

Active Learning in Multi-objective Evolutionary Algorithms for Sustainable Building Design

Siamak Safarzadegan Gilan
Georgia Institute of Technology
Atlanta, GA
safarzadegan@gatech.edu

Naman Goyal
Georgia Institute of Technology
Atlanta, GA
ngoyal32@gatech.edu

Bistra Dilkina
Georgia Institute of Technology
Atlanta, GA
bdilkina6@gatech.edu

ABSTRACT

Residential and commercial buildings are responsible for about 40% of primary energy consumption in the US. The design of a building has tremendous effect on its energy profile, and recently there has been an increased interest in developing optimization methods that support the design of high performance buildings. Previous approaches are either based on simulation optimization or on training an accurate predictive model to replace expensive energy simulations during the optimization. We propose a method, suitable for expensive multiobjective optimization in very large search spaces. In particular, we use a Gaussian Process (GP) model for the prediction and devise an active learning scheme in a multi-objective genetic algorithm to preferentially simulate only solutions that are very informative to the model's predictions for the current generation. We develop a comprehensive and publicly available benchmark for building design optimization. We show that the GP model is highly competitive as a surrogate for building energy simulations, in addition to being well-suited for the active learning setting. Our results show that our approach clearly outperforms surrogate-based optimization, and produces solutions close in hypervolume to simulation optimization, while using only a fraction of the simulations and time.

CCS Concepts

•Applied computing → Multi-criterion optimization and decision-making;

Keywords

Building Design; Multi-objective optimization; Active learning

1. INTRODUCTION

Residential and commercial buildings are responsible for about 40% of primary energy consumption in the United States, and reducing this consumption will play an important role in taking practical steps toward a sustainable society and the reduction in usage of the corresponding non-renewable resources [23, 36]. Enhancement of the energy efficiency of buildings is certainly a key task

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '16, July 20 - 24, 2016, Denver, CO, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4206-3/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2908812.2908947>

contributing to an immediate reduction of energy consumption and carbon emissions [40]. In particular, the design of the building has a major impact on its energy footprint. There are several factors affecting the building energy efficiency, such as floor plan design, building orientation, construction materials, daylight and solar control measures, and activity-related parameters. These have to be considered during the conceptual stage of a project, since any attempts at improving energy efficiency in the later stages could be more costly or impossible. Several software packages have been developed to accurately simulate building energy consumption (e.g. EnergyPlus, DOE-2, and Green Building Studio) and to support decision makers at the design phase to produce more energy-efficient structures. However, in reality, simulation tools are mainly used to validate the performances of the final design of a building rather than exploring multiple design possibilities [11].

Recently, optimization approaches have been proposed to support building design. To achieve optimal energy performance and reduce manual efforts of exploring design options, several optimization techniques, such as Genetic Algorithms [4, 20, 34, 32, 21, 2], have been proposed. Most computational tools for building design optimization support workflows, which directly combine genetic algorithms with accurate building simulation software [4, 20, 32, 34]. The major disadvantage of such approaches is that simulations are computationally expensive. One way to address this is to replace the expensive simulations with fast surrogate methods for estimating the energy consumption of the proposed design. A couple of recent papers have investigated the use of genetic algorithms coupled with a predictive model of energy consumption. The predictive model is pre-trained on simulated data, and is used to quickly evaluate candidate solutions without directly interfacing with the simulation during search. [21] combined a feed-forward neural network model with NSGA-II [8] genetic algorithm to optimize building design for energy consumption and comfort. [2] followed a similar approach based on a neural network and a genetic algorithm to optimize decisions about retrofitting a building. Although these approaches already improve over a simulation-optimization strategy, the use of neural networks as the predictive model necessitates large amounts of training data for accurate prediction over many design variables. Furthermore, approaches based on a pre-trained surrogate model that remains static during the optimization phase can easily suffer from bad predictive performance as the search uncovers configurations that are significantly different from the training data.

The computational short comings described above are common to all large-scale multi-objective expensive black box optimization problems, which arise in many application areas. We propose a novel method that integrates the machine learning and optimization components more tightly, based on a specific form of *active*

learning to improve the surrogate model during evolutionary optimization. To achieve savings in computation, we train our predictive model on a small set of simulated samples. However, during the optimization, additional configurations are simulated and added to the training data. In particular, we use Gaussian Process (GP) model for prediction and a multi-objective genetic algorithm for the optimization. In contrast to previous approaches that usually pick individuals that are most uncertain, at each new generation, our active learning strategy selects to simulate the offspring that resolve the most uncertainty about the remaining offspring, leveraging GP’s properties. While our proposed framework is applicable in the context of any multi-objective population-based optimization approach, we present an instantiation of our method using the well-known NSGA-II algorithm.

We develop a comprehensive and publicly available benchmark for building design optimization, consisting of a realistic commercial building with many design parameters, along with the corresponding energy consumption and cost. We evaluate 5 machine learning approaches that have been proposed for building energy prediction on our dataset, and show that the Gaussian Process model is a competitive predictive approach to building energy consumption, in addition to being well-suited for the active learning setting. We compare our approach against the previously proposed 2-stage approach of (passive) learning and optimization, as well as against a baseline where each configuration explored during search is simulated. Our results show that our approach produces solution with competitive Pareto front hypervolume, while using only a fraction of the simulations.

