An information driven hybrid evolutionary algorithm for optimal design of a Net Zero Energy House

Scott Bucking*, Radu Zmeureanu, Andreas Athienitis

[a] *Centre for Zero Energy Building Studies, Department of Building, Civil, and Environmental Engineering, Concordia University, 1455 de Maisonneuve Blvd., West, Montréal, QC, Canada H3G 1M8*

Abstract

Building performance simulation (BPS) is a powerful tool to estimate and reduce building energy consumption at the design stage. However, the true potential of BPS remains unrealized if trial and error simulation methods are practiced to identify combinations of parameters to reduce energy use of design alternatives. Optimization algorithms coupled with BPS is a process-orientated tool which identifies optimal building configurations using conflicting performance indicators. However, the application of optimization approaches to building design is not common practice due to time and computation requirements. This paper proposes a hybrid evolutionary algorithm which uses information gained during previous simulations to expedite and improve algorithm convergence using targeted deterministic searches. This technique is applied to a net-zero energy home case study to optimize trade-offs in passive solar gains and active solar generation using a cost constraint.

*Corresponding author

Email addresses: s_bucki@encs.concordia.ca (Scott Bucking), radu.zmeureanu@concordia.ca (Radu Zmeureanu), aathieni@encs.concordia.ca (Andreas Athienitis)

Keywords:
optimization, evolutionary algorithm, information theory, net-zero energy home, solar building

Abbreviations

BPS	Building Performance Simulation
BW	Box-Whisker Plot
DE	Differential Evolution
DHW	Domestic Hot-Water
EA	Evolutionary Algorithm
EUI	Energy Use Intensity
GA	Genetic Algorithm
GenOpt	Generic Optimization Program
HJ	Hooke-Jeeves search
MARR	Minimal Acceptable Rate of Return
MIHEA	Mutual Information Hybrid Evolutionary Algorithm
NPV	Net-Present Value
NZEH	Net-Zero Energy House
PSO	Particle Swarm Optimization
PSOIW	Particle Swarm Optimization Inertial Weight
PV	Photovoltaic Panels

1. Introduction

Building performance simulation coupled with optimization techniques is a powerful tool to identify optimal pathways to improve the energy, cost and environmental performance of new buildings. To reduce energy consumption and maximize solar energy use in new buildings, pivotal design decisions must be made within a narrow time frame before the solidification of the final design. These design-stage decisions commit 80–90% of a building's life-cycle operational energy demand (Ramesh et al., 2010; UNEP-SBCI, 2007). In North America, energy used to construct and operate buildings accounts for some 40% of total energy use (DOE, 2009). Reductions in building energy use has the largest economical greenhouse gas abatement potential estimated to be in the range of 5.3 to 6.7 $GtCO_2-eq/yr$, representing 18 to 35% of the total abatement potential by 2030 (Parry et al., 2007).

Optimization techniques in concert with BPS offer the following benefits: (i) automated search and discovery of potential optimal designs which best achieve desired performance objectives; and (ii) consideration of conflicting system level design trade-offs.

Since each building simulation problem has a unique set of constraints, climate conditions, shape characteristics (Hachem et al., 2011) and occupant usage characteristics, optimization studies must inevitably be performed on a case-by-case basis. Reducing time requirements for optimization studies while improving search resolution is an important research area of BPS.

The utilization of information obtained during the search process still remains unexplored in building optimization research. This paper proposes a data-mining technique within the optimization process. A new algorithm is presented to extract and strategically apply information gained using sub-searches to improve search resolution and expedite algorithm convergence for building simulation problems.

This algorithm is applied to a net-zero energy house (NZEH) design case study. A NZEH generates as much renewable energy on-site as it consumes over a year (Torcellini et al., 2006). Residential buildings in Canada are ideal case-studies since they are sparsely occupied buildings, with relatively low energy use intensity compared to other building types (NRCan-OEE, 2009). They offer large surfaces, such as walls and roofs, for solar panel installation to offset energy consumption. Due to growing interest, an international task-force was established to determine NZE building definitions and simulation approaches (IEA/ECBCS, 2013). NZE building design requires an integrated approach involving passive solar design, improved envelope insulation and air-tightness, renewable energy generation, and control strategies to regulate solar gains. The process of balancing passive solar with energy efficiency and renewable energy generation involves many interacting design aspects and requires a systematic optimization approach to reduce costs and achieve the NZE target.

This paper contains the following sections. Section 2 reviews previous studies related to the application of optimization algorithms in BPS. Section 3 presents the proposed methodology, and the algorithm is applied to a case study in section 4. Discussions of results are presented in section 5, followed by conclusions.

2. Review of Optimization Methodologies Applicable to Building Performance Simulation

In this section, suitable optimization approaches for building simulation studies are reviewed. Few previous researchers have incorporated information obtained during the optimization search process to identify specialized search strategies for building simulation problems. Therefore, a more general overview of methods and algorithms which have proven to be versatile in BPS applications are presented; some search approaches are applied in the proposed methodology.

The following optimization approaches are discussed: (i) deterministic searches,

(ii) population-based searches, and (iii) hybrid search approaches.

A deterministic search operates on individual building representations to identify optimal regions by changing the value of variables using small increments or decrements. Two deterministic searches are discussed: i) hill-climbing search, and ii) Hooke-Jeeves search. These searches are called deterministic as a search operation on the same individual for a given optimization problem will always result in the same search outcome.

In a hill-climbing search, building design variables are incrementally changed to improve an objective function. Typically, the order in which variables are searched and the particular building design representation being searched will greatly affect the search outcome. Renders (1994) recommended integrating a hill-climbing search into the mutation operator of a genetic algorithm or as a forked process interwoven into the search algorithm. Bucking et al. (2010) demonstrated that performing a hill-climbing search on weakly interacting variables at the start of the hybrid algorithm and locking them inside an EA improves algorithm performance and search resolution for solar building optimization studies.

The Hooke-Jeeves (HJ) search (1961), a member of the general pattern search family (Audet and Dennis, 2002), is a deterministic search which explores defined step-sizes in each continuous design variable coordinate. The algorithm selects the design variable, for a given step-size, that best improves fitness. If fitness is not improved, then the process is repeated to find the best step-size improvement in the other design variable coordinates. When no further improvements are made, the step-size is decreased, as previous step-sizes are assumed to be too large. Decreasing step-sizes requires the algorithm to be constantly converging. This disadvantage can be overcome by combining the HJ algorithm with other global searches, as demonstrated by Wetter and Polak (2004).

Population-based algorithms perform operations on populations of representative building designs. Often they are called metaheuristics due to their nature of finding near optimal solutions to a wide range of problems. Two popular population-based search algorithms previously used in BPS were genetic or evolutionary algorithms and particle swarm optimizations.

