x} \in \mathcal{B}^8 \subset \mathbb{Z}_{\geq 0}^8, \tag{6c}$$

which defines the optimization problem that is examined. The constraint functions g_j in (6b) are assumed to be prohibitively expensive to evaluate due to underlying numerical computations. In contrast, the objective functions in (6a) are assumed to have known analytical expression that allow for cheap evaluation.

In order to determine the quality of a solution set in multi-objective problems, the hypervolume indicator measure is commonly employed.

Given a finite set of t objective points, $\mathcal{F} = \{\boldsymbol{f}^{(1)}, \ldots, \boldsymbol{f}^{(t)}\} \subset \mathbb{R}^m$ the hypervolume indicator (Zitzler and Thiele 1999), denoted HV, measures the volume of the subspace dominated by $\mathcal{F}$, given a from above bounding reference point $\boldsymbol{r} \in \mathbb{R}^m$:

$$\mathrm{HV}(\mathcal{F}, \boldsymbol{r}) := \lambda_m(\cup_{\boldsymbol{f} \in \mathcal{F}} [\boldsymbol{f}, \boldsymbol{r}]), \tag{7}$$

where λ_m is the Lebesgue measure on $\mathbb{R}^m$.

Typically, the objective values are normalized to $[0, 1]^m$, so that we emphasize all objective functions equally while working with the hypervolume indicator. Furthermore, a reference point $\boldsymbol{r} = \mathbf{1}^m$ can be used. This, however, implies that the minimum and maximum values of each unconstrained objective functions are known beforehand, which may be reasonable considering that the evaluations of the objective functions are cheap. If this is the case, objective values can be normalized by:

$$\hat{f}_i(\boldsymbol{x}) := \frac{f(\boldsymbol{x}) - f_{i*}}{f_i^* - f_{i*}}, \quad i = 1, \ldots, m, \tag{8}$$

where $f_i^* = \max_{\boldsymbol{x} \in \mathcal{B}^8} f_i(\boldsymbol{x})$ and $f_{i*} = \min_{\boldsymbol{x} \in \mathcal{B}^8} f_i(\boldsymbol{x})$. If these values are not known on beforehand, the normalized objective values may be recomputed from (8) whenever a new minimum or maximum solution is encountered.

2.5 Examined algorithms

Three different optimization algorithms are employed to solve the problem defined in (6), as introduced here. Additional details on the implementation are provided in Appendix 2.

2.5.1 Bayesian algorithm

Gelbart et al. (2014) proposed a Bayesian optimization approach to expensive, unknown-constraint optimization. To solve a one-dimensional optimization problem with n unknown constraints, the authors suggested using a separate Gaussian process (see Rasmussen 2004) surrogate model for the objective and each constraint, while employing a constraint-weighted acquisition function of the form:

$$a(\boldsymbol{x}) := \mathrm{EI}(\boldsymbol{x}) \prod_{j=1}^{n} \mathrm{Pr}(-g_j^{\mathrm{GP}}(\boldsymbol{x}) \geq 0), \tag{9}$$

where EI is the standard expected improvement acquisition function, Pr is the probabilistic probability function for constraint satisfaction, and g_j^{GP} is the Gaussian process surrogate model of constraint function g_j. In this work, this method is expanded upon to handle the multi-objective setting with cheap objective evaluations.

Considering the cheapness of the objective functions, EI in (9) may be replaced with actual improvement I, i.e., $\mathrm{I}(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{best}$, as there is no need in constructing surrogates for non-expensive functions.

We now wish to extend (9) to the multi-objective setting. By using the hypervolume indicator definition as stated in (7), the *hypervolume improvement* may be defined as:

$$\mathrm{HVI}(\boldsymbol{f}(\boldsymbol{x}); \mathcal{F}, \boldsymbol{r}) := \mathrm{HV}(\{\boldsymbol{f}(\boldsymbol{x})\} \cup \mathcal{F}, \boldsymbol{r}) - \mathrm{HV}(\mathcal{F}, \boldsymbol{r}). \tag{10}$$

Finally, the acquisition function of (9) is restated as:

$$a(\boldsymbol{x}) := \mathrm{HVI}(\boldsymbol{f}(\boldsymbol{x}); \mathcal{F}, \boldsymbol{r}) \prod_{j=1}^{n} Pr(-g_j^{\mathrm{GP}}(\boldsymbol{x}) \geq 0), \tag{11}$$

now adapted to the cheap and multiple-objective setting, as in problem (6). A flowchart of the proposed Bayesian algorithm is presented in Fig. 2.

A common procedure to optimize the acquisition function in Bayesian optimization is to start with initialization of randomized points and thereafter utilize local search. However, a by-product of the definition of HVI (10) is that as soon as a decently estimated Pareto front is found, the HVI is equal to zero for many $\boldsymbol{x}$. This generally risks trapping the search algorithm in these regions. In order to alleviate this problem, we implement a pattern search (Torczon 1997) algorithm including a tie-breaking strategy for inputs with zero HVI. See Appendix 2 for details.