2. BACKGROUND

2.1 Multi-objective optimization

We formulate the building design problem as a Multi-Objective Optimization (MOO) problem, where we try to simultaneously optimize n objective functions f_1, f_2, \dots, f_n over a design space E . We use the notation $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$ to denote the vector of all objective function values on input $x \in E$. We also use $\mathbf{f}(E)$ to denote the objective space of the MOO problem.

Pareto Frontier. Since there is no single optimal solution of a MOO problem, the objective is to find the Pareto frontier which gives the best compromises amongst the multiple objectives. A point $y'' \in \mathbf{f}(E)$ is preferred to $y' \in \mathbf{f}(E)$, written as $y'' \succ y'$, if y'' is no worse than y' in any objective and is better in at least one. The Pareto frontier is then given as:

$$\mathbf{f}(\mathbf{P}) = \{y' \in \mathbf{f}(E) : \{y'' \in \mathbf{f}(E) : y'' \succ y', y'' \neq y'\} = \emptyset\}$$

Hypervolume: a solution quality metric for the Pareto front. A fundamental problem with the Pareto front solution of a MOO problem is to compute the quality of the generated solution. One quality metric that is often used is the Hypervolume metric [42], i.e. the volume enclosed by the Pareto frontier in the objective space. The hypervolume of Pareto front P is the volume enclosed by the origin and $\mathbf{f}(P)$, assuming all the f_i to be non-negative (that can be easily established by shifting if the minimum is known). The hypervolume of an arbitrary set $S \subset E$ is defined similarly after all dominated points in S have been removed. When all the objectives are to be minimized (which can be established by negating any maximizing objective), a solution set with a *smaller* hypervolume is preferred over other solution sets.

MOO solution approaches. Multi-objective optimization has always been an active area of research due to its applicability in real-world problems and the inherent challenges it brings about.

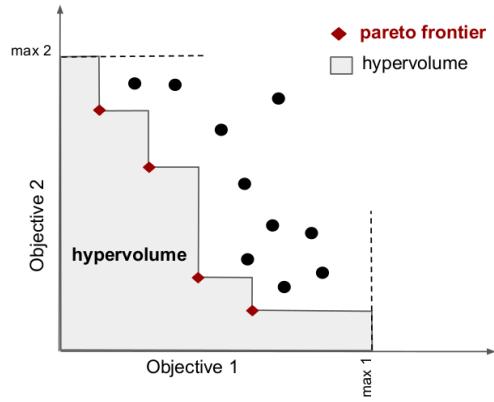


Figure 1: Hypervolume indicator and Pareto frontier for a two objective (both minimization) optimization problem. For minimization problem, the lower hypervolume would indicate better Pareto solution set.

Population-based evolutionary algorithms such as NSGA-II [8] have proven to be extremely effective for this class of problems. In addition to having multiple objectives, real engineering optimization problems do not typically have analytical objective functions, and need rather time-consuming simulations or experiments to evaluate the objectives, adding yet another layer of complexity. Such problems are referred to as expensive black-box optimization problems. Using simulation optimization, which leverages methods for multi-objective optimization directly by simply evaluating each candidate solution using the true (and expensive) objective functions, is usually prohibitive in such settings. Instead, surrogate-based optimization approaches have been proposed (see [15] for a review).

Surrogate functions are computational models constructed with data, collected by running (hopefully few) expensive evaluations of the true objective/fitness function. In the simplest form of surrogate-assisted evolutionary optimization, the search proceeds using solely the surrogate for all evaluations, assuming it can provide sufficiently accurate fitness evaluations. However, serious limitations of this assumption have highlighted the need for model management, which is mainly achieved by using the surrogate together with the real fitness function. One popular way of using surrogates for fitness evaluations is on the individual level, where the real fitness function is used for fitness evaluations for some of the individuals in a generation. Hence, selection is based on a mix of surrogate and real-fitness values among the offspring candidates.

How to choose which of the individuals are to be evaluated using the true expensive objectives is a key research question that has been addressed in many different ways. Three main groups of ideas are to evaluate 1) the individuals that potentially have a good fitness function, 2) a representative set of individuals with respect to the population, or 3) those that have high degree of uncertainty in their surrogate prediction. The idea that evaluation of individuals with uncertain surrogate prediction may be effective in improving the approximation accuracy of the surrogate is similar to the idea of active learning [6]. The estimation of the prediction uncertainty can be achieved with different methods, such as the average distance to the closest data samples used for constructing the surrogate, or using an ensemble of surrogate models. The most often used surrogate model for estimating model uncertainties are Gaussian Processes [26], which have been employed in numerous surrogate-assisted evolutionary approaches [38, 10, 16]. However, choosing to evaluate individuals simply based on their uncertainty

is quite myopic. Ideally, one wants to make sure that the surrogate makes good predictions on all individuals, and especially locally on the ones in the current population.

2.2 Gaussian Processes

Among Bayesian tools that are applicable to real-world statistical problems, Gaussian process (GP) modeling has been shown to be a powerful approach for inference over functions and a practical probabilistic kernel machine, as it can inherently quantify the uncertainty associated with prediction due to its probabilistic nature (unlike other kernel methods like SVM). In many applications, GPs perform competitively to other regression approaches, including in our motivating application in building design as we will show in section 4.2.