The first algorithm selected for discussion from the group of population-based algorithms is the Genetic Algorithm (GA), from the Evolutionary Algorithm (EA) family. GAs have become popular due to their ease of implementation and proven ability to solve multi-modal and multi-objective problems. Computational pseudo-evolution was first demonstrated by Goldberg (1989) using biological inspirations. Performing genetic operations, such as mutations and crossovers, on representations in combination with selection operators emulate the 'survival of the fittest' found in biological evolution. Eiben and Rudolph (1999) described members of the EA family as "adaptive systems having a 'basic instinct' to increase the average and maximum fitness of a population." Genetic algorithms are a well-studied metaheuristic. Wang et al. (2006) used a GA to perform a multi-objective optimization using life-cycle cost and exergy on a green building with a polygonal-shaped floor plan. Caldas (2008) used a GA to simultaneously optimize building geometry, energy efficiency and visual comfort. Many modifications exist combining the best elements of other search strategies from the evolutionary algorithm family such as Differential Evolution (DE) (Price et al., 2005). Literature refers to a modified GA by its more general family name, EA. EAs have been scaled to building optimization problems with many design variables. For example, Kämpf et al. (2010a) optimized the solar radiation availability for a grid of buildings. A benefit of EAs is the flexibility to include sub-specialized search strategies. For example, multi-island EAs allow for the population in one generation to be divided into sub-populations, or islands, where specialized sub-population search can be performed. This approach is useful to deconstruct large optimization problems into smaller, easier to solve problems. Ooka and Komamura (2009) utilized a multi-island EA to design, and control an HVAC system for a hospital in Japan.

A Particle Swarm Optimization (PSO) (Kennedy et al., 2001) is fundamentally different from evolutionary cycles found in EAs. Instead of forming a new population of individuals each iteration, the existing population is allowed to gravitate towards other more fit individuals, or particles, in the population. This attraction effect is a form of directed mutation also found in DE (Kennedy et al., 2001). Particles are updated using the best local and global particles in the swarm. PSO competes favourably with other optimization algorithms. For example, Elbeltagi et al. (2005) compared five evolutionary based algorithms and found PSO to outperform the other algorithms for a discrete design problem, with regards to reproducibility of optimal solutions and ability to scale with increasing problem sizes. PSOs are the primary population-based search approach used in the Generic Optimization Program (GenOpt) commonly used in building optimization studies (Wetter, 2011).

More recently, researchers have combined the strengths of population-based and deterministic algorithms into a hybrid approach. Population-based algorithms identify near optimal regions; deterministic searches intensify the search process around near optimal landscapes. Although hybridization can occur at different levels (Feoktistov, 2006), the most common approach is to augment a population-based search with a local deterministic search. The GenOpt tool performs a HJ search on the optimal individual resulting from a PSO (Wetter, 2011). This algorithm was found to have better convergence properties for non-multimodal problems compared to a hybrid DE algorithm (Kämpf et al., 2010b).

Based on the evaluation of the reviewed algorithms, an EA and hill-climbing search was selected for the proposed methodology.

3. Methodology

In this section, two evolutionary algorithms are proposed. EAs allows for the required flexibility to incorporate search strategies based on information obtained during the optimization process. In addition, effective search strategies are borrowed from other optimization algorithms and incorporated into the proposed EA. For example, pseudo-differential gradients originating from DE were explored as a mutation operator. Hill-climbing searches from the deterministic family are examined to perform searches on isolated design variables. The proposed optimization algorithms are discussed in the next section. Before providing details, some mathematical terminology used in the methodology is reviewed.

The formal goal of a minimization study is to find a design variable vector, $\mathbf{x}$, such that:

$$min\{f(\mathbf{x})\} \tag{1}$$

where: $\mathbf{x}$ is the design variable vector $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T$, in design space $\mathbf{X} \subset \mathbb{R}^N$; the objective or fitness function, $f()$, evaluates set of design variables onto an 'objective' vector $\mathbf{y} = (y_1, y_2, \cdots, y_M)^T$ where $f_i \in \mathbb{R}^M$, $y_i = f_i(\mathbf{x})$, $f_i : \mathbb{R}^N \to \mathbb{R}^1$ for $i = 1, 2, \cdots, M$, describes the objective or solution space $\mathbf{Y} \subset \mathbb{R}^M$; $min\{f(\mathbf{x})\}$ is subject to L constraints $g_i(\mathbf{x}) \leq 0$ where $i = 1, 2, \cdots, L$; feasible design vectors set $\mathbf{x}|g_i(\mathbf{x}) \leq 0$ form the feasible design space $\mathbf{X}^*$, and corresponding objective vectors set $\mathbf{y}|\mathbf{x} \in \mathbf{X}^*$ form feasible objective space $\mathbf{Y}^*$; for a minimization problem, a design vector $\mathbf{a} \in \mathbf{X}^*$ is Pareto optimum if no design vector $\mathbf{b} \in \mathbf{X}^*$ exists such that $y_i(\mathbf{b}) \leq y_i(\mathbf{a}), i = 1, 2, \cdots, M$.

The proposed algorithms require discrete variables. This is beneficial as discrete variables improve the convergence properties of the optimization algorithm by shrinking the solution space.

3.1. Proposed Optimization Algorithms

This section describes the proposed optimization algorithms. Clojure (Hickey, 2012), a LISP programming language, was used to integrate mixed optimization strategies into an evolutionary algorithm.

Figure 1 presents the integration of BPS with a typical optimization algorithm. Upper and lower limits of design variables are first defined. These limits define the entire possible set of designs available to the optimization algorithm. Once algorithm and design variables are defined, the optimization algorithm can be initiated. Design representations from the algorithm are converted into simulation files. Simulation files are evaluated using a building simulation tool to determine the performance of each design in question. Simulation results are interpreted and assigned fitness values before entering the algorithm. Databases are used by the optimization algorithm to store relevant simulation information. Building representations in the algorithm are improved upon until a terminal criterion is satisfied.

In the following sections, two algorithms, a modified evolutionary algorithm (section 3.1.1) and an information-driven hybrid evolutionary algorithm (section 3.1.2)

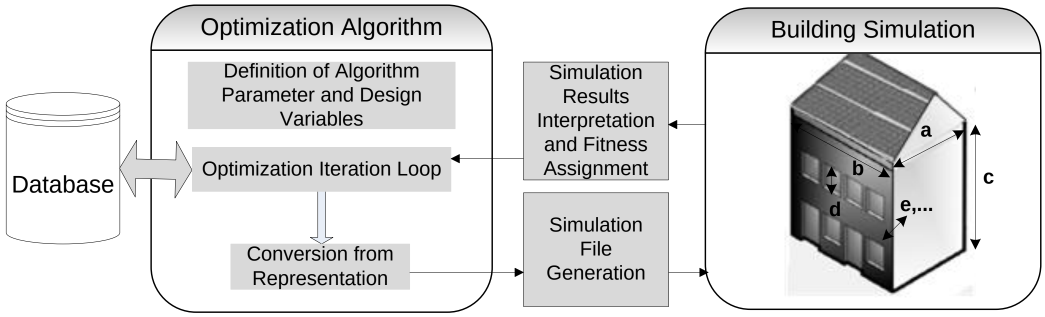

Figure 1: Integration of BPS with an optimization algorithm

are proposed. The performance of both algorithms are benchmarked and discussed in later sections.

3.1.1. *Proposed Modified Evolutionary Algorithm (EA)*

This section describes several improvements to an EA for solar building design optimization problems; a more exhaustive review of EA design can be found in Eiben and Smith (2003). These algorithms are most similar to the Genetic Algorithm described in section 2. However, the more general family name, EA, is used since these algorithms leverage search strategies from other algorithm subclasses such a Differential Evolution algorithms.

A modified EA was developed and configured to estimate algorithm performance. Investigating the performance of the modified EA algorithm provided longitudinal data from which the hybrid-EA can be compared to. Figure 2 presents the evolutionary cycle common to an EA.

