COMBINATION OF OPTIMISATION ALGORITHMS FOR A MULTI-OBJECTIVE BUILDING DESIGN PROBLEM

Mohamed Hamdy, Ala Hasan and Kai Siren
HVAC Technology, Helsinki University of Technology, Espoo, Finland

ABSTRACT

Recently, a combination between simulation and optimisation has been important for many of HVAC design problems. Long execution time is usually needed for one simulation run; consequently huge time will be required for the optimisation process. The aim of this study is to evaluate how combinations between optimisation algorithms can achieve faster and/or better solutions for multi-objective optimisation problems.

Two optimisation approaches are suggested and tested for an HVAC-building optimisation problem. These approaches are based on combinations between two deterministic algorithms and a genetic algorithm using MATLAB environment. The results indicate that significant time could be saved by applying these approaches compared with using genetic algorithm alone.

INTRODUCTION

Multi-objective optimisation produces a range of optimal solutions that gives an opportunity for the decision maker to select the appropriate solution(s). Using multi-objective optimisation in the first stages of building and HVAC system design would allow the designer to explore some favourable concepts, which could be out of the traditional way of design. Such solutions can produce savings in a multi-objective problem (e.g. energy consumption, investment cost or both) compared with a reference design. This is normally done by combining a simulation tool with an optimisation tool. One of the major problems that limits the use of simulation-based optimisation is the long execution time needed for simulation because most of multi-objective optimisation algorithms need large numbers of evaluations (simulation-runs) to obtain feasible solutions.

Most of researches focus on the optimisation process aiming to attain reasonable results disregarding the consumed time. That may be acceptable in academic studies. However, it does not match the practical implementation. It is important to create new generations of energy building simulation-optimisation tools able to give a range of optimal solutions for designer in short time.

The purpose of this study is to evaluate the benefits from combining optimisation algorithms to achieve faster and/or better solutions for building and HVAC system design problems. For this purpose, a combination between simulation and optimisation is first created by combining IDA ICE 3.0 (Building performance simulation program) with MATLAB 2008a optimisation tools. Then combinations of deterministic and genetic optimisation algorithms are investigated for the quality of the results and the required execution time compared with using genetic algorithm alone.

Various studies suggested different methods to avoid the need for long time in the optimisation process. For example, Nielsen (2002) used a simple thermal model instead of using building simulation program to perform fast yearly energy analysis and thermal simulations on buildings using a limited amount of input data describing the building constructions and systems. Hasan et al. (2008) considered the detached house as a single zone in the simulation carried out by IDA ICE 3.0 (Building performance simulation program) reducing the time of simulation, and therefore, reducing the total time needed for the simulation-optimisation process.

The current study proposes two combinations between optimisation algorithms (PR-GA and GA–RF) attempting to reduce the time of optimisation process by lowering the number of simulation-runs. This feature allows dealing with complicated and detailed building design problems.

Abbreviations

dIC	Difference in the initial investment cost (€)
GA_RF	Combination between optimisation algorithms: genetic algorithm with refine process.
Gen	Generation
Pop	Population
Pre	Preparation
PR_GA	Combination between optimisation algorithms: preparation process and genetic algorithm.

IDA ICE – MATLAB COMBINATION

''Using commercial optimisation libraries reduces the tedious task of testing and benchmarking of algorithms needed for hard-coded implementation'' (Mourshed et al., 2003). MATLAB optimisation libraries include many effective optimisation algorithms. In addition, using MATLAB gives the designer a good opportunity to use the other features available in MATLAB environment such as (Excel Link, Database, data analysis, plotting functions, curve fitting functions, GUI graphical user interface, etc.).

For less simulation efforts with more facilities, it is useful to use available building simulation tools. IDA ICE 3.0 is whole-building dynamic simulation program that makes simultaneous performance assessments of all issues fundamental to building design: shape, envelope, glazing, HVAC systems, controls, lighting, indoor air quality, thermal comfort, energy consumption, etc. IDA ICE 3.0 has been chosen as one of the major 20 building energy simulation programs, which were subjected to analysis and comparison in (Crawley DB et al., 2005).

