MOBO A NEW SOFTWARE FOR MULTI-OBJECTIVE BUILDING PERFORMANCE OPTIMIZATION

Matti Palonen[1], Mohamed Hamdy[1], and Ala Hasan[2]
[1]Aalto University, Espoo, Finland
[2]VTT Technical Research Centre of Finland, Espoo, Finland

ABSTRACT

This paper introduces a new software developed for building performance optimization. MOBO is a generic freeware able to handle single and multi-objective optimization problems with continuous and discrete variables and constraint functions. It can be coupled to many external (simulation) programs. It has a library of different types of algorithms (evolutionary, deterministic, hybrid, exhaustive and random), and is able to handle multi-modal functions and have automatic constraint handling. The input is fed by a GUI. The user can write the input by algebraic formulas using standard symbols. The output can be viewed by two graphs that show the progress of the optimization. A beta version of MOBO is available for download and use.

INTRODUCTION

By building optimization, it is possible to find optimal values of decision variables, among huge numbers of possible combinations, which are able to achieve defined conflicting objective functions and at the same time satisfy specified constraint functions. Various decision variables can be considered in the building envelope, the heating, ventilating and air conditioning (HVAC) systems, the centralised/on-site energy generation systems etc. Examples of the objectives are: minimization of environmental impacts (energy consumption, carbon emissions etc.), cost (investment cost, operating cost, life-cycle cost), equipment size (energy generation units, HVAC system etc.), and/or maximization of indoor air quality, energy efficiency, etc. These can be achieved individually, as single objectives, or simultaneously, as multi-objective optimization. The constraint functions may indicate satisfying, or not violating, different criteria (e.g. thermal comfort level, total investment cost limit, primary energy limit etc.).

Currently, there are many building optimization tools available with different features. However, we think that there is still a need for a new tool that should be a generic freeware and can fill the shortages recognised in the available tools. These were our main motivations for developing MOBO, a **M**ulti-**O**bjective **B**uilding **O**ptimization tool.

In this paper, we will first review available tools, then describe the features of MOBO, and finally give an example of implementation.

REVIEW

This section reviews some available optimization tools that have been used for building performance optimization. Table 1 gathers some main features of the reviewed tools. These tools can be classified into two categories: customized and generic tools.

Customized optimization tools

Opt-E-Plus, GENE_ARCH, BEopt™, TRNOPT, MultiOpt2, and jEPlus+EA are examples of tools customized mainly for building energy performance optimization. These tools are combinations of optimization algorithms/approaches and building performance simulation engines.

In **Opt-E-Plus** (Ellis et al., 2006), EnergyPlus simulation engine was coupled with a number of optimization strategies. Opt-E-Plus is a collection of input and output files, system directories, and computer routines that use an XML data model to transfer information among its various components. It allows distributed programming and supports selection of automation and optimization strategies. Opt-E-Plus doesn't support multidisciplinary optimization and the visualization of its tradespace is limited (Flager et al., 2008).

GENE_ARCH (Caldas, 2006) has scalable geometry generation functionality and good visualization capabilities. It is customized to couple DOE2.1E for building performance simulation and genetic algorithms for optimization. GENE_ARCH was used to find energy-efficient architecture solutions (Caldas, 2008 and 2011).

BEopt™ (Christensen et al., 2005) includes a graphical user interface (GUI) that allows the user to select from a range of predefined and discrete building options (heating, ventilating, and air-conditioning system type, envelope constructions, etc.) to be used in the optimization process. This allows the user to rapidly generate and visualize the design space through a browser, but its flexibility is limited as a result of having predefined building options and its inability to identify a wide range of objective functions. DOE2.2 and TRNSYS are the simulation engines of BEopt™. BEopt™ was used to evaluate the energy and cost savings potential from constructing efficient new homes and Net Zero-Energy Homes in the USA (Christensen et al., 2005).

TRNOPT (TESS. 2004 component libraries type 758) is an interface that couples the TRNSYS simulation with the generic optimization tool GenOpt (Wetter, 2001) in order to minimize a single cost function. TRNOPT's optimization advantages are limited by GenOpt's features indicated next. Furthermore, TRNOPT does not allow changes in the building simulation file. This limits its ability to optimize building elements such as external-wall insulation thickness, etc. TRNOPT was used by Calise F. (2010) and Fraisse et al., (2010) for HVAC system optimization.