In Figure 2, a set of binary genomes, or simplified representations of building designs, form the population. The population is initialized by randomly creating the specified population size and the fitness of each individual is evaluated; in this paper an energy simulation program evaluates building energy use. This population becomes the parent population as it enters the evolutionary cycle. Parent se-

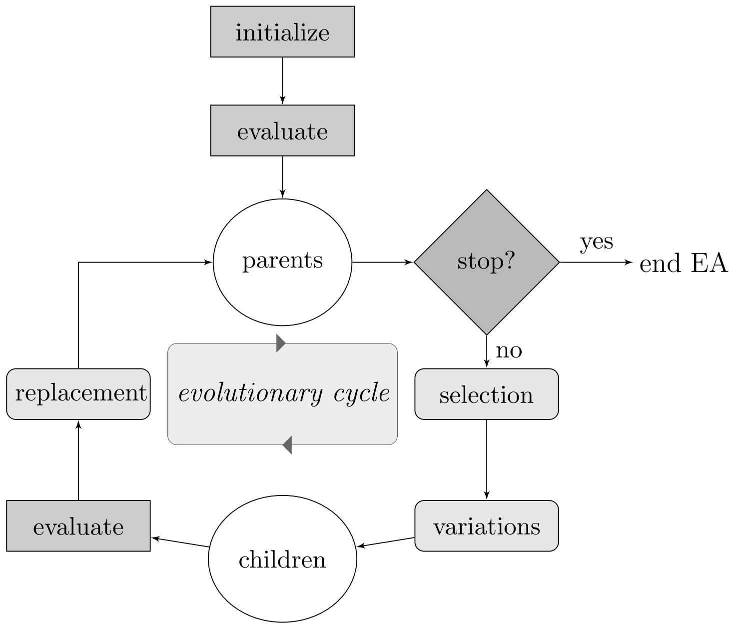

Figure 2: Overview of an evolutionary algorithm

lection is used to select genomes for variation operators such as recombination and mutations. The fitness of new individuals, called children, is evaluated. Survivor selection, or replacement, selects which genomes from the old and new population will survive in the next generation. The process is repeated until a termination criterion is reached, typically a set number of evolutionary cycles sometimes called iterations or generations. Individuals are elite if there exist no other individual in the present population with a better fitness. Elitism is an algorithm feature where a specified number of elite individuals pass to the next generation.

Two types of recombination were used. The first method shared data between two parents on a bit-by-bit basis using a uniform crossover and the second method shared information on a variable-by-variable basis. Uniform recombination on a variable-by-variable basis, shown in Figure 3, was beneficial as it was unlikely that a binary string representing a sensitive design parameter would be transferred from a parent to a candidate child for longer binary representations; from experience, representations with greater than 50 bits for building optimization problems. An algorithm parameter defined the probability of selecting recombination method 1 over method 2. This parameter defined which recombination method was used at each crossover instance. Grey-coded binary representations required that adjacent parameters differ by one bit. This ensured that representations with similar binary encodings had similar design parameters settings.

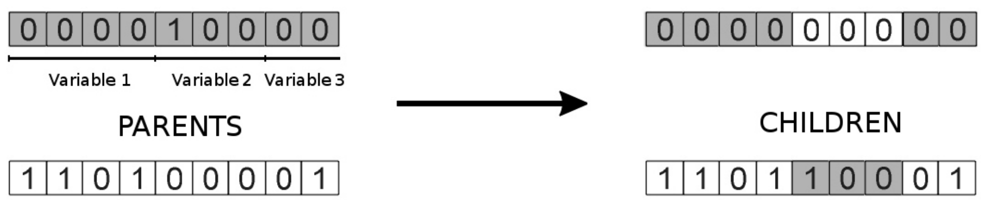

Figure 3: Variable uniform recombination

Two types of mutation operators were explored: (i) a binary mutation operator, and (ii) a differential mutation. A binary mutation operated on a binary genome by flipping bits with a probability of 2% and returned the resulting representation. The diversity of the population can be increased by using a higher mutation rate but at the detriment of possibly losing progress made within evolutionary cycles. The second method used was a differential mutation. Differential operators are the primary evolutionary mechanism found in DE and PSO algorithms. This modified discrete mutation operated on a single parent using gradient information from three unique, randomly selected individuals from the population, see modified version of differential mutation (Storn and Price, 1995) adapted to work within a binary EA. A scaling factor determined the scaling of the gradient difference used in the operator. The mutation rate was identical to the probability of mutation used in the bit-flip operator. After the differential mutation, the resulting continuous representation required rounding to conform to the specified variable step-sizes. Thus, the representation was rounded back into a discrete vector before conversion into a binary format. If values exceeded specified ranges within the differential mutation, they were randomly reset to an allowed value, as recommended by Feoktistov (2006). An algorithm parameter defined the probability of selecting mutation method 1 over method 2.

A *SQLite* database (SQLite, 2012) stored design variable sets, algorithm parameters and building performance metrics such as breakdowns of annual energy consumption from energy simulations. *SQLite* allows for concurrent writes from simultaneous building simulations originating from multi-core and distributed computers. To save computation time, a database query confirmed if an identical representation has been simulated previously before calling the energy simulation tool. SQL queries allowed for the quick recollection of previously simulated design parameter sets and corresponding energy consumption. Database queries were used to data-mine information as described in the following section.

3.1.2. Incorporation of Mutual Information into a Hybrid Evolutionary Algorithm (MIHEA)

The proposed EA from the previous section was augmented with a module to data-mine previous simulation information. This hybrid EA was developed to extract information regarding variable interdependencies and strategically deploy deterministic searches to improve algorithm performance.

EAs are best suited for finding near-optimal solutions and there is no guarantee that searches will resolve to absolute minima. Deterministic searches are better suited for resolving local minima, or search intensification. In building optimization, interactions between variables are treated as a hindrance when they could improve the search process. For example, weakly dependent design variables might be susceptible to deterministic searches. Similarly, if interactions are identified between sub-clusters of design variables, sub-population search strategies might expedite the search process.

A hill-climbing algorithm was used for the deterministic search. A hill-climbing search increments or decrements each design parameter such that fitness is improved. The difficulty lies in identifying which design variables may be weakly interacting and thus susceptible to deterministic searches within the present landscape of the solution space. Mutual information calculations, a concept originating from information theory (Cover and Tomas, 2006), identified weakly interacting variables.

By definition, mutual information is a measure of dependency between two random variables (Cover and Tomas, 2006). Due to its Bayesian roots, the updating of mutual information throughout the optimization search reduces the uncertainty in interaction calculations and builds confidence in selected variables for deterministic searches.

One effective way to extract variable interdependencies is to use the mutual information shared between two design variables denoted by $I(X_i, X_j)$ in equation 2 (Cover and Tomas, 2006), noting that x_i belongs to the set X_i ($x_i \in X_i$) and x_j belongs to the set X_j ($x_j \in X_j$).

$$I(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \cdot \log_2 \left(\frac{p(x_i, x_j)}{p(x_i) \cdot p(x_j)} \right) \tag{2}$$

Probability calculations are made using representations of previously simulated individuals, saved in the database. The functions $p(x_i)$ and $p(x_j)$ are the marginal probability functions of discrete random variables X_i and X_j for a given performance range. Similarly, $p(x_i, x_j)$ is the joint probability for discrete variables X_i and X_j for a specified performance range. From $p(x_i, x_j)$, $p(x_i)$, and $p(x_j)$ the mutual information common to variables X_i and X_j can be calculated.

