MULTIOBJECTIVE OPTIMIZATION OF BUILDING DESIGN USING GENETIC ALGORITHM AND ARTIFICIAL NEURAL NETWORK

Laurent Magnier†, Liang Zhou, Fariborz Haghighat[1]
[1] *BCEE department Concordia University, Montreal, Canada*

ABSTRACT

Building optimization, especially using multiple objectives, is a time-consuming process. The GAINN approach presented in this study first uses simulation-based artificial neural network to characterize building behaviour, and then combines it with a genetic algorithm for optimization. This process has proven to enable fast and reliable optimization.

GAINN was improved in this study by integration of multiobjective evolutionary algorithms (MOEAs). Two new MOEAs named NSGAINN and PLAGUE were designed specifically for the presented methodology. They are both based on NSGA-II but take advantage of extremely quick calculations. They were tested over bench test functions, and compared with NSGA-II based on dominated space measure. Results will be presented and discussed here.

Finally, a past case study using GAINN methodology will be re-optimized with developed MOEA. Improvement in results compared to classical weighted sum method will be shown and discussed.

INTRODUCTION

Modern building design is a complex task, involving many different approaches, parameters, and conflictive objectives. For instance, a building which pretends to be sustainable needs to be at the same time affordable, environment-friendly, and comfortable for occupants. The notion of environment-friendly itself is complex involving not only primary energy consumption but embodied energy, water use, proportion of green energy, etc. Despite this growing complexity, building owners continue to ask for best designs and for alternatives of decision. All this makes building design a very hard task, which should be optimized in a multiobjective point of view. A special care has to be taken and specific tools are to use.

In many cases however, design is based on rules of thumbs and on a limited number of simulations. This method although widely used has two major drawbacks. First, a rather limited range of choices is covered and proposed to final decision maker. Secondly and more important, optimal designs are very unlikely to be reached. A real optimization tool should be used.

As far as optimization techniques are concerned, genetic algorithm (GA) appears to be extremely efficient. It has been used with success in many studies such as Wang et al. (2006) or Huang and Lam (1997). GA however suffers from an important limitation which is the high number of evaluations required to reach optimum. This drawback gets even more important for building simulation, where each simulation can take up to hours.

This paper will present an efficient approach to overcome this limitation and will then expand the methodology to multiobjective optimization.

DESCRIPTION OF GAINN METHODOLOGY

Genetic algorithm (GA) is an optimization technique belonging to the family of evolutionary algorithms. It is based on Darwin's law of evolution and uses reproduction and mutation to improve a population of solutions. It is gradient-free, able to optimize non-differentiable functions, and it can handle conflictive objectives. Despite those qualities, GA use is limited for building design due to the high number of evaluations/simulations required, which makes it time-consuming. This shortcoming has been highlighted in several studies such as Wang et al. (2006).

Artificial Neural Network (ANN) is a response surface approximation method, used to approximate complex behaviour in a simple function. Its structure is based on human brain, using different neurons and weighted

