An Evolution-Based Generative Design System: Using Adaptation to Shape Architectural Form

by

Luisa Gama Caldas

M.Sc. in Architecture
The Bartlett Graduate School
University College London, University of London, UK (1995)

Submitted to the Department of Architecture,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in
Architecture: Building Technology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Signature of Author...
Department of Architecture
August 10, 2001

Certified by ....
Leslie K. Norford
Associate Professor of Building Technology, Department of Architecture
Thesis Supervisor

Accepted by...
Stanford Anderson
Chairman, Departmental Committee on Graduate Students
Head, Department of Architecture

Doctoral Committee

Leslie K. Norford
Associate Professor of Building Technology, Department of Architecture
Thesis Supervisor

William J. Mitchell
Professor of Architecture and Media Arts and Sciences
Dean, School of Architecture and Planning

Julie O'B. Dorsey
Associate Professor of Architecture and Computer Science and Engineering

An Evolution-Based Generative Design System:
Using Adaptation to Shape Architectural Form

by

Luisa Gama Caldas

Submitted to the Department of Architecture on August 10, 2001,
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Architecture: Building Technology

Abstract

This dissertation dwells in the interstitial spaces between the fields of architecture, environmental design and computation. It introduces a Generative Design System that draws on evolutionary concepts to incorporate adaptation paradigms into the architectural design process. The initial aim of the project focused on helping architects improving the environmental performance of buildings, but the final conclusions of the thesis transcend this realm to question the process of incorporating computational generative systems in the broader context of architectural design.

The Generative System [GS] uses a Genetic Algorithm as the search and optimization engine. The evaluation of solutions in terms of environmental performance is done using DOE2.1E. The GS is first tested within a restricted domain, where the optimal solution is previously known, to allow for the evaluation of the system's performance in locating high quality solutions. Results are very satisfactory and provide confidence to extend the GS to complex building layouts. Comparative studies using other heuristic search procedures like Simulated Annealing are also performed.

The GS is then applied to an existing building by Álvaro Siza, to study the system's behavior in a complex architectural domain, and to assess its capability for encoding language constraints, so that solutions generated may be within certain design intentions.

An extension to multicriteria problems is presented, using a Pareto-based method. The GS successfully finds well-defined Pareto fronts providing information on best trade-offs between conflicting objectives. The method is open-ended, as it leaves the final decision-making to the architect. Examples include finding best trade-offs between costs of construction materials, annual energy consumption in buildings, and greenhouse gas emissions embedded in materials.

The GS is then used to generate whole building geometries, departing from abstract relationships between design elements and using adaptation to evolve architectural form. The shape-generation experiments are performed for distinct geographic locations, testing the algorithm's ability to adapt buildings shape to different environments. Pareto methods are used to investigate what forms respond better to conflicting objectives. New directions of research are suggested, like combining the GS with a parametric solid modeler, or extending the investigation to the study of complex adaptive systems in architecture.

Thesis Supervisor: Leslie K. Norford
Title: Associate Professor of Building Technology, Department of Architecture

Acknowledgments

I would like to thank the several institutions that contributed to funding this research: The Fundação para a Ciência e a Tecnologia, Praxis XXI program, Portugal; the Fundação Cultural Luso-Americana, Fulbright program; the V. Kahn-Rassmussen Foundation; and MIT, through a Rosenblith Fellowship.

This work is first dedicated to my parents. To my father, who has always guided me through life with his clarity of view, his sharp mind and his uncompromising honesty. To my mother, who has that kind of wisdom that one only gains if one passes through life with sensitivity and awareness.

Then to my sisters and brother, Ana, Gabé and Miguel, for always being there for me. And to those late night talks with Miguel, drinking whiskey and talking about all kinds of things, some of them so important for this thesis, until four or five in the morning.

I would like to thank Les Norford, my advisor, for always believing in my ideas and my work, making me come to MIT, and for all the valuable advice he gave me throughout this academic endeavor; Bill Mitchell, for bringing a precious contribution to this research drawing from his extensive knowledge of the field and theoretical background; Julie Dorsey, for looking at my work with her inquisitive mind and forcing me to face the higher-level questions; Peter Testa, for his creative advice and conversations about the use of generative systems in architecture; Álvaro Siza, for accepting to participate in this research and providing all the necessary information; Jorge Bastos and the late Frederico George, for pushing me towards doing this Ph.D.

I would also like to thank Steven Selkowitz, Robert Sullivan, Eleanor Lee and Chas Ehrlich at Lawrence Berkeley National Laboratory, in California, who welcomed me and gave me precious help when I was giving my first steps in the field.

A word to Renée at MIT, for always supporting me through difficult times.

My friends at MIT, Berkeley and Lisbon, Ana, Axel, the late Catarina, Christoph, Clara, Dan, Elsa, Erik, Franco, Guilherme, João, José, Monika, Pedro, Rita, Yaz, and so many others. And finally, João, for our long discussions about what all this really meant at the end.

Luisa

Table of Contents

CHAPTER 1 Introduction

1.1 Introductory note

This thesis dwells in the interstitial spaces between the fields of architecture, environmental design and computation. Its nature is that of an enclave, a boundary space between areas that too often stay within the limits of their own paradigms, both in academic and professional realms.

In this work, a method is proposed to incorporate evolutionary systems and adaptation paradigms into the architectural design process. Departing from a description of abstract relationships between design elements, adaptation processes are used to evolve architectural form. During this thesis, adaptation was directed towards environmental behavior, looking for shapes that harvest daylighting and reduce thermal exchanges with the external environment. Many other criteria are possible to be considered, as the most important contribution of this thesis may be the method it proposes, more than the specificities of the particular experiments that were developed to validate that method.

Our final model for including the adaptation paradigm in architectural design can be expressed as:

initial rules + randomness + adaptation [reaction + action | behavior] + frozen accidents

Initial rules relate to a description of the design problem that the architect formulates, in terms of relationships between the different design elements, variables and their constraints, compositional rules, etc. The architect can incorporate in this initial framework many of the higher-scale issues that relate the architectural artifact with other external forces.

Randomness is introduced as a major driving force in mechanisms of adaptation such as those present in several natural and artificial systems (Holland, 1975). The introduction of a randomized procedure like Genetic Algorithms [that act as the search engine for this

Generative System] aimed at introducing stochastic, nondeterministic processes into the system, and also represented a move towards the adaptation paradigm.

Genetic algorithms [GAs] were chosen from the outset because 'adaptation' was assumed as a central concept to the work. However, adaptation itself can be further decomposed in two major approaches, which we called reactive and active adaptation (Popper, 1992).

Popper describes reactive adaptation as a passive process, where there is basically a population with multiple variants, and better-adapted individuals are selected and others eliminated. Selection pressure is thus exerted from the exterior environment only. Genetic Algorithms tend to dwell in this approach to adaptation. Even though Popper's notion is a philosophical one and was not developed within the realm of artificial adaptive systems, some useful inferences can be taken from this alternative view to inform further interpretations of what adaptation in artificial systems can be about.

Popper's active interpretation of adaptation puts emphasis instead on individuals' activities, as constantly active problem-solvers. Although survival is still their main concern, they develop behaviors and means of action that help them to overcome the problems they find, improve their adaptation to the outside environment, or even generate their own microenvironment. They may do so individually, or act collectively with other individuals, either in a cooperative or a confrontational way. They may look for niches to include themselves, where their characteristics can better be taken advantage of, and thus make their survival probabilities increase, or they may actively help construct those niches. This view of adaptation puts an emphasis on action and behavior instead of simple reaction to the environment.

Simply resorting to GAs will not supply all the necessary mechanisms to move towards active adaptation. For that, one must instead rely on classifier systems or models like 'Echo,' proposed by Holland in the last chapter he added to the second edition of 'Adaptation in Natural and Artificial Systems,' published by MIT Press in 1992 [seventeen years after its first edition in 1975]. The work described in that chapter was later further developed into the book 'Hidden Order: How Adaptation Builds Complexity,' published in 1995. These models include behavior or actions taken by individuals over their neighbors or over the environment itself. This allows that from simple rules and initial order, complex patterns can emerge due to the combined action of different individuals, leading to the emergence of what are called complex adaptive systems. Without those mechanisms, the system is limited to reactive adaptation, as in genetic algorithms: there are populations of solutions, based on different combinations of a restricted and predetermined number of variables, and then some pressure exerted from the exterior to assign a fitness to each individual, which will impinge on its ability to survive.

Finally, the latest process we propose is the inclusion of what Gell-Mann (1995) calls 'frozen accidents'. As Gell-Mann says, "Most single accidents make very little difference to the future, but others may have widespread ramifications, many diverse consequences all traceable to one chance event that could have turned out differently. Those we call frozen accidents". Those mechanisms are, both in natural and artificial systems, related to the operation of chance, which is associated with indeterminacy. GAs do include randomization and are not deterministic procedures. However, we would like to propose a different formalization of the concept of 'frozen accident' within an architectural adaptive domain. Instead of relying solely on the guided stochastic processes of selection, recombination and mutation of GAs to introduce 'accidents,' an external agent is allowed to interfere with the search and make it follow a different path from what the initial adaptive processes could lead it. This external agent is the architect's preference for some solutions [or features of a certain solution] over the others. To achieve that, a process similar to that of Design Galleries (Marks et al., 1997) is proposed.

Using Design Galleries, the architect can be presented with the solutions found until that point during the course of the adaptive search process, and will manually choose the ones that he preferred, thus changing the previous course of the adaptation system. Or the architect's preference can be combined in some way with the fitness value of the solutions to continue the search. Another option could be that the architect could 'freeze' part of a solution that he finds particularly interesting, while letting the other elements of the design free for further exploration. Conversely, he should be able to 'unfreeze' any elements that he does not find satisfying anymore, after the solution has further evolved.