Therefore, simulation-based optimisation is performed in the current study by combining IDA ICE 3.0 with MATLAB.

OPTIMISATION PROBLEM

In order to test the suggested two optimisation approaches, PR-GA (preparation process and genetic algorithm.) and GA–RF (genetic algorithm with refine process), a multi-objective optimisation problem is formulated. The aim is minimisation of the annual energy consumption needed for space heating and the difference in the initial investment cost (dIC) for a single-family detached house.

Five design variables are selected to be optimised: three continuous variables (insulation thickness of the external wall, roof and floor) and two discrete variables (U-value of the windows and type of heat recovery).

The house is considered as a single zone with initial U-value in accordance with the Finnish National Building Code (C3, 2003). There is a heating system in the house. No cooling system is considered, this is a typical case for Finnish houses. The heating system is a direct electric heating system where heating energy is supplied by two means: electric radiators inside the zone and an electric heater in the air-handling unit (AHU). The heating system is always ON with a set temperature of 21°C. Further details and descriptions of the detached house and the reference case can be found in (Hasan et al., 2008).

Genetic algorithm (Deb, 2001) in MATLAB Genetic and Direct Search Toolbox is employed in the current study. GA is the core of the two suggested optimisation approaches (PR-GA and GA–RF). Furthermore, GA is used alone, for comparison purpose. For all studied cases presented in the current paper, GA is implemented with crossover fraction 0.9 and elite count 2.

All the optimisation algorithms (GA, Fmincon, and Fminimax), implemented in the current study, are developed to be able to deal with the two types of variables (discrete and continuous). Table (1) presents the design-variables types and their nominal, minimum and maximum values.

Table (1)
Design variables

Design Variables	Type	Nominal value	Min. Value	Max. Value
Wall Insulation Thickness (m)	Continuous	0.122	0.122	0.522
Ceiling Insulation Thickness (m)	Continuous	0.299	0.299	0.799
Floor Insulation Thickness (m)	Continuous	0.165	0.165	0.565
U-Values of the Windows (W/m^2K)	Discrete (two options)	1.4	1	1.4
Heat Recovery Efficiency (%)	Discrete (two options)	70	70	80

A brute-force search method was implemented in (Hasan et al., 2008) to check the results obtained by optimisation. This brute-force is also used in the current study for the same purpose. The brute-force search is an exhaustive search that systematically enumerates all possible candidate solutions. In order to make the brute-force search feasible, the previous study (Hasan et at., 2008) limited the size of the problem using some indications from the optimisation results. In this way, Hasan et al. (2008) succeeded to obtain a feasible brute-force which covers the effective range of solutions (heating energy range was from 8340 kWh/a to 14498 kWh/a where the maximum and minimum corresponding difference in investment cost were 5548 and zero Euro respectively). Fig.1 presents the feasible brute-force and the bounds of the space-solution as well as utopia point. This figure indicates that the investment to reduce the space heating energy less than 8340 kWh/a is not effective. Therefore, the value (6000 Euro) is considered as a maximum budget for the investment in the current study.

The default Genetic algorithm (Deb, 2001) in MATLAB Genetic and Direct Search Toolbox cannot deal with constraint functions. Therefore, 6000 Euro is used as a constraint function through the deterministic algorithms that are suggested to be combined with the genetic algorithm. Using constraint function through the suggested approaches helps to explore in restricted space of solution searching for an optimal or near optimal solutions.

The maximum and minimum values of the design variables (Table 1) are used as upper and lower bounds for the optimisation problem. These bounds give space-solution, which are wider than the feasible brute-force.

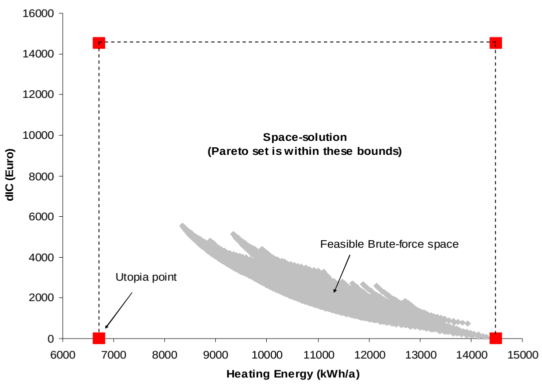

Figure 1: Spaces solution and brute-force

FIRST APPROACH (PR_GA)

Optimisation and Simulation

In this section, PR-GA approach is introduced where PR denotes preparation and GA denotes genetic algorithm (Genetic and direct search toolbox, MATLAB 2008a).