† Corresponding Author, email: lmagnier@gmail.com

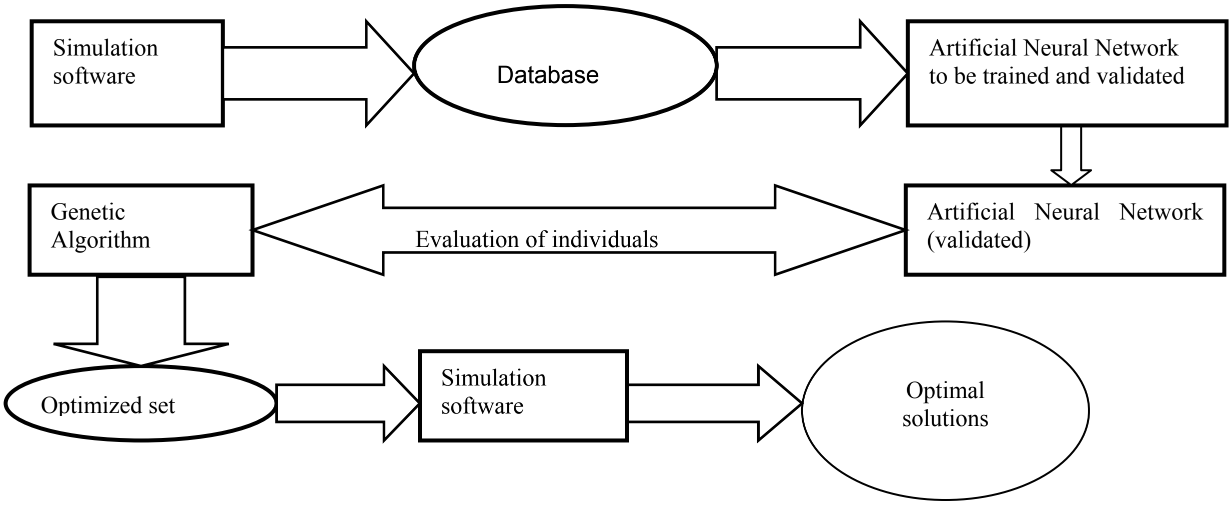

Figure 1: Scheme of GAINN methodology

connections. A set of data is required to train the ANN and validate it; it can afterward approximate a system in a fast and accurate way. ANNs have proven their effectiveness for building applications in the past, such as in Pala et al. (2007). One drawback of ANN is however that approximation function is generally too complicated to be optimized by classical mathematical means.

GAINN stands for Genetic Algorithm Integrating Neural Network. Its basic idea is to first train an ANN to approximate a given system, and then use this approximation as evaluation function inside a genetic algorithm. This enables much faster evaluations and therefore significantly reduces optimization time. The whole methodology is described in figure 1 for a building application. The first step is to select a building and parameters to be optimised. Using a simulation software (which can be either TRNSYS, CFD, EnergyPlus, or any other), a database is created for the artificial neural network. Latin Hypercube Sampling is recommended for that step in order to minimize number of samples while keeping good representation (Lee et al 2006). Using the database, ANN is trained and validated. The next step is optimization by itself using genetic algorithm where evaluation of individuals is done by the ANN, saving a significant amount of time. Finally, the optimized solution set provided by the GA can be tested using original program for accuracy.

GAINN methodology by itself is not new, it has been used for instance in 1993 by Morimoto et al. for plant growth optimization. In terms of engineering and most specifically building engineering however, this method has been widely unexploited. While genetic algorithms are used quite often, few studies have used GAINN to reduce computation time. When used, GAINN demonstrated its ability to reduce computation time significantly; a reduction of as much as 94% has been found for a CFD-based optimization by Zhou et al. (2007b). To author knowledge, GAINN has however never been used combined with a true multiobjective optimization algorithm. The purpose of this study was thus to develop a specific multiobjective genetic algorithm to be integrated in GAINN, and study the overall improvement in the methodology. Two algorithms were created and will be presented in the next section.

DEVELOPMENT OF MULTIOBJECTIVE GENETIC ALGORITHMS

Notion of Pareto-optimality

A real-world problem and particularly an engineering problem can rarely be reduced to a single objective. Several different aspects and issues have to be studied, and therefore should be included in any optimization. Various methods can be used to deal with multiple and conflicting objectives. The most common one consists in aggregating all objectives into one sum, and optimize this sum. This method has been widely used in the past and is still often used due to its simplicity (Xu and Wang (2007) for instance). It however suffers from strong limitations such as:

- It is extremely dependent on weights set for each objective;
- It is dependent on initial situation;
- There is no proof that the set of weights chosen will actually lead to an optimal solution.

The approach used by MultiObjective Evolutionary Algorithms (MOEAs) and which will be used in this study is totally different since it is based on the notion of dominance. This notion was originally proposed by Edgeworth and later generalized by Pareto (1964). It can be described as follows: *"An individual dominates another if and only if it has as good as this one for all objectives, and better for at least one objective"*. It can be described in a mathematical way:

For a multiobjective optimization problem of the form:
Minimize $[f_1(x), f_2(x), \ldots, f_k(x)]$
Where x€F is a vector of decision variables and f_i objectives functions.
A vector x*€F is non-dominated if there does not exist any other vector y€F such that $f_i(x) \leq f_i(x^*)$ for all i and $f_j(x) < f_j(x^*)$ for at least one j.