2.5.2 NSGA-II algorithm

As a benchmark to the Bayesian algorithm, the NSGA-II (Deb et al. 2002), which is a standard genetic algorithm often employed in multi-objective optimization, is included in the study.
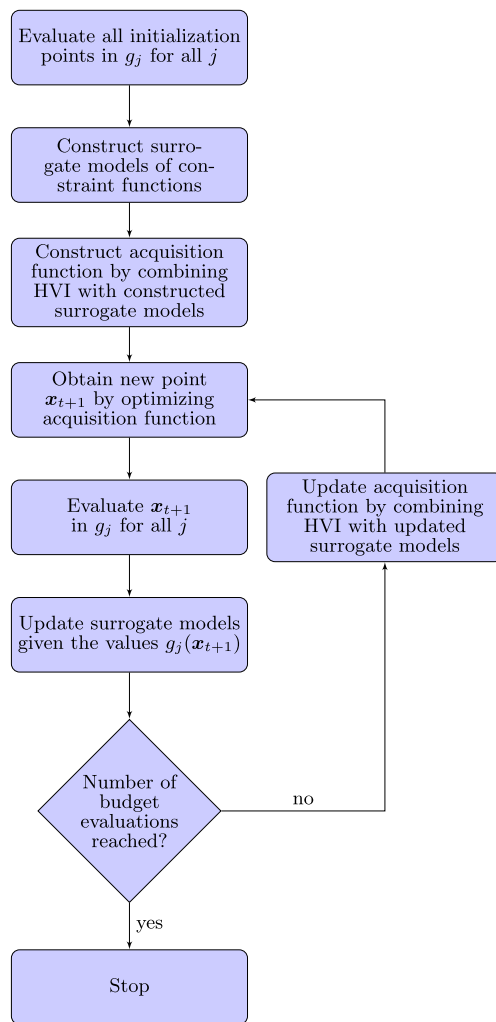
Fig. 2 Flowchart of the proposed Bayesian algorithm

In the NSGA-II algorithm, each solution candidate (i.e., input) is viewed as an individual with a set of properties (a specific design choice), which may be manipulated by mathematical operators representing biological phenomena such as selection, reproduction (crossover), and mutation.

The performance of the NSGA-II algorithm has been evidenced to depend strongly on its hyperparameter configurations (Andersson et al. 2015), such as population size, crossover rate, and mutation rate parameters, implying that some kind of tuning procedure should be performed in practical applications.

2.5.3 Random search algorithm

Finally, a randomized search algorithm is also included as a baseline. Random search optimization works by randomly sampling the input space and evaluating the solutions in sequence. Random search is considered a naive strategy as no information from previous search history is utilized in selecting the next evaluation points. It has however been evidenced

that random search generally compares favorably to grid search (Bergstra and Bengio 2012), i.e., sweeping parameters over a manually specified input subset, especially in situations where only a small number of dimensions bears high influence on the objective function(s).

As the Bayesian algorithm incorporate mechanisms as to avoid evaluating an input with non-positive HVI, the random search algorithm is slightly modified as to avoid wasting evaluations on redundant inputs. For each new evaluation, the objective values of a randomly sampled input are evaluated. If the set of objective values does not yield a positive improvement, the input is re-sampled and the process repeats. Once an input with positive improvement in the objectives is found, the constraint functions are evaluated.

2.5.4 Experimental setup

The three algorithms are examined by running 25 independent optimization runs with an allowed budget of 1000 expensive constraint evaluations.

For the Bayesian algorithm, Gaussian process surrogates were used for the constraint function modeling, using the Matèrn 3/2-kernel (Rasmussen 2004), with a single length-scale parameter. This kernel is a sensible default given the assumption that no prior information exists of the constraints functions being modeled. The hyperparameters of the Gaussian process are determined by maximum likelihood estimation optimized by L-BFGS-B algorithm (Byrd et al. 1995). Input parameter bounds are normalized to the $[0, 1]^8$ hypercube. For constraint evaluation where g_1 is not fulfilled, implying that g_2, g_3 are not available at this input, the corresponding surrogate models are not updated.

The simple structure of the objective functions allows minimum and maximum values of each objective to be discovered by any suitable optimization procedure, whereafter the normalization strategy of (8) is employed.

In each iteration, the acquisition function in (11) is optimized by the pattern search algorithm including the tie-breaking strategy described in Appendix 2, using a budget of 5000 acquisition function evaluations.

For the NSGA-II algorithm, a tuning procedure is first performed as an attempt to mitigate the risk of choosing poor hyperparameters by varying algorithm parameters population size, crossover rate, and mutation rate according to values stated in Table 4. Each of the 27 configurations is run 10 times with seeded initialization. From these results, the highest performing configuration in terms of final hypervolume is selected and then evaluated along the Bayesian algorithm and the random search algorithm. For inputs where constraint evaluations are undefined, that is g_2, g_3 at inputs where g_1 is not fulfilled, the NSGA-II algorithm is provided with the value 0.0 for both constraint functions. This is motivated by the internal

Table 4 Examined NSGA-II hyperparameter values

Hyperparameter	Examined values
Population size	{50, 100, 200}
Crossover rate	{0.6, 0.8, 1.0}
Mutation rate	{0.001, 0.01, 0.1}

sorting procedure included in NSGA-II, where multiple constraints are summed to quantify the degree of infeasibility of a solution; a zero value hence avoids shifting this measure in any direction.