By default, the genetic algorithm creates a random initial population using a creation function. The next generation of the population is computed using the non-dominated rank and a distance measure of the individuals in the current generation. In other words, the next generation will track relatively the first generation. Random creation for the initial population and the dependence on this random behaviour usually need a large number of trials in order to achieve good results.

Setting of good initial population could produce best fitness at each generation and little diversity for the algorithm. In addition, it makes the algorithm to focus on specified area of the space-solution, which is near to the optimal Pareto-front. The main problem is how to prepare this initial population and how long time required for that. Using deterministic algorithm, to generate calculated initial population, can overcome the disadvantage in the random initial population of GA. Therefore, deterministic algorithm is proposed to run before the multi- objective genetic algorithm in order to prepare trusted initial population for the GA.

High quality results are not instantly required in this stage, where the GA will complete the optimisation task. Stopping criterion with low quality of deterministic algorithm results can limit the time consumed in the preparation phase. Five cases are suggested to compare the quality of the results obtained by using GA with its random initial-population (cases 2, 3, 4, and 5) and PR_GA with its calculated initial-population (case 1). These cases are summarised in Table (2).

Table (2)
Optimisation algorithm settings

Case	Algorithm	No. Pre Run	Size Pop	No. Gen	Number of Simulation Runs*
1	PR_GA	207	36	10	567
2	GA	0	36	16	576
3	GA	0	20	30	600
4	GA	0	20	55	1100
5	GA	0	25	50	1250

**Number of Simulation Runs = Pre + (Pop X Gen)*

PR_GA approach has been adopted in case1. In the first step; PR_GA algorithm calls Fmincon solver, from the MATLAB 2008a optimization toolbox (Waltz, 2006), to minimise the first objective (space heating energy) using the upper limit of the second objective as a constraint function. The upper limit of the dIC (second objective) corresponds to the maximum additional budget (6000 Euro). In the second step, Fmincon is used to minimise the second objective (dIC) under the upper limit of the first objective (heating energy). The upper limit for the heating energy corresponds to the maximum requirements of the heating energy in this problem. Selection of a proper stopping criterion is a very important issue in the phase. Fast stopping criterion is implemented since the quality of the results is not the target of this step.

Special programming code has been developed to record all the iterations occurred during the previous optimisation process. The developed code ranks the iterations and selects some of them to be initial values for Fminimax function (multi-objective optimisation function provided in MATLAB 2008a optimisation toolbox) in order to create a number of new optimal solutions as close as possible to the Pareto front.

Fminimax function uses the two ‘’near to optimum’’ points, obtained in the first two steps, as an indication for the upper and lower bounds for the Pareto-front. However, that does not mean that Fminimax is restricted to explore in between the two mentioned points. These two points just help the Fminimax to survey in a predictable range of optimal solutions.

Using Fmincon and Fminimax functions, as demonstrated above, is called preparation phase. In the current approach, preparation phase consumed 207 simulation-runs. Then the developed ranking-code picked out 36 individuals as a good initial population for the next phase (GA phase). After that, GA performed 10 generations consuming 360 simulation-runs Table (2). In this way, 567 simulation-runs were needed to produce 106 optimal decisions on the local Pareto-front of case 1.

On the other hand, GA performed in the next cases 2, 3, 4, and 5 using random initial population created by the default creation function. The same size of population (36), as in case 1, was selected for case 2. Case 2 consumed 576 simulation-runs in 16 generations. 600 simulation-runs (20 Pop X 30 Gen) were executed in case 3. Theses numbers of simulation-runs are close to the number of simulation-runs executed in case 1 (567 simulation-runs). Therefore, it is reasonable to compare between the results of these first three cases; see Fig. 2.