A Gaussian Process [26], as a stochastic process, provides a powerful tool for probabilistic inference on an infinite-dimensional space of functions, $\{f(x)\}$, where any finite collection of random variables have a joint multi-variate Gaussian distribution, and the random variables in GP represent the value of $f(x)$ at x . Generally, GP is classified as a non-parametric supervised learning method defining a prior on the space of functions, which could be written as $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$, and is defined by its mean $m(x)$ and covariance function $k(x, x')$. Given the training dataset $\{X_{tr}, Y_{tr}\} = \{(x_i, y_i), i = 1, \dots, n\}$, where n is number of samples, the prediction (mean value) and confidence (standard deviation) for a new data point x' are calculated as:

$$\hat{y}(x') = K(X_{tr}, x')C^{-1}Y_{tr} \quad (1)$$

$$\sigma_{\hat{y}(x')}^2 = k(x', x') - K(X_{tr}, x')C^{-1}Y_{tr} \quad (2)$$

In the above equations, C is the covariance matrix of the training dataset and $K(X_{tr}, x') = [k(x_1, x'), \dots, k(x_n, x')]$ is the vector of covariances between the training data and the data point x' .

Common covariance functions are linear, squared exponential, γ -exponential, and Matérn with automatic relevance determination (ARD). Matérn covariance functions with ARD [1], where the degree of Matérn covariance function is $d = \{3, 5\}$, simplifies to:

$$k_d(x, x') = g_d(\sqrt{dr})\exp(-\sqrt{dr}) \quad (3)$$

where $g_3(z) = 1 + z$, $g_5(z) = 1 + z + z^2/3$, and r defines the distance between the two points as follows: $r = \sqrt{(x - x')^T P (x - x')}$, where P is a diagonal matrix with ARD parameters $\{\ell_1, \dots, \ell_D\}$ and D is the data input dimension.

GPs are inherently dense representations and need all the training data set for inference. Exact inference on GP for a data set of size n has time complexity $O(n^3)$ and space complexity $O(n^2)$, due to size of the kernel matrix. However, there have been several approximate or online methods proposed recently to overcome the computational problem of GPs, making it applicable to large-scale problems as well [24, 13]. Many of these methods approximate the process by sparsifying the GP and keeping the rank of covariance matrix constant by learning basis vectors. Other approaches include partitioning the dataset into local groups, and building a committee of expert GP models to tackle the computation problem [29]. Both of these efficient variations on the original GP can potentially significantly reduce the computational burden of our proposed framework and enhance its scalability to large-scale real-world applications such as sustainable building design.

3. PROPOSED METHOD

In order to facilitate fast building design optimization that does not require many simulations, we propose a novel individual-level

model management in GP-assisted evolutionary algorithms for large-scale expensive black-box optimization. First, the predictive model is pre-trained with a small number of simulated samples. Then during the optimization some offspring are carefully selected to be simulated and added to the training dataset.

Active Learning with Gaussian Processes

The main goal of active learning is to query points which are the most informative to the learning algorithm. The uncertainty estimate at test points offered by GP has been used as the active learning selection criteria in numerous settings. However, the information it maximizes is local to the single selected test point. Instead, here we employ a more effective active learning strategy, based on selecting the test point which resolves model uncertainty on other test points we need to predict on. Since all these meta-heuristic evolutionary optimization algorithms are iterative and they gradually improve the quality of the populations, the overall performance of the surrogate model contributes more to the final performance of the optimization framework, than labeling few of the most uncertain points.

Specifically, GPs allow us to directly evaluate the effect of adding a new data point on the overall prediction accuracy of the model. [27] studied a method for active learning with GPs based on the minimization of the generalization error. The goal is to choose the query \hat{x} that when added minimizes the overall variance of the estimator. The variance on a reference data point ξ after adding a query candidate \hat{x} to the training dataset is defined by $\sigma_{\hat{y}(\xi)}^2(\hat{x})$. Therefore, the change in variance is:

$$\begin{aligned} \Delta\sigma_{\hat{y}(\xi)}^2(\hat{x}, X_{tr}) &= \sigma_{\hat{y}(\xi)}^2 - \sigma_{\hat{y}(\xi)}^2(\hat{x}) \\ &= \frac{(K(X_{tr}, \xi)^T C^{-1} K(X_{tr}, \hat{x}) - k(\hat{x}, \xi))^2}{(k(\hat{x}, \hat{x}) - K(X_{tr}, \hat{x})^T C^{-1} K(X_{tr}, \hat{x}))} \end{aligned} \quad (4)$$

In the above equation, C is the covariance matrix of the training dataset and $K(X_{tr}, s) = [k(x_1, s), \dots, k(x_n, s)]$ represents the vector of covariances between the training data and a sample s . Therefore, $K(X_{tr}, \xi)$ and $K(X_{tr}, \hat{x})$ represent the training data covariance with the reference point ξ and the query candidate \hat{x} , respectively. The overall impact of adding \hat{x} is measured as the average of $\Delta\sigma_{\hat{y}(\xi)}^2(\hat{x}, X_{tr})$ over a set of reference points. High value means high reduction in variance, and so such candidate points should be preferred for active labeling. If the search space is small, the reference set could be the whole search space, but this is rarely the case in real applications. In the context of evolutionary algorithms, the reference set has another very natural realization. At each generation of the algorithms we need to make predictions for the new cohort of offspring, and these predictions will play a critical role in selecting who survives to the next generation. Therefore, we propose to use the current offspring as the reference set for our active learning selection. In other words, we exploit GP to minimize the mean estimated posterior variance over the input distribution of interest, which is the offspring population of each iteration.