If variables X_i and X_j are independent, then $p(x_i, x_j) = p(x_i) \cdot p(x_j)$ and $I(X_i, X_j) = 0$, indicating that no information is shared. Larger values of $I(X_i, X_j)$ indicates that more information is shared between variables X_i and X_j. Given these relations, $I(X_i, X_j) \geq 0$.

Finally, equation 3 calculates the total information that design variable X_i shares with all other design variables. Note that deterministic searches work best on variables that are loosely coupled to other variables in the model, i.e. variables with the lowest I_i. The identification and strategic searching of weakly interacting variables is an improvement over population-based optimization searches such as EAs.

$$I_i = \sum_{j=1}^{N} I(X_i, X_j) \qquad \text{where, } j \neq i \tag{3}$$

Information depends on the fitness of the set of design vectors used for the calculation even through fitness is not explicitly used in mutual information calculations. For example, in a building simulation problem, information calculated for a set of design vectors which are evaluated in a range of annual energy use intensity (EUI) of $[800, 1200)$ MJ/m^2 would be different than information calculated from design vectors evaluated within $[400, 800)$ MJ/m^2. Mutual information tends to increase as EUI decreases since building designs with lower energy consumption tend to have more strongly coupled variables to achieve a given performance level.

Figure 4 and Table 1 presents the proposed mutual information hybrid EA

(MIHEA). The evolutionary cycle was identical to Figure 2 except for the addition of a data-mining module which identified weakly-interacting variables and performed a hill-climbing search on the elite individual in the present population. The data-mining of variable interactions was repeated every two generations as determined by the 'datamine?' decision block. After the formation and evaluation of the child population, the elite member of the previous population entered the data-mining module.

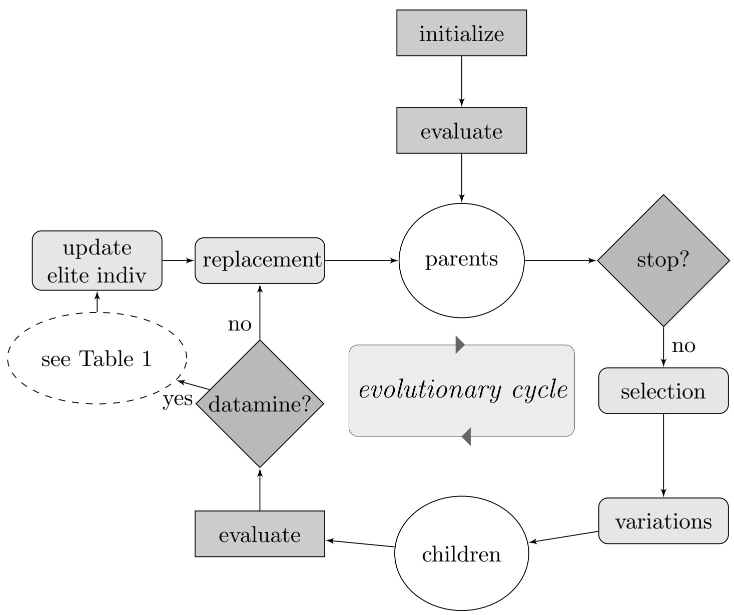

Figure 4: Overview of the proposed mutual information evolutionary algorithm (MIHEA)

Variables were selected for the hill-climbing search using two criteria: (i) the mutual information shared with other design variables, and, (ii) the frequency that each variable had been deterministically searched in all previous generations. Mutual information calculations used at most 100 unique individuals from the database ordered by improving fitness to calculate interactions. The MIHEA selected variables for hill-climbing searches using a tournament selection operator

Table 1 Information-driven deterministic hill-climbing search

Precondition: **a** is a grey-coded binary string and the elite individual in the population

```
function MIDETSEARCH(a)
    a ← binary2discrete(a)                    ▷ Note: a = (a_1, ···, a_N)^T
    data ← getnBestIndiv(n=100)               ▷ Select 100 fittest individuals from database
    I ← calcMI(data)                          ▷ Calculate and sum mutual information
    freq_vars ← calcFreq()                    ▷ Calculate frequency of previously searched variables
    vars ← tournSelect(I, freq_vars)          ▷ Select variables using tournament
    for var ∈ vars do                         ▷ Hill-climbing increments and decrements variable var
        b ← hillclimb_inc_dec(a,var)          ▷ Conduct hill-climbing search
    return discrete2binary(b)                 ▷ Convert discrete representation to binary
```

to identify variables with low total mutual information, see equation 3, and a low frequency of being previously hill-climbed. Tournament operators ensured that the same variables were not searched repeatability every generation but still gave preference to variables that were weakly interacting.

The follow section describes how the proposed algorithms were benchmarked.

3.2. Optimization Algorithm Performance Comparison

Comparing the performance of the proposed optimization algorithms was challenging because both proposed EA and MIHEA algorithms depend on stochastic processes and simulations in this study were conducted in batches on multi-core processors.

The performance of the proposed EA and MIHEA were compared to GenOpt's particle swarm inertial weight (PSOIW) algorithm (Wetter, 2011). Initial populations were randomized for each optimization run to ensure that algorithms were compared under different initial fitness landscapes. Identical design variables and variable step-sizes were used to constrain algorithms to the same solution spaces.

The following measures compared algorithm performance: (i) sensitivity of algorithm configurations, (ii) repeatability studies, and (iii) convergence analysis. The sensitivity study compares the sensitivity of each algorithm to its initial configuration. In addition, this study determines which initial configuration resulted in the best algorithm performance. A repeatability study explores how consistently each algorithm will find optimal or near optimal solutions and the expected fitness value for each algorithm given a single optimization run. The repeatability study also compares algorithms to determine reductions in computational and time requirements. Because the optimization algorithm used in the study depends on stochastic processes, a significant sample of optimization runs is required to conduct the repeatability study. Finally, a convergence analysis compares how quickly each algorithm converges to optimal landscapes from a random initial population.

In the following case study, we compare the performance of the proposed EA and MIHEA to the GenOpt PSOIW algorithm. The proposed EA and MIHEA are also compared separately to estimate the performance improvement from augmenting the EA with information-driven deterministic searches.

4. Case Study: Net-Zero Energy House

The case study involves the optimization of a net-zero energy home (NZEH) located in Montréal, Québec. The energy model was calibrated using monitored data from an occupied near-NZEH located near Montréal (Doiron et al., 2011) to ensure that the model used for the optimization case study reflects the energy balances of a NZEH.

The model was calibrated using the ÉcoTerra house, one of 15 houses in the Canada Mortgage and Housing Corporation EQuilibrium Housing Demonstration Initiative, is a two-story, detached home located in Eastman, Québec, Canada. Alouette Homes prefabricated the home and Natural Resources Canada, Canada Mortgage and Housing Corporation, and Hydro Québec partially funded the project. A balance of passive solar design strategies, roof-top building-integrated photovoltaic panels (PV) and a geothermal heat-pump provided on-site renewable energy generation (Chen et al., 2010a,b). Since the site-NZE definition was selected (Torcellini et al., 2006), the primary energy factor associated with electricity from the grid was not considered.