For the random search algorithm, no modifications are needed.

In each of the 25 runs, the algorithms are provided with 100 initial evaluations, including both objective and constraint functions, with inputs generated by sampling from a seeded uniform distribution over the defined input domain of (6c). Initialization inputs are shared for all algorithms per individual run, with the exception of the NSGA-II algorithm using hyperparameter configurations with the population size parameter set to 50, hence using only the first 50 initialization inputs.

All algorithms are implemented in Python 3. The hypervolume computations are carried out using the pygmo-library (Biscani and Izzo 2019). For the constraint surrogate models, the Bayesian algorithm uses the default implementations provided in the pyGPGO-library (Jiménez and Ginebra

2017). The NSGA-II algorithm uses the default implementation provided in the Platypus-library (Hadka 2015).

3 Results and discussion

The performance of the three examined algorithms has been compared in terms of rate of improvement and final-evaluation normalized hypervolume.

As a reference point, $r = \mathbf{1}^8$ is used, which implies that the hypervolume is bounded to [0, 1] with larger values indicating higher quality solution sets.

In the tuning procedure for the NSGA-II algorithm a configuration with population size 50, crossover rate 0.6, and mutation rate 0.1, was seen to exhibit the best results in terms of final hypervolume. Hereafter, NSGA-II refers to this configuration, which is compared along the other two algorithms.

For each algorithm, Fig. 3 shows the 25-run hypervolume mean as a function of constraint evaluation, while Table 5 lists the mean hypervolume obtained after the final evaluation.

As shown in Fig. 3, the best performance is by far achieved by the Bayesian algorithm followed by that of the NSGA-II algorithm. After the random initialization phase, the Bayesian algorithm quickly improves its hypervolume. Already at constraint evaluation 150, the Bayesian algorithm has surpassed the end-of-the-run results for both the NSGA-II and random search algorithms. At around
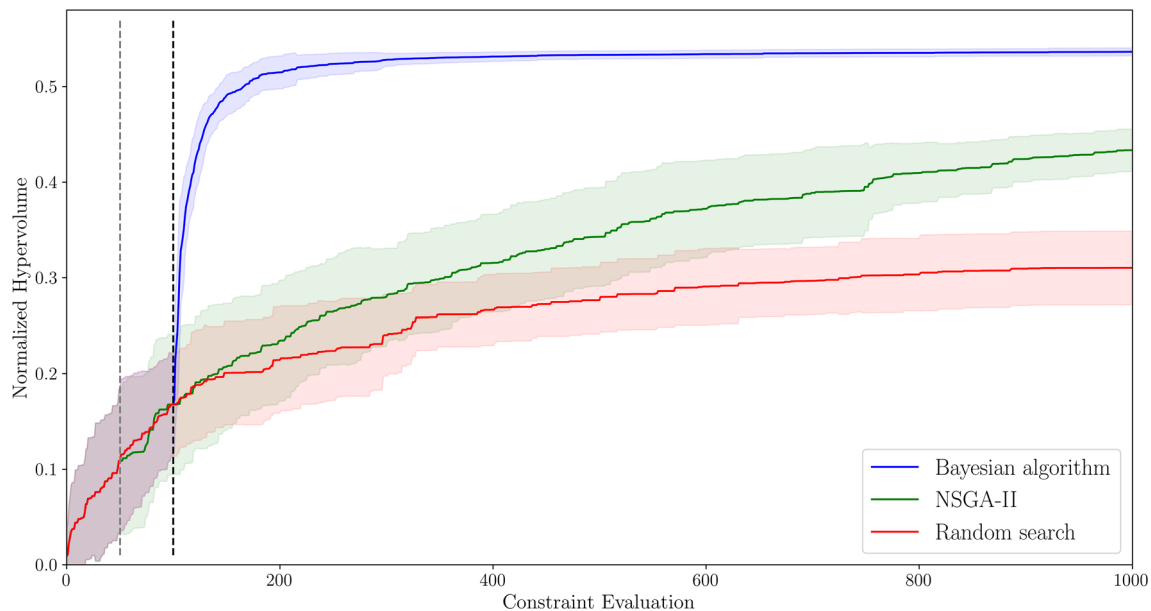
Fig. 3 Normalized hypervolume improvement as a function of constraint evaluation for the three examined algorithms, averaged over 25 runs with seeded initialization. Solid lines indicate the average, and upper/lower limits of the shaded areas represent ±1 standard deviation from the average. For the Bayesian and the random search algorithms, each run is initialized with 100 inputs generated by seeded randomized sampling (marked by the dashed black line). For the NSGA-II algorithm, only the first 50 inputs per seed (marked by the gray dashed line) are used due to its population-size hyperparameter being set to 50