Multiopt2 (Chantrelle et al., 2011) is a commercial customised multi-objective optimization tool for TRNSYS 17. The tool has a GUI for defining the optimization problem. However, it is limited by only one optimization algorithm NSGA-II (Deb et al., 2000).

jEPlus+EA (jEPlus+EA, 2011) is designed with an aim to remove the barrier to entry into the field of optimization, for existing jEPlus users at least. Example of jEPlus+EA's limitations is it can only use parameters defined in a valid jEPlus project as variables. This means all variables are considered as discrete during optimization. If a jEPlus project parameter tree has multiple branches, only the first branch will be used for optimization. **jEPlus+EA** is used in (Porritt et al., 2012)

Generic Optimization tools

GenOpt (Wetter, 2001), ModelCenter (by Phoenix Integration), modeFRONTIER (Nardin et al., 2009), DAKOTA (Design Analysis Kit for Optimization and Terascale Applications toolkit, Adams et al., 2011), iSIGHT (Velden and Koch, 2010) as well as MATLAB Optimization and Direct Search Toolboxes, are all examples of generic optimization tools. They were developed to allow coupling to any computer software. These tools have different features as indicated by Table 1. They allow performing parametric and sensitivity analyses using different visualization methods. The literature shows combinations of the above-mentioned tools with building performance simulation programs including EnergyPlus, DOE, TRNSYS, IDA-ICE etc (Wetter and Wright, 2004, Flager et al., 2008, Suga et al., 2010, Kayo and Ooka, 2010, and Hamdy et al., 2011). Excluding GenOpt and DAKOTA, the rest of the tools are commercial ones.

All the indicated generic tools are expected not to be user-friendly to inexpert users, which is due to the different configuration of the model files in each simulation program and the input and output files structure. GenOpt, was written in an effort to simplify and standardize coupling with different simulation programs through its open interface on both the simulation program side and the optimization algorithm side. Therefore, we can find that it was coupled with different simulation programs, e.g. EnergyPlus (Djuric et al., 2007), IDA-ICE (Hasan et al., 2008), ISOLAB (Bigot et al., 2010), Dymola/Modelica (Ali et al., 2010), and TRNSYS (Fraisse et al., 2010). The absence of multi-objective optimization algorithms and automatic constraint handling in GenOpt are its main disadvantages. Besides, it is not able to check for errors in user's inputs interactively.

Table (1)
Customized and Generic Optimization Tools

Optimization Tools		Q1	Q2	Q3	Q4	Q5
Customized	Opt-E-Plus	Yes	No	No	No	No
	GENE_ ARCH	Yes	Yes	No	No	No
	BEopt™	Yes	No	No	No	No
	TRNOPT	No	Yes	No	No	Yes
	MultiOpt2	No	Yes	Yes	?	Yes
	jEPlus+EA	No	Yes	No	Yes	No
Generic	GenOpt	Yes	No	No	Yes	Yes
	Model-Center	No	Yes	Yes/ No*	Yes	No
	modeFRO NTIER	No	Yes	Yes	Yes	Yes
	DAKOTA	Yes	Yes	Yes	Yes	Yes
	iSIGHT	No	Yes	Yes	No	Yes
	MATLAB Optim-ization Toolboxes	No	Yes	Yes/ No*	Yes	No

Q1: Is it a freeware?
Q2: Does it include multi-objective algorithms?
Q3: Does it handle constraint functions automatically?
Q4: Does it allow parallel computing?
Q5: Can it handle discrete and continuous variables simultaneously?
*Yes/No means that the answer is ''yes'' for some algorithms and ''no'' for the other algorithms in the reviewed optimization tool.

MAIN FEATURES OF MOBO

We think that there is a clear demand for an optimization tool that should be a generic freeware and not be limited to implementations with a specific simulation program.
In addition to single objective, MOBO is able to handle multiple objective optimization problems with constraint functions without a need to define penalty functions. It is developed in a way that most of the algorithms include automatic constraint handling. However, this feature does not limit the user to define penalty functions if he or she wishes.

Graphical User Interface for Defining the Optimization Problem

MOBO has a Graphical user Interface (GUI) for defining the optimization problem. MOBO can handle both continuous and discrete variables. For both variables a preprocessing function can be easily added by using standard algebraic symbols. The software supports approximately 50 functions that can be used in the formulas. Examples of these functions are *sin, cos, sqrt, exponent* etc. In MOBO, the input (usually the optimization problem and simulation software parameters) is given through the GUI, which checks that the input is correct interactively.