This process of combining initial rules representing a problem formulation, computational processes of randomness and adaptation, and the architect's capability to add his preference to the evolution progress can lead to an interesting and novel approach to the architectural design process. If one combines such a system with a powerful CAD modeling tool, able to generate parametric, free-form geometries, such a system could represent a relevant new approach to the design process. A key issue involved is that of representation of architectural solutions using symbolic languages. If a building can be fully described using a 3D CAD model and text [like materials specifications], there is theorethically no reason why it cannot be fully described by a single string of symbols [numbers, mathematical operations and words], what in a GA environment would represent a chromosome. If we add to this the concept of design variables and data structures (Mitchell, 1977), we can have a purely symbolic representation of an entire building that can be manipulated using a generative design system like the one proposed here. So, departing from initial rules and abstract relationships, and by manipulating data structures, any potential architectural shape can emerge, depending on the initial rules and the flexibility of the CAD platform used to encode any type of geometry. The architectural shapes created this way can then evolve through the processes of adaptation previously described. Further theoretical implications of the development and use of such a system in architectural design will be provided in the conclusions.

Given the general picture of where do we consider this system and method can go, we should now add a note about how far into its development we went in this dissertation. The actual implementations that were performed for this work are still much behind what our final aims may be, but many important steps were already taken.

Genetic algorithms were chosen from the beginning because 'adaptation' was assumed as a central concept to the work. However, just including a GA as a search and optimization procedure does not ensure by itself a real move towards including adaptive systems in architectural design. GAs can be used just like any other reasonably reliable heuristic method for optimization studies, with the exception that they provide populations of solutions instead of single ones. That was what happened during most of the initial chapters of this dissertation, since one of the important tasks we were facing was the testing of the system, to assess its reliability and robustness. GAs were used for improvement of solutions [a term we prefer to 'optimization,' as suggested by Holland (1992)], but to a large extent the use of the adaptation paradigm was limited to 'reactive adaptation'.

Towards the end of this dissertation, adaptation starts to take off, mainly in the shape generation experiments, which helped further clarifying many of the concepts that were in the background of this dissertation, although in a latent, implicit way. The 'conclusions and future work' section will provide a more clear view of the work that was actually finalized and all the new developments that remain to be done.

As Jorge Luis Borges said, Kafka created his own predecessors, as without Kafka's work many of the endeavors from others preceding him would have had a different meaning, or might even have been forgotten. Hopefully, this thesis and the future work attached to it may contribute to create its own Kafka in the future.

1.2 Thesis outline

This thesis describes the conceptual development and implementation of a computational generative design system based on evolutionary concepts. The initial aim of the first chapters was to develop and test a system that could evolve certain aspects of an architectural design solution so as to improve its environmental performance, namely in terms of using natural light, reducing the subsequent need of artificial lighting, and controlling heat gains and losses so that smaller amounts of energy were required to keep the building's interior within a comfort range.

The Generative System [GS] is composed of two main modules: a search algorithm, that would look for high performance solutions, and a simulation program, that would evaluate the quality of those solutions in terms of the criteria defined by the user. For the search algorithm, a Genetic Algorithm [GA] was chosen, for reasons that will later be explained in the chapter that deals with the system implementation [chapter 3]. The GA is responsible for introducing the evolutionary paradigm into the work. For the simulation module, the choice relied on DOE2, a thermal and lighting simulation program that combines good accuracy of results with reasonable computational times.

Subsequent chapters involve the testing and validation of the system [chapter 4], and comparison with other search methods that could represent alternative approaches and formalizations to the ones that constituted our choice [chapter 5]. After enough confidence with the system was gained, several applications were proposed and implemented. From the initial, almost schematic building layouts used for testing the Generative System, to the application of the GS to an existing piece of architecture by one of the most prominent contemporary architects [chapter 7], many new questions were raised concerning the limits of the application of generative computational systems to the architectural design process, the possibility of incorporating architectural intentions into computational systems, and the conflicts between aesthetics [or formal] realms, and concerns about performance measures and architecture sustainability issues.

Recognizing the limitations of using a single objective function, or figure of merit, to evaluate the performance of such a complex artifact like a piece of architecture, the work was further expanded to include multiple evaluation criteria [chapter 8], using a challenging method that avoids combining multiple objective functions into a single number, but instead provides a frontier of best trade-offs between several criteria [Pareto fronts], ultimately leaving to the architect the decision of which factors are most relevant to his own design evaluation.

Finally, the last piece of this work uses the methods developed to generate and evaluate new building geometries [chapter 9]. Instead of departing from an existing building design, and introducing performance improvements by manipulating a few variables like dimensions of some elements, material characteristics, etc., the work expanded into the more interesting realm of researching shape adaptation to the external environment. Given an initial schematic layout for a building, how would its overall three-dimensional geometry adapt to the external climatic conditions it had to deal with? To some extent, a building is an interface between an outside environment and an inside environment, where people will reside. Are those external forces enough to provide feedback information that would make the building shape adapt to their influence, while responding simultaneously to the inside requirements? That was one of the main questions that governed the research presented in that chapter. A final extension related the Pareto-front approach with shape adaptiveness. If the same 3D shape is simultaneously responding to a number of different pressures and requirements, it might be useful to look at what kind of shapes respond better to each of those molding factors, to be able to better assess the compromises made to try to meet all of them simultaneously.

In the conclusions section [chapter 10], it will become evident that even though many conclusions can be drawn from the work and experiments developed, many new questions appeared during the process. Instead of the presented applications serving as a justification for the purpose of previously developed work, they themselves acted as triggers for further developments and extensions of the research, pushing it to other limits and boundaries. As Jung said, 'the work in progress becomes the artist's faith.' The 'further work' section will thus reflect all the unanswered questions, all the new paths of development that emerged from the experiments performed or from our initial ideas, but that due to time limitations were not possible to be further pursued. This includes the study of complex adaptive systems applied to urban configurations [what happens when a number of buildings are put together and can 'act' over their neighbors to improve their own microenvironment], the use of variable length chromosomes [that could allow for the emergence of new elements that were not included in the initial user-defined scheme, if that proved to improve the adaptation level of the overall system], the use of dynamic constraints [which would improve the degree of adaptiveness and diversity in new solutions, by removing the need for constraining from the outset all variables by established limits], the creation of a parametric solid modeling interface to the Generative System [which would allow a much more intuitive relation between the architect and the GS, and permit the appearance of a new wealth of shapes and geometries, that could probably not be initially predicted by the architect], the use of design galleries [to include the architect's own preferences in the course of the generative search – the 'frozen accidents'], and other developments that we envision as paths for further exploring the issues of environmental adaptiveness in architecture.

1.3 Some background

Herbert Simon, in his book 'The Sciences of the Artificial' (1969), claims that any artifact, or man-made object, can be seen as an interface between an inner and an outer environment. The inner environment is the substance and organization of the artifact itself, and the outer environment the surroundings in which it operates. If the inner environment is appropriate to the outer environment, or vice-versa, the artifact will serve its intended purpose and it can be said to be adapted to some situation. This also applies to the living systems that have evolved through the forces of organic evolution.

In most large and complex systems, it is possible to use this separability of inner and outer environments to explain the adaptation of a design object in terms of its purpose or function. This means that we can often predict the behavior of an artifact from knowledge of the system's goals and of its outer environment, with only minimal assumptions about the inner environment. A corollary of this is that we can often find quite different inner environments accomplishing identical or similar goals in identical or similar environments (Simon, 1969).

This abstract interpretation of the process of design is also applicable to architecture. The stated corollary implies that many different architectural designs can answer a particular problem with similar efficacy, a fact well known to architects. For this reason, the Generative System described in this dissertation adopts a method where the result is not a single solution, but a number of alternative solutions that all perform reasonably well in terms of the defined goals. Hence one of the reasons behind the choice of the evolutionary paradigm, and particularly of genetic algorithms. GAs deal in parallel with a population of solutions and, as a result, provide a population of solutions instead of a single one, contrarily to many other search methods. This nicely fits with Simon's proposition that many different individuals may be equally adapted to a given environment, a concept that in itself draws from Darwinist theories.

One of the problems in architecture is that it is difficult to state clearly the goals that a design has to achieve, and also what are the procedures to achieve those goals. An architectural design problem is thus an ill-defined problem in its nature.

Most early applications of mathematical models and their respective computational implementations to architecture problems were thus confined to limited problem settings, where only a subset of the large complexity of issues interacting in an architecture problem were dealt with. This related to proportion issues, floor plan layout studies, research on urban densities, solar shading studies, etc.

Some of this work made explicit use of optimization techniques, while others just used series of simulations to reach results. Here a distinction should be made between simulation and optimization (Radford et al., 1978). In simulation studies, a scenario-by-scenario method is used, where a limited number of different configurations are generated by changing some of the parameters of interest, and then some conclusions can be drawn from these parametric studies. It is often a manual procedure, where only a few alternatives are evaluated because it tends to be a time consuming and tedious process. The other option is to make formal use of optimization methods, so that the computer performs the search according to some problem definition, choice of objective function, determination of the solution space by placing upper and lower bounds in the variables, and application of a search and optimization procedure to the location of high-performance or optimal solutions in that space.

Many early studies were done where simulation was the method applied, such as the well-known study by Leslie Martin and Lionel March (1966) that concluded that similar densities to high-rise buildings could be achieved using low-rise construction.

Other problems, such as floor plan layout problems, where for a given initial perimeter one wants to find the optimal spatial distribution that assigns to each space the required area while respecting several adjacency requirements, were already making use of optimization techniques in the 60's and 70's (Mitchell, 1974). A more extensive review on previous work on this area will be provided in chapter 2.