Table 5 Mean normalized hypervolume across 25 runs per algorithm at the final constraint evaluation

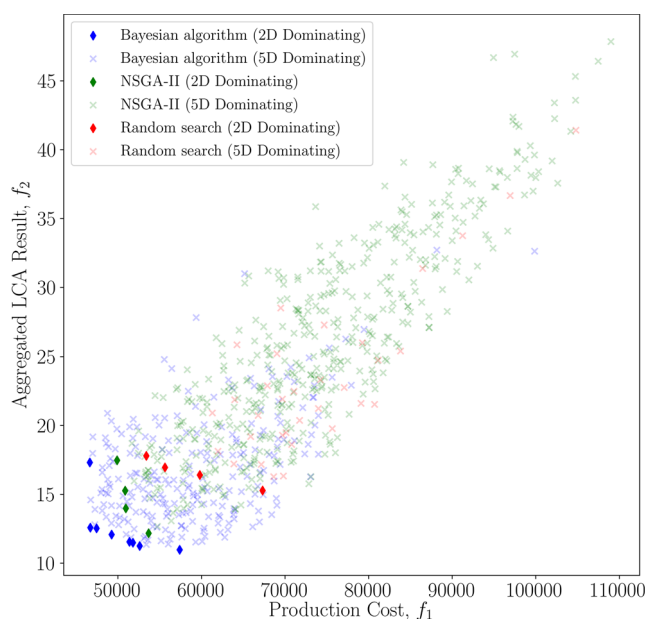| Algorithm	Final hypervolume
Bayesian algorithm	0.536($\pm$0.00415)
NSGA-II	0.433($\pm$0.0223)
Random search	0.310($\pm$0.0386)

Estimated standard deviation within parentheses

evaluation 400, the Bayesian algorithm improves only with a very small rate, barely noticeable on the linearly scaled plot. It should also be noted that the Bayesian algorithm exhibits the lowest variance among the algorithms across the runs, as relevant to practical applications in the expensive setting where multiple runs in general cannot be afforded. Using the end-of-run final hypervolume means listed in Table 5, the Bayesian algorithm yields a 72.8 % larger volume than the random search algorithm.

In Fig. 3, the NSGA-II algorithm is seen to distinctly outperform the random search algorithm from approximately constraint evaluation 700 and onwards, displaying zero overlap with random search in the 1 standard deviation. And, as displayed in Table 5, it achieves an approximately 39.7 % increase in final hypervolume with respect to the random search result, however with a larger variance compared to the Bayesian algorithm. It should be pointed out that the NSGA-II algorithm was subjected to a hyperparameter tuning procedure prior to algorithm evaluation, which is not desirable for practical applications in expensive constraint evaluation settings.

Finally, the random search algorithm exhibits a comparatively low rate-of-improvement, and displays the largest variance exceeding all algorithms at the final constraint evaluation.

In order to demonstrate the applicability of the proposed approach in a more practical sense, results are analyzed in more detail for the two objective functions subjectively deemed the most important: the production costs f_1, and the aggregated LCA result f_2. Figure 4 displays the complete set of solutions belonging to the Pareto front spanning the final hypervolume, projected to the cost/LCA-subspace, for the best run per algorithm. In Fig. 4, solutions along the estimated 2D dominating fronts can for the Bayesian algorithm be seen to dominate solutions for the other two algorithms.

It can also be observed that the whole set of Pareto solutions (i.e., the estimated 5D Pareto front) obtained by the Bayesian algorithm appears to cluster in a better region (lower left corner in Fig. 4) of the subspace, compared with solutions of the NSGA-II and the random search algorithms which show more spread. This, together with the previous observations made on the respective performance of the algorithms,

Fig. 4 Set of algorithm solutions projected to the objective dimensions f_1 and f_2, for the best (highest final hypervolume indicator) run per algorithm. Opaque diamond markers represent solutions on the estimated 2D Pareto front while faded cross markers represent the remaining solutions belonging to the estimated 5D front

indicates that the Bayesian algorithm is capable of finding higher quality trade-offs among the five objectives.

A more detailed view of the solutions on the estimated 2D Pareto fronts of Fig. 4 is shown in Fig. 5.

Among the three annotated solution points produced by the Bayesian algorithm, representing the attained trade-offs between objectives f_1 and f_2, we notice that values for the input parameters x_{5-8} remain fixed; variability in this subspace hence seems to originate only from the x_{1-4} parameters. Also, considering the objectives f_{3-5}, we observe some variability among the solutions in that the objectives f_3 and f_4 seem negatively (positively) correlated with f_2 (f_1) while f_5 exhibits the opposite behavior.

However, as evident in Fig. 4, a lot of additional solutions are found in the very near vicinity of annotated 2D dominating solutions, and may offer more desirable trade-offs in the 5D with an apparent very low penalty in the (f_1, f_2) space. In a practical setting, where we assume that more than two objectives are of interest, alternative visualization techniques, e.g., parallel-coordinates plots (Inselberg 1985), could be employed.

Such visualization of the results, preferably including interactive features, is particularly important to analyze in multi-objective optimization as it can provide valuable insights for a designer or a decision maker when selecting among different designs.