For adding different functions, the GUI also supports formulas and functions in the formulas. MOBO can handle up to four different function types that are:

- *Objective Function(s)*
- *Constraint Function(s):*
  - *Less or equal to 0*
  - *Equal to 0*
- *Other*

The function type *Other* is a function that will not be optimized but can be referenced from the functions that are optimized to build more complex objective- or constraint -functions. All these functions can be added as a formula or/and the value of the function can be read from the result file of the simulation program.

Graphical User Interface for the Optimization Progress

The progress of the optimization can be viewed with two different graphs: the first graph shows the values of the decision variables as a function of the simulation number, and the second graph shows a bi-objective function space.

Interface for Simulation Programs

MOBO can be used to run external (simulation) programs that calculate the values of the objective and possible constraint functions. The interface can be used with several tools such as IDA-ICE and TRNSYS etc. It handles the input and output through text files. The user needs to define the command that starts the external program, template file and the names of the input and the output files of the external program. Since the interface is generic, that is, it can be used with different programs, the user needs to specify the locations where to write the values of the variables (in input file) and the locations in the output file where to read the function values.

Interface for Adding New Algorithms and Functions

Programming interface for adding new functions can be used to extend available algorithms and functions in MOBO. The user can add new functions and optimization algorithms with MOBO API for Java.

Portability

Since the software is written with Java programming language, it can be by used with a number of different platforms such as Windows, Linux and Mac OS.

Parallel Simulation Execution

Modern microprocessor architecture is usually based on one processor that can handle multiple threads. These threads can run on parallel. Before starting the optimization procedure, the software will automatically detect the number of threads (or processors) in the system. Then, the optimization procedure will run multiple simulations (threads) on parallel. This parallel computing feature will lower the optimization time with the factor that is equal to number of threads available. All the algorithms in the new tool (expect the algorithm of Hooke and Jeeves) can use parallel computing.

Algorithms

Table 2 shows the algorithms available in MOBO and gives an overview of their features.

Genetic Algorithms (GA)

MOBO comes with a number of Genetic Algorithms (GA) for doing multi- or single objective optimization. These algorithms are NSGA-II (Deb et al. 2002), Pareto Archive NSGA-II (aNSGA-II) (Hamdy et al., 2012), and OMNI-optimizer (Deb and Tiwari, 2005) Both binary and real coded implementations are included. Binary GA can be used to solve problems with both discrete and continuous variables, while real coded GA can be used to solve problems with continuous variables only. All the GAs included in the software has an automatic constraint handling strategy. This means that the user does not need to define any penalty function.

The absence of optimization algorithms that can handle multi-modal single- or multi-objective problems (with single optimization run) seems to be a feature of many available optimization tools. In addition, most of the tools available consider diversity of the Pareto-set through the objective space only. For these reasons, we added the OMNI-optimizer algorithm. The OMNI-optimizer considers the diversity of the decision variables and can handle multi-modal problems.

Algorithm of Hooke and Jeeves

The algorithm of Hooke and Jeeves is a direct search algorithm, which is considered as a local search method. Algorithm of Hooke and Jeeves is a single objective optimization algorithm and can handle only continuous variables. In our implementation, an automatic constraint handling strategy is included, so that a penalty function is not needed in a constrained problem.

Hybrid Algorithm

Hybrid single objective optimization algorithm can handle both unconstrained- and constrained optimization problems. This algorithm has also an automatic constraint handling strategy included.

Hybrid algorithm uses first GA to find a good initial point for the Hooke and Jeeves algorithm.

Brute-Force Algorithm
The Brute-Force search is an exhaustive search method that can sample the whole solution space. The user has to set a step for the continuous variables. The user has to make a pre-evaluation of the total number of the simulations that will be generated in order to assure the control of the total optimization time.

Random Search Algorithm
The idea of the random search algorithm is to produce a number of randomly generated points. It is probably the easiest optimization algorithm to implement and can be used to solve any kind of optimization problem since it does not include any heuristics. However, its effectiveness has to be tested, as it will depend on the difficulty of the handled problem.