4.1. Objective function

The objective of the study was to minimize the net-annual energy consumption of a near net-zero energy home. Heating, cooling, fan loads, PV generation and lighting loads were simulated using EnergyPlus (Crawley et al., 2000). The objective function used for this case study was the annual net-electricity consumption of the building, see equation 4,

$$f(\mathbf{x}) = Q_{heat}/COP_H + Q_{cool}/COP_C + E_{elec} - E_{PV} \tag{4}$$

where: $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T$ is a design variable vector; $f(\mathbf{x})$ is the annual net-electricity consumption of the building (kWh); COP is the average annual coefficient of performance of the ground-source heat pump in heating and cooling mode, 3.77 and 2.77 respectively; Q is the annual heating and cooling load (kWh); E_{elec} is the annual electricity consumption in lighting, domestic hot-water (DHW), appliances and plug-loads (kWh) and; E_{PV} is the electricity generated by the roof-top photovoltaic panels (kWh). When $f(\mathbf{x}) < 0$ this implies the net-generation of electricity, or a positive-energy house.

This case study used twenty-six discrete variables, see Table 2. Note that

Table 2: Sample of influential variables for NZEH case study

Variable	Units	Min.	Max.	No. Steps	Description
azi	degrees	-45	45	32	Building orientation/azimuth
aspect	–	0.7	2.2	8	Aspect ratio (south facing width to depth ratio)
wall_ins	m^2K/W	3.5	13.0	8	Effective resistance of wall insulation
ceil_ins	m^2K/W	5.6	15.0	8	Effective resistance of ceiling insulation
base_ins	m^2K/W	0.0	7.0	8	Effective resistance of basement wall insulation
slab_ins	m^2K/W	0.0	2.3	4	Effective resistance of slab insulation
ovr_south	m	0.00	0.45	4	Width of Southern Window Overhangs
pv_area	%	0	90	8	Percent of PV area on roof
pv_eff	%	12	15	4	PV efficiency
roof_slope	degrees	30	45	8	South facing roof/PV slope
wwr_s	%	5	80	8	Percent of window to wall ratio, south (also N,E,W)
GT_s	–	1	4	4	Glazing type, south (also N,E,W)
heating_sp	°C	18	25	4	Heating setpoint
cooling_sp	°C	25	28	4	Cooling setpoint
FT	–	1	2	2	Window Framing Types (1:Wood, 2:Vinyl)
slab_th	m	0.1	0.2	8	Concrete slab thickness
vwall_th	m	0.00	0.35	8	Concrete wall thickness (basement)
zone_mix	L/s	0	400	4	Air circulation rate between thermal zones
infil	ACH	0.025	0.179	8	Natural infiltration rate

variable descriptions are shown for the south orientation only; also, the PV slope is equal to the roof slope. Design of experiment techniques (Goos and Jones, 2011) and previous studies (da Graca et al., 2012; Kolokotsa et al., 2011; O'Brien, 2011; Wang, 2005) aided in identifying influential design variables. Table 3 shows the binary encoding used in the representation for a sample of variables. Equation 5 demonstrates the translation of a partial representation from binary to vector space using the encodings of Table 3.

Table 3: Sample of grey-coded binary representation of design variables

Variable: aspect		Variable: wall_ins		Variable: ceil_ins	
encoding	value, –	encoding	value, m^2K/W	encoding	value, m^2K/W
000	0.7	000	3.50	000	5.60
001	0.9	001	4.86	001	6.94
011	1.1	011	6.21	011	8.29
010	1.3	010	7.57	010	9.63
110	1.6	110	8.93	110	10.97
111	1.8	111	10.29	111	12.31
101	2.0	101	11.64	101	13.66
100	2.2	100	13.00	100	15.00

$$\text{“}\overbrace{\underbrace{010}_{\text{aspect}}\ \underbrace{110}_{\text{wall\_ins}}\ \underbrace{000}_{\text{ceil\_ins}}\ \ldots\text{”}}^{\text{Binary Representation}} \rightarrow \overbrace{(1.3, 8.93, 5.60, \ldots)}^{\text{Vector Representation}} \tag{5}$$

Electric lighting ensured that a minimum illuminance of 200 lx was present in all occupied spaces regardless of the window-to-wall ratio. A heat recovery ventilator with an efficiency of 60%, taken from manufacturer specifications, maintained the ventilation rate at 0.3 air-changes per hour in all occupied spaces. Roller shades were automatically deployed if exterior solar radiation on the exterior window surface exceeded 150 W/m^2 and if exterior temperature on the window exceeded 20 °C. These values ensured that blinds were closed if there was potential for zone overheating.

4.2. Cost Constraint

This section describes the formulation of a cost constraint used in the case-study. A cost constraint required the algorithm to minimize net-energy consumption cost-effectively. Establishing a cost-constraint ensured that algorithm identified cost-effective design trade-offs between passive-solar design and renewable energy generation. If the cost-constraint was exceeded, a barrier function was applied to the objective function and net-energy consumption was set to infinity.

Incremental cost of materials and operational energy costs over the life-cycle is shown in equation 6. A cost constraint of $90,000 was determined based on published cost premiums of NZEHs in Canada (CMHC, 2009). Costs were evaluated over the life-cycle of the building. Hence, initial, operational, and replacement costs are evaluated using the net-present value (NPV) of each design. Cost calculations were performed by post-processing energy simulation results.

$$
\begin{aligned}
g(\mathbf{x}) &= C_{NPV} + E_{NPV} + R_{NPV} - S_{NPV} \\
&\leq \$90,000
\end{aligned} \tag{6}
$$

where: C_{NPV}: is the capital costs of materials and equipment in Canadian dollars; E_{NPV}: is the operational energy costs calculated from energy simulation results; R_{NPV}: is the replacement cost for materials and equipment; and S_{NPV}: is the salvage or residual value using a linear depreciation method.

Materials were scheduled for replacement based on an expected serviceable lifetime (RSMeans, 2011). A marginal electricity rate of 7 cents with an escalation rate of 2.0% was used (Hydro-Québec, 2010). Note that all monetary amounts refer to Canadian dollars. Life-cycle costs were calculated over a 30 year time horizon.

The NPV of each term is calculated using:

$$
\text{NPV} = \sum_{t=0}^{N} \frac{C_t}{(1+a)^t} \tag{7}
$$

where: C_t: is the future net-cash flow at year, t (Net meaning $C_t = cash_{out} - cash_{in}$); a: is the minimal acceptable rate of return (MARR); and N: is the number of years considered in the life-cycle.

Equation 8 specified the minimal acceptable rate of return used for net-present value calculations.

$$
a = (1+r)(1+i) - 1 \tag{8}
$$

where: r is assumed bank rate, a 2.14% return from a 10 year GIC from 2002 to 2012 (Bank of Canada, 2009); i is the annual inflation rate, 2.0% in Canada (Bank of Canada, 2009); a is the calculated minimal acceptable rate of return, 4.18%.

Initial costs were broken down as follows:

$$\begin{aligned} C = {} & \text{wallinsCost} + \text{ceilinsCost} + \text{baseinsCost} + \text{slabinsCost} + \\ & \text{roofCost} + \text{overhangCost} + \text{concrCost} + \text{PVCost} + \\ & \text{winCost} + \text{airtightCost} \end{aligned} \tag{9}$$

where: C is the total material cost; insCost is the cost of wall, ceiling, basement and slab insulation; winCost is the cost of windows based on glazing area; roofCost is the incremental cost of additional roof framing beyond 30 degrees slope; overhangCost is the cost of overhangs; concrCost is the cost of concrete walls and slab for passive thermal storage; PVCost is the cost of PV panels and inverters; and airtightCost is the incremental cost associate with tighter envelopes. These costs were specified from RS-Means data (RSMeans, 2011, 2012).