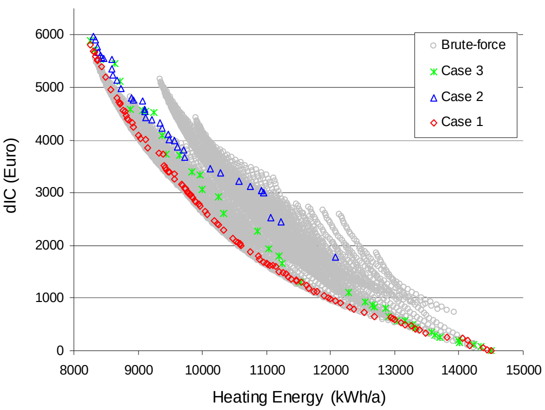

Figure 2: PR_GA results compared with GA results using close number of simulation-runs

In supplementary tests, the proposed approach (case 1) was tested against two cases (case 4 and case 5) which have large number of simulation-runs; 20 Pop X 55 Gen = 1100 simulation-runs for case 4 and 25 Pop X 50 Gen = 1250 simulation-runs for case 5. Larger number of generation and less size of population were implemented in the last two cases (case 4 and case 5) trying to limit the effect of the random initial population. The comparison between the results of suggested approach (case 1) and cases 4 and 5 are presented in Fig.3.

Decision and result analysis

Fig. 2 and 3 give a visual base to judge the quality of the results produced in the five cases. In case 2, the GA used its default creation function to create the initial population. This function creates a random initial population. Furthermore the optimisation process is performed without constraint function. As a result, most iterations were out of the feasible brute-force area (see Fig. 4). In case 3, increasing the number of generations was the idea to avoid the above-mentioned problem. More number of generations was assumed in cases 4 and 5 to get feasible results covering the whole brute-force area.

Final Pareto is a Pareto-front created from the aggregation of all obtained solutions of the five cases based on non-dominated sorting code. Final Pareto picks up the best solutions from all obtained decisions; see Final Pareto Fig. 5. The participation of the solutions in the final Pareto is proposed to be the numerical criterion to evaluate the capability of the cases to achieve good quality of results. Table 3 shows the number of optimal solutions provided by each case on its local Pareto as well as how many of these solutions participate in the Final Pareto.

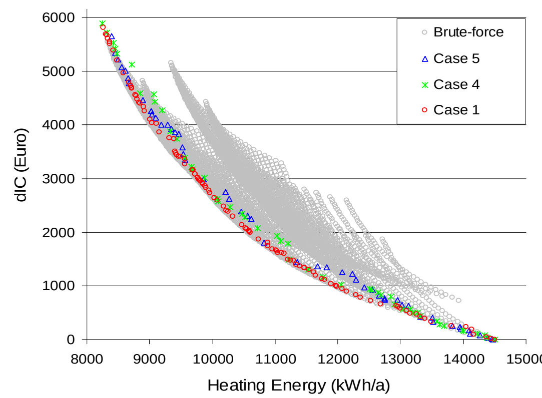

Figure 3: PR_GA results compared with GA results using higher number of simulation-runs

Figure 4: GA (case 2) works out of feasible brute-force

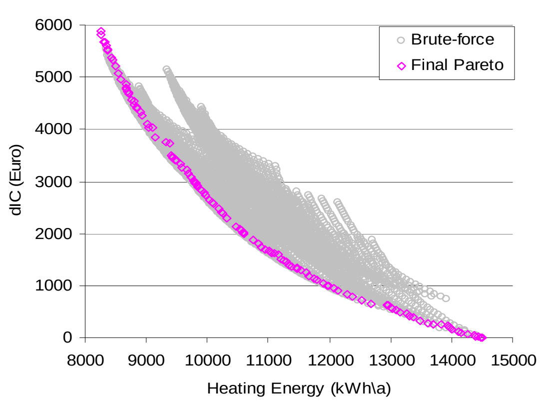

Figure 5: Final Pareto for the five cases

Table (3)
Local Pareto and Final Pareto

Case	Algorithm	The Optimum on the local Pareto	The Optimum on the Final Pareto
1	PR_GA	106	100
2	GA	37	0
3	GA	40	6
4	GA	49	10
5	GA	54	13

Fig.6 shows the participation of each case in the Final-Pareto. Proposed approach (case 1) participated with the largest number of solutions in the Final-Pareto (100 optimal decisions) although it has the lowest number of simulation-runs (567). Fig. 7 shows the execution time elapsed in each case. Fig. 6 and 7 present the advantage of the suggested approach (case 1) on the other four cases (case 2, 3, 4, and 5).