One advantage of employing Bayesian algorithms in general is that prior domain knowledge may be incorporated
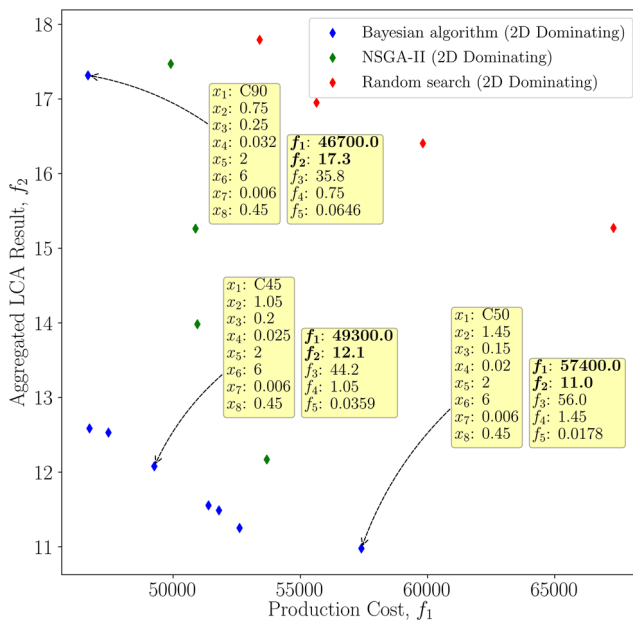
Fig. 5 Scatter plot of estimated 2D Pareto front after projection to the f_1 and f_2 subspaces, for the best (highest final hypervolume indicator) run per algorithm. Three particular solutions yielded by the Bayesian algorithm are annotated with solution input-variables x_{1-8} and objectives f_{1-5}

directly by specifying the hyperparameters. For instance, if the order of differentiability of the target function (here constraint) is known, this prior information could guide the process of selecting the Gaussian process kernel. It should be noted that in this work, no prior knowledge or tuning procedure was used in the selection of Gaussian process hyperparameters. Another advantage is that once the optimization is completed, one callable probabilistic surrogate model per constraint over the input domain has been constructed. This provides the possibility to retrieve estimates of both constraint values and corresponding uncertainties for new inputs, providing information about constraint behavior and unexplored regions.

In order to tackle the multi-objective setting, we have used a normalized hypervolume approach, which has the advantage that the objective importance does not need to be quantified on beforehand. Furthermore, the iterative nature of the proposed Bayesian algorithm allows the HVI factor in (11) to be modified during optimization (e.g., by means of rescaling the objective normalization constants, which governs relative importance of each objective in terms of hypervolume contribution). This allows a decision maker to influence the search procedure dynamically according to his or her preferences, without requiring rework from the designer.

The good performance attained by the proposed Bayesian algorithm in this work together with capabilities demonstrated in previous works leads us to believe that this approach has a large potential to generalize its results to more complex problem settings. For instance, problems involving FE computations (e.g., 3D model of a bridge), could be expected to exhibit similar properties to those of the RC beam problem while further pronouncing evaluation expensiveness.

The proposed algorithm's sensitivity to increasing problem dimensionality remains to be explored. While increased dimensionality is expected to increase problem difficulty for literally any algorithm, Bayesian methods are, in general, especially sensitive to the curse-of-dimensionality problem. Previous works have tackled this issue by learning subspace projections reducing the effective dimensionality (Moriconi et al. 2019), or to decrease time complexity by sparse approximation methods, in turn allowing for a larger number of evaluations (Quiñonero-Candela and Rasmussen 2005). The potential benefits of pursuing such approaches would depend on the actual problem considered.

The approach of this paper to estimate the entire Pareto front will become intractable at some point as the number of objectives increases. In such a case, one may have to resort to actively incorporating the decision makers into the process to learn the decision makers' preferences online; see, e.g., Astudillo and Frazier (2020). However, in most use cases for structural design, the actual number of objectives is usually fairly low.

In the explored case of RC beam optimization, imposed constraints included the number of bar layers, and bending and shear resistances. Naturally, one can expect some correlation in these constraints. Further improvements on the Bayesian algorithm proposed in this paper could hence be to include correlation among the surrogate models, as has been previously suggested (Shah and Ghahramani 2016).

Wu et al. (2019) propose a method for multi-fidelity Bayesian optimization. The multi-fidelity setting has interesting parallels to the structural optimization in that FE simulations of constraints could be evaluated using a variable mesh resolution, hence introducing multiple noise levels of the constraint functions, and should be explored in future work.

Finally, in the problem setting at hand, all input parameters were of ordinal type. In practice, design cases could of course include categorical values (e.g., material types or types of cross sections such as I-beam or T-beam), or even conditional ones, such as dimensions dependent on the choice of cross section. Bayesian optimization has previously been successfully extended to such input domains in the context of hyperparameter optimization domain (Sjöberg et al. 2019), and we expect the results therein to be transferable.