5. Results and Discussion

To ensure that the EA and PSOIW algorithms were operating properly, the sensitivity of several algorithm configurations were explored. The algorithm settings which resulted in the lowest fitness values were selected for future optimization runs, see run no. 1 of Tables 4 and 5.

Parallelization of building simulations to multi-core processors was used extensively for this study. Parallel simulations can greatly reduce optimization time requirements but do so with diminishing returns, as per Amdahl's law of compu-

Table 4: Parametric run for various algorithm parameters, EA

EA Parameters	Run 1	Run 2	Run 3	Run 4	Run 5
Representation	62 bit binary string	–	–	–	–
Population Size	10	–	–	–	–
Recombination *	60% Method 1	60% Method 1	60% Method 2	80% Method 2	60% Method 2
Mutation ⊙	60% Method 2	60% Method 2	60% Method 1	60% Method 1	80% Method 2
Mutation Prob	2.0%	3.0%	2.0%	2.0%	1.0%
Scaling Factor	0.7	0.5	0.5	0.5	0.1
No. Generations	35	–	–	–	–
Fitness (kWh)	−1481	−1400	−1367	−1104	−934

* Recombination: Method 1: Bit-by-bit Uniform; Method 2: Variable Uniform

⊙ Mutation: Method 1: Bit-by-bit Mutation; Method 2: Differential Mutation

–: No change as compared to Run 1

Table 5: Parametric run for various algorithm parameters, GenOpt PSOIW

GenOpt PSOIW Parameters	Run 1	Run 2	Run 3	Run 4	Run 5
Representation	Discrete	–	–	–	–
Topology	gbest	–	–	–	–
Population Size	10	–	–	–	–
Neighborhood Size	5	–	–	–	–
Cognitive Acceleration	2.8	1.0	3.4	1.8	2.8
Social Acceleration	1.3	1.0	1.5	1.8	2.3
Max Velocity Discrete	4	3	3	4	2
Initial Inertia Weight	1.2	–	1.6	1.4	–
Final Inertia Weight	1.0	–	1.4	1.2	–
No. Iterations	35	–	–	–	–
Fitness (kWh)	−1205	−1003	−1171	−1202	−861

–: No change as compared to Run 1

tational parallelization (Amdahl, 1967). To identify the optimal population size or number of particles, a parallelization simulation study was performed. Figure 5 shows that five simultaneous building simulations allows for an optimal speed-up of four times compared to a sequential simulation strategy. The improvement factor of Figure 5 shows that it is most computationally efficient to conduct energy simulations in batches of five. Since a population of five individuals was insufficient to maintain population diversity within the evolutionary and PSOIW algorithms, a population of ten individuals was selected. Thus, two simulation batches of five individuals were required per algorithm iteration and they were approximately time equivalent to two separate energy simulations.

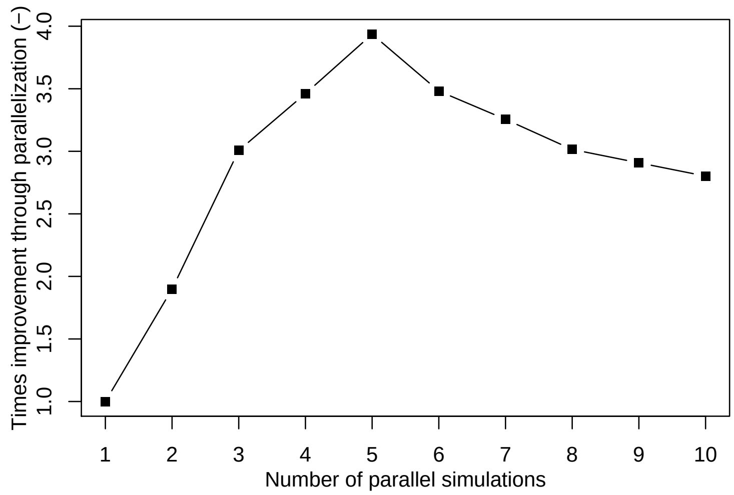

Figure 5: Simulation scalability test on NZEH energy model

Table 6 shows the results of the repeatability study. The results in Table 6 represent the expected fitness value for each algorithm given a single optimization run. This data was built using 20 repeated optimization runs. A sample size of 20 repeated optimization runs yielded 97% statistical power using a p-value of 5%. One standard deviation of data is shown with the average fitness value of optimal solutions.

Table 6: Expected optimal fitness for the proposed EA, proposed MIHEA and PSOIW based on 20 repeated optimization runs, NZEH case study

	Proposed EA	Proposed MIHEA	GenOpt PSOIW
No. of energy simulations	350	364	350
No. of deterministic searches	0	14	0
No. of simulations batches	70	70	70
Algorithm generations/iterations	35	28	35
Mean fitness (kWh)	-1250 ± 172	-1411 ± 119	-1112 ± 213

In Table 6 the expected optimal value of the proposed EA is slightly improved over the PSOIW. A larger disparity was observed when comparing the MIHEA to the PSOIW algorithm. The MIHEA algorithm found designs which had 20% lower fitness values with less variance. Since simulations were conducted in batches on multi-core processors, each algorithm was allowed an equal number of simulation batches rather than an equal number of building simulations. Recall that each batch consisted of five energy simulations. Thus the proposed EA and PSOIW were allowed 70 simulation batches over 35 algorithm iterations. Since MIHEA required one batch of six deterministic searches every other generation the total number of generations was reduced to 28 for a total of 70 simulation batches. MIHEA required 14 more energy simulations than the other algorithms because simulation batches of six were used for deterministic searches instead of batches of five for each algorithm generation. However, the computational requirements are equivalent across all compared algorithms.

Table 7 shows the optimal NZEH parameter sets for the case study. The optimal design shown in Table 7 generated a net of 1491 kWh of electricity and was found using MIHEA. The cost constraint was sufficiently large to allow for the full roof-surface to be covered in PV panels and achieve the NZE target. To achieve this optimal design required integrated design approach. A balance of passive solar strategies, such as: air-tight envelopes (0.025 ACH natural infiltration rate), sufficient wall envelope insulation values (8.56 m^2K/W), appropriate south-facing window-to-wall percentage (48%), sufficient air circulation between zones to distribute solar gains (133 L/s) and sizing of thermal mass (0.25 m central thermal storage wall in basement). Thermal mass allowed storage of solar gains and interacted with solar gain control strategies. Blind control strategies and exterior shading allowed for a larger window-to-wall fraction while maintaining acceptable visual comfort. The identification of trade-offs between passive solar design,

Table 7: Optimization results with MIHEA: Optimal design for case study

VARIABLE	DESCRIPTION	UNITS	OPTIMAL VALUES
azi	Building orientation/azimuth	degrees	0
aspect	Aspect ratio (south facing width to depth ratio)	–	1.3
wall_ins	Effective resistance of wall insulation	m^2K/W	8.93
ceil_ins	Effective resistance of ceiling insulation	m^2K/W	10.97
base_ins	Effective resistance of basement wall insulation	m^2K/W	5.08
slab_ins	Effective resistance of slab insulation	m^2K/W	1.39
ovr_south	Width of Southern Window Overhangs	m	0.34
pv_area	Percent of PV area on roof	%	90
pv_eff	PV efficiency	%	15
roof_slope	South facing roof/PV slope	degrees	45
wwr_s	Percent of window to wall ratio, south	%	48
wwr_n	Percent of window to wall ratio, north	%	10
wwr_e	Percent of window to wall ratio, east	%	10
wwr_w	Percent of window to wall ratio, west	%	10
GT_s	Glazing type, south (also N,E,W)	–	2
FT	Window Framing Types (1:Wood, 2:Vinyl)	–	2
slab_th	Concrete slab thickness	m	0.2
vwall_th	Concrete wall thickness (basement)	m	0.251
zone_mix	Air circulation rate between thermal zones	L/s	133
infil	Natural infiltration rate	ACH	0.025
	Fitness of Individual (kWh)		-1491

energy efficiency and active solar electricity generation is a significant application of the proposed optimization algorithm.