EXAMPLE

As a demonstration of the implementation, MOBO is used to handle a multi-objective optimization problem of a single-family detached house (Figure 1) located in Helsinki-Finland. The problem was solved in previous studies using different single and multi-objective optimization algorithms (Hasan et al., 2008, Hamdy et al., 2009, and Palonen et al., 2009).
The problem includes five design variables (Table 3): three continuous variables (additional insulation thickness in the external wall, roof and floor) and two discrete variables (types of windows and heat recovery). Space heating energy and additional investment costs are the two objective functions to be minimized.
The house is considered as a single zone for the energy calculation. It's initial U-values are in accordance with the Finnish National Building Code C3-2003. The house is heated by direct electricity (electric radiators inside the house and an electric heater in the air-handling unit). As a typical case for Finnish houses, no cooling system is implemented.

Table 2. Algorithms available in MOBO and their characteristics.

	PROBLEM					VARIABLES		
Algorithm	Single	Multi-	Constrained	Multi-modal	Automatic Constraint handling	Discrete	Continuous	Parallel Computing
Binary NSGA-II	x	x	x		x	x	x	x
BINARY Pareto Archive NSGA-II	x	x	x		x	x	x	x
Binary OMNI-Optimizer	x	x	x	x	x	x	x	x
Real Coded NSGA-II	x	x	x		x		x	x
Real Coded Pareto Archive NSGA-II	x	x	x		x		x	x
Real Coded OMNI-Optimizer	x	x	x	x	x		x	x
Hooke-Jeeves	x		x		x		x	
Hybrid Algorithm	x		x		x	x	x	x
Brute-Force	x	x	x	x		x	x	x
Random Search	x	x	x	x		x	x	x

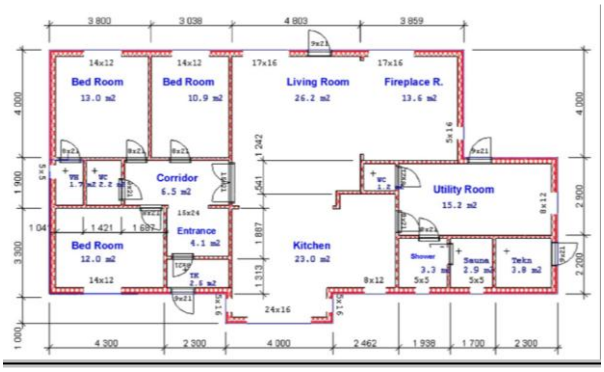

Figure 1. The detached house (Shemeikka and Laitinen, 2005)

Table (3)
Design variables of the optimization problem

Design Variables	Type	Min. Value	Max. Value
Additional insulation thickness in external walls (m)	Continuous	0	1
Additional insulation thickness in roof (m)	Continuous	0	1
Additional insulation thickness in floor (m)	Continuous	0	1
Window's *U*-values (W/m^2K)	Discrete (two options)	1	1.4
Heat recovery efficiency (%)	Discrete (two options)	70	80

The problem is optimized here using three algorithms: the Brute-Force algorithm, Random-Search algorithm and Pareto-Archive NSGA-II algorithm. The Brute-Force algorithm used a step of 0.05 m for the continuous variables (insulation thickness). This resulted in 32000 simulations. The Random-Search algorithm applied 600 simulations. The Binary aNSGA-II was run twice. In the first run, no constraint function was imposed, while in the second run, a maximum additional investment of 6000 euros was considered as a constraint function. There were no attempts to find optimal parametric values for the GA. For both runs, the mutation probability was set to $1/L$ (where L is the number of bits used to represent each of the solutions), the adult population size to six, and the number of generations to 100. Two crossover probabilities (0.8 and 0.9) were set to the first and second runs, respectively.

Program Set-up for the Problem

The set-up of the BI-objective constrained example in MOBO and its operation is shown here.

First, the design variables, their bounds and possible pre-process functions are added. This is demonstrated in Figure 2. Then, the two objective functions and the constraint function are constructed as shown in Figure 3. Twelve functions of type *other* are used to read the monthly energy consumptions. The objective function *Eheating* is the energy consumption for the whole year and is to be minimised, as well as the objective function *Cost*. Finally, the constraint functions *C1* is defined such that *Cost- 6000* should equal to or be less than zero. In the simulation tab (Figure 4), the location of the model file, names of the input- and output files of the simulation software and the command that starts the simulation are given. In the algorithm tab (Figure 4), the Pareto Archive NSGA-II algorithm is selected and the parameters for the algorithm are given. In this example, we used the IDA-ICE software for the building simulation. The progress of the output during the optimization run can be seen in Figure 5.