The preparation phase succeeded to suggest a proper population size for the second phase (GA algorithm). Since the combination between the preparation and genetic algorithm can be done automatically, the less expert user can use this approach without need to assume the size of population.

Furthermore, the proposed approach can determine the minimum acceptable size of population and give the user the ability to select another size higher than the minimum, if large number of optimal solutions is required. That can save huge time in unacceptable optimisation trails.

In Table 3, the comparison between case 1 and case 5 indicates that 100 optimal solutions on the Final-Pareto come from the local Pareto of case 1 where only 13 optimal solutions come from the local Pareto of case 5. Since the elapsed time of one simulation-run is equal to 50 sec, using the proposed approach can conserve 570 (683 simulation-runs * 50 sec) minutes = 9.5 hours, where 683 simulation-runs is the difference between the required simulation-runs for case 5 and case 1 (1250_{case_5} -567_{case_1}).

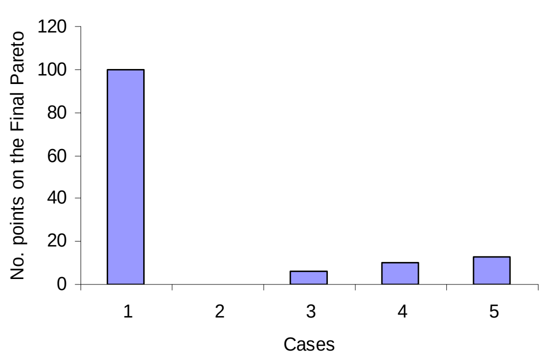

Figure 6: Participation of cases 1, 2, 3, 4 and 5in the Final Pareto

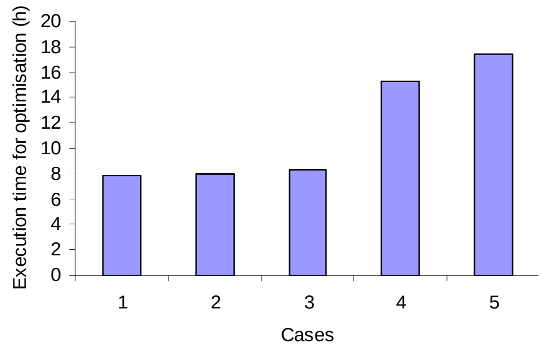

Figure 7: Execution time required for cases 1, 2, 3, 4 and 5

SECOND APPROACH

Optimisation and Simulation

In this section, GA_RF approach is introduced where GA denotes genetic algorithm, RF denotes refine or improving the results.

GA_RF is a combination between GA (Genetic and direct search toolbox, MATLAB 2008a) and sequential quadratic programming (SQP) method to improve or refine some of GA Pareto points as well as to enhance Pareto-front with additional refined solutions. The combination between GA and Fminimax (optimisation toolbox, MATLAB 2008a), is one example for this combination, which has been created through this work. It mostly provides more accurate results than the results produced by using GA alone. The main difference between GA–Fminimax combination (presented in this paper) and GA-Fgoalattain hybrid algorithm (provided in MATLAB toolbox) is that the former works on especial picked out decisions (GA Pareto decisions which have a significant difference in dominating for one of the objectives).

Dealing with the picked out optimal decisions instead of all the solutions of the Pareto makes this approach (GA_RF) using less number of simulation-runs for the purpose of refine the results. In addition, this GA_RF attempts to generate more than one refined solution from each preselected optimal decision based on non-dominated rank for each generated individual. Therefore, it can be said that GA_RF combination not only improves Pareto solutions but also enhances these solutions. Fig. 8 presents how this approach, presented by case 6, can improve the results, obtained by default GA (20Pop X 40Gen), and multiply the results from 38 to 120 solutions on the Pareto front.