Algorithm

We exemplify our model management in the context of non-dominated sorting genetic algorithm (NSGA-II) [8], however any evolutionary algorithm can benefit from this scheme. The algorithm proceeds in generations, where a new population of candidate configurations is produced at each step. We evaluate the effect of adding each candidate to the training dataset by Eq. 4 using the rest of the new candidate solutions as reference points. When there are more than one expensive objective functions, we have a GP surrogate for

Algorithm 1: Optimization coupled with active learning using NSGA-II and GPs.

```

input :  $\{X_{tr}, Y_{tr}\} = \{(x_i, y_i), i = 1, \dots, n\}$ : samples used
       for pre-training GP
        $n_p$  = population size
        $n_{mg}$  = maximum number of generations
        $O$  = objective functions
        $\delta$  = threshold for active query selection

1 Initialize  $P_0$  population of size  $n_p$ 
2  $X_{tr} \leftarrow X_{tr} \cup \text{ActiveQuery}(P_0, X_{tr})$ 
3 Retrain GPs using  $X_{tr}$ 
4 Evaluate  $P_0$  using GPs
5 for  $p \leftarrow 1$  to  $n_{mg}$  do
6   Generate new population  $P_p$  of size  $n_p$ 
7    $X_{tr} \leftarrow X_{tr} \cup \text{ActiveQuery}(P_p, X_{tr})$ 
8   Retrain GPs using  $X_{tr}$ 
9   Evaluate  $P_p$  using GPs
10  Combine  $P_p$  and  $P_{p-1}$ 
11  Extract new population using fast non-dominated sorting
12 end for

13 Function  $Q = \text{ActiveQuery}(P, X_{tr})$ :
14    $Q \leftarrow \emptyset$ 
15   for  $\hat{x} \in P$  do
16      $R \leftarrow P - \{\hat{x}\}$ 
17     for  $\xi \in R$  do
18       | Compute  $\Delta\sigma_{\hat{y}_k(\xi)}^2(\hat{x}, X_{tr}), \forall k \in O$ 
19     end for
20     if  $\text{AVG}_{\xi \in R} [\otimes_{k \in O} [\Delta\sigma_{\hat{y}_k(\xi)}^2(\hat{x}, X_{tr})]] > \delta$  then
21       |  $Q \leftarrow Q \cup \hat{x}$ 
22     end if
23   end for

```

each and hence variance reduction in each of the objectives. We can combine, denoted by \otimes , the variance reduction across objectives in several alternative ways such as the product or one of the forms of generalized mean. In our experiments for building design, we take the maximum across all expensive objectives. We, then, take the average of the multi-objective variance reductions across all reference points. If a candidate solution's average variance reduction exceeds a specified threshold δ , then this configuration is simulated and added to the training set. Not shown in the algorithm, one may wish to further limit the number of simulated individuals per generation, say to no more than 50% of the offspring size. After updating the GP models with the newly labeled individuals, the remaining individuals are evaluated using the predictive model. Finally, selection is applied to form the new population. The pseudocode of optimization coupled with learning is given in Algorithm 1.

Properties like non-parametric presentation, easy model selection through kernel selection, analytical marginal and prediction distribution, etc., make the GP very attractive and efficient for our proposed optimization framework. However, other Bayesian regression methods could also be translated to our framework. One of the reasonable alternatives to GP can be the student-t process proposed by [28]. In addition to all the advantages of GP, the student-t process has the ability to adapt to the situations where there is a non-stationary covariance structure with different surrogate model smoothness in different regions of the design space, and therefore, it requires kernels with different length scales at different regions. Exploring the advantages of student-t process with our framework is a topic for future research.

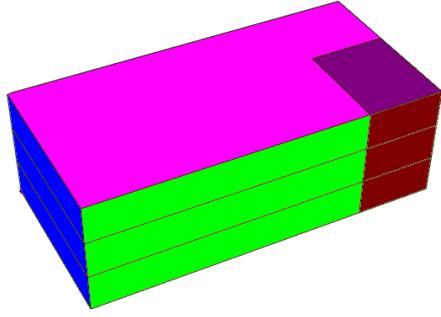


Figure 2: Schematic view of the building geometry of the benchmark model

Related Work

Most related to our work are approaches that use GP-based surrogates for multi-objective optimization. One of the successful solutions proposed so far is the PAREGO model proposed in [16]. In this approach, the multi-objective function is translated to a scalar cost function using Tchebycheff function with different parameter values at each iteration, and a GP is fitted to predict the object function. Even though they have used GP in this paper, any method with confidence prediction capability might be used as well. In this algorithm, the new point to be evaluated through the expensive simulation is selected based on the expected improvement of the solution. One of the main disadvantages of PAREGO is that the prediction models need to be re-trained from the scratch at each iteration for the subsample of the population, which can easily add up to the error due to the subsampling process and increase the time needed to train the model. However, in our proposed method, the GP can be updated incrementally and efficiently.

[43] presented the active learning-based framework, PAL, for finding Pareto-Optimal solution where the input space is discrete or can be reasonably discretized and subsampled. This approach classifies each design candidate from the search space into either Pareto optimal or not, iteratively. To this aim, they deployed GP trained on a small subsample of the design space to evaluate each candidate and the uncertainty bound associated with it. The main drawback of this algorithm is its use of all data points in the search space to choose the next sample to be added to the training set, which clearly limits the applicability of this method.