This definition obviously leads to not only a single solution, but to a set of non-dominated solutions which are all called Pareto-optimal. The final answer of the optimization is thus a curve (or a surface), giving a range of different solutions. As an example, if one wants to optimize energy consumption and building cost, MOEA will provide a solution with the lowest cost, a solution with the lowest energy consumption, and dozens of optimized trade-off in between. Building owner would thus be able to see what would be the consequence of decreasing one objective against another. Therefore, in addition to be independent of assumptions (weights) and of initial situations, MOEAs based on Pareto dominance widen the range of solutions and offers a variety of choices to final decision maker.

First algorithm: NSGAINN

The purpose of creating new MOEAs in this study is to have them specifically designed to take advantage of GAINN fast evaluations. While in classical GAs the number of evaluations is a somehow limiting parameter, MOEAs to be used with GAINN must not limit themselves in terms of that. Evaluations are almost instantaneous and this asset should be used. Further studying classical GAs behaviour, it appears that the most time-consuming step (apart from evaluation) is generally the sorting of population. Based on that, the following algorithms were developed with the idea of significantly increase overall population fitness before proceeding to complete population sorting, no matter how many evaluations are involved.

The first MOEA developed with this idea is NSGAINN, for Non-Dominated Sorting Algorithm for Integrated Neural Network. It is based on Deb's Non-dominated Sorting Genetic Algorithm II (NSGA-II) but differs from it for the last 20% of the optimization. Reader is referred to original study for details about NSGA-II behaviour and genetic operators (Deb 2000). Pseudo-code of NSGAINN can be written as follows:

If CPU time < 80% maximum CPU time
- Similar behaviour as NSGA-II (tournament selection, SBX and polynomial mutation, NSGA-II sorting and selection)

If CPU time ≥80% maximum CPU time
- Select parent by tournament selection
- Produce 4 children by mate (probability of ½ for SBX on each variable)
- Introduce children in population only if they are fit enough (see hereafter)

End

As can be seen, during the last 20% of a run, 4 children are produced by each couple, and only fit-enough offspring are introduced in the population. After creation of offspring, one the two following selection occurs, according to probabilities shown in figure 2 (square ruled):

- Family Selection 1 (FS1): In this selection, offspring are compared with their parents. If they are not dominated by them, they are included in current population.
- Family Selection 2 (FS2): In this selection, offspring are compared with the whole family. Offspring are included in population only if they are non-dominated and if they dominate at least one parent.

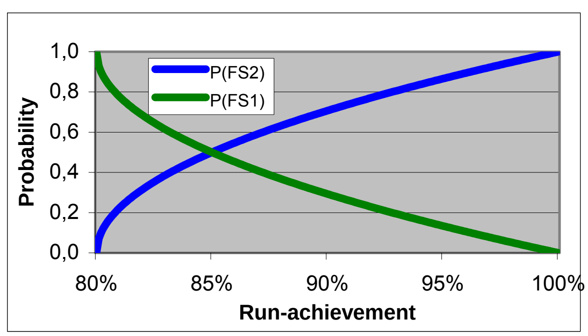

Figure 2: Probabilities for family selection

This behaviour and especially FS2 are expected to force improvement of overall fitness before proceeding to sorting (which is computed when N children are selected, where N is population initial size). Doing this selection only for the last part of optimization ensures that population is close enough to Pareto front, and that the effort of selecting offspring will be both necessary and worthy. Finally, producing 4 children lowers needs for random number in couple selection, explore couple's possibilities more widely, and make family sorting more relevant.

Second algorithm: PLAGUE

The second algorithm created has a different approach. Its name stands for Polyobjective Looped Algorithm using Genetics and Uncompleted Extinction (PLAGUE). It is based on the idea of letting the population progress in fitness and in size, and then applying a drastic reduction, named *plague* referring to the lethal disease. Pseudo-code of the algorithm can be written as follows:

While population size is inferior to five times the initial size (N)

- Generate offspring by recombination and mutation;
- Add offspring if they are not dominated by their parents;
- Remove children-dominated parents from population.

