AUTOMATED MULTIVARIATE OPTIMIZATION TOOL FOR ENERGY ANALYSIS[a]

Peter G. Ellis[1], Brent T. Griffith[1], Nicholas Long[1], Paul Torcellini[1], and Drury Crawley[2]
[1]National Renewable Energy Laboratory, Golden, CO
[2]United States Department of Energy, Washington, DC

ABSTRACT

Building energy simulations are often used for trial-and-error evaluation of "what-if" options in building design—a limited search for an optimal solution, or "optimization." Computerized searching has the potential to automate the input and output, evaluate many options, and perform enough simulations to account for the complex interactions among combinations of options. This paper describes ongoing efforts to develop such a tool. The optimization tool employs multiple modules, including a graphical user interface, a database, a preprocessor, the EnergyPlus simulation engine, an optimization engine, and a simulation run manager. Each module is described and the overall application architecture is summarized.

INTRODUCTION

Building energy simulations are often used for trial-and-error evaluation of "what-if" options in building design—a limited search for an optimal solution. With today's computer power, the bottleneck is no longer simulation run time, but rather the human time to handle input and output. Human-driven methods are inefficient, and require skills, experience, and time. Computerized optimization has the potential to automate the input and output, evaluate many options, and perform enough simulations to account for the complex interactions among combinations of options. This paper describes ongoing efforts to develop such an automated optimization tool.

Building design problems are inherently multivariate and multicriteria. Multivariate optimization is much more difficult than the simpler problem of minimizing a single variable. The objective, or performance index, also must include energy performance and the cost implications of design options. However, as pointed out by Papamicheal (1993), the relative importance of cost and performance cannot be explicitly specified because it represents a qualitative judgment. The implication is that the optimization search is best formulated as a multicriteria, or multiobjective, search for a set, or Pareto-optimal front, of optimal solutions. Figure 1 uses one possible choice of metrics for cost and performance (solutions that are down and to the left are better) to diagram a Pareto front. A designer who is presented with such results then has a range of possible solutions (which are all optimal) that can be used to inform decision-making. The preferred search algorithms for finding the Pareto-optimal front can separately and simultaneously minimize both cost and performance. This is opposed to the more common approach of attempting to aggregate and weight different metrics into a single performance index.

Numerous researchers have studied the application of optimal searches for building design. The GenOpt program implements a large number of search algorithms. The most prominent is called "Generalized Pattern Search Hooke-Jeeves" (Wetter 2004). Gradient-based methods are not well suited for building design applications. Wetter and Wright (2003) point out that the approximate solutions produced by energy simulation can lead to discontinuous results, which cause problems for gradient methods. For buildings, search methods need to handle discrete variables and should attempt to identify a broad portion of the Pareto-optimal front. Genetic algorithms are applicable to discrete variables and have been studied (in the buildings context) by Wright et al. (2002), Huang and Lam (1997), Coley and Schukat (2002), and Caldas and Norford (2003). Wetter and Wright (2003) proposed

---

[a] This work has been authored by an employee or employees of the Midwest Research Institute under Contract No. DE-AC36-99GO10337 with the U.S. Department of Energy.

combining single objective genetic algorithms with generalized pattern search algorithms.

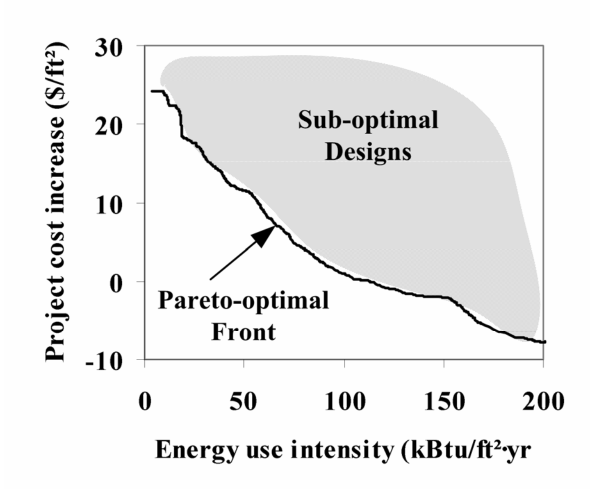

Figure 1 Pareto front of optimal solutions

OPTIMIZATION TOOL

The authors developed an optimization tool as a research platform to answer broad, energy-related questions about the commercial building sector for the U.S. Department of Energy (DOE). In particular, DOE has a goal to set the conditions for marketable zero-energy buildings by 2025. Whole-building energy modeling and optimization are considered important tools for achieving this goal, as they can identify common solutions for energy efficiency that can be replicated across the commercial building sector.