One common characteristic of these approaches was that they tended to isolate a small problem from within a larger problem set, and study it using systematic or mathematical approaches. The relative quality of the different solutions had also to be quantifiable, usually in the form of a figure of merit. The attempt is thus to establish a well-defined sub-problem within a larger, ill-defined one.

1.4 Environmental issues

Energy consumption in building has also been approached as an area in which optimization models could be used. A review of previous research done in the area will be provided in chapter 2. Particularly, fenestration design has interested a number of researchers, since this is an area that can have a relatively large impact on a building's performance, and can to a certain extent be isolated from other more complex problems in an architecture project [needless to say, this 'separation' is in itself arguable, and many architects would disagree with such an approach]. Fenestration design also poses a number of challenges in terms of predicting the complex interactions between daylighting requirements, heat gains and heat losses, what makes it an interesting topic for applying search and optimization methods.

However, a point should be raised about whether building energy efficiency can be considered a well-defined problem. Certainly one can perform whole-building simulations, and account for energy consumption for lighting, cooling, heating, ventilation fans, etc., to get a final number that reflects the total consumption in the building throughout the year. But this single number is itself plagued with complexities.

Take the example of a building located in a very cold climate. Reducing window sizes may be an adequate energy conservation strategy. Savings in energy may offset the increased spending in artificial lighting [daylight levels will be reduced]. However, one should also account for the fact that people usually prefer natural light and a view out. Those are, nevertheless, largely non-quantifiable issues to be included in the evaluation process. On the other hand, can the architect start playing with other variables, like materials characteristics or space layout, to overcome some of these problems? If that is true, then the initial, 'well-defined' problem is no longer an isolated, simple and easily quantifiable problem, but it has necessarily to interact with other factors that contribute to the characterization of the architectural artifact.

For these reasons, we prefer not to use the concept of optimization applied to energy efficiency in buildings. As mentioned before, many respected researchers on complex optimization problems are starting to put forward the term 'improvement' instead of optimization (Holland, 1992). We consider that the Generative System proposed here should not be regarded as an 'optimization tool' but instead as a powerful design aid tool that can lead to design improvement, and with which the architect must engage in an interactive loop. The architect can run the Generative System, retrieve some useful information from its results, which can also act as a diagnosis tool of existing problems in the current building solution, make some changes to the project or to the constraints of the GS if they fail to meet other criteria [like those related to aesthetics or user preferences, for example], run the GS again, get new results and information, and so on. This is not a system that aims at providing definite, 'optimal' answers, but is instead a novel tool to help producing quality architecture that simultaneously has a sustainable dimension to it.

It should be added that another of the driving forces behind the development of this system was a willingness to break some prevailing prejudices among many architects that low-energy, sustainable architecture has some particular aesthetics characteristics, that in a way tend to 'tag' the architects that engage in this type of practice [the so-called bioclimatic or 'green' architecture]. This misconception comes only from a lack of understanding of the real issues involved in producing sustainable architecture. By applying the Generative System to a building designed by a respected contemporary architect, we also intended to demonstrate that any piece of architecture, even those highly driven by aesthetic concerns, can have a sustainable dimension, without any decrease in the intrinsic architectural quality of the work.

On the other hand, the attitude of somewhat unawareness of environmental-related issues in architecture is progressively changing, as more pressure is put on architects to deal with those issues. Buildings that are poorly adapted to their environment have to heavily rely on mechanical and electrical system to maintain reasonably appropriate indoor environmental conditions [the other option, of course, is to make occupants suffer from discomfort within the spaces]. However, those artificial systems carry with them other inconveniences apart from high energy consumption. HVAC systems can cause poor indoor air quality by lack of appropriate air distribution or by lack of system maintenance [like regularly replacing filters, which are often costly], can lead to problems such as the well-known sick building syndrome, and at extreme cases can even be sources of dangerous illnesses like legionnaires disease, which has recently plagued Murcia, in Spain.

On the other hand, users often prefer more natural environments, where they can have access to daylighting and open windows for ventilation, having some degree of control over their environment. Although in many situations it may be very hard to implement architectural solutions that rely solely on natural strategies for environmental control, improving a building's adaptiveness to its environment may reduce the need for mechanical systems, and during the milder seasons of the year the building may even function on its own.

This short introduction aimed at clarifying some other dimensions of the concepts and goals behind the development of the Generative System presented in this dissertation. Much remains to be said, but the relevance of the work will further emerge from the discussion of the results achieved in each of the chapters, and from the 'conclusions and further work' section. In addition, section 1.5 will provide some larger-scale background on the necessity to move towards a more sustainable architecture.

1.5 Research sponsors

The research presented in this dissertation was sponsored by four sources: the Portuguese Ministry of Science of Technology [Fundação para a Ciência e a Tecnologia], within the Praxis XXI program; the Fulbright program; an MIT's Rosenblith fellowship; and the V. Kahn-Rassmussen Foundation, within the framework of the 'Sustainable Urban Housing in China' project, developed at MIT.

Much of the work presented here is applicable to the 'Sustainable Urban Housing in China' project, as all the experiments done for Chicago are equally applicable to Beijing, one of the cities targeted by that project. Even though a weather file for Beijing was not available in the format required by the Generative System, Chicago's climatic conditions are similar to those of Beijing [as will be demonstrated in chapter 8], so results can be extrapolated with no significant errors.

In the next section, a detailed account is given of current European Union policies regarding energy efficiency in buildings, to provide some of the background framework within which this work was supported, and also to partly justify the relevance of the research herein developed. The case of Portugal is reviewed in more depth, as even though the Praxis XXI program is partially financed by the EU, this particular project does make part of Portugal's efforts to increase its energy-efficiency standards and contribute to the reduction of global climatic change.

1.6 Energy conservation policies in the European Union

1.6.1 General considerations

There is currently a strong commitment from the European Union to improving the environmental performance of buildings, by reducing energy consumption levels and subsequently greenhouse gas emissions, towards the goals set by the Kyoto protocol. Some issues that are currently being discussed are presented below, to support the relevance of the work presented in this dissertation.

Article 3 of the Kyoto Protocol states: “The Parties included in Annex I shall, individually or jointly, ensure that their aggregate anthropogenic carbon dioxide equivalent emissions of the greenhouse gases do not exceed their assigned amounts, ... with a view to reducing their overall emissions of such gases by at least 5 per cent below 1990 levels in the commitment period 2008 to 2012”. Within this framework, the European Union has agreed to reducing its emissions level by 8%, against, for example, 7% for the USA and 6% for Canada. To achieve this goal, it will be necessary to implement effective goals tackling all the main sectors of energy consumption, and thus greenhouse gases emissions. The building sector, responsible for more than 40% of the energy consumed in the UE [1997 data], is a prioritary target.

The European Communities Commission has very recently presented, in May 11, 2001, a proposal for new legislation from the European Parliament and Council, regarding the energy efficiency of buildings (CCE, 2001).

In the introduction, some points mentioned in the Green Book of the Commission are stated:

- The European Union is each time more dependent on energy imports from outside the Union. Based on current predictions, if no active measures are taken, the dependency from energy imports will grow from 50% in 2000 to 70% in 2030.

- There is little margin to influence this scenario from the production point of view. The main measures have to tackle the demand side, mainly in the sectors of energy efficiency in buildings and transportation. The EU is one of the main energy consumption areas in the world.

- Greenhouse gases emissions are currently increasing in the EU, making it difficult to achieve the levels predicted by the Kyoto Protocol.

Given these initial considerations, the fact that 40.7% of the energy consumed in the EU in 1997 was in buildings, both in the residential and services sectors, makes them a preferential target to decrease energy demand in the Union. The total energy consumption in the EU in 1997 was 930 Mtoe. From those, about 380 where consumed in buildings, making them the largest end-users of energy in overall terms. Heating, lighting, and electric appliances and equipment represent the main end-use categories.

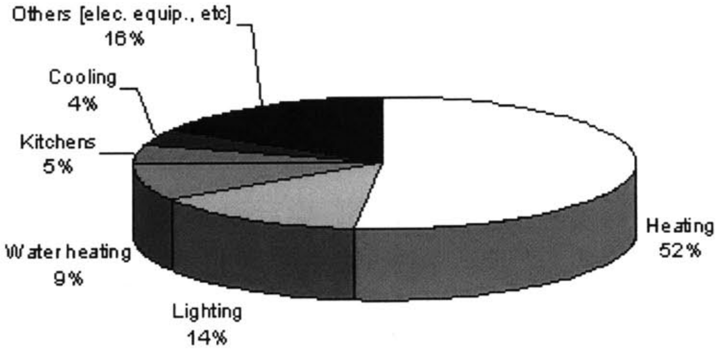


Figure 1.1 Energy end-used in services buildings in the European Union. The services sector includes office, schools, hospitals, sports buildings, hotels, restaurants, commerce, etc. It does not include industrial buildings.

Lighting and other forms of electricity consumption are particularly relevant, because although electricity is a clean form of energy at the end-user level, its production implies more greenhouse gases emissions than any other form of energy, due to the low efficiencies involved in the process.

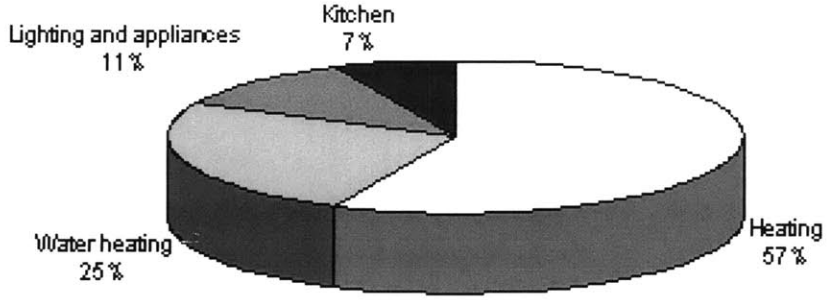


Figure 1.2 Energy end-used in residential buildings in the European Union

Many studies (MURE, 1998) suggest there is a larger opportunity for economically viable energy efficiency in the building sector than in any other. However, the ongoing support projects for the development and promotion of new technologies has not given the expected outcome in terms of realized efficiency. For that reason, the Commission wants now to implement more concrete measures.