4 Conclusions

In this paper, a state-of-the-art constrained Bayesian optimization algorithm has been adapted to a RC beam optimization problem incorporating multiple objectives. Contrasting previous approaches, the Bayesian algorithm proposed in this work explicitly utilizes that objective functions are cheap to evaluate while constraint functions have expensive evaluations, as constraint function evaluations include expensive numerical computations, which are common characteristics in structural engineering design problems.

The efficiency and applicability of the Bayesian algorithm were studied on a benchmark problem consisting in optimizing a RC beam over eight design parameters with respect to five objective functions covering economic, environmental and social, buildability, and performance aspects. Additionally, design constraints were imposed to ensure that the configuration of the beam could be built and that it satisfied the required bending and shear capacity according to structural design codes. It was shown that the Bayesian algorithm could effectively find high-quality estimated Pareto sets for the considered benchmark problem using a very limited number of constraint function evaluations, and that it outperformed NSGA-II and random search algorithm by a large margin.

The good performance exhibited by the Bayesian algorithm opens up possibilities for its application to more complex cases in the field of structural engineering. The limited number of function evaluations required is particularly interesting to enable performing structural calculations using a finite element analysis software, whose computational cost usually limits the number of possible evaluations and hinders their application in structural design optimization problems.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results In order to facilitate the replication of the results presented in this paper, the structural design benchmark example is described in detail, the values of design parameters are given in Table 1, and the data used for the evaluation of the objectives is detailed in Appendix 1. Additionally, algorithm implementation aspects and parameter settings unspecified in the previous sections are declared for in Appendix 2.

Appendix 1: Data used for objectives evaluation

The unit values per construction material and activity used for evaluating the economic (production cost), environmental and social (aggregated LCA result), and buildability (construction time) objectives are detailed in Table 6.

Appendix 2: Algorithm implementation details

In the hypervolume computations, we have employed normalization of objective values. The objective function minimum and maximum values used per objective are declared in Table 7.

2.1 Bayesian algorithm implementation

For the Gaussian process surrogate models, the pyGPGO-library (Jiménez and Ginebra 2017) was used with the software version given in Jiménez (2019). Default values for the Gaussian processes with associated initial values for hyperparameters and bounds were used in the maximum likelihood estimation process.

In order to optimize the acquisition function, a pattern search algorithm was used. The core idea of the pattern search is that it evaluates neighboring points, chosen randomly, and moves in an ascent direction, while maintaining a tabu list of already evaluated points. Since the HVI is 0 for many inputs (as described in Section 2.5.1), a tie-breaking strategy is devised to allow inter-comparison of such inputs. For inputs when the HVI factor of (11) is equal to 0, this factor is replaced with a convex combination of the objective functions, i.e., $\sum_{i=1}^{m} \alpha_i f_i(\boldsymbol{x})$, where $\sum_{i=1}^{m} \alpha_i = 1$ and each $\alpha_i \geq 0$ is chosen randomly each time the acquisition function is called. The pattern search algorithm always favor a input with HVI greater than zero over an input with a convex combination.

In the implementation, the pattern search is initialized with 25 inputs randomly sampled from the input domain, each one marking the start of a series of evaluations hereby denoted as *trails*. For each trail, the initial input

Table 6 Unit values used for objectives evaluation

	Material/activity	Value	Unit	Source
Material prices	Concrete	$11.36\, f_{ck} + 1585$	(SEK/m^3)	Wikells (2014)[2]
	Steel	14.4	(SEK/kg)	Wikells (2014)
	Form	123.0	(SEK/m^2)	Wikells (2014)
Construction times	Concreting	0.7	(h/m^3)	Wikells (2014)
	Steel placing	$1.42 \cdot 10^{-2} \Phi_s^{-1.415}$	(h/kg)	Ljungkrantz et al. (1997)[3]
	Formwork	1.1	(h/m^2)	Wikells (2014)
Labor costs	All	733	(SEK/h)	Wikells (2014)
LCA[1]	Concrete	$0.0193\, f_{ck} + 0.407$	(m^{-3})	Thinkstep (2019)[4]
	Steel	0.014	(kg^{-1})	Thinkstep (2019)
	Form	0.055	(m^{-2})	Thinkstep (2019)
	Transport	0.00043	(t^{-1} km^{-1})	Thinkstep (2019)
Distance to site	Concrete	30	(km)	
	Steel	100	(km)	
	Form	30	(km)	

[1] LCA values correspond to the aggregated result for the 16 environmental and social impact categories using the normalization and weighting factors defined in the European Product Environmental Footprint (Sala et al. 2017, 2018), and data from GaBi life cycle database (Thinkstep 2019). Material values (concrete, steel and form) concern the life cycle stages A1–A3 (material production), while the transport value is used for stage A4 (transportation to the site)

[2] Equation obtained by linear regression on four available values from C25/30 to C50/60 ($R^2 > 0.995$)

[3] Equation obtained by power regression using the time per ton vs. bar diameter values given in Ljungkrantz et al. (1997) ($R^2 > 0.995$)