This project grew out of a related study, also commissioned by DOE, which is referred to as the *Assessment of Commercial Sector Opportunities* (see Griffith and Crawley 2006 in these proceedings). To perform the *Assessment*, we developed software tools for automatically creating and running many simulations. The *Assessment* automatically generated thousands of EnergyPlus input files that swept a variety of design options over every TMY2 weather file location in the country. More than 200,000 simulations for the *Assessment* were run on NREL's computing resources. Much of its application architecture was reused and further developed to form the underpinnings of the optimization tool.

Because extensive computing resources are required for useful optimizations, the optimization tool has been mainly developed as an in-house research tool that uses a supercomputer.

Overview

The optimization tool employs multiple modules, including: (1) a graphical user interface (GUI) for selecting options and viewing results, (2) a database for storing component performance data and costs, (3) a preprocessor to convert high-level input parameters into a detailed building model, (4) the EnergyPlus whole-building simulation engine to analyze the model, (5) an optimization engine to select design options, and (6) a simulation run manager to handle simulation runs on different computing resources. The *Assessment* pioneered the initial efforts for developing and integrating the preprocessor, input database, simulation engine, and simulation run manager. Each module is described in detail below and diagrammed in Figure 2.

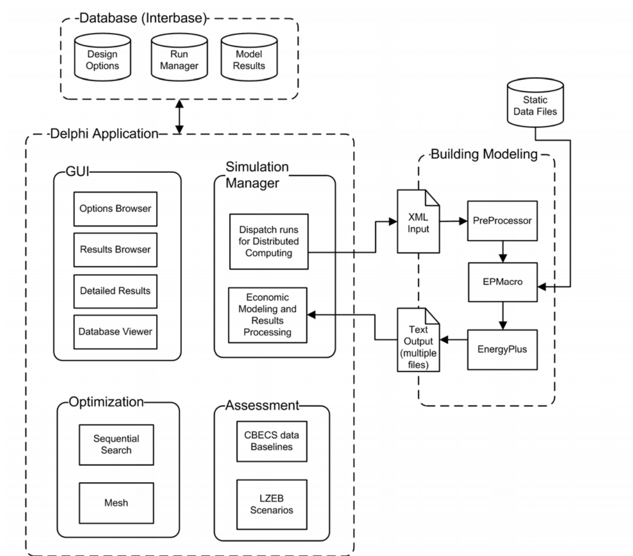

Figure 2 Block diagram of optimization tool

The program architecture is modular to allow multiple programmers to work at the same time. Modules are programmed in Fortran95 or in object-oriented Delphi.

Graphical User Interface

The GUI features a building creation wizard, an options browser, and a results browser. The building creation wizard presents a series of tabs with high-level parameters that define the overall type and location of the building. The options browser allows the user to select from 40 major design options that are grouped into categories of program, form, fabric, and equipment (see Figure 3). Program parameters include lighting density, plug and process load density (electrical equipment), people density, infiltration rate, illuminance setpoints for automatic dimming controls, and density of tubular daylighting devices (TDDs). Form parameters include aspect ratio, orientation, window glazing fraction, and overhang depth. Fabric parameters include choice of constructions for exterior walls, interior walls, roofs, windows, and skylights. Equipment parameters include choice of HVAC system type and photovoltaic system efficiency. All options are selected as discrete values. Cost data are also associated with each option.

The GUI provides a results browser that renders all the simulations on a chart of percent energy savings versus percent cost savings (see Figure 4). Each point represents an individual annual simulation. The user simply clicks on a point in the graph to view the individual results for that simulation. The 3D wireframe view (DXF) of the building model is also available.

The GUI is coded in Delphi 7 and uses model-driven architecture to take advantage of many vendor-provided software components for database interactions, GUI controls, DXF views, and fast array and matrix math.

Simulation Engine

The optimization platform uses EnergyPlus as the simulation engine. EnergyPlus is a whole-building energy simulation program developed by DOE (Crawley et al. 2004) that can model the complex interactions that are important for optimization. EnergyPlus can also simulate many state-of-the-art and cutting-edge technologies that will be used in tomorrow's zero-energy buildings. The ability to model technologies such as TDDs, radiant heating, and underfloor air distribution, along with features for thermal comfort, will allow the optimization tool to include these options in the simulations. (Some of these technologies are not yet implemented in the GUI and preprocessor.)