The proposed measures are:

a) Establishing a common methodology in all EU countries for assessing the integrated energy efficiency of buildings;
b) Enforcing minimum requirements for the energy efficiency of new buildings and in retrofitting of existing large buildings [above 1000m^2];
c) Establishing a certification system for new and existing buildings, based on the requirements mentioned above. In public buildings, those certificates of performance have to be publicly displayed;
d) Periodic inspection and evaluation of heating and cooling systems.

a) Establishing a common methodology in all EU countries for assessing the integrated energy efficiency of buildings

The integrated assessment of building energy efficiency is being studied in countries outside the Union [USA, Canada, Australia, New Zealand], to be included in building codes and best practice standards. Given that most buildings in the Union [at least in the northernmost countries] already have fabric insulation, the other factors to be assessed include lighting, HVAC systems, ventilation energy, heat recovery, and passive measures like building positioning and orientation, solar heat gains, and the use of renewable energies. In Germany, France, UK, Netherlands and Italy this integrated approach already exists to some degree, and in some countries is enforced by law. The current directive looks at a standardization of those methods and extension to the remaining member countries, while considering local factors like climatic characteristics.

b) Enforcing minimum requirements for the energy efficiency of new buildings and in retrofitting of existing large buildings [above 1000m^2]

Due to the low rate of building renovation [the life cycle of buildings is usually between 50 and 100+ years], the new measures will have to include existing buildings to have the desired impact. Large buildings [above 1000m^2] will have to meet minimum requirements when undergoing retrofitting.

c) Establishing a certification system for new and existing buildings, based on the requirements mentioned above.

One of the main imperfections in the real estate market in terms of energy efficiency is the conflict of interests between landlord and tenant, or developer and buyer. As it is the tenant or buyer who will have to pay the energy bill, the incentive for the landlord or developer to invest in energy efficiency is low. To make that investment more attractive, a system should be implemented to provide future building users with information about its efficiency, which will then influence the rent or cost of the building. This information will be available in the form of 'energy certificates,' both for services and residential buildings, new or existent, and be shown at the time of a rental or sales contract, with the maximum issuing date of 5 years before.

For public buildings, these certificates must be displayed and accessible to the general public, as public buildings are seen as potential displays of new technologies and design concepts, and should thus constitute examples to the public. Information must also be available about the functioning of the HVAC systems, the range of admissible internal temperatures, internal temperatures in the building at the moment [show through reliable instrumentation], and other relevant indoor environment parameters like relative humidity. This will make the performance of the building more transparent, promote public awareness about energy efficiency and comfort standards, and in general, it is expected, lead to energy savings.

New-building certification is currently mandatory in the UK, Germany and Denmark. In relation to existing buildings, only Denmark has specific regulations, but voluntary systems exist in a number of member countries. In Denmark, calculations based on data from 3.5 years show a return on investment above 13%, a very high cost-effectiveness.

d) Periodic inspection and evaluation of heating and cooling systems.

Periodic heating systems inspection [mainly of furnaces] is already mandatory in 10 member countries. The remaining ones have voluntary programs and information systems for owners. The goal is to have all member countries introducing mandatory inspection. Cooling systems [air conditioning] must be submitted to similar regular inspection measures.

1.6.2 Economic potential of energy efficiency measures in buildings in the EU

It is estimated that there is a potential to save about 22% in energy consumption in buildings using only cost-effective measures. Cost-effectiveness is measured as having a return period less than eight years, which compares well with other alternative investments, including investing in energy production. This can be achieved by savings in heating energy, water heating, air conditioning and lighting, by improving the respective systems, and by applying energy reducing measures regarding the building envelope, including windows. It does not include non-installed equipment in the services sector [like office equipment], which is supposed to account for about 20% of electric energy consumed in offices, for example.

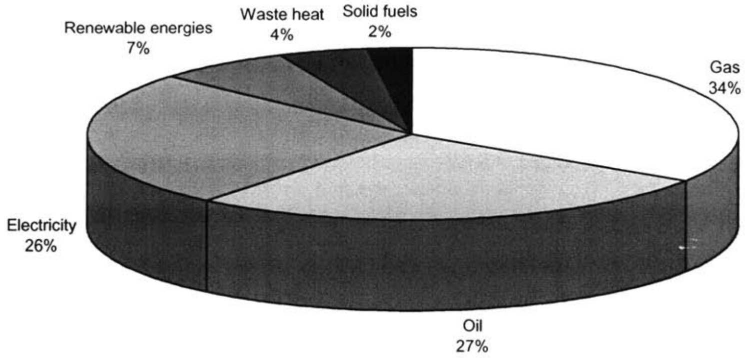


Figure 1.3 Energy consumption in the European Union by type of end-use fuel. Fourteen percent of the electricity consumed is from renewable sources.

The European commission aims at achieving, in the building sector, savings above 55Mtoe in energy consumption, what is equivalent to reducing CO_2 emissions by 100Mt/year, about 20% of the EU commitment in relation to the Kyoto protocol.

a) Potential economic benefits from improving the building envelope

The last EUROSTAT inquiry (published in 1999) about energy consumption in residential buildings shows the following results regarding building insulation measures [the numbers correspond to the percentage of enquired participants who had a specific measure applied in they house]:

Type of insulation	**FIN**	**S**	**DK**	**IRL**	**UK**	**D**	**NL**	**B**	**F**	**L**	**A**	**GR**	**P**	**I**	**E**
No insulation	0	0	1	13	15	0	14	21	21	55	39	77	-	-	-
Roof insulation	100	100	76	72	90	42	53	43	71	35	37	16	-	-	-
Cavity wall	100	100	65	42	25	24	47	42	68	2	26	12	-	-	-
Floor insulation	100	100	63	22	4	15	27	14	24	5	11	6	-	-	-
Double glazing	100	100	91	33	61	88	78	62	52	20	53	8	3	-	-

Note: Data for Spain and Italy are still missing, and Portuguese data are incomplete