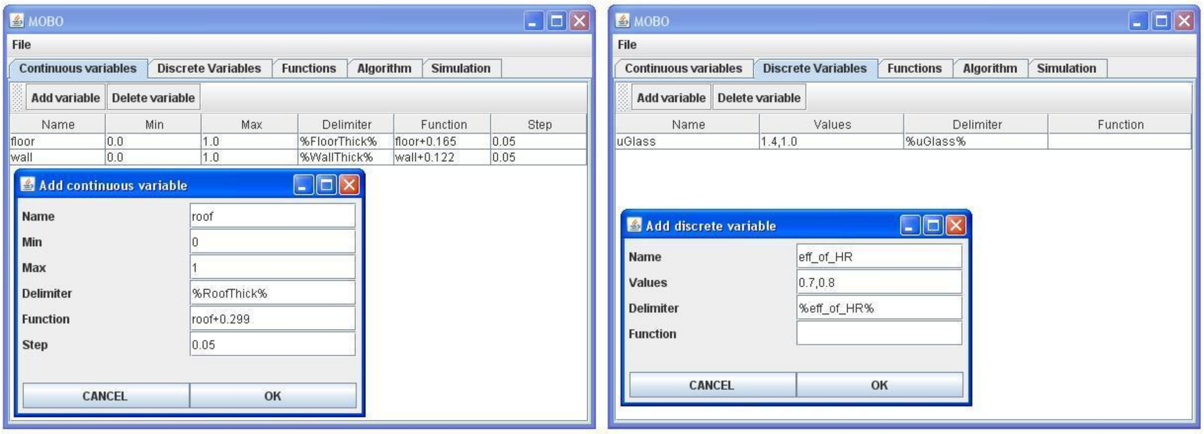

Figure 2. Set-up of the continuous and discrete variables.

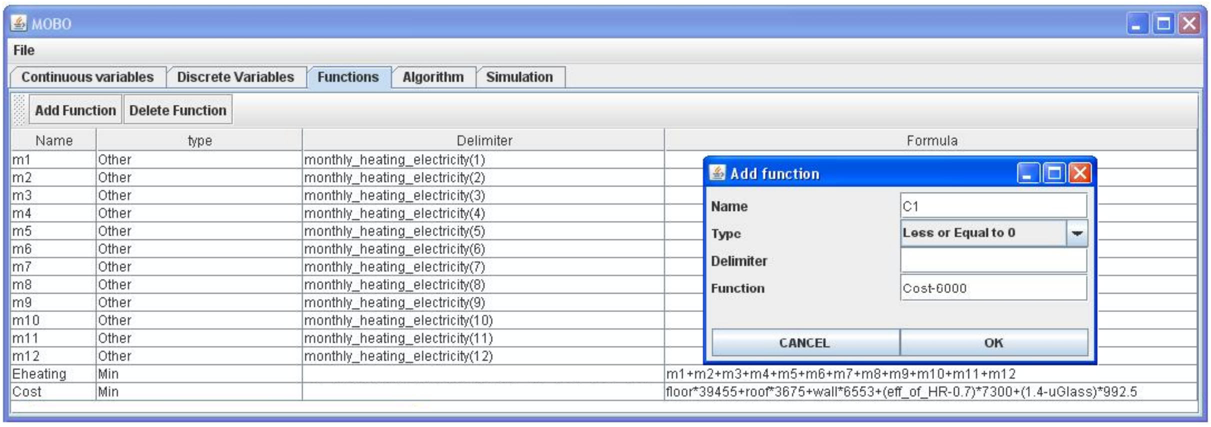

Figure 3. Set-up of the objective functions and the constraints function.

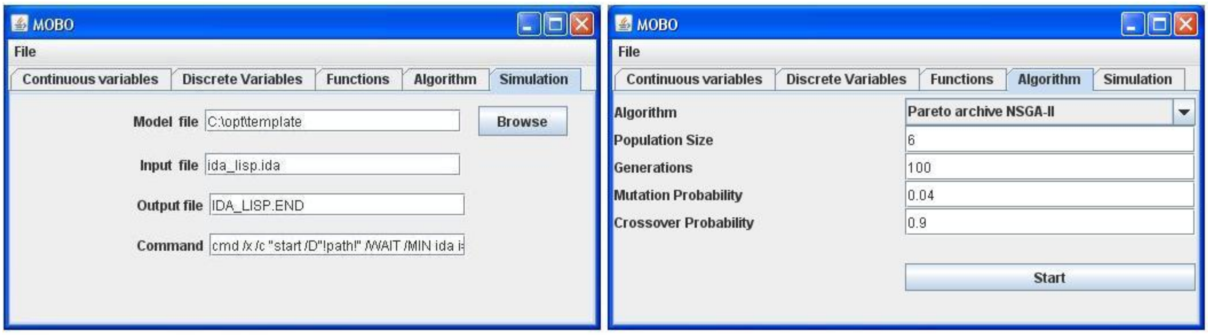

Figure 4. Set-up of the connection with the simulation program and the selection of the algorithm.