Because the optimization process is computationally intensive and demands hundreds (or even thousands) of individual simulation runs, we have looked for ways to decrease the run time of EnergyPlus simulations to expedite the overall process. Adjusting factors such as minimum system time step, minimum temperature convergence, and maximum number of warm-up days significantly decreases the run time of the program by up to a factor of 10 with little penalty in accuracy. We also use multipliers for thermal zones to help reduce run times. This greatly reduces the run times for multistory buildings and buildings with hundreds of TDDs. We continue to explore further opportunities to decrease the simulation run times.

Preprocessor

Because simulation engines such as EnergyPlus require detailed input data about the building, we use a preprocessor to reduce the low-level input data to a set of high-level input parameters. Unlike most automated, macro-based procedures that search and replace parameters one by one, the high-level parameters can have a one-to-many relationship. The preprocessor implements a large number of deterministic assignments that map high-level parameters to the low-level simulation input. For instance, a parameter to vary the number of floors can completely change the number, size, and location of almost every surface in the EnergyPlus model. It also changes the number of thermal zones and configuration of the HVAC system. Finally, it can affect the HVAC sizing.

High-level parameters are implemented in the preprocessor for every option in the GUI options browser. We use an XML text file to exchange data between the GUI and the preprocessor based on an XML Schema that we developed for this purpose. The preprocessor then uses the high-level parameters to "autobuild" the EnergyPlus input file, including all the envelope geometry, zoning, internal loads, schedules, and HVAC system. Deterministic assignments are used to default simulation input details that are not explicitly defined by the parameters.

Database

The database module serves multiple purposes by storing design options and input parameters selected in the GUI, input libraries for EnergyPlus, cost data, simulation run queue, and simulation results. All data except for the EnergyPlus input libraries (which are text files) are stored in a relational database management system.

The EnergyPlus input libraries comprise a collection of macro files, or include files, that provide segments of EnergyPlus input for location-dependent input objects, schedules, HVAC systems and component performance data, and report variables. Some macro files that were distributed with the release version of EnergyPlus are also included.

We consulted practicing architects and engineers to help formulate cost estimates. We also used data from RSMeans (2004) with regional adjustment factors.

The database module has an input mechanism to add or change data and allows for user input of cost information.

After simulations are completed, the results are also stored in the database so that individual simulation results can be browsed in the GUI.

Optimization Engine

The optimization engine module is the heart of the optimization tool. It determines which design options are to be simulated and then analyzes the results in the context of the performance objective that is to be minimized. The optimization engine then decides which options to keep and which to change before it launches a new set of simulations.

Early pilot work for the project used GenOpt (Wetter 2004) as the optimization engine. GenOpt is a general-purpose engine that enables a selection of optimization algorithms, including generalized pattern search, gradient search, particle swarm, and multidimensional parametric runs (enumeration) to be run. We later developed our own optimization engine that supports multiple optimization algorithms, including full enumeration and sequential search. Because the algorithms are programmed in a modular fashion within the optimization engine, more algorithms can be added easily.

The full enumeration, or "brute force," algorithm evaluates every possible combination of options. It is guaranteed to always identify the true optimal solution, but it is the most time-consuming and computationally intensive algorithm. Full enumeration is primarily used to check the results of the other algorithms.

The sequential search algorithm is the same optimization method used in BEOpt (Christensen et al. 2006). The method was probably first described by Meier (1982) and used in a different, whole-sector context. It is similar to one used by Davis Energy Group in a Pacific Gas and Electric ACT2 project (DEG 1993) and to the "energy code multiplier method" available in EnergyGauge-Pro (FSEC 2001). Christensen developed this method specifically to search for a "path" to zero-energy homes. The Pareto-optimal front is defined by connecting the points for building designs that achieve various levels of energy savings at minimal cost (establishing the lower bound of results from all possible building designs). The search technique moves along the path in steps. At each step along the path, individual simulations are performed to evaluate all options across a range of categories (wall type, ceiling type, window glass type, HVAC type, etc.) and searches for the most cost-effective option. Based on the results from the previous step, the most cost-effective option is selected as an optimal point on the path and put into a new building description. The process is then repeated.