Table 1.1 Results from EUROSTAT enquiry regarding use of insulation in buildings

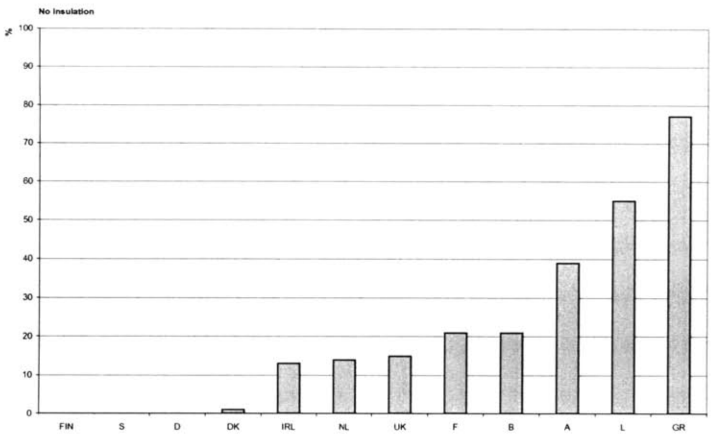


Figure 1.4 Percentage of buildings with no insulation for each EU country

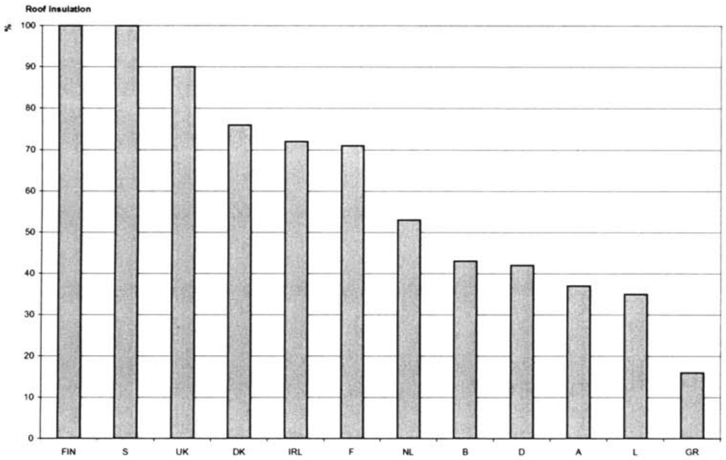


Figure 1.5 Percentage of buildings with roof insulation for each EU country

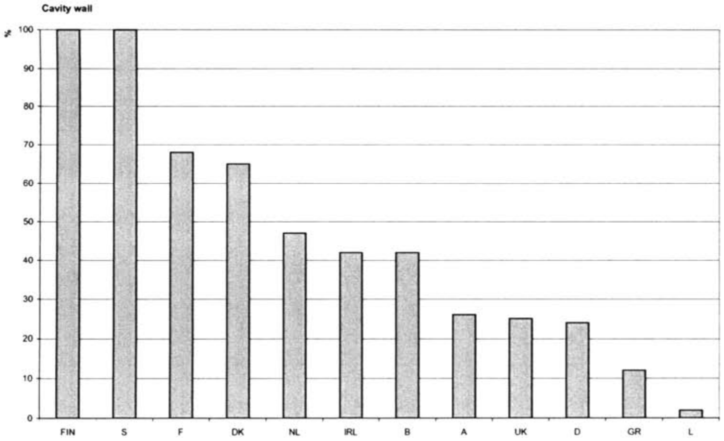


Figure 1.6 Percentage of buildings with cavity walls for each EU country

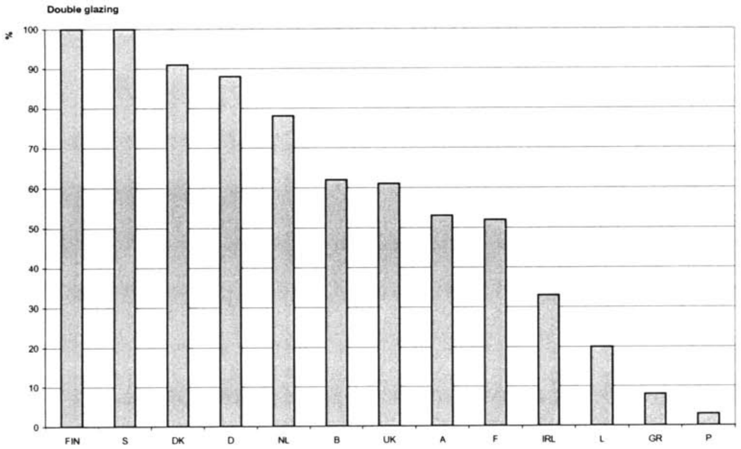


Figure 1.7 Percentage of buildings with floor insulation for each EU country

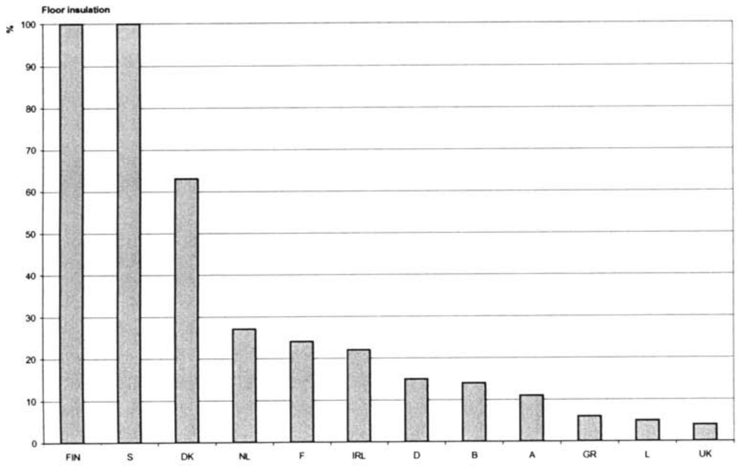


Figure 1.8 Percentage of buildings with double-glazing for each EU country

It can be seen that there are significant differences between the several countries, which only part can be explain by climatic characteristics. Portugal appears as the country with the lowest percentage of installed double-glazing units, although the data are still incomplete.

A study was made where the current Danish energy efficiency code was applied to the other 14 members, with a correction to account for climatic differences, based on degree-days. The results suggest that there are significant savings to be made in many countries, with Spain, Italy and Portugal showing the highest increases.

b) Potential economic savings related to lighting

For residential buildings, lighting represents 4% of the total energy consumption [9 Mtoe]. For the services sector, lighting consumption is 18Mtoe, or 14% of the total. There is a potential for 30%-50% savings in lighting, mainly in the services sector, by improving the use of natural lighting in buildings, and using more efficient lighting systems and better controls. This amount of savings [6 to 9 Mtoe] represents a significant part of the potential energy savings in the building sector. The recently launched GreenLight European program has shown that most lighting energy savings have a high economic cost-effectiveness. This fact may be of significance for the development of the work presented in this dissertation, where natural lighting use is an important factor.

c) Potential economic savings related to cooling and air conditioning

Air conditioning is an area that is rapidly increasing both in the services and the residential sectors. Although currently it only represents about 0.7% of the total energy consumption in buildings, the tendency is for a rapid increase if no measures are to be taken, which may include improved efficiency for HVAC systems and building-design related techniques. The potential for cost-effective savings is about 25%.

d) Potential economic savings related to building design and orientation

It is predicted that new buildings can have up to a quarter of the heating needs of standard current buildings. Even the ones that already comply with high insulation standards can reduce their energy consumption up to 60%, by resorting to bioclimatic strategies like careful orientation, natural lighting and cooling, solar control, active and passive solar systems, etc. Existing buildings where many key issues are already established can still take advantage of favorable conditions in the cases where they exist but are currently unexploited.

1.6.3 Conclusions on the current EU proposal

The conclusions of the current EU proposal are that many economical and technical factors are involved in increasing the energy efficiency of the building sector. Some of the member countries have opted for an integrated system to characterize the energy performance of buildings, using simple, aggregated indicators of energy performance. This provides the architect / designer with an enhanced degree of flexibility, since he can decide what factors to trade-off against each other to achieve a final degree of performance.

Currently, many member states have established minimum requisites for new buildings, but with very different standards for distinct countries. The objective is to set a common methodology and standards among all the countries.

Since renovation of existing buildings is seen as an important source of energy reduction, a certification method is proposed, that may provide incentives for landlords to invest in energy efficient retrofitting and correct some of the asymmetry of interests between the owner and the user of the building.

1.6.4 The case of Portugal

As one of the less developed and industrialized countries in the EU, Portugal is one of the states allowed to increase its emissions of greenhouse gases until the commitment period of 2008-2012. Within the 'burden sharing' agreement between EU countries, Portugal can increase its emission by 27%. Current predictions, based on an economic growth of 3.3% a year, show Portuguese emissions increasing by 52% in 2010, vastly above the aimed target. Transportation is the main cause of this increase, followed by the building sector. Industry, even though the main energy consuming sector in the country, is expected to decrease its contribution to greenhouse gases emissions (from 40% in 1990 to 37% in 2010).

The building sector is expected to increase its contribution from 17% in 1990 to 20% in 2010. To better understand these numbers, it is necessary to look at the domestic and services sectors separately.

The domestic sector, which in 1990 contributed 10% of the total emissions, will keep this percentage. Since the total country emissions will increase by 52%, this means the domestic sector will also increase by 52%. The carbon intensity of the sector (emissions per unit area) keep the same values because the decreases caused by the higher efficiency of the installed equipment and the use of cleaner energy sources (like natural gas), is counteracted by the predicted increase in consumption due to higher comfort standards, and also to the 25% increase in residential units predicted for the period under consideration.

The services sector, on the other hand, is expected to increase its contribution from 7% to 10% of the total energy consumed in Portugal. It shows the highest increase in carbon intensity, what clearly shows its disagreement with the overall policy of the country. It is predicted that the total construction area in the sector will increase by 100% in the period under study, with an increase in energy consumption of 200%.

1.6.4.1 Energy consumption patterns

a) Residential sector

In 1988, energy consumption in the residential sector accounted for 15% of the total energy consumed in the country. During the period 1990-1995, the number of housing units in Portugal grew from 2,937,123 to 3,140,371, a 7% increase. In the same period, the number of housing units with heating increased from 50.6% to 72.6%. The number of electrical appliances per household also increased. These numbers reflect the growing economy of the country, greater economic power of the population and an increase in comfort standards.

Table 1.2 and figure 1.9 show the evolution of energy consumption by end use in the residential sector for the period 1990-2010. The values for 1990 and 1995 come from inquiries to the population made by the DGE [Direcção Geral de Energia, Portugal]. The numbers for the remaining years are projections developed by DGE (GASA, 2000).

	1990	**1995**	**2000**	**2005**	**2010**
Heating	13763	30645	33850	38078	38562
Cooling	-	65	2350	5375	9444
Lighting and equip.	12287	25110	27276	30747	33424

Table 1.2 Evolution of energy consumption by end use in the residential sector for the period 1990-2010 (TJ)

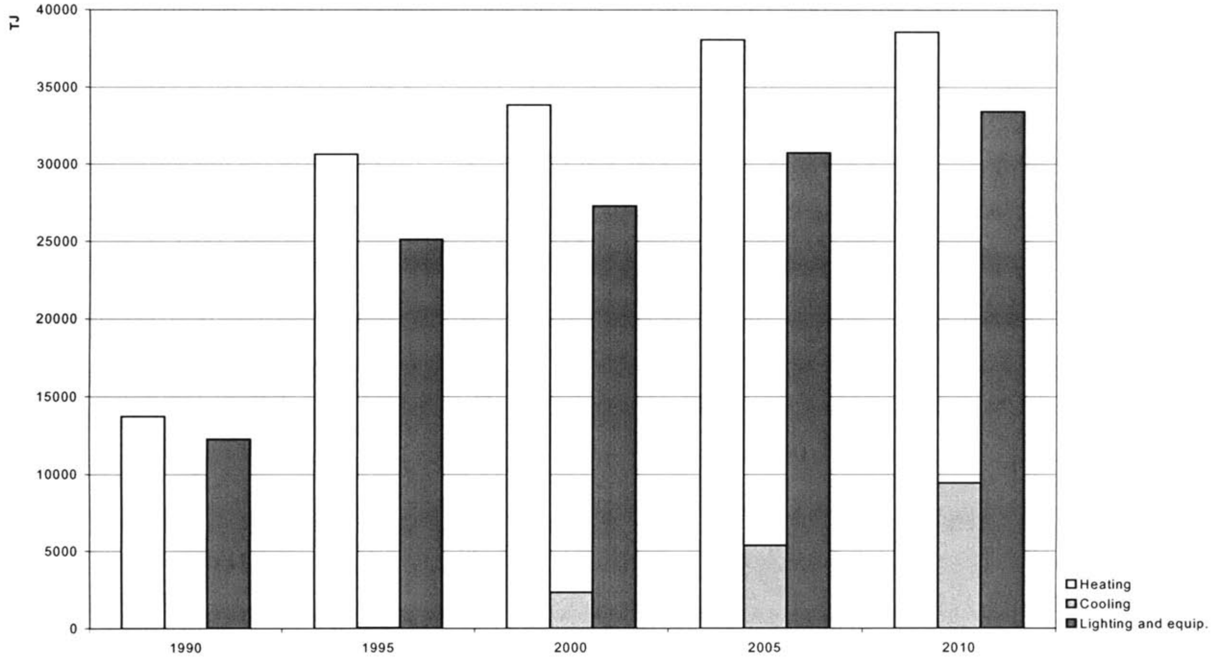


Figure 1.9 Evolution of energy consumption by end use in the residential sector for the period 1990-2010 (TJ)

During the period 1990-2010, heating energy is expected to increase by 180%, and electricity for lighting and appliances by 172%. Energy for cooling, which practically did not exist in 1990, is expected to be up to almost 9500 TJ by 2020, in an explosive growth.