Table 8 shows the deterministic search probability for a sample of design variables from the case study. The search probability is defined as the probability that a given design variable will be searched deterministically within the MIHEA. The probability of selecting a variable for a deterministic search with no prior information is $1/N$, where N is the number of design variables. The actual search probability was calculated by post-processing previous MIHEA optimization runs. The variables with the highest deterministic search probability were the sizing of renewable energy generation, such as PV efficiency, area of PV coverage, roof/PV slope and heating/cooling setpoints. Variables that were rarely selected for deterministic searches were the solar orientation of the building (azimuth) and the aspect ratio (ratio of south facing width to depth ratio). Both variables were tightly coupled to other design variables. The optimization of coupled variables is best handled in the EA.

Table 8: Search probability of design variable within MIHEA for Case Study

Variable	Description	Search Probability (%)
pv_eff	PV efficiency	5.4
pv_area	PV area	5.3
roof_slope	Roof and PV angle	5.1
set_heat	Heating setpoint	4.8
set_cool	Cooling setpoint	4.7
aspect	Aspect ratio	1.6
azi	Building orientation	0.6

Box-whisker (BW) plots compared the distribution of optimization results for each optimization algorithm (Fig. 6). BW plots allow for side-by-side comparisons of the convergence characteristics of each algorithm using five important statistical properties of the optimization datasets. In the BW plots, the dashes represent extremes of the data points (starting point of initial population and final optimized population). The thick line inside the box represents the mean quartile of the set. The lines of the box represent the lower and upper quartiles of the set where 50% of data points reside. The algorithm with the lowest mean fitness has the best convergence properties. Bean plots (Kampstra, 2008) were superimposed onto this Figure to show the individual fitness distribution throughout the search using Gaussian kernel density functions (Scott, 1992). The three dotted lines represent the global maximum, minimum and mean of the dataset. These lines are intended to simplify visual comparison of results.

Figure 6 shows the convergence analysis results for the case study using 20 optimization runs.

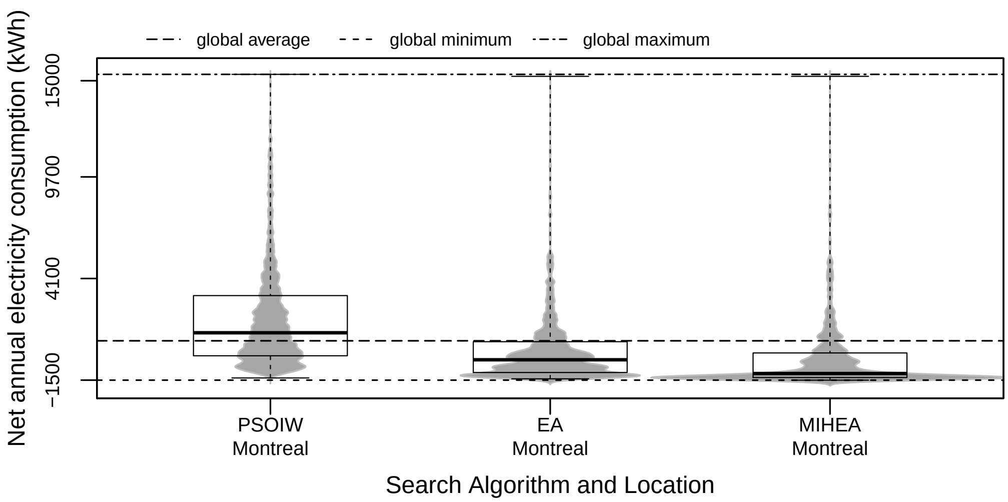

Figure 6: Box-whisker plot for 20 optimization runs

Both EA and MIHEA found better optimal designs and evolved more individuals closer to the optimal landscape than the PSOIW. Note, the best individual from repeated PSOIW optimization was close to the EA solution; however both EAs were able to converge to the near-optimal landscape using fewer fitness evaluations which led to surplus individuals, as illustrated by spiking in the distribution. Note that this spike is absent in the PSOIW algorithm. MIHEA identified optimal solutions using only 22 generations compared to the 35 required by the proposed EA and PSOIW.

6. Conclusions

In this paper a hybrid evolutionary algorithm is proposed for minimizing solar building energy consumption. A net-zero energy house was used as a case-study to demonstrate the algorithm. Optimization approaches are required to identify cost-effective trade-offs between passive solar design and renewable energy generation. The MIHEA algorithm utilized information regarding variable interactions during the optimization process to identify opportunities for deterministic searches. This augmentation is valuable as EAs are strong at optimizing interdependent variables but have difficulties optimizing weakly coupled design variables—a strength of deterministic searches. Results suggest that this approach improves the reproducibility of near optimal solution set while requiring less computational resources.

The proposed MIHEA algorithm is applicable to any problem that involves various strengths of design variable interactions including several weakly interacting design variables. Building energy simulation tools used for performance evaluations of solar buildings, such as ESP-r or EnergyPlus, are ideal case studies as they involve solving sets of sparse matrices (Clarke, 2001) or iterative solvers applied to loosely-coupled heat balance equations (DOE, 2011). However, the proposed algorithm may be useful for other fields. Furthermore, using mutual information calculations to identify variables that may be susceptible to deterministic searches is not specific to an evolutionary algorithm. The approach could have equally been integrated into the PSOIW algorithm or a different algorithm entirely.

The information gained using the proposed optimization strategy is applicable to practicing energy modellers. For example, knowing which sets of design variables require simultaneous tuning and which design variables can be selected in isolation is useful information for energy modellers attempting to model high performance buildings.

Acknowledgments

This work was funded in part by NSERC and by the Faculty of Engineering and Computer Science at Concordia University.

References

Amdahl, G., 1967. Validity of the single processor approach to achieving large-scale computing capabilities. In: Proceedings of AFIPS. New York, US, pp. 483–485.

Audet, C., Dennis, J.E., 2002. Analysis of generalised pattern searches. SIAM Journal of Optimization 13 (3), 889–903.

Bank of Canada, 2009. Rates and Statistics: Interest rates, Canadian interest rates. Accessed January, 2013.
URL http://www.bankofcanada.ca/rates/interest-rates/

Bucking, S., Athienitis, A., Zmeureanu, R., O'Brien, W., Doiron, M., 2010. Design optimization methodology for a near net zero energy demonstration home. In: Proceedings of EuroSun International Conference on Solar Heating, Cooling and Buildings. EuroSun, Graz, Austria.