End

When population size is superior or equal to 5 times N: Proceed to "Plague"

- Sort the population based on rank and crowding distance, select the 85%•N fittest individuals except extrema, and include them in next population (for 2 objectives, or 80%•N for 3 objectives);
- For each objective, select the 5%•N best individuals over the entire population and include them in next population;
- Take 5%•N individuals randomly over the entire population, regardless of their rank, and include them in next population.

End

As can be seen, population continues to grow until size has been multiplied by five. This growth is controlled by parental replacement of children-dominated parents, and by inclusion of offspring only if they are not dominated by their parents. The idea is to let population explore different directions, while keeping a constant fitness pressure, and replacing obviously bad individuals. *"Plague"* selection, in turn, ensures fitness improvement, explores more carefully minima, and adds diversity by selecting random individuals. A last and important aspect of PLAGUE is that it works with a slightly smaller initial population than NSGA-II (80 instead of 100). In order to compensate this, the very last *plague* sorting keeps all non-dominated individuals in the solution set. This leads to final solution sets composed of up to 400 individuals.

ANALYSIS

This section will study efficiencies of proposed algorithms compared with that of NSGA-II. NSGA-II has been chosen as base MOEA due to its good performance for most tests problems. Regarding termination criterion, comparisons based on a fixed number of generations are often used but are not possible here due to MOEAs behaviour. This would be unfair and advantage PLAGUE too much. Comparisons based on number of evaluations would of course be meaningless in the present case. It was finally decided to perform tests on a CPU-time basis. Despite some shortcomings, this criterion is assumed to be relevant since all three algorithms are based on a mostly similar code and simulations are done on a same computer.

Metric used for comparison is Zitler's S-metric, representing the amount of space dominated by solution set (Zitler and Thiele 1998). The main advantage of this metric is that it is able to represent at the same time convergence and spreading of solutions. The higher S-metric is, the better the solution set is. It should however be noted that maximum reachable value is problem dependent (different for each test function) and never expected to be 100%. S-metric is here computed based on a random sampling of one million points in a volume with bounds equal to true Pareto front's bounds. In order to remain fair for the three algorithms, the metric will be calculated two times for PLAGUE, first with the actual solution set, and then with a solution set reduced to 100 points. Algorithms parameters are summarized in table 1.

	NSGA-II	NSGAINN	PLAGUE
Population size	100		80
Crossover type	Simulated Binary Crossover with η=20		
Crossover probability	0.9		
Mutation type	Polynomial mutation with η=20		
Mutation probability	1/(number of variables)		
Termination criterion	Time : 15 seconds for ZDT suite (30 for ZDT6) 120 seconds for DTLZ suites		
Number of variables	10 variables for all problems except for ZDT 1,2&3 (30 variables)		

Table 1: Parameters of tests

Tests functions are two-objective ZDT suites (Zitler et al. 2000) and three-objective DTLZ suites (Deb at al. 2001), recognized to be challenging in many ways. Each test is done 10 times and the average of results is taken. Run-times were deliberately chosen very short in order to see differences between MOEAs. All runs were computed on a computer with Genuine Intel(R) CPU T2300 @1.66GHz and 1GB of RAM, equipped with Windows XP. MOEAs were programmed in MATLAB 7 with a mostly common code (and same possible weaknesses).

RESULTS

Results are summarized in Table 2 and plotted in figures 3 and 4. PLAGUE algorithm performs better than the two others for bimodal functions, and generally better for three-objective ones (higher S-value). Efficiency is influenced by number of objectives because for three-objective problems individuals are more likely to be non-dominated, and thus PLAGUE family sorting may become less efficient. Regarding size of solution set, it does not appear to influence S-metric for two-objective problems (PLAGUE and PLAGUE100 have very similar results), but it is of significant importance for DTLZ suites. Overall, PLAGUE constitutes a significant improvement compared to NSGA-II for both convergence and exploration of objectives space.