4. EXPERIMENTAL RESULTS

4.1 Building Design Problem

We aim to create a benchmark building model with a considerable number of decision variables that closely matches real-case design scenarios. Our benchmark is based on an open-plan side-lit building plan (OD), shown in Figure 2, developed as one of four UK office buildings archetypal models analyzed in [18]. The major drawback of the original model is that many of the design parameters were applied uniformly to the whole building. In practice, designers often assign different building materials, window types, etc., to different building parts in order to meet project requirements. In our benchmark, different parameters were assigned to each of the four individual exterior faces. The parameters and their associated values are presented in Table 1. All the other parameters in the model have been fixed to default values. For the meanings and the details of values, please refer to the description in [18]. Our

Table 1: Parameter list and values for the developed benchmark building model

Parameter	Values
Orientation	{0,45,90,135,180,225,270,315}
Overhang	{none, overhang}
Heating Set Point	{21,22,23}
Cooling Set Point	{23,24,25}
Building Fabric: Floors	{BF1,BF2,BF3,BF4,BF5}
Building Fabric: Roof	{BF1,BF2,BF3,BF4,BF5}
Building Fabric: Walls*	{BF1,BF2,BF3,BF4,BF5}
Glazing Ratio: Walls*	{25,50,75}
Glazing Coating: Walls*	{Non-Reflective, Reflective}

*Assigned to the four exterior walls of the model independently.

benchmark model has a total of 2,916,000,000 possible designs, forming a large search space.

We considered two performance measures for a design: heating energy and cooling energy. Energy consumption was simulated using the EnergyPlus toolbox [7]. EnergyPlus [7] is a complete energy analysis and thermal load simulation program, that is available for free for all main computer platforms. EnergyPlus evaluates the energy consumption of any kind of building defined by the user in an input file specifying the design parameters. Both the heating and cooling energy objectives are returned by the EnergyPlus simulation run. To evaluate different predictive models for energy consumption in our benchmark, we generate two datasets of size 5000 (train) and 1000 (test) samples using Latin hypercube sampling (LHS) [14]. To this aim, we used the jEPlus parametric study toolbox [39] for accelerating the simulation process and generating the input files for EnergyPlus. Further details, the models, scripts and precompiled datasets are made available online (url <http://www.cc.gatech.edu/~bdilkina/buildings/index.html>.)

All the experiments were done on 8 core machine, where whenever possible we performed simulations in parallel on all 8 cores.

4.2 Energy Consumption Prediction

Exploiting data-driven models to predict the energy consumption of buildings from their design variables has drawn the attention of many researchers in recent years. Many variations of supervised regression approaches have been utilized for this purpose (for a survey see [41]): neural networks [22, 21, 35, 37]; support vector machines (SVM) [9, 19], decision trees [36, 31], random forests [30], Gaussian Processes [40, 17], and others.

We first use our benchmark dataset to evaluate several models for predicting energy consumption using the building design parameters as input features: ensemble methods of random forest (RF) [3] and least squares boosting (LSBoost) [12]; neural network (NN) [12]; support vector regression (SVR) [33]; and Gaussian process (GP) [26]. The details of the methods and their parameters are listed in Table 2. Moreover, in the experiments, we used MATLAB’s toolboxes for NN, RF, and LSBoost, the LIBSVM toolbox [5] for SVR, and the GPML toolbox [25] for GP. The performance was evaluated based on root mean square error (RMSE) and correlation coefficient (R^2).

Input variables were normalized to the range [0, 1] for better training performance. The model selection was done with 10-fold cross validation on the 5000 samples, and performance was measured on the additional 1000 samples, reported in Table 3. The Neural Network performs slightly better than GP on RMSE, and they both have the best correlation coefficient. While GP is competitive on performance with big enough training set, it also has a fast learning curve as can be seen in Fig. 3. The learning curve is

Table 2: ML approaches and their parameters

Model	Learning Parameters
NN	Multilayer feed-forward structure with one or two hidden layers of {10,20,30,40,50} neurons, Levenberg-Marquardt back propagation algorithm, hyperbolic tangent sigmoid function and linear transfer function for hidden and outer layers.
RF	Number of trees of {10,20,30,40,50}.
LSBoost	Weak learner of tree, Number of trees of {10,20,30,40,50}.
SVR	Regularization $C \in [2^{-5}, 2^5]$, RBF kernel with $\gamma \in [2^{-15}, 2^3]$, $\epsilon \in [0.01, 1]$.
GP	Covariance function: Matern kernel with $\nu = \{3/2, 5/2\}$ and automatic relevance determination, Gaussian likelihood, exact inference.

Table 3: Accuracy of Heating Energy Prediction

Model	Training		Testing		
	AU	RMSE	R^2	RMSE	R^2
RF	5809	0.9731	5615	0.9718	
LSBoost	4257	0.9841	4364	0.9822	
NN	1405	0.9982	1306	0.9984	
SVR	5791	0.9704	5746	0.9690	
GP	1406	0.9983	1331	0.9984	

obtained by training the GP model based on different number of samples and then measuring its performance on 1000 independent testing samples. These results confirm that GP is an appropriate choice for energy regression in the context of building design.

4.3 Comparison with passive learning and simulation techniques

We conducted a comparative study between simulation-based optimization (Opt-Sim), optimization using predictive models (Opt-PL), and our method, active learning for optimization (Opt-AL). After studying the learning curve for GP, we set the training size for Opt-PL to 2000 as a good compromise between quality and time.