Caldas, L., 2008. Generation of energy-efficient architecture solutions applying GENE_ARCH: An evolution-based generative design system. Advanced Engineering Informatics 22 (1), 59–70.

Chen, Y., Athienitis, A., Galal, K., 2010a. Modeling, design and thermal performance of a BIPV/T system thermally coupled with a ventilated concrete slab in a low energy solar house: Part 1, BIPV/T system and house energy concept. Solar Energy 84 (11), 1892–1907.

Chen, Y., Galal, K., Athienitis, A., 2010b. Modeling, design and thermal performance of a BIPV/T system thermally coupled with a ventilated concrete slab in a low energy solar house: Part 2, ventilated concrete slab. Solar Energy 84 (11), 1908 – 1919.

Clarke, J., 2001. Energy Simulation In Building Design, 2nd Edition. Butterworth-Heinemann, Oxford, UK.

CMHC, 2009. CMHC equilbrium homepage. Retrieved May, 2012.
URL `http://www.cmhc-schl.gc.ca/en/co/maho/yohoyohe/heho/eqho/index.cfm`

Cover, T., Tomas, J., 2006. Elements of Information Theory, 2nd Edition. Wiley, Hoboken, NJ.

Crawley, D. B., Pedersen, C. O., Lawrie, L. K., Winkelman, F. C., 2000. Energy-plus: Energy simulation program. ASHRAE Journal 42, 49–56.

da Graca, G. C., Augusto, A., Lerer, M. M., 2012. Solar powered net zero energy houses for southern europe: Feasibility study. Solar Energy 86 (1), 634–646.

DOE, 2009. Buildings Energy Data Book, Chapter 1.2 Building Sector Expenditures. United States Department of Energy.

DOE, Oct. 10, 2011. EnergyPlus Engineering Reference: The Reference of EnergyPlus Calculations (in case you need to know). DOE.

Doiron, M., O'Brien, W., Athienitis, A., 2011. Energy performance, comfort, and lessons learned from a near net zero energy solar house. ASHRAE Transactions 117 (2), 1–12.

Eiben, A. E., Rudolph, G., 1999. Theory of evolutionary algorithms: a bird's eye view. Theoretical Computer Science 229 (1-2), 3–9.

Eiben, G., Smith, J., 2003. Introduction to Evolutionary Computing, 1st Edition. Springer, London, UK.

Elbeltagi, E., Hegazy, T., Grierson, D., 2005. Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics 19 (1), 43–53.

Feoktistov, V., 2006. Differential Evolution: In search of solutions, 1st Edition. Springer, New York, USA, Chapter 9.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition. Addison-Wesley, New York.

Goos, P., Jones, B., 2011. Optimal Design of Experiments: A Case Study Approach, 1st Edition. Wiley.

Hachem, C., Athienitis, A., Fazio, P., 2011. Parametric investigation of geometric form effects on solar potential of housing units. Solar Energy 85 (9), 1864 – 1877.

Hickey, R., 2012. Clojure programming language homepage. Accessed May, 2012.
URL `http://www.clojure.org/`

Hooke, R., Jeeves, T.A., 1961. 'Direct search' solution of numerical and statistical problems. Journal of the Association for Computing Machinery 8 (2), 212–229.

Hydro-Québec, 2010. Hydro-Québec: Annual Report 2010. Accessed January, 2013.
URL `http://www.hydroquebec.com/publications/en/annual_report`

IEA/ECBCS, 2013. IEA Task 40/ECBCS Annex 52 Homepage. Accessed March 2013.
URL http://task40.iea-shc.org/

Kämpf, J. H., Montavon, M., Bunyesc, J., Bolliger, R., Robinson, D., 2010a. Optimisation of buildings' solar irradiation availability. Solar Energy 84 (4), 596–603, international Conference CISBAT 2007.

Kämpf, J. H., Wetter, M., Robinson, D., 2010b. A comparison of global optimization algorithms with standard benchmark functions and real-world applications using EnergyPlus. Journal of Building Performance Simulation 3 (2), 103–120.

Kampstra, P., 2008. Beanplot: A boxplot alternative for visual comparison of distributions. Journal of Statistical Software 28 (1), 1–9.

Kennedy, J., Eberhart, R., Shi, Y., 2001. Swarm Intelligence, 1st Edition. Morgan Kaufmann Publishers, San Francisco, USA.

Kolokotsa, D., Rovas, D., Kosmatopoulos, E., Kalaitzakis, K., 2011. A roadmap towards intelligent net zero- and positive-energy buildings. Solar Energy 85 (12), 3067 – 3084.

NRCan-OEE, 2009. Energy use data handbook. Accessed Nov 2012.
URL http://oee.nrcan.gc.ca/publications/statistics/handbook09/pdf/handbook09.pdf

O'Brien, W., 2011. Development of a solar house design methodology and its implementation into a design tool. Ph.D. dissertation, Concordia University.

Ooka, R., Komamura, K., 2009. Optimal design method for building energy systems using genetic algorithms. Building and Environment 44 (6), 1538–1544.

Parry, M., Canziani, O., Palutikof, J., van der Linden, P., Hanson, C. (Eds.), 2007. Fourth Assessment Report: Climate Change 2007: The AR4 Synthesis Report. Geneva: IPCC.
URL http://www.ipcc.ch/ipccreports/ar4-wg1.htm

Price, K., Storn, R., Lampinen, J., 2005. Differential Evolution: A Practical Approach to Global Optimization, 1st Edition. Springer-Verlag, Heidelberg, DE.

Ramesh, T., Prakash, R., Shukla, K., 2010. Life cycle energy analysis of buildings: An overview. Energy and Buildings 42 (10), 1592–1600.

Renders, J., 1994. Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In: Proceedings of IEEE World Congress on Computational Intelligence. Vol. 1. IEEE, Orlando, FL, pp. 312–317.

RSMeans, 2011. RSMeans building construction cost data. Retrieved April 2011.
URL http://www.rsmeans.com

RSMeans, 2012. RSMeans Electrical cost data. RSMeans Company, Kingston, MA.
URL http://www.rsmeans.com

Scott, D. W., 1992. Multivariate Density Estimation. Theory, Practice and Visualization, 1st Edition. Wiley, New York.

SQLite, 2012. SQLite homepage. Accessed April 2012.
URL http://www.sqlite.org/

Storn, R., Price, K., 1995. Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces. Tech. rep., International Computer Science Institute, Berkley, USA:.

Torcellini, P., Pless, S., Deru, M., Crawley, D., June 2006. Zero energy buildings: A critical look at the definition. In: ACEEE Summer Study. NREL, pp. 1–12.

UNEP-SBCI, 2007. Buildings Can Play Key Role in Combating Climate Change. United Nations Environment Program.
URL http://www.unep.org/Documents.Multilingual/Default.asp?DocumentID=502&ArticleID=5545

Wang, W., 2005. A simulation-based optimization system for green building design. Ph.D. dissertation, Concordia University.

Wang, W., Rivard, H., Zmeureanu, R., 2006. Floor shape optimization for green building design. Advanced Engineering Informatics 20 (4), 363–378.

Wetter, M., 2011. GenOpt, generic optimization program, user manual, version 3.1.0. Technical report, Lawrence Berkeley National Laboratory, Berkeley, CA.
URL http://gundog.lbl.gov/GO/index.html

Wetter, M., Polak, E., 2004. Building design optimization using a convergent pattern search algorithm with adaptive precision simulations. Energy and Buildings 37, 603–612.