NSGAINN is generally less efficient than PLAGUE but still better than NSGA-II. In very complex problems such as ZDT6, DTLZ 5, or DTLZ6, it gets better than the two others due to its behaviour forcing improvement of overall fitness. NSGAINN is expected to get extremely efficient for very complex objective functions. This kind of function may appear using ANN approximation.

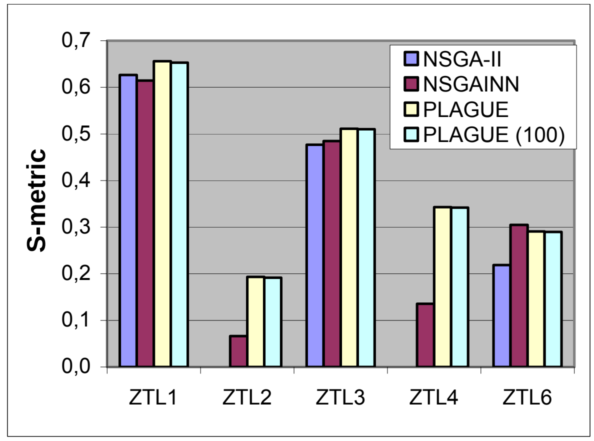

Figure 3: S-metric for the three MOEAs for ZDT suites

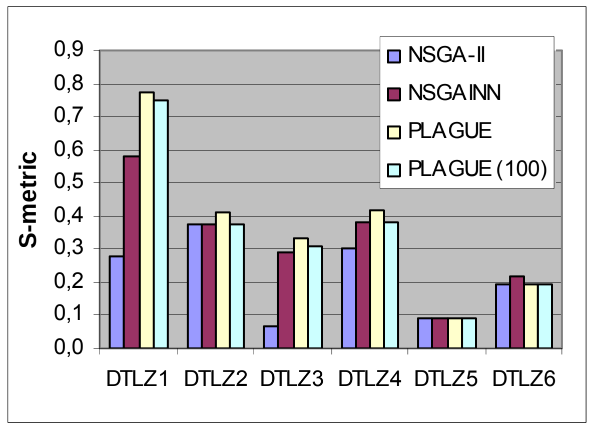

Figure 4: S-metric for the three MOEAs for DTLZ suites

	S-metric (dominated space)				Number of evaluations		
Function	NSGA-II	NSGAINN	PLAGUE	PLAGUE (100)	NSGA-II	NSGAINN	PLAGUE
ZTL1	0.626	0.615	0.656	0.653	7600	6796	14870
ZTL2	0.000	0.066	0.193	0.192	4406	4451	27306
ZTL3	0.477	0.485	0.511	0.510	7830	7222	15830
ZTL4	0.000	0.135	0.343	0.342	5520	6546	21097
ZTL6	0.219	0.305	0.291	0.290	13175	18160	35036
DTLZ1	0.279	0.581	0.775	0.751	58082	79344	163760
DTLZ2	0.376	0.375	0.410	0.374	81026	105550	100640
DTLZ3	0.065	0.289	0.334	0.310	60935	80654	172070
DTLZ4	0.303	0.382	0.418	0.383	71156	104120	99269
DTLZ5	0.092	0.093	0.093	0.091	79281	103610	123590
DTLZ6	0.195	0.219	0.196	0.193	79737	112760	107530

Table 2: S-metric and number of evaluations for NSGA-II, NSGAINN, and PLAGUE (standard and reduced to 100 solutions)

Regarding number of evaluations, both PLAGUE and NSGAINN require more of them than NSGA-II (up to 6 times more for PLAGUE). This was expected and somehow wished, since those are specifically designed to be integrated into GAINN methodology where evaluations are extremely fast. Nonetheless and quite surprisingly, NSGAINN was found in one case (ZTL3) to require fewer evaluations than NSGA-II, for better results.

SIMULATION

In the last part of this paper, we will propose an application of the improved methodology on a previous study. That study has been previously undertaken by Zhou (2007a, 2007b, 2007c), and was based on GAINN methodology. Only final optimization part will be discussed here, comparing results using original weighted-sum method with results using proposed MOEAs.