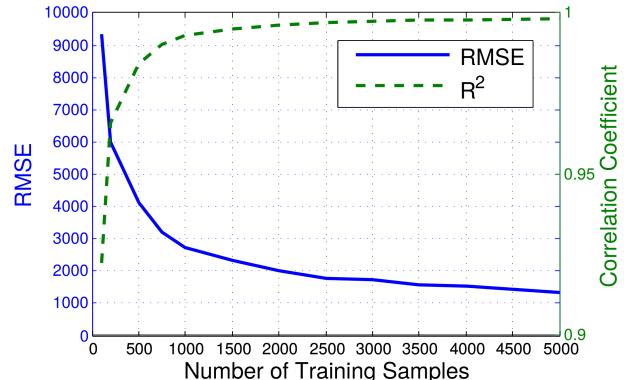


Figure 3: GP learning curve: Number of training samples vs. Accuracy on Heating Energy prediction.

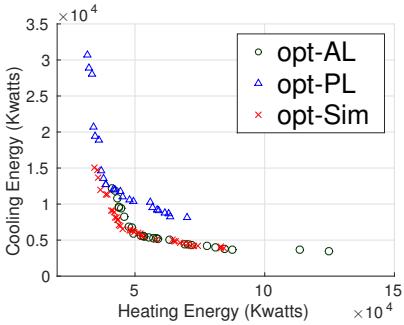


Figure 4: The Pareto front between heating and cooling energy obtained by Opt-Sim, Opt-PL and Opt-AL. Best seen in color.

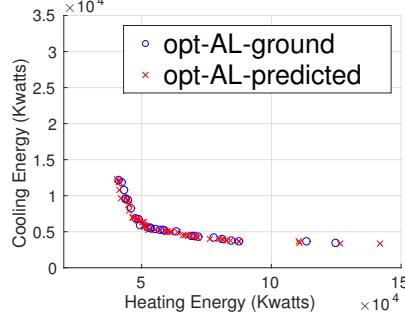


Figure 5: opt-AL last generation predicted and ground Pareto front

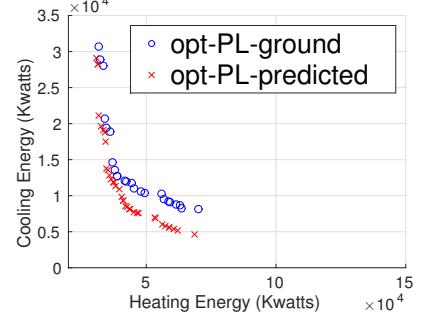


Figure 6: opt-PL last generation predicted and ground Pareto front

Approach	# of Simulations				Time (Min)	Mean Hypervolume (x e ⁹)	Hypervolume diff. from opt-Sim
	Before Opt.	During Opt.	After Opt.	Total			
Opt-Sim	-	2000	0	2000	867	3.570	0.00%
Opt-AL	500	299	50	849	414	3.683	3.17%
Opt-PL	2000	-	50	2050	933	3.716	4.09%
Opt-PL	1000	-	50	1050	462	3.835	7.42%
Opt-PL	500	-	50	550	220	3.926	9.97%

Table 4: Number of simulations and run-times for Opt-Sim, Opt-PL and Opt-AL.

For optimization using NSGA-II, the maximum number of generations was 40 and the population size was set to 50, which implies that for Opt-Sim there could be up to 2000 simulations during the optimization procedure. Bearing in mind that the goal of this paper is to propose a faster method for building energy optimization, for Opt-AL we start with just 500 training samples, we limit the total number of additional queries to 1000 (at most half of each generation), and set the threshold δ for active learning to $0.2 * 10^4$ (a value that depends on the magnitude of the objective function). The Pareto curve of Heating Energy vs Cooling Energy of the final non-dominated solutions obtained using all three approaches is shown in Figure 4. The number of simulations and run-times have been summarized in the Table 4.

Our results indicate that our approach Opt-AL has produced a very competitive Pareto curve in comparison with the other two approaches (in terms of ground truth energy), while taking less than one third of their running time. It can be also observed that the discrepancy between the predicted energy and the simulated (ground truth) energy for the Pareto solutions is much smaller for Opt-AL than for Opt-PL (Figs. 5 and 6). Evaluating the Pareto solutions obtained by Opt-PL trained only on the 500 initial samples used in Opt-AL, we obtain inferior solutions and much greater discrepancy between predictions and ground truth (results omitted). Hence, the active learning component, which simulated an additional 299 building design configurations (on average) during the optimization, is critical to the performance of Opt-AL.

Figure 7 compares all three approaches on the hypervolume metric of Pareto frontier across various number of simulations. In order to vary the number of simulations used by the optimization approaches, we limit Opt-PL and Opt-AL simulations by giving smaller number of initial simulations. The same is not possible for opt-Sim, so we limit the simulation by breaking the generation

earlier for opt-Sim. Results show that our method Opt-AL outperforms Opt-PL, using similar or smaller number of simulations.

4.4 Comparison with PAL

We compared our method against PAL [43] on the building optimization problem. We used PAL’s implementation provided by the authors of paper and ran it for building optimization configuration space. In the PAL paper, the authors have shown the results on problems that have total possible configurations of maximum 1023. It is very small compared to the building optimization problem, which has 2,916,000,000 configurations.

Since PAL classifies all points in the search space as Pareto optimal, non-Pareto optimal, and unclassified based on their evaluations and associated uncertainty bounds using a GP trained on a small subset of the search space, it needs all the design space to be loaded in memory. Thus the memory and CPU requirement make it not possible to use PAL for a problem like building design where design search space is very large.

One alternative is to take small samples from the design space and run PAL over it and hope that the samples contain the Pareto front points. To examine this, we took 5,000 LHS samples from the whole design space and tried PAL on it. While PAL does a very good job at finding Pareto front on these 5,000 samples, Fig. 8 shows that for a problem like our building design it is a very small sample of the whole design space that is far from the Pareto frontier.