The search technique handles special cases that are caused by interactions between options (beyond just the diminishing returns accounted for in the basic sequential search technique). Special cases with negative interactions are handled by looking back along the path and continually re-evaluating previously rejected options to properly identify the potential of large-savings options and options that involve trade-offs between categories. The search technique does not assume that once an option is selected that it stays selected. The technique also tracks points from previous steps and checks to see whether they may be better results than the current step. Positive interactions are accommodated by allowing the user to define combined options (which are linked to ensure that such potentially synergistic combinations are evaluated during optimization). The method has been validated by comparison to full enumeration.

The sequential search technique has several advantages. It searches for points near the Pareto-optimal front in a single optimization analysis, i.e., minimum-cost building designs at different energy savings levels, not just a global optimum. It also stores multiple near-optimal designs that are identified at each particular energy savings level. This may provide interesting design alternatives. The method is amenable to using distributed computing because for each step, an entire set of simulations can be run at the same time.

Disadvantages of the technique have not been explicitly identified. Areas to investigate may include coarse and fixed resolution for continuous parameters, scalability problems for very large numbers of design options, and its inability to ensure that synergistic but separate options are selected. These are all critical issues for shifting from residential to commercial analysis. The advantage of the technique is that relatively few iterations are required to arrive at the solution.

The resulting tool has the advantages of providing near-optimal solutions.

The sequential search allows for multi-objective and multicriteria optimizations. The principle behind the algorithms is to find the steepest slope from the previous point. Currently the slope is the change in cost versus the change in energy. However, if needed, more objectives can be added and tensors could be used to calculate the slopes.

Simulation Run Manager

Depending on the number of options that are selected in the GUI, the processor time needed for the optimization engine to resolve an answer can be very long. However, because each step in the optimization process uses multiple independent simulations, it favors a distributed computing approach.

NREL has two computing resources for distributed computing. The Computational Sciences Center maintains a Linux cluster, dubbed Lester, of 126 dual-processor nodes that run the 64-bit Linux operating system. We can use a custom 64-bit build of EnergyPlus for Linux to run up to 252 simultaneous simulations on the cluster. In reality, the cluster is shared with other users at NREL, so actual availability is less.

NREL also maintains a distributed computing network that uses the Condor workload management system (Thain et al. 2005). The Condor system identifies idle computers on the laboratory's local area network and takes advantage of the unused processing power to run jobs in the background. We currently have about 40 Windows machines enrolled in the Condor network, but we plan to increase this number in the future.

A simulation run manager handles the distribution of jobs launched by the framework and farms them out for processing on the computing resources, including Lester, Condor, and the user's local machine. The simulation run manager also monitors the status of running jobs and retrieves the simulation results when the job is done.

Lester and Condor are valuable computing resources that have allowed us to develop and test the optimization engine with a reasonable turnaround time for results. For example, an optimization that requires 18 iterations for 545 simulations (each simulation can take up to 2 minutes) can be completed in 2.5 hours, which averages to approximately 16.5 seconds per simulation.

Although distributed computing certainly expedites the optimization process, it is not required to run the tool. The optimization will just take longer to run the simulations individually.

VALIDATION

The main validation task is to verify that the input data are accurate and that the optimization tool reliably identifies the true optimum combination of parameters and design options. We have used the full enumeration algorithm, which will always find the true optimum (among the selected options), to check the results of the other optimization algorithms. The outcome of this showed that the sequential search reasonably approximates the Pareto-optimal front; enumeration did not yield any surprises. However, there are practical limitations to using full enumeration because the number of cases to evaluate becomes far too large for most real-world problems.

Thus far our testing has identified some deficiencies in the sequential search algorithm under certain scenarios. Competing design options can sometimes result in the true optimum solution not being correctly identified. One technology may not be useful unless it is combined with two or more other technologies. For example, TDDs by themselves may not be more efficient unless they are combined with daylighting controls. Fortunately, in this particular case, the sequential search algorithm *will* find this combination because daylighting is always an advantage. However, other combinations may not be found. The issue is in the number of degrees of freedom, which is currently limited to one.

The optimization tool is also being tested in a real-world application under another NREL project by using the tool in the design phases of a small retail building. Cost data are still problematic, especially for HVAC systems and equipment. Costs are also volatile. As mentioned earlier, cost is key to the entire optimization procedure. Working on real-world projects is an important approach for collecting cost data.