Design to be optimized is ventilation system of a standard office room in summer, with two occupants and four underfloor air distribution (UFAD) diffusers (figure 5). Objectives studied are thermal comfort represented by Predicted Mean Vote (PMV), Indoor Air Quality represented by ventilation effectiveness ε_v, and energy consumptions for cooling and for fans ($E_{cooling}$ and E_{fan}). Control variables are temperature of supply air T_s, speed of supply air V_s, distance from diffuser to occupant, and distance from return grill to contaminant source.

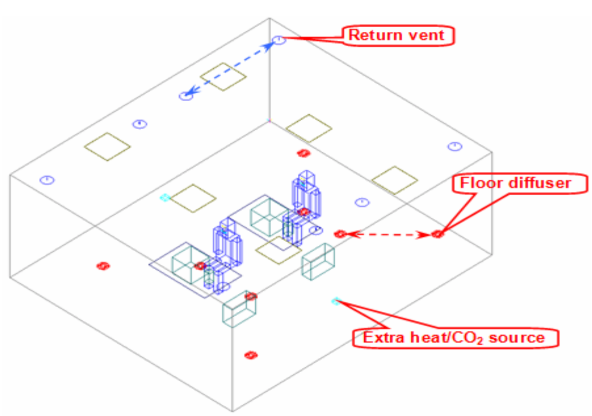

Figure 5: Office room used in Zhou study

In the original study, Zhou used Computational Fluid Dynamics (CFD) to simulate the room. Those CFD simulations were validated with experimental values, and created a database, using Latin Hypercube Sampling method. An ANN was then trained by this database, validated, and implemented in a classical GA.

This GA was run for a maximum of 100 generation, with a population of 100 real coded individuals, using Scattered Crossover with a rate of 0.8, Uniform Adapted-feasible Mutation with a rate of 0.2, an Elite size of 2, stochastic uniform parental selection and fitness-based replacement for survivor selection. Function to optimize was the following weighted sum, where weights were $w_{pmv} = 0.5$, $w_{iaq} = 0.25$, $w_{cooling} = 1$, and $w_{fan} = 0.5$, and where PT represents penalty terms due to possible local discomfort.

$$Min \left[\begin{array}{l} w_1\left(\sum_i \left(\frac{|PMV_i|}{PMV_{\max}} \right) \right) + w_2 \frac{\varepsilon_{v\max}}{\varepsilon_v} + \\ w_f\left(\frac{E_{fan}}{E_{fan\max}} \right) + w_4\left(\frac{E_{cooling}}{E_{cooling\max}} \right) + PT \end{array} \right]$$

For the current study, NSGA-II, PLAGUE, and NSGAINN will be applied, with the same parameters as before, on the ANN found by Zhou. Problem is simulated as a three-objective optimization, studying PMV, ventilation effectiveness, and energy consumption combining consumptions of cooling and fan. Time limit is set to 60 seconds.

RESULTS

With a 60 seconds run-time, there is almost no difference between solutions found by PLAGUE, NSGA-II and NSGAINN. The main purpose of this section was not to compare the three MOEAs with each other but to compare MOEA with GA using weighted sum. For clarity purpose, only original study's and PLAGUE's solution sets are displayed. Solution set of original study is actually limited to one single point since that study was based on weighted-sum with one specific set of weights, leading to only one so-called optimal solution. Figures 6 to 9 plot respectively PMV Vs energy consumption, ventilation effectiveness VS energy consumption, PMV Vs ventilation effectiveness, and a 3D view of those three. Energy consumption corresponds to sum of $E_{cooling}$ and E_{fan}.