[4] Equation obtained by linear regression on six available values from C20/25 to C50/60 ($R^2 > 0.995$)

is perturbed by a single smallest possible step along a randomly selected dimension. The acquisition value of this new input is computed and then compared (according to the precedence rule of the tie-breaking strategy) to the former solution, where the better solution is kept as the reference point for future perturbations. Each trail keeps a tabu list which prohibits the re-visiting of inputs already evaluated in the individual trail and also inputs evaluated in the global evaluations (i.e., inputs used in any of the surrogate models). Using a total budget of 5000 acquisition function calls, each initial trail is allowed 200 calls, and is evaluated by the stepping procedure until either its budget is reached or a local maxima is obtained. If all initial trails have been evaluated and the total budget is not exhausted, trails are evaluated until convergence in the order of best-yet solutions. If the total budget is still not exhausted, new

Table 7 Values per objective function used in computation of normalized hypervolume

Objective index	Max. value	Min. value
f_1	$1.72 \cdot 10^5$	$2.02 \cdot 10^4$
f_2	$8.54 \cdot 10^1$	2.46
f_3	$1.05 \cdot 10^2$	$2.11 \cdot 10^1$
f_4	1.50	$5.00 \cdot 10^{-1}$
f_5	$5.41 \cdot 10^1$	$4.85 \cdot 10^{-3}$

trails are initialized and evaluated sequentially, until the total budget is spent.

2.2 NSGA-II algorithm implementation

The examined NSGA-II algorithm implemented uses the default implementation provided in the Platypus library (Hadka 2015). As crossover operator, the half-uniform crossover (HUX) operator is used. As mutation operator, probabilistic mutation (PM) with distribution index 20.0 is used.

References

Andersson M, Bandaru S, Ng A, Syberfeldt A (2015) Parameter tuning of MOEAs using a bilevel optimization approach. In: Gaspar-Cunha A, Henggeler Antunes C, Coello CC (eds) Evolutionary Multi-Criterion Optimization. Springer, Cham, pp 233–247. https://doi.org/10.1007/978-3-319-15934-8_16

Astudillo R, Frazier P (2020) Multi-attribute Bayesian optimization with interactive preference learning. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, PMLR, vol 108, pp 4496–4507

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(10):281–305

Bernardo J, Bayarri M, Berger J, Dawid A, Heckerman D, Smith A, West M (2011) Optimization under unknown constraints. Bayesian Stat 9(9):229. https://doi.org/10.1093/acprof:oso/9780199694587.001.0001

Biscani F, Izzo D (2019) esa/pagmo2: pagmo 2.10. https://doi.org/10.5281/zenodo.2529931

Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput 16(5):1190–1208. https://doi.org/10.1137/0916069

Calandra R, Seyfarth A, Peters J, Deisenroth MP (2016) Bayesian optimization for learning gaits under uncertainty. Ann Math Artif Intell 76(1):5–23. https://doi.org/10.1007/s10472-015-9463-9

Deb K, Pratap A, Meyarivan T (2001) Constrained test problems for multi-objective evolutionary optimization. In: Zitzler E, Thiele L, Deb K, Coello Coello CA, Corne D (eds) Evolutionary multi-criterion optimization, vol 1993. Springer, Berlin, pp 284–298. https://doi.org/10.1007/3-540-44719-9_20

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE T Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017

European Committee for Standardization (2004) EN 1992-1-1:2004. Eurocode 2: design of concrete structures - part 1-1: general rules and rules for buildings. CEN, Brussels

European Committee for Standardization (2005) EN 1990:2002. Eurocode - basis of structural design. CEN, Brussels

European Committee for Standardization (2019) EN 15804:2012+A2:2019 - Sustainability of construction works - environmental product declarations - core rules for the product category of construction products. CEN, Brussels

Gelbart MA, Snoek J, Adams RP (2014) Bayesian optimization with unknown constraints. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, UAI 2014. AUAI Press, pp 250–259

Hadka D (2015) Platypus-multiobjective optimization in python. https://platypus.readthedocs.io/, accessed 21 January 2020

Imani M, Ghoreishi SF (2020) Bayesian optimization objective-based experimental design. In: Proceedings of the 2020 American Control Conference (ACC)

Inselberg A (1985) The plane with parallel coordinates. Vis Comput 1(2):69–91. https://doi.org/10.1007/BF01898350

International Federation for Structural Concrete (2009) Bulletin 51: Structural concrete - textbook on behaviour, design and performance, 2nd edn. Lausanne

Jahjouh M, Arafa M, Alqedra M (2013) Artificial bee colony (ABC) algorithm in the design optimization of RC continuous beams. Struct Multidiscip Optim 47(6):963–979. https://doi.org/10.1007/s00158-013-0884-y

Jiménez J, Ginebra J (2017) pyGPGO: Bayesian optimization for python. J Open Source Softw 2(19):431

Jiménez J (2019) pyGPGO Python Package. https://github.com/josejimenezluna/pyGPGO/tree/0c665304623d638ac01a2674eccc3e4699dda40d, accessed 20 January 2020