RESULTS

It should be noted that the main target of the implementation example is the demonstration of the use of MOBO. However, a small description of the performance of the above-mentioned algorithms will also be made.

Figure 6 depicts the results obtained from MOBO. We can notice that the 32000 simulations of the Brute-Force algorithms cover the solution space and determine the global Pareto-front solution. The whole history of the Random Search algorithm solutions is also shown. It can be noticed that some of these solutions converge close to the global Pareto-front in its middle section. However, it could not generate many points on the right-hand side of the search space.

In its both runs, the aNSGA-II algorithm generated a diverse set of solutions close or on the global Pareto-front. Only the non-dominated solutions of these two algorithms are shown in Figure 6. It can be noticed that solutions were generated that are very close to the extreme solutions. However, in the first run there was room for little improvement to the second extreme value (maximum cost). It is interesting to see that the Random Search found few points that actually dominate the aNSGA-II points in the middle section. It is worthwhile mentioning that these are results of running those stochastic algorithms only once. However, the results are of high quality as seen.

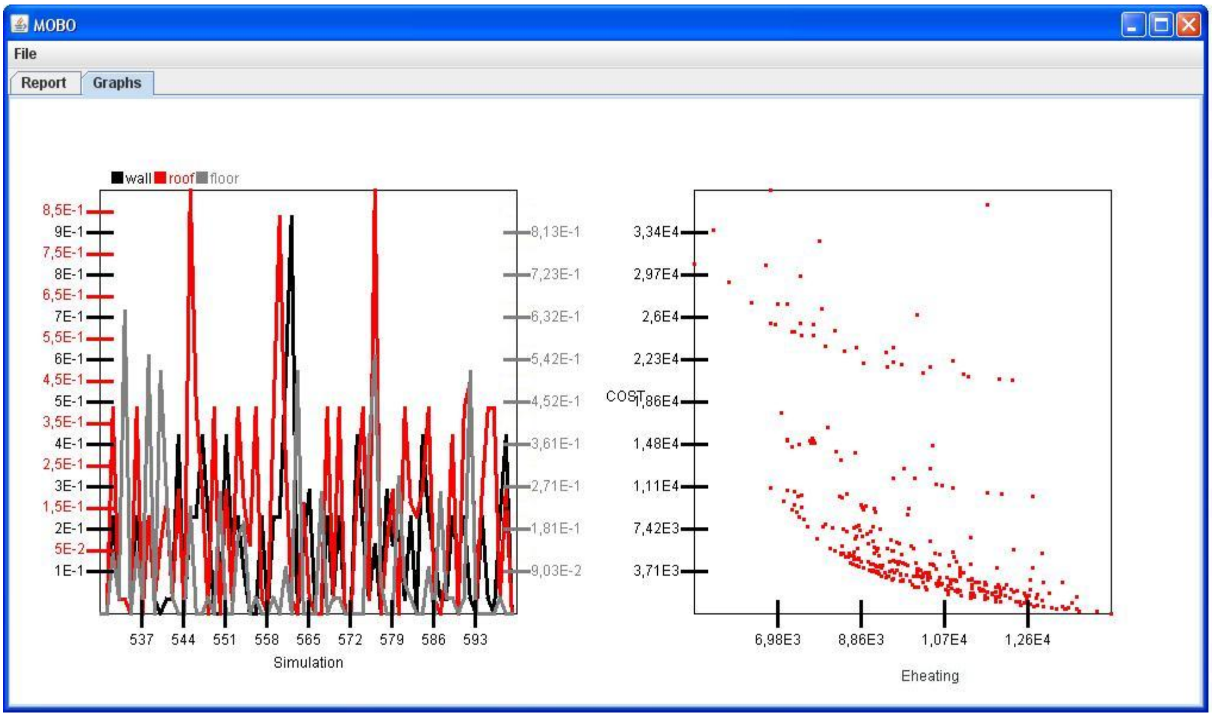

Figure 5. Online output of MOBO.