CONCLUSION

An optimization tool has been implemented to evaluate the cost and performance trade-offs to support decision-making on commercial building design projects. The tool currently requires considerable computing resources and is intended for in-house research to assist in DOE-funded research in support of the goal of zero-energy buildings. As computing power increases, developing a non-research version of the framework for practitioners in the private sector will become more feasible. Collecting and verifying cost and performance input data remain important challenges. The sequential substitution search algorithm is effective; however, improved search algorithms and more efficient methods of validating them are needed.

ACKNOWLEDGMENTS

This work was supported by the Building Technologies Program within the Office of Energy Efficiency and Renewable Energy at the U.S. Department of Energy. The authors would like to thank the numerous people who contributed effort, ideas, and reviews to the study, including Ron Judkoff, Craig Christensen, Mary Colvin, and Michael Deru.

REFERENCES

Caldas, L.G. and Norford, L. 2003. "Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems." *Journal of Solar Energy Engineering*. August 2003, Vol. 125, pp. 343−351.

Christensen, C. et al. 2006. "BEOpt: software for identifying optimal building designs on the path to zero net energy." *Proceedings of ISES 2005 Solar World Congress*, Orlando, FL, August 6−12, 2005.

Coley, D.A. and Schukat, S, 2002. "Low-energy design: combining computer-based optimization and human judgment." Building and Environment 37: 1241−1247.

Crawley, D.B. et al. 2004. "EnergyPlus: an update." *Proceedings of the SimBuild 2004 Conference*, August 4−6, 2004, Boulder, CO.

DEG. 1993. *ACT2 Stanford Ranch Site Final Design Report*. Davis, CA: Davis Energy Group.

FSEC. 2001. EnergyGauge Pro. Cocoa, FL: Florida Solar Energy Center. http://energygauge.com/FlaRes/features/pro.htm.

Griffith, B., and Crawley, D. 2006. "Methodology for analyzing the technical potential for energy performance in the us commercial buildings sector using detailed energy modeling." *Proceedings of the SimBuild 2006 Conference*, August 2006.

Huang, W., and Lam, H.N. 1997. "Using genetic algorithms to optimize controller parameters for HVAC systems." Energy and Buildings 26 (1997), pp. 277−282.

Meier, A.K. 1982. *Supply Curves of Conserved Energy*. Lawrence Berkeley National Laboratory, LBL-14686.

Papamichael, K. 1993. "The limits of intelligence in design." For the 4th International Symposium on System Research, Infomatics and Cybernetics, Baden-Baden, Germany, August 3−4. Also LBL-31742.

RSMeans. 2004. *2005 Cost Data*. RSMeans. Kingston, MA. Along with other titles including Assemblies Cost Data, Square Foot Costs, Facilities Maintenance and Repair Cost Data, Green Building: Project Planning & Cost Estimating.

Thain D., Tannenbaum, T., and Livny, M. 2005. "Distributed Computing in Practice: The Condor Experience." *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2−4, pp. 323−356.

Wetter, M. 2004. *GenOpt Generic Optimization Program User Manual, Version 2.0.0*. Berkeley, CA: Lawrence Berkeley National Laboratory.

Wetter, M. and Wright, J. 2003. "Comparison of a generalized pattern search and a genetic algoritm optimization method." Eighth International IBPSA Conference. August 2003, Eindhoven, The Netherlands.

Wright, J.A., Loosemore, H.A., and Farmani, R. 2002. "Optimization of building thermal design and control by multi-criterion genetic algorithm." Energy and Buildings 43 (2002) 959−972.