Another alternative that we tried was to reduce the design space by fixing some of the design variables that might have lower impact on the heating and cooling energy and limiting the search space to 46,875 which is still much bigger than the problems tackled in the original paper. PAL provides a parameter, ϵ , that controls the number of simulations during search. Higher values of ϵ use fewer simulations during the iterations but generate a high number of pos-

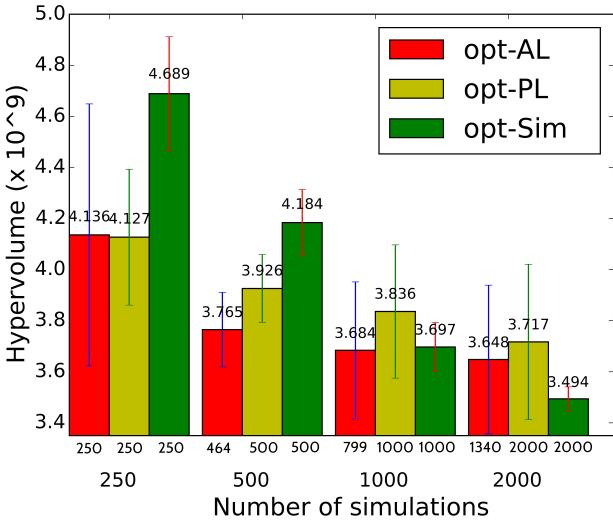


Figure 7: Mean hypervolume (smaller is better) of Pareto front for Opt-AL, Opt-PL and Opt-Sim over 5 runs (with std. error bars). To achieve total simulations close to 250, 500, 1000 and 2500, we set the initial simulation samples used for training to 125, 250, 500 and 1000 for Opt-AL, and to 250, 500, 1000, 2000 for Opt-PL, while for Opt-Sim we limit the number of NSGA generations to 5,10,20 and 40 respectively.

sible Pareto front points with lower confidence. Lower values of ϵ use many more simulations and produce limited Pareto frontier points with higher confidence. Since our goal is to give better design choices to the building designers, we take all the Pareto front points predicted by PAL and simulate each of them and generate the final Pareto solution from them. There was no value of epsilon that resulted in the total number of simulations during iterations and final Pareto points smaller than 2,000. The lowest achieved was $\epsilon = 0.06$ which used 1388 simulations during iterations and predicted 1333 as Pareto front points, much higher than the simulation requirements used by Opt-AL, Opt-PL or Opt-Sim.

5. CONCLUSION

The results presented in this work indicate that combining active learning and optimization approaches is a promising direction for achieving scalable building design optimization. The use of genetic algorithms like NSGA-II allows us to optimize multiple objectives, such as energy and cost. Future extensions to this work will explore how to automatically tune the parameters that guide the rate of active learning. We will explore how the size of the starting training set impacts the final performance. Also we would like to investigate whether other predictive models in addition to GPs could be easily employed under our optimization + active learning framework. We would like to extend our benchmark with other building performance measures, such as environmental impact and comfort.

In this work we have shown results for the building optimization problem, but particular scheme of combining active learning with genetic algorithms is rather generic. It can be easily applied to other similar multi-objective optimization problems that have large decision spaces. Our approach falls in the class of methods that address multi-objective optimization problems, in which evaluating the objective functions is prohibitively expensive. We have also shown that Pareto active learning (PAL) applicability is limited to small search spaces since it chooses the next sample to add to the

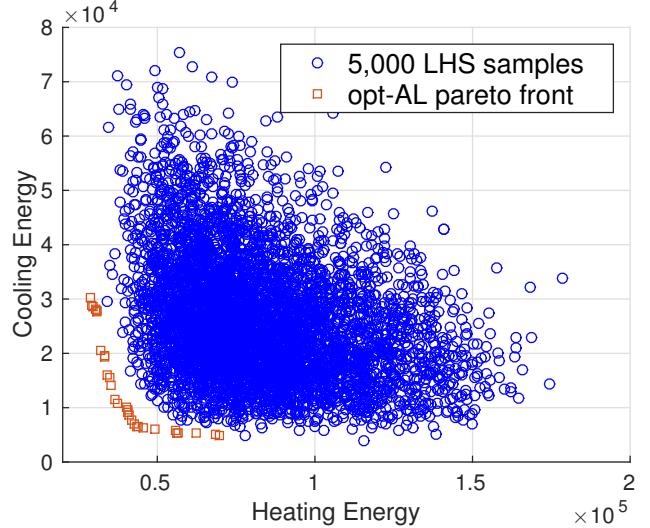


Figure 8: 5000 samples objective space against the opt-AL Pareto front on full design space.

training set needs to evaluate a covariance matrix over all points in the search space. In contrast, our method uses the current population as the reference samples in the active learning selection, and hence can potentially scale up to larger search spaces.

6. ACKNOWLEDGMENTS

The authors would like to thank Mahdi Roobzahani for technical assistance and Kyungki Kim for collaborating in the development of the building design model. B.D. is supported by NSF grant "CompSustNet: Expanding the Horizons of Computational Sustainability" (CCF-1522054).