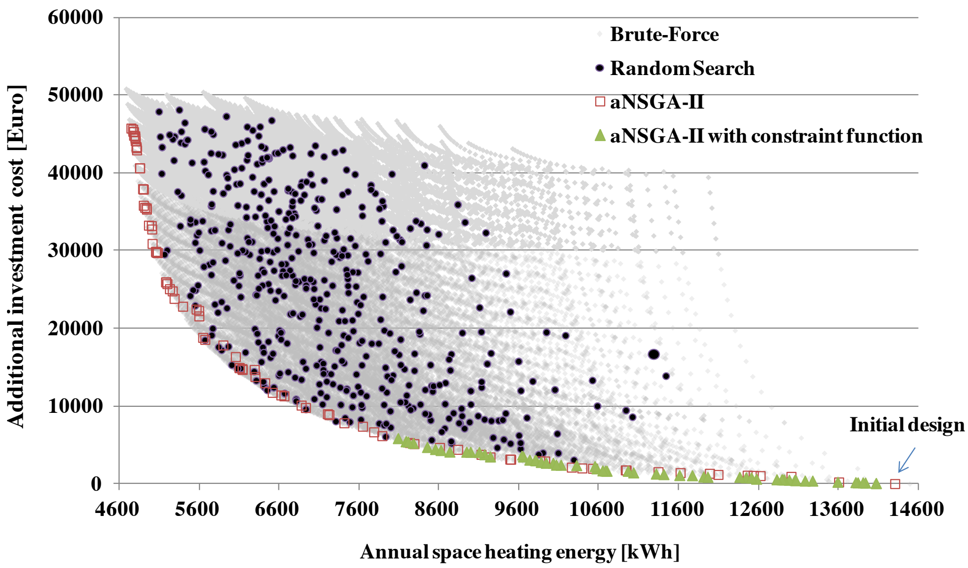

Figure 6. MOBO results of the four search/optimization runs

CONCLUSIONS

This paper presents the features of a new building optimization freeware MOBO, which can handle multi- and single-objective optimization problems with automatic constraint handling ability, as well as single objective problems. MOBO is a generic optimization tool that can be coupled to many building simulation programs. On the other hand, it tries to minimize associated complications when making the coupling. It has a Graphical User Interface (GUI) by which the coupling can be established, the input is given and the output is viewed. Through the GUI, the user is able to use simple algebraic symbols and calculator functions to describe the continuous and discrete variables, as well as the objective and constraint functions. There is a library including different optimization algorithms, which can be extended by adding other algorithms by the user. Currently there are 10 algorithms of different types: evolutionary with real and binary coding, deterministic, hybrid, exhaustive and random. The implementation of MOBO is demonstrated by a building performance optimization example, which was solved using three algorithms. The beta version of the software is available for download from the following link http://www.ibpsa-nordic.org/tools.php.

ACKNOWLEDGEMENTS

Acknowledgements are due to the Academy of Finland for funding this work.

REFERENCES

Adams, B.M., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Eldred, M.S., Hough, P.D., Lefantzi, S., Swiler, L.P., and Vigil, D.M. 2011. "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.2 Theory Manual," Sandia Technical Report SAND-9106.

jEPlus+EA, 2011. available online at : http://www.iesd.dmu.ac.uk/~jeplus/wiki/doku.php?id=docs:jeplus_ea

Ali M., Vukovic V., Sahir M.H., Fontanella G. 2012. Energy Analysis of Chilled Water System Configurations using Simulation-based Optimization, Energy and Buildings, 59 (4), pp. 111–122.

Bigot, D., Miranville, F., Boyer, H., Bojic, M., Guichard S., Jean A. 2013. Model optimization and validation with experimental data using the case study of a building equipped with photovoltaic panel on roof: Coupling of the building thermal simulation code ISOLAB with the generic optimization program GenOpt, Energy and Buildings, 58 (3), pp. 333-347

Caldas L. 2006. GENE_ARCH: an evolution-based generative design system for sustainable architecture. I.F.C. Smith (Ed.): EG-ICE 2006, LNAI 4200, pp. 109-118.

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

Caldas L. 2011. Generation of Energy-Efficient Patio Houses: Combining GENE_ARCH and a Marrakesh Medina Shape Grammar. Artificial Intelligence and Sustainable Design. The AAAI 2011 Spring Symposium (SS-11-02).

Calise F. 2010. Thermoeconomic analysis and optimization of high efficiency solar heating and cooling systems for different Italian school buildings and climates. Energy and Buildings 42 (7) 992–1003.