Regarding first relevancy of the results from a physical point of view, displayed shapes are very logical. For instance, variation of PMV with energy decrease is the same as one could expect, with the lowest PMV for higher energy cost and *vice versa*. Similarly, ventilation effectiveness has a relevant variation with energy, with increase of effectiveness as energy consumption decreases. (One should only look at the bottom line of

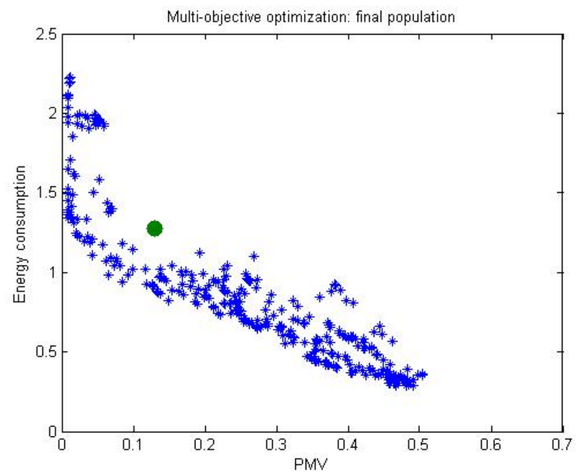

Figure 6: PMV Vs Energy consumption

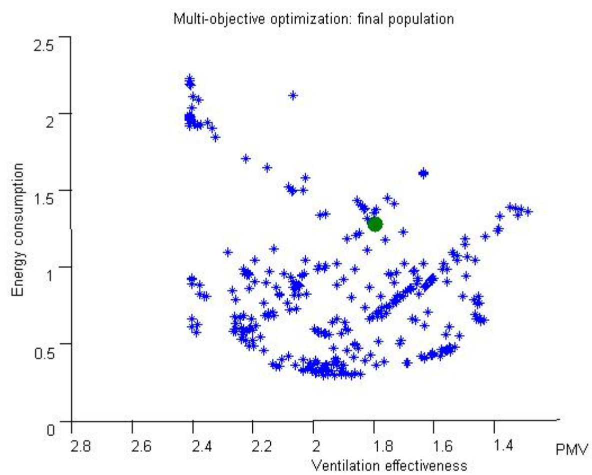

Figure 7: Ventilation effectiveness Vs Energy consumption

points, since other points are due to 3D projection). Even if those variations cannot totally validate the model, they show that ANN behaves properly, and that GAINN gives relevant results.

Regarding efficiencies of the two optimization methods, we can see in the figures that solution from original weighted-sum optimization is dominated by MOEAs solution sets. It is worse at the same time in terms of PMV, energy consumption, and ventilation effectiveness. This result is logical since even if Zhou undertook sensitivity analysis (i.e. trying different sets of weights) to find a good trade-off, this process is nothing but trial-and-error from a multiobjective point of view, leading to possibly good but rarely optimal solutions. MOEAs are true multiobjective optimization and will always give better results than classical GAs. It is however necessary to highlight that in the present optimization, penalty terms due to local

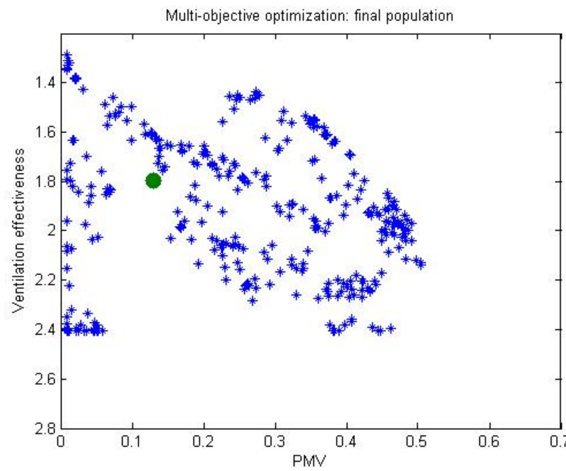

Figure 8: PMV Vs Ventilation effectiveness

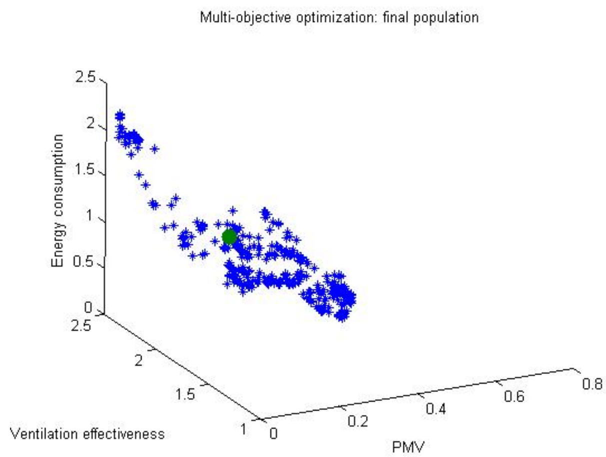

Figure 9: PMV Vs Ventilation effectiveness Vs Energy

discomfort (local draft mostly) were not included. After carefully checking the solution set though, local discomfort does not seem to appear for any configuration found.