7. REFERENCES

- [1] M. Abramowitz, I. A. Stegun, et al. *Handbook of mathematical functions*, volume 1046. 1965.
- [2] E. Asadi, M. G. d. Silva, C. H. Antunes, L. Dias, and L. Glicksman. Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application. *Energy and Buildings*, 81:444–456, 2014.
- [3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [4] L. Caldas. Generation of energy-efficient architecture solutions applying GENE_ARCH: An evolution-based generative design system. *Advanced Engineering Informatics*, 22(1):59–70, Jan. 2008.
- [5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [6] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.
- [7] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and buildings*, 33(4):319–331, 2001.

- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. on Evolutionary Computation*, 6(2):182–197, 2002.
- [9] B. Dong, C. Cao, and S. E. Lee. Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings*, 37(5):545–553, 2005.
- [10] M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *Evolutionary Computation, IEEE Transactions on*, 10(4):421–439, 2006.
- [11] F. Flager and J. Haymaker. A comparison of multidisciplinary design, analysis and optimization processes in the building construction and aerospace, 2009.
- [12] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*. Springer, 2 edition, 2009.
- [13] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*, pages 282–290, 2013.
- [14] R. L. Iman. Latin hypercube sampling. In *Encyclopedia of Quantitative Risk Analysis and Assessment*. John Wiley & Sons, Ltd, 2008.
- [15] Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61 – 70, 2011.
- [16] J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *Evolutionary Computation, IEEE Transactions on*, 10(1):50–66, 2006.
- [17] J. Z. Kolter and J. Ferreira. A large-scale study on predicting and contextualizing building energy usage. In *AAAI Conference on Artificial Intelligence*, 2011.
- [18] I. Korolija, L. Marjanovic-Halburd, Y. Zhang, and V. I. Hanby. Uk office buildings archetypal model as methodological approach in development of regression models for predicting building energy consumption from heating and cooling demands. *Energy and Buildings*, 60:152–162, 2013.
- [19] Q. Li, Q. Meng, J. Cai, H. Yoshino, and A. Mochida. Applying support vector machine to predict hourly cooling load in the building. *Appl. Energy*, 86(10):2249–2256, 2009.
- [20] S.-H. E. Lin and D. J. Gerber. Designing-in performance: A framework for evolutionary energy performance feedback in early stage design. *Automation in Construction*, 38:59–73, Mar. 2014.
- [21] L. Magnier and F. Haghigat. Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network. *Building and Environment*, 45(3):739–746, 2010.
- [22] A. H. Neto and F. A. S. Fiorelli. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. *Energy and Buildings*, 40(12):2169–2176, 2008.
- [23] L. Pérez-Lombard, J. Ortiz, and C. Pout. A review on buildings energy consumption information. *Energy and buildings*, 40(3):394–398, 2008.
- [24] A. Ranganathan, M.-H. Yang, and J. Ho. Online sparse gaussian process regression and its applications. *IEEE Transactions on Image Processing*, 20(2):391–404, 2011.
- [25] C. E. Rasmussen and H. Nickisch. The gpml toolbox version 3.4. 2014. <http://www.gaussianprocess.org/gpml/code>.
- [26] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [27] S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: Active data selection and test point rejection. In *DAGM-Symposium*, pages 27–34, 2000.
- [28] A. Shah, A. G. Wilson, and Z. Ghahramani. Student-t processes as alternatives to gaussian processes. *arXiv preprint arXiv:1402.4306*, 2014.
- [29] E. Snelson and Z. Ghahramani. Local and global sparse gaussian process approximations. In *Intl. Conference on Artificial Intelligence and Statistics*, pages 524–531, 2007.
- [30] A. Tsanas and A. Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Build.*, 49:560–7, 2012.
- [31] G. K. Tso and K. K. Yau. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, 32(9):1761–1768, 2007.
- [32] D. Tuhus-Dubrow and M. Krarti. Genetic-algorithm based approach to optimize building envelope design for residential buildings. *Building and Env.*, 45(7):1574–1581, 2010.
- [33] V. Vapnik, S. E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, pages 281–287, 1997.
- [34] W. Wang, R. Zmeureanu, and H. Rivard. Applying multi-objective genetic algorithms in green building design optimization. *Building and Env.*, 40(11):1512–1525, 2005.
- [35] J. Yang, H. Rivard, and R. Zmeureanu. On-line building energy prediction using adaptive artificial neural networks. *Energy and Buildings*, 37(12):1250 – 1259, 2005.
- [36] Z. Yu, F. Haghigat, B. C. Fung, and H. Yoshino. A decision tree method for building energy demand modeling. *Energy and Buildings*, 42(10):1637 – 1646, 2010.
- [37] G. Zemella, D. De March, M. Borrotti, and I. Poli. Optimised design of energy efficient building façades via evolutionary neural networks. *Energy and Buildings*, 43(12):3297–3302, 2011.
- [38] Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *Evolutionary Computation, IEEE Transactions on*, 14(3):456–474, 2010.
- [39] Y. Zhang. Use jeplus as an efficient building design optimisation tool. In *CIBSE ASHRAE Technical Symposium, London*, pages 18–19, 2012.
- [40] Y. Zhang, Z. O’Neil, T. Wagner, and G. Augenbroe. An inverse model with uncertainty quantification to estimate the energy performance of an office building. In *Intl. Building Performance Simulation Association Conference*, 2013.
- [41] H.-X. Zhao and F. Magoules. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, 16(6):3586 – 3592, 2012.
- [42] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pages 862–876. Springer, 2007.
- [43] M. Zuluaga, G. Sergent, A. Krause, and M. Püschel. Active learning for multi-objective optimization. In *Intl. Conference on Machine Learning*, pages 462–470, 2013.