Chantrelle, F. B., Lahmidi, H., Keilholz, W., El Mankibi M., Michel P. 2011. Development of a multicriteria tool for optimizing the renovation of buildings. Applied Energy, 88 (4), pp. 1386–1394.

Christensen, C., Horowitz, S., Givler, T., Courtney, A. 2005. BEopt: software for identifying optimal building designs on the path to zero net energy. Proceedings of ISES 2005 Solar World Congress, Orlando, Florida, USA.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation. 6 (2): 182–197.

Deb, K, Tiwari, S. 2005. Omni-optimizer: A Procedure for Single and Multi-objective Optimization. In Proc EMO. Mexico. p. 47-61.

Djuric, N., Novakovic, V., Holst, J., Mitrovic, Z. 2007. Optimization of energy consumption in buildings with hydronic heating systems considering thermal comfort by use of computer-based tools. Energy and Buildings 39, pp. 471–477.

Ellis, P.G., Griffith, B., Long, N., Torcellini, P. Crawley, D. 2006. Automated multivariate optimization tool for energy analysis, IBPSA SimBuild Conference, 2-4 August, Cambridge, Massachusetts.

Flager, F., Welle, B., Bansal, P., Soremekun, G., Haymaker, J. 2008. Multidisciplinary Process Integration and Design Optimization of a Classroom Building. ITcon, 14 (August): pp. 595-612.

Fraisse, G., Boichot, R., Kouyoumji, J.-L., Souyri, B. 2010. Night cooling with a Ventilated Internal Double Wall. Energy and Buildings, 42 (3), pp. 393–400.

Hasan, A., Vuolle, M, Siren, K. 2008. Minimisation of life cycle cost of a detached house using combined simulation and optimization. Building and Environment. 43(12): pp. 2022-2034.

Hamdy, M., Hasan, A., Sirén K. 2009. Combination of optimization algorithms for a multi-objective building design problem. 11th IBPSA Conference, Glasgow-UK.

Hamdy, M., Hasan A., Sirén K. 2011. Impact of adaptive thermal comfort criteria on building energy use and cooling equipment size using a multi-objective optimization scheme. Energy and Buildings, 43 (9); pp. 2055–2067.

Hamdy, M., Palonen, M., Hasan, A. 2012. Implementation of Pareto-Archive NSGA-II Algorithms to a nearly-Zero-Energy Building Optimization Problem. 1st Building Simulation and Optimization Conference (BSO12), IBPSA-England, Loughborough Uni., UK.

Kayo, G., Ooka, R. 2010. Building energy system optimizations with utilization of waste heat from cogenerations by means of genetic algorithm. Energy and Buildings 42: pp. 985–991.

Nardin, L., Sørensen, K., Hitzel, S., Tremel, U. 2009. modeFRONTIER© , a Framework for the Optimization of Military Aircraft Configurations. MEGADESIGN and MegaOpt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design. Volume 107, pp 191-205.

Palonen, M., Hasan, A., Siren, K. 2009. A genetic algorithm for optimization of building envelope and HVAC system parameters. 11th IBPSA Conference, Glasgow-UK.

Porritt, S.M., Cropper P.C., Shao, L., Goodier C.I. 2012. Ranking of interventions to reduce dwelling overheating during heat waves, Energy and Buildings, 55 (12), pp. 16-27.

Shemeikka J., Laitinen A. 2005. Specification of RET-single family house. Technical Research Centre of Finland VTT, Building and Transport. Version 2.0, 2.3.2005 [in Finnish].

Suga K., Kato S., Hiyama, K. 2010. Structural analysis of Pareto-optimal solution sets for multi-objective optimization: An application to outer window design problems using Multiple Objective Genetic Algorithms. Building and Environment 45 (5) pp. 1144–1152.

TESS, 2004. TRNOPT v16 Installation, Run-Time Instructions, and License Agreement. Thermal Energy Systems Specialists, Madison, WI. http://www.tess-inc.com

Wetter, M., Wright, J. 2004. A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. Building and Environment 39(8): pp. 989-999.

Wetter, M. 2011. GenOpt®-A Generic Optimization Program. Proc. IBPSA's Building Simulation 2001 Conference, August 13-15, 2001 in Rio de Janeiro.

Velden, A., Koch, P. 2010. Isight Design Optimization Methodologies. ASM Handbook Volume 22B Application of Metal Processing Simulations.