As for type of energy used in the residential sector, Table 1.3 and figure 1.10 provide information about that, in terms of site energy. The origin of the data is the same as before (GASA, 2000)

	1990	1995	2000	2005	2010
Wood	84524	66315	66476	65832	62291
Electricity	17780	31097	36065	40939	46829
LPG	22769	32333	33412	33567	32179
City gas	1676	1910	-	-	-
Natural gas	-	-	4103	8165	11682
Coal	995	871	-	-	-
Solar	-	-	879	1130	1549

Table1.3 Energy type for the residential sector for the period 1990-2010 (TJ)

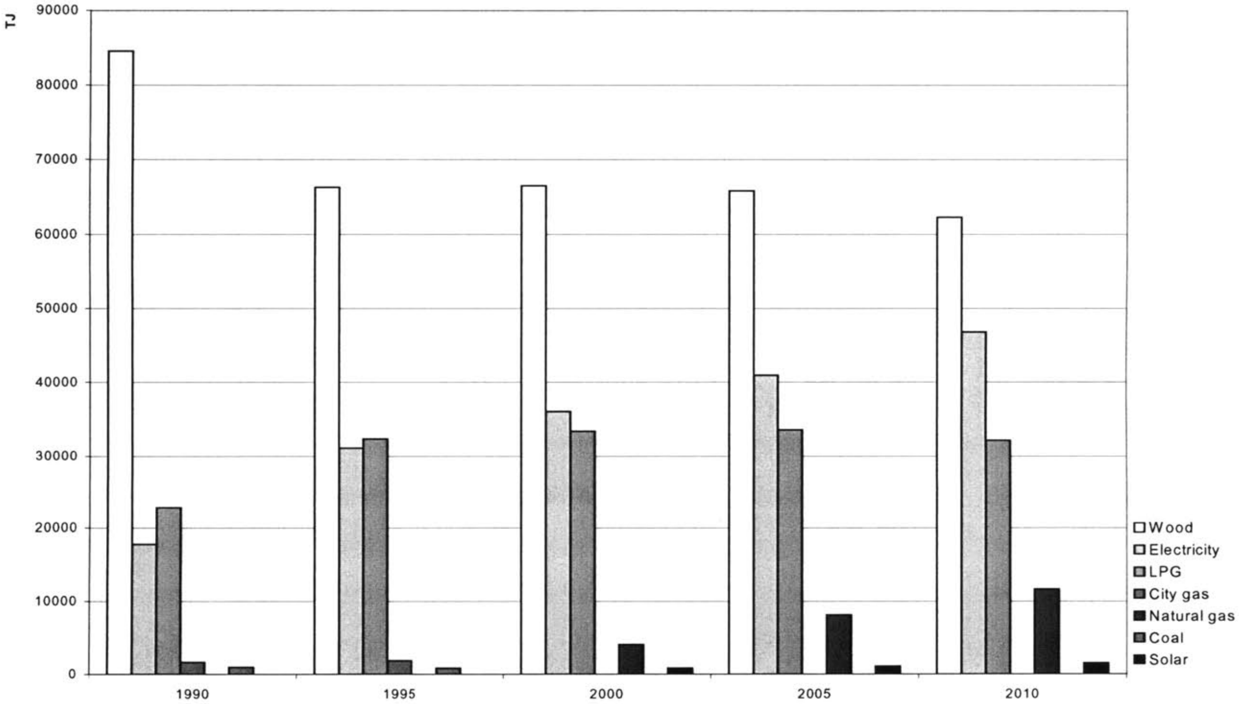


Figure 1.10 Energy type for the residential sector for the period 1990-2010 (TJ)

It is possible to see that wood is the main combustible used for heating, but its use will decrease by about 26%, while LPG, also used for heating, will increase by 41%. Electricity use will drastically increase by 163%, both for lighting, appliances/equipment, cooling and also heating. New forms of energy start emerging by 2000, like natural gas [whose contribution quickly grows] and solar energy, while previous ones like city gas and coal tend to disappear.

b) Services sector

The energy consumption of the services sector is expected to grow by 198% from 1990 to 2010. Electricity use is the main responsible for this fact, going from 5.5 TWh to 16.8 TWh, a 204% increase. Natural gas use also grows by 154% in the period 2000-2010. Table 1.4 and figure 1.11 Illustrate energy type for the services sector.

	1990	1995	2000	2005	2010
Electricity	19955	25412	39485	48276	60628
LPG	2980	4800	5443	6113	6113
Gasoil	2030	4708	2991	2884	2857
Fueloil	2024	1814	1884	1884	1759
Oil	1153	534	1699	1638	1623
Natural gas	-	-	3894	7746	9881
Solar	-	-	377	586	796

Table1.4 Energy type for the services sector for the period 1990-2010 (TJ)

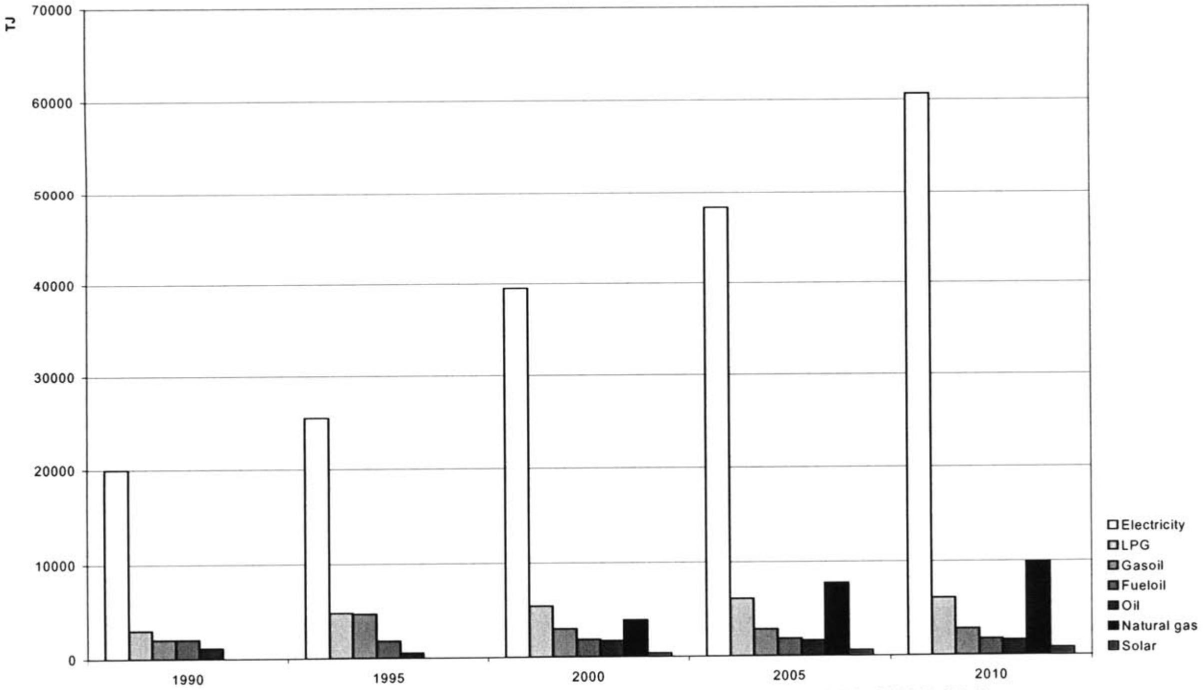


Figure 1.11 Energy type for the services sector for the period 1990-2010 (TJ)

1.6.4.2 Greenhouse gas emissions

a) Residential sector

GGE will increase by 55% in the residential sector in the period 1990-2010, from 5863 to 9083 Kt CO_2 equiv. The main contributors are electricity [72%] and natural gas. The decrease in wood use is not enough to counteract the other sectors. Electricity has the highest emission of GG per unit energy, which can be seen from the life cycle analysis of the different types of energy until they reach the final user (GASA, 2000).

As for the relative contribution of end-use areas that are a GGE source, they can be seen in figure 1.12.