For each selected point, Function tolerance (TolFun) specifies the minimum tolerance for the objective function. After a successful poll, if the difference between the function value at the previous best point and function value at the current best point is less than the value of function tolerance, the algorithm halts. Using a proper value for TolFun can avoid many un-useful simulation-runs. For example, asking the optimisation algorithm for high accurate results could need huge number of simulation-runs. Since the two objectives are the dIC and energy consumption, it is evident that improving the results by saving 1 Euro or 1 kWh/a of heating energy does not merit consuming much long time. Many of simulation-runs were saved by using this option through employing the refine approach after GA.

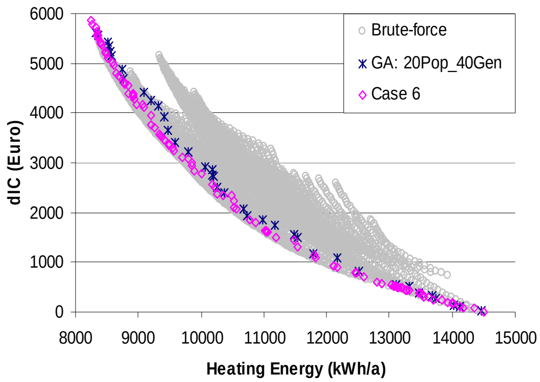

Figure 8: GA_RF can improve the quality of the results

This combination uses modified GA algorithm based on original GA algorithm provided in MATLAB 2008a (genetic and direct search toolbox). This modification has been developed to enable GA and Fminimax to deal with both discrete and continuous variables. In addition, other modifications were required to exchange the design variables between two different optimisation algorithms (GA and Fminimax).

Decision and result analysis

Since the rate of improving the results decreases from a previous generation to a next one using GA, it is important to determine appropriate stopping criterion in order to reduce the needed number of simulation-runs. Actually, it is not always convenient to use maximum number of generation for this purpose. Because the appropriate number of generations depends on the type of the problem, the size of population and the quality of the random initial population, which is not predictable.

On the other hand, using hybrid function with GA could be a good solution for a number of optimization problems. However, it is not recommended for building optimisation problems because it consumes large number of iterations.

In order to test this approach, 49 solutions obtained by GA alone (20Pop_55Gen = 1250 simulation-runs) (case 7) are compared with 120 solutions obtained by GA_RF (case 6) where the number of iterations is 1221 and 1250for case 6 and 7 respectively. The mentioned comparison is presented in Fig.9. Refinement process is stared from the results which are obtained by GA (20PopX 20 Gen = 400 simulation-runs) followed by 821 simulation-runs for refine process by Fminimax. Table 4 shows the participation of the results of each case in the Final-Pareto. The Final Pareto is a Pareto-front created from the aggregation of the obtained solutions of the two cases (cases 6 and 7) based on non-dominated sorting code.

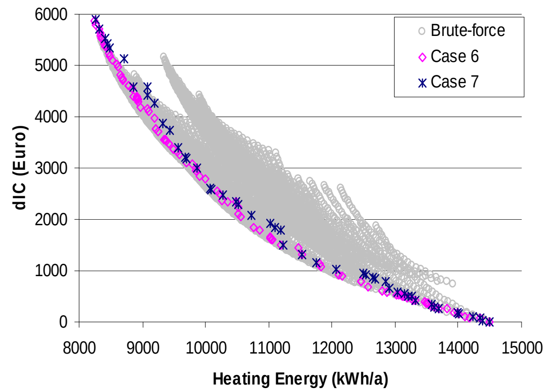

Figure 9: Comparison between the results of cases 6 and 7

Table (4)
Local and Final Pareto for case 6 and 7

Case	Algorithm	Total No. Simulation Run	Optimal decisions on	
			local Pareto	Final Pareto
6	GA_RF	1221	120	114
7	GA	1250	49	5

Final Pareto indicates that most of the solutions come from GA_RF results (Fig. 10). This means the quality of GA_RF results is mostly higher than the quality of the GA's results. Fig. 11 shows the execution time elapsed in case 6 and 7. In addition, GA_RF provide large number of solutions on its local Pareto (120 optimal solution) while only 49 optimal solution are provided in case 7 using GA alone.