In addition to his, the most obvious and important improvement from using MOEAs is the range of results, which is not limited to one point but counts at least one hundred of them. From a practical point of view, it means that in a single run MOEA provides one hundred different designs, all Pareto-optimal. This is a demonstration of multiobjective optimization capability in terms of choices, and global vision of a problem. A designer looking at these results is able to know what would be for instance the consequence of lowering energy consumption in terms of comfort and ventilation effectiveness, or vice versa. There is therefore an improvement not only in terms of number of solutions, but more generally in terms of information given to building designers and owners. Building owners can have a clear view of the problem and base their final decision knowing the problem completely. The time costly optimization is also much more attractive since more results are provided.

CONCLUSIONS

This paper presented GAINN methodology, integrating artificial neural network and genetic algorithm to save computation time. This methodology was enhanced by use of multiobjective genetic algorithm. Two new algorithms were created specifically for GAINN: NSGAINN, and PLAGUE. These GAs were tested against NSGA-II and both displayed a significant improvement in results. PLAGUE was found to be generally the best of the three MOEAs while NSGAINN has shown great abilities for very complex three-objective functions. Finally, developed algorithms were applied on a ventilation design coming from a previous study. Results display important improvement in both convergence and range of choices compared to weighted sum GA.

REFERENCES

Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T. (2000). ‘A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II’ KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

E. Zitler, K. Deb, and L. Thiele (2000), ‘Comparison of Multiobjective Evolutionary Algorithms: Empirical Results’. *Evolutionary Computation Journal*, 2(8)125--148.

Huang, W.; Lam, H. N. (1997). ‘Using genetic algorithms to optimize controller parameters for HVAC systems’, *Energy and Buildings*, 26 277-282.

Lee et al. (2006), ‘Comparison of Latin Hypercube Sampling and Simple Random Sampling Applied to Neural Network Modeling of HfO2 Thin Film Fabrication," *Transactions on Electrical and Electronic Materials*, 7-4, 210-214.

Morimoto, T.; Takeuchi, T.; Hashimoto, Y., (1993), ‘Growth optimization of plant by means of the hybrid system of genetic algorithm and neural network,’ IJCNN '93-Nagoya. *Proceedings of 1993 International Joint Conference on Neural Networks*, 3 2979-2982

Pala, M., Caglar, N., Elmas, M., Cevik, A., and Saribiyik, M. (2008), ‘Dynamic soil-structure interaction analysis of buildings by neural networks’, *Construction and Building Materials* 22-3 330-342.

Pareto V., (1964). *Cours D’économie politique*, volume I and II. F. Rouge, Lausanne

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

Xu X. and Wang S. (2007), ‘Optimal simplified thermal models of building envelope based on frequency domain regression using genetic algorithm’, *Energy and Buildings* 39 - 5 525-536.

Zitzler, E., and Thiele, L. (1998), ‘Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Case Study,’ In Eiben, A. E., et al., *Proc. 5th International Conference: Parallel Problem Solving from Nature—PPSNV*, Amsterdam, The Netherlands, Springer, pp. 292–301.

Zhou, L. and Haghighat, F. (2007a), ‘Optimization of ventilation systems in office environment, Part I: methodology’, *Proceedings of IAQVEC conference*, Sendai, Japan.

Zhou, L. and Haghighat, F. (2007b), ‘Optimization of ventilation systems in office environment, Part II: results and discussions’, *Proceedings of IAQVEC conference*, Sendai, Japan.

Zhou, L. and Haghighat, F. (2007c), ‘Optimization of ventilation systems in office environment, Part III: near optimal solutions and discussions’, *Proceedings of IAQVEC conference*, Sendai, Japan.