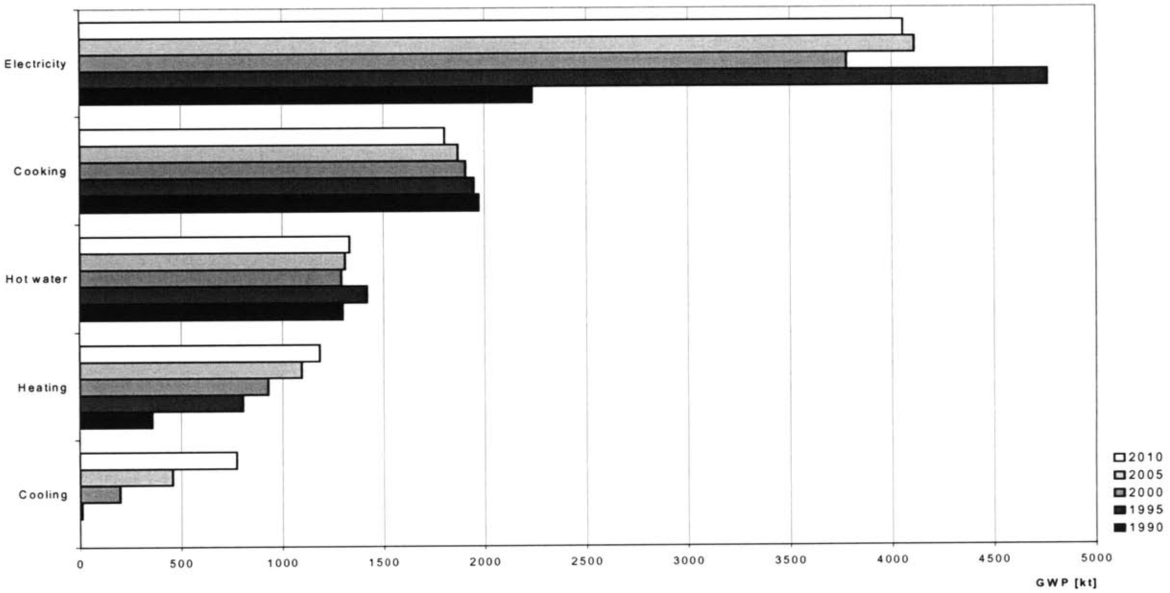


Figure 1.12 Relative contribution of end-use areas for GGE in the residential sector in Portugal

Electricity use for lighting and appliances was already the main contributor by 1990, and is expected to increase by 81% by 2010. Heating has the highest percentual increase [231%], from a quite low contribution in 1995 to becoming almost as high as hot water in 2010. Cooling, non-existent in 1990, has an extremely fast growth that shows, as heating does, the increase in comfort standards of the population and enhanced economic capacity.

b) Services sector

It is predicted that the Global Warming Potential [GWP] of the services sector in Portugal will increase by about 105% until 2010. Electricity (98%), LPG (105%) and natural gas (154% from 2000 to 2010), are the main responsible for this growth. As can be seen in figure 1.13, electricity is by large the main contributor to the GWP of services buildings.

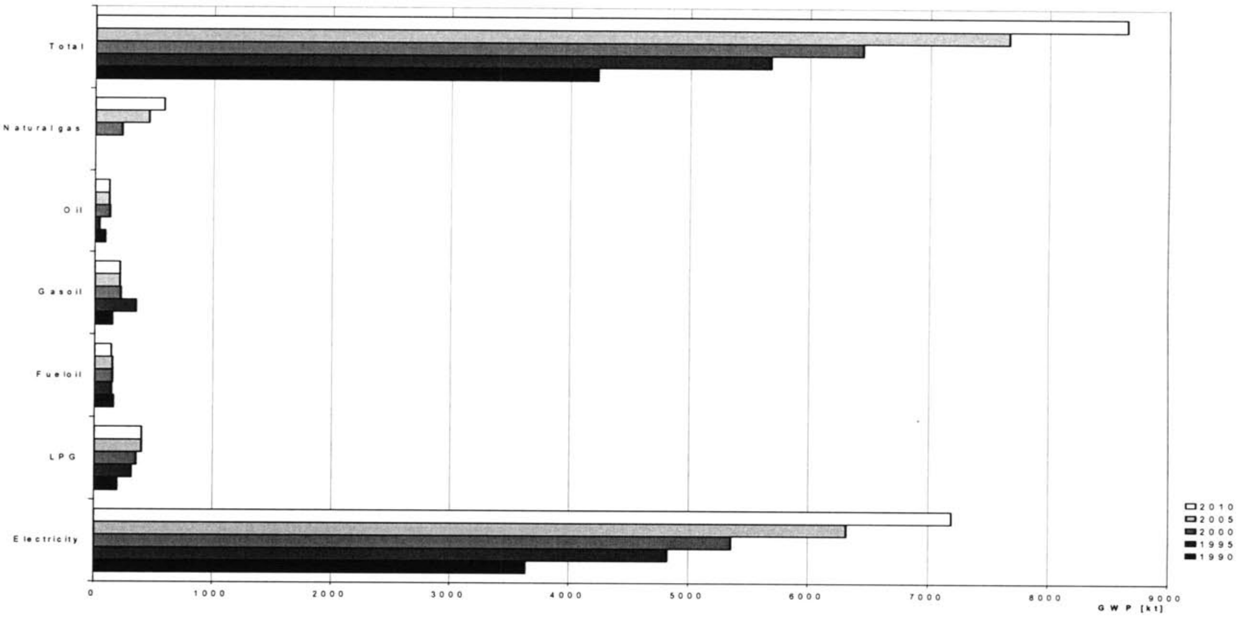


Figure 1.13 Global Warming Potential [GWP] in the services sector, according to type of energy

Unfortunately there was no information available on GWP by end use [heating, cooling, lighting, etc.] in the services sector. Some insight into this area may be gained by looking at results from a GASA study [GASA, 2000], where a number of energy efficient measures was simulated and assessed in terms of their potential for reducing GGE by 2010.

The following measures were considered for the services sector:

a) Revising the current Building Energy Code and make it 40% more restrictive, for new constructions;

b) Retrofitting existing buildings [50% of those built until 1990], improving wall and roof insulation and applying double glazing units;

c) Active solar energy: installing 400,000 m^2 of solar panels for water heating, replacing gas or electricity;

d) Lighting efficiency measures: replacing incandescent bulbs for compact fluorescent ones; using high frequency ballasters in fluorescent lamps.

Table 1.5 below shows the predicted effectiveness of each of these measures in reducing CO_2 emissions until 2010.

Measures	**CO_2 reduction potential [kt CO_2]**	**% Δ services sector**
A	142	1.6
B	31	0.36
C	99	1.14
D	347	4.0
Total	**619**	**7.2**

Table 1.5 Impact of the measures considered in reducing GGE.

It can be seen that although the impact of the measures considered is low in general, the largest contribution comes from strategies to improve artificial lighting efficiency [4% reduction of the expected GEE increase by 2010]. This suggests lighting plays a significant role in the electricity consumption in the services sector. Strategies for improving daylighting use in services buildings could thus also play an important role in reducing electric energy consumption in services buildings and thus greenhouse gas emissions, since they look at decreasing the need for artificial lighting instead of just improving its efficiency.

1.6.5 Conclusions

From the analysis presented it is possible to see that both for the European Union in general and for Portugal in particular, improving energy efficiency in buildings is a matter of current concern. For the EU, 40.7% of the energy consumed in 1997 was in buildings. In Portugal, current prediction place GEE increasing by 52% in the period 1990-2010, well above the 27% determined by the 'burden sharing' agreement between EU countries to meet the Kyoto Protocol commitment. The building sector in expected to increase its contribution to GEE from 17% to 20% due to the services sector, where growth in electricity consumption is the main factor.

For Portugal, energy inefficiency is also a source of economic problems. For example, in 2000 Portugal paid twice as much for its energy bill than in the previous year [including electricity, oil and refinery products, coal and natural gas] (Martins, 2001). This was partly due to the increase in the price of oil, but also because of the increase in energy consumption, which grew by 8% in the first trimester of 2001, almost three times more than the country's economic growth, which stayed below 3%. That is, to produce a unit of wealth, the country spent three times more energy, what is a good indicator of the existing inefficiency. That is even more significant if one considers the fact that Portugal imports a large part of the energy it consumes, what makes it very vulnerable to fluctuations in energy costs.

Finally, apart from Italy, Portugal has the most expensive electricity in the EU for residential use (Diário de Noticias, 2001). It costs almost 100% more than in the UK, Greece or Switzerland [not an EU member], 33% more than Spain, and has a similar value to that of Germany, where purchase power is much higher. So, for the general population, decreasing electric energy consumption means significantly reducing energy bills, since electricity accounts for about 59% of the utility bills paid by a Portuguese household [GASA, 2000][1].

Given this framework, the research developed for this dissertation may be relevant in that it helps to contribute to find new ways of improving buildings energy efficiency, looking at strategies as daylighting use and the best balance between lighting, heat gains and heat losses, for different building uses, layouts, and geographical locations.

[1] For the industry, electricity in Portugal is 70% more expensive than in Finland or Switzerland, and about the same as in the UK.

CHAPTER 2 Literature review

2.1 Introduction

Several of the chapters in this dissertation include their own literature review section. Chapter 3, which describes the development of the Generative System, will mention previous studies using optimization methods for energy-related architectural problems. Chapter 5, where a comparison is done between Genetic Algorithms, Simulated Annealing and Tabu Search, reviews a number of studies doing similar comparative studies. Chapter 8, on multicriteria optimization and Pareto-Based approaches, includes a quite thorough review of the topic.

In this chapter, a review of the work that was most influential to the development of this project will be done, together with a description of work by others that use somewhat similar concepts and methods to those described here, or whose approximation may bring new insights for further developments of this Generative System. Some earlier applications of optimization techniques to architecture-related problems are also briefly reviewed.

2.2 Background

Research on decreasing energy consumption in buildings usually relies on parametric studies, where a series of simulations are performed after manually changing the value of some variable under study to assess its impact on overall building behavior.