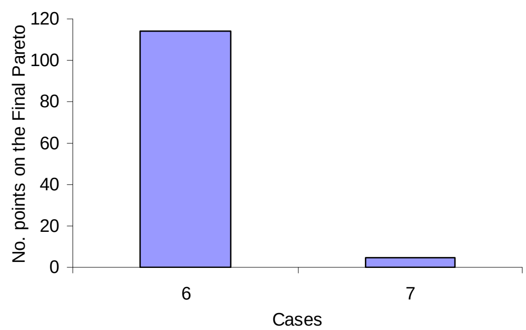

Figure 10: Participation of cases 6 and 7 in the Final-Pareto

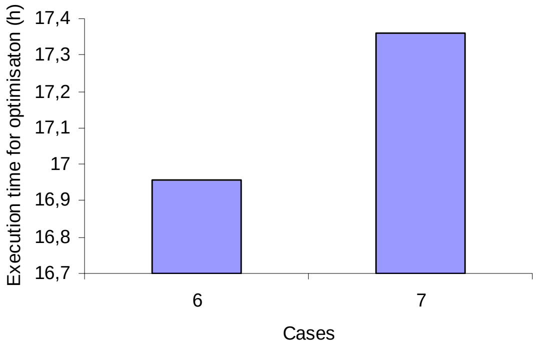

Figure 11: Execution time required for cases 6 and 7

CONCLUSION

Two optimisation approaches (PR_GA and GA_RF) are developed and tested for a two-objective optimisation problem. The first objective is the space heating energy. The second objective is the difference in investment cost for five design variables. The two objectives are considered as nonlinear functions. The results are verified by means of applying the investigation on a simple building-model which is studied in a recent published paper. The value of the current study, good quality optimal-solutions based on low number of simulation runs, can be expanded by reference to more complex building-models.

The obtained results indicate that the two proposed approaches can be applied successfully for this type of problems and can also achieve more accurate results and/or need less time compared with using the default GA (provided in MATLAB 2008a genetic and Direct search Toolbox) alone.

PR_GA combination is recommended to reduce the execution time needed for the optimisation process. GA_RF combination could be a good choice when high quality results are required. It also gives a good approach to stop the optimisation process according to a clear criterion. The two methods can be combined to form a PR-GA-RF approach which will retain the good features of the two proposed methods.

Additional work and graphical user interface (GUI) are still needed to make the proposed approaches able to be used by engineers with no statistical background.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial support of the Finnish National Technology Agency (TEKES), as part of the MASI programme, as well as the following supporting companies: Optiplan Oy, Pöyry Building Servicees Oy, Saint-Gobain Isover Oy, Skanska Oy, and YIT Oyj.

REFERENCES

C3. 2003. Ministry of Environment Finland, National Building Code of Finland C3, Thermal insulation in the building, regulation.

Crawley, D.B, Hand, J.W, Kummert, M., Griffith, B.T. 2005. Contrasting the capabilities of building energy performance simulation programs, In Proceeding of the 9th IBPSA conference, Montreal, Canada.

Deb, K. 2001. Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons.

Hasan, A., Vuolle, M., Siren, K. 2008. Minimisation of life cycle cost of a detached house using combined simulation and optimisation. Building and Environment, Volume 43, No 12, December 2008, Pages 2022-2034

Mourshed, M., Kelliher, D., Keane, M. 2003. ArDot a tool to optimise environmental design of building, IRUSE (Informatics Research Unit for Sustainable Engineering), Department of Civil & Environmental Engineering, National University of Ireland Cork, Ireland.

Mendes. 2003. Matlab based simulation tool for building thermal performance analysis, Eighth International IBPSA Conference, Eindhoven, Netherlands,

Nielsen, T.R., 2002. Optimisation of buildings with respect to energy and indoor environment, PhD dissertation, Technical University of Denmark, Denmark

Waltz, R. A., Morales, J. L., Nocedal, J., Orban, D. 2006. An Interior Algorithm For Nonlinear Optimisation That Combines Line Search And Trust Region Steps, Mathematical Programming, Vol 107, No. 3, Pp. 391–408.