Ljungkrantz C, Möller G, Petersons N (1997) Betonghandbok. Arbetsutförande : projektering och byggande [Concrete handbook. Work execution: planning and construction], 2nd edn. Svensk byggtjänst, Sweden, https://byggtjanst.se/bokhandel/kategorier/konstruktion-och-stomme/Betonghandbok---Arbetsutf.-Utg-2/

Maher ML (1987) Expert systems for structural design. J Comput Civ Eng 1(4):270–283. https://doi.org/10.1061/(ASCE)0887-3801(1987)1:4(270)

Mathern A, Mara V, Patiño Quinchía J, Poneta P, Hermosilla C (2013) Flexible assembly methods, Project report D4.19. PANTURA Project - flexible processes and improved technologies for urban infrastructure construction sites, European Commission 7th Framework Programme

Mergos PE, Mantoglou F (2020) Optimum design of reinforced concrete retaining walls with the flower pollination algorithm. Struct Multidiscip Optim 61(2):1–11. https://doi.org/10.1007/s00158-019-02380-x

Miettinen K (2012) Nonlinear multiobjective optimization, vol 12. Springer, Boston. https://doi.org/10.1007/978-1-4615-5563-6

Moriconi R, Deisenroth MP, Kumar K (2019) High-dimensional bayesian optimization using low-dimensional feature spaces. arXiv:190210675

Nakayama H (2005) Multi-objective optimization and its engineering applications. In: Branke J, Deb K, Miettinen K, Steuer RE (eds) Practical approaches to multi-objective optimization, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Dagstuhl, no. 04461 in Dagstuhl Seminar Proceedings, http://drops.dagstuhl.de/opus/volltexte/2005/234

Parrish KD (2009) Applying a set-based design approach to reinforcing steel design. Dissertation, University of California, Berkeley

Penadés-Plà V, García-Segura T, Yepes V (2019) Accelerated optimization method for low-embodied energy concrete box-girder bridge design. Eng Struct 179:556–565. https://doi.org/10.1016/J.ENGSTRUCT.2018.11.015

Quiñonero-Candela J, Rasmussen CE (2005) A unifying view of sparse approximate Gaussian process regression. J Mach Learn Res 6:1939–1959

Rasmussen CE (2004) Gaussian processes in machine learning. In: Bousquet O, von Luxburg U, Rätsch G (eds) Advanced lectures on machine learning. ML 2003. Lecture notes in computer science, vol 3176. Springer, Berlin, pp 63–71. https://doi.org/10.1007/978-3-540-28650-9_4

Sala S, Crenna E, Secchi M, Pant R (2017) Global normalisation factors for the environmental footprint and life cycle assessment. Publications Office of the European Union, Luxembourg. https://doi.org/10.2760/88930

Sala S, Cerutti AK, Pant R (2018) Development of a weighting approach for the environmental footprint. Publications Office of the European Union, Luxembourg. https://doi.org/10.2760/945290

Shah A, Ghahramani Z (2016) Pareto frontier learning with expensive correlated objectives. In: Proceedings of the 33rd International Conference on Machine Learning, JMLR, vol 48, pp 1919–1927

Sjöberg A, Önnheim M, Gustavsson E, Jirstrand M (2019) Architecture-aware Bayesian optimization for neural network tuning. In: International Conference on Artificial Neural Networks. Springer, pp 220–231, https://doi.org/10.1007/978-3-030-30484-3_19

Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp 2951–2959

Snoek J (2013) Bayesian optimization and semiparametric models with applications to assistive technology. Dissertation, University of Toronto

Tamaki H, Kita H, Kobayashi S (1996) Multi-objective optimization by genetic algorithms: a review. In: Proceedings of IEEE international conference on evolutionary computation, pp 517–522. https://doi.org/10.1109/ICEC.1996.542653

Thinkstep (2019) GaBi ts - Software for life cycle assessment. https://www.gabi-software.com, accessed 14 January 2020

Torczon V (1997) On the convergence of pattern search algorithms. SIAM J Optim 7(1):1–25. https://doi.org/10.1137/S1052623493250780

Vlennet R, Fonteix C, Marc I (1996) Multicriteria optimization using a genetic algorithm for determining a Pareto set. Int J Syst Sci 27(2):255–260. https://doi.org/10.1080/00207729608929211

Wikells (2014) Sektionsfakta-NYB 14/15 - Teknisk-ekonomisk sammanställning av byggdelar [Technical-economic overview of building components]. Wikells byggberäkningar, Växjö. https://byggtjanst.se/bokhandel/kategorier/projektering-upphandling/sektionsfakta-nyb-18191/

World Economic Forum (2016) Shaping the future of construction - a breakthrough in mindset and technology. Geneva

Wu J, Toscano-Palmerin S, Frazier PI, Wilson AG (2019) Practical multi-fidelity Bayesian optimization for hyperparameter tuning. arXiv:190304703

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE T Evol Comput 3(4):257–271. https://doi.org/10.1109/4235.797969

Zuluaga M, Sergent G, Krause A, Püschel M (2013) Active learning for multi-objective optimization. In: Proceedings of the 30th International Conference on Machine Learning, PMLR, vol 28, 462–470