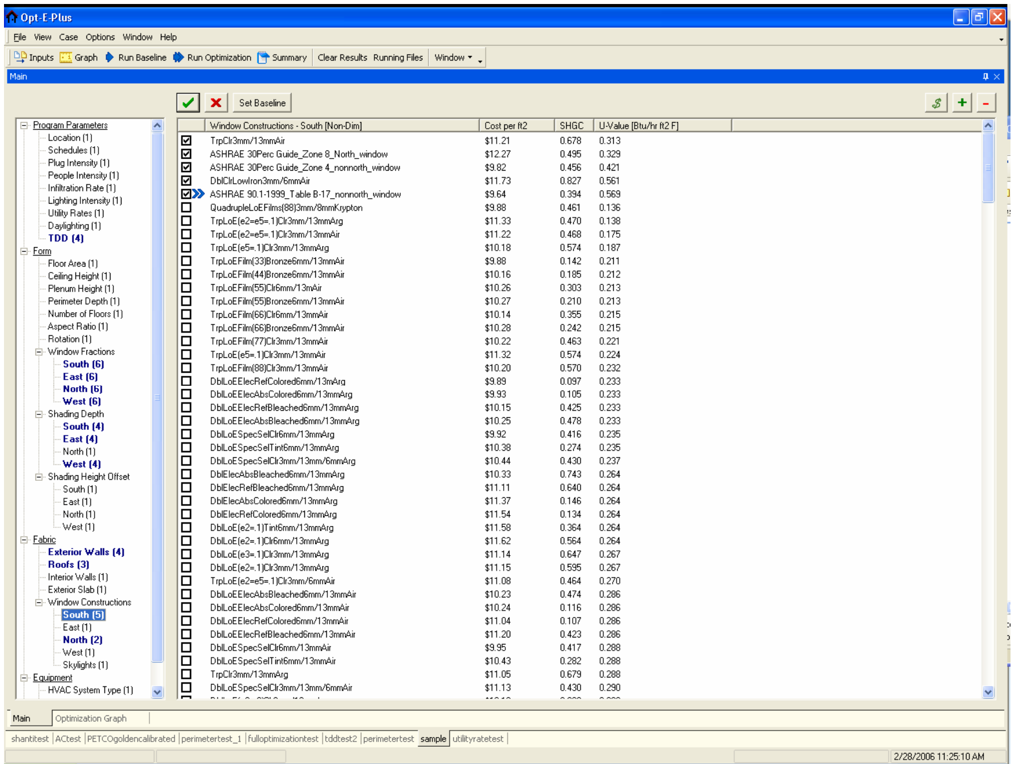

Figure 3 Screen shot of the options browser

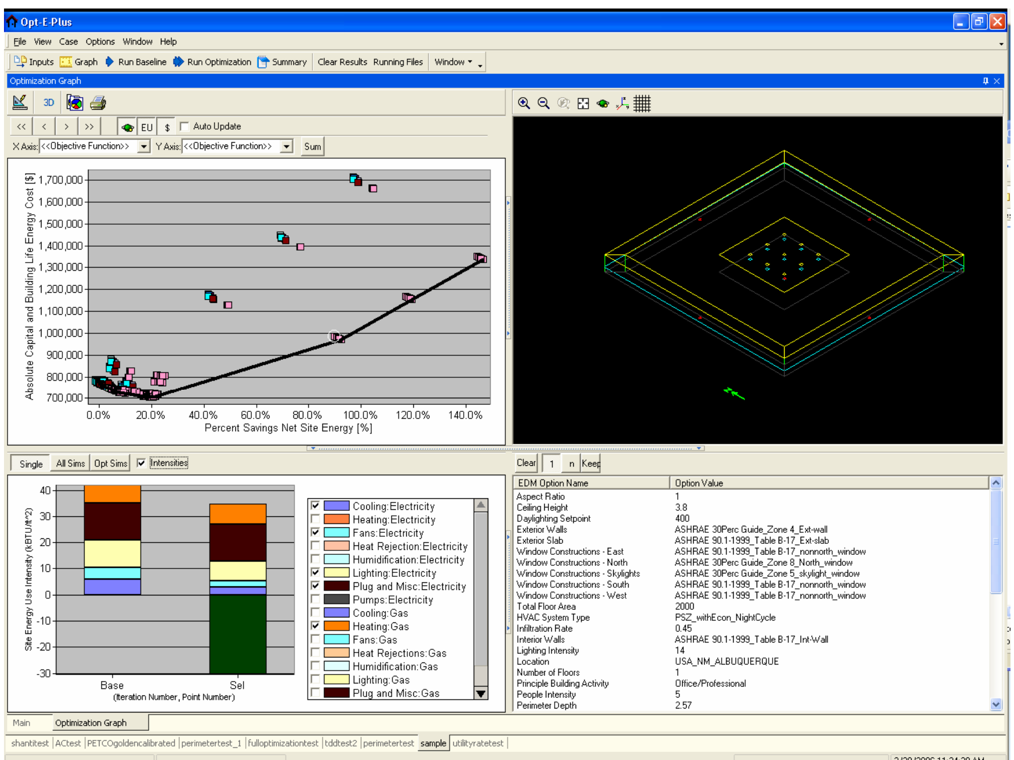

Figure 4 Screen shot of the results browser