Parametric studies on the use of dynamic, adaptable building envelope materials, like electrochromic glazings, have also been quite extensively done, mainly at Lawrence Berkeley National Laboratory [LNBL] (Sullivan et al., 1997; Moeck et al., 1998, Reilly et al., 1991). In an unpublished Masters thesis (Caldas, 1995), I had also made extensive use of parametric studies to study dynamic building envelopes including electrochromic glazings, where the interest for architecture adaptation to the exterior environment was already a main driving point in the research. After that work had been completed, the initial contact with the Genetic Algorithms concept (Goldberg, 1989) led to new speculations about the way the adaptation paradigm might be embedded in architecture, not only at the operation stage, as was the case with adaptive materials, but earlier on, at design stage. Some experimental work being developed by John Frazer at the Architectural Association in London (Frazer, 1995) also provided some speculative background for the ideas that were in the genesis of this dissertation. The initial idea was to use evolutionary computation, like genetic algorithms, to explore new façade configurations including changeable materials that would increase the building's adaptation to the external environment. As the work developed, the idea of including adaptive materials in the research was abandoned [although it is still considered pertinent, and a possible path for future work], and new concepts, such as those of geometric shape manipulation, emerged and were investigated. More sophisticated views of adaptation mechanisms were gained from Holland (1996) and Gell-Mann (1995), namely on the study of complex adaptive systems, and will be later discussed in the 'further work' section, as they suggest interesting potential expansions for the work developed in this dissertation.

Given this initial background, the literature review will first focus on Genetic Algorithms concepts and on studies by others applying these methods to architecture related problems. It will then expand to other heuristic methods like Simulated Annealing, and finally will provide an overall view of some previous applications of other optimization methods to architecture.

2.3 Genetic algorithms

A genetic algorithm is a search technique adequate for searching noisy solution spaces with local and global minima. Because it searches from a population of points, not a single point, the probability of the search getting trapped in a local minimum is limited. GAs start searching by randomly sampling within the solution space, and then use stochastic operators to direct a hill-climbing process based on objective function values (Goldberg 1989).

Under GA terminology, a solution to a problem is an individual and the group of solutions existent at each stage is a population. Each time a new population of individuals is created it is called a generation. In binary GAs like the one used in this study, each individual is represented by a binary string called a chromosome, which encodes all the parameters of interest corresponding to that individual. A chromosome is formed of ailleles, the binary coding bits. The fitness of any particular individual corresponds to the value of the objective function at that point.

Genetic operators control the evolution of successive generations. The three basic genetic operators are reproduction, crossover and mutation. The probability of a given solution being chosen for reproduction is proportional to the fitness of that solution. Crossover implies that parts of two randomly chosen chromosomes will be swapped to create a new individual. Mutation involves randomly changing an allele in a solution to look for new points in the solution space. Although there are more elaborate versions of these operators, the basic principles remain similar for most GAs.

A genetic algorithm starts by generating a number of possible solutions to a problem, evaluates them and applies the basic genetic operators to that initial population according to the individual fitness of each individual. This process generates a new population with higher average fitness than the previous one, which will in turn be evaluated. The cycle will be repeated for the number of generations set by the user, which is dependent on problem complexity. Despite their apparent simplicity, GAs have proved to have high efficacy in solving complex problems which other, more conventional optimization methods, may have difficulties with, namely by being trapped in local minima.

Genetic algorithms have been used in building applications related to energy consumption, mostly to optimize the sizing and control of HVAC [Heating, Ventilating and Air Conditioning] systems (Huang et al. 1997, Wright 1996, Dickinson et al. 1995). At the structural analysis level, GAs have been used for design optimization of trusses (Camp et al. 1998, Galante 1996, Goldberg 1989), beams (Wang et al. 1996, Marcelin et al. 1995) and columns (Ishida et al. 1995). Several applications of evolutive design to floor layout problems have also been found (Gero et al. 1998, Jo et al. 1998, Damsky et al. 1997, Pham et al. 1992), either using standard GAs or 'evolved genes.'

In this section we looked for examples, mostly in building applications, where a non-linear optimization problem was approached by coupling a GA with a problem-dependent simulation software working like a 'black box,' in much the same way as we did in our problem.

Huang at al. (1997) used Genetic Algorithms to optimize the control parameters of an HVAC system. The tuning of the digital controllers of the system involves the selection of controller settings [like temperature settings] and controller sampling time. The genetic algorithm was coupled with a HVAC simulation package, HVACSIM+ ©, which was used to perform a detailed simulation of the system for each control-parameter configuration. The objective function to be minimized was a combination of three performance indicators [overshot, settling time and mean squared error]. The HVACSIM+ software was treated as a 'black-box' simulator.

Dickinson et al. (1995) used a GA combined with a hill-climbing procedure to optimize fixed parameters in a building heating control system, and also dynamic scheduling inputs that varied throughout the duration of the run. A building heating systems simulation package [QUEST ©] was used to calculate the fitness function in a 'black box' approach.

Wright (1996) also used Genetic Algorithms to optimize design variables in a HVAC system. The HVAC system variables are the physical dimensions of the system components [discrete], the operating point of the system as represented by the controller set-points, and the fluid state parameters such as the maximum mass flow rates [continuous]. The design constraints cover the upper limit of the fluid velocities, the control range of the fin-tube coils, the water circuit configuration of the coils, the maximum supply air to room temperature difference, and the failure of the simulation. Failure of the simulation would indicate the undersizing or oversizing of one or more components. The objective function was the capital cost of the system. Although a full HVAC simulator was used to determine the feasibility of the system, the final fitness value was calculated as a linear sum of the component costs of the system, so it is a different case from the two others described above that use non-linear fitness functions.

In terms of structures, Camp et al. (1998) coupled a GA with a Finite Element Analysis Program [FEAP] to optimize two-dimensional structures. The program allows a number of useful features like design checking using the American Institute of Steel Construction Allowable Stress Design specifications and a comprehensive database of available structural steel members. The objective function is total weight of the structure, which is a non-linear function of several outputs provided by the FEAP software.

Marcelin et al. (1995) used a GA to optimize the behavior of composite beam structures. The aim of the research was to improve the damping ability of a composite beam structure without significantly increasing its weight or decreasing its stiffness characteristics. Variables were the selection of materials and combination layout of the several layers: number and thickness of layers, and stacking sequence. The constraints were the stiffness characteristics and weight. The fitness function was non-linear, the factor of dampness achieved.

Wang et al. (1996) approached the problem of finding the support locations of a uniform beam to maximize the fundamental natural frequency of the beam using a Genetic Algorithm. The length, bending rigidity and mass density of the beam are fixed. Variables are the types of support considered [elastic and rigid] and the location of the supports. The fitness function is the fundamental natural frequency of the beam, calculated using the Rayleigh-Ritz method.

Many other applications of Genetic Algorithms in the context of buildings exist, like optimization of steel roof trusses (Koumousis et al., 1994), elevator installation design (Yoshiaki et al., 1996) and space layout problems (Gero, 1998; Pham, 1992; Jo, 1998; Damsky, 1997). Investigations on GA applications for floor plans layout may gain some relevance for this work when it is expanded to include the generation of new spatial configurations within a fixed perimeter, as will be discussed on the section on 'further work.'

Attempts have also been made to apply GAs to the design of three-dimensional shapes, to help designers at early stages of design mainly by proposing unexpectedly innovative shapes. The difficulty of finding appropriate objective functions for design evaluation has made some of these attempts remain at an experimental level, with selection and crossover being performed manually by the designer (O'Reilly et al., 1998).

A more sophisticated surface-generation system is MoSS: Morphogenetic Surface Structure (Testa, 2000), which uses Lindenmayer systems [not GAs] in three dimensions to grow surfaces by recursively applying rules to a basic shape defined by the user. It currently runs as a plug-in for Alias|Maya ©, and allows the 'environment' to influence surface growth by means of boundaries or objects that attract or repel direction of growth.

Again, MoSS has no evaluation procedure attached to it, so the user must decide which surfaces are more adequate to his purpose.

2.4 Simulated Annealing

Simulated annealing is based on the analogy between the annealing of solids and the problem of solving large combinatorial optimization problems. It leads a search progression based on a type of modified descent method, where nonimproving solutions may also be accepted with some probability. A detailed description of the simulated annealing algorithm will be provided in chapter 5.

Not so many examples were found in the literature of applications of Simulated Annealing to building-related problems. A particularly interesting study is presented by Monks et al. (1998), using a combination of Simulated Annealing and steepest descent to optimize the acoustical performance of architectural spaces such as auditoria. The variables used relate to the physical characteristics of the space, like room geometry [for example, tilt of reflectors and other surfaces], and materials characteristics [materials are chosen from a given library, and can be combined to form a given component]. The objective function used is a combination of six outcome measures, an evaluation approach developed by Beranek, and known as the Objective Rating Method. The space performance is calculated by an acoustical simulation software, that uses an hybrid method of simulating the early sound with beam tracing and late sound with a statistical approximation. The concept the authors use is that of 'goal-based' or 'inverse design,' where the desired acoustic performance is set a priori, and the material and geometric parameters of the environment are determined by the optimization algorithm in order to approach that goal. This work relies on the use of computer graphics both for the user to interactively define constraints [geometric and materials] and for visualization of results of the acoustical simulations and of each of the outcome measures. This work is a very sophisticated and complete piece of research, and its inclusion of surface tilts manipulation influenced our initial steps towards shape generation experiments.

Another interesting approach is that of 'shape annealing' (Cagan and Mitchell, 1993), a design generation technique that combines concepts from shape grammars and stochastic optimization [usually simulated annealing] to produce optimally directed designs.

To better understand the concept of shape annealing, that of shape grammars must be explained first. Shape Grammars [SG] (Stiny, 1980) are used to create shape designs via iterative application of grammar rules, which act on a shape design and transform it into a new one. A SG consists of four components: a finite set of rules that perform various transformations on the current design, a finite set of shapes, a finite set of labels used to attach additional information to the shapes, and an initial shape as the starting point for the grammar, which may be the null shape.

The shape annealing method can thus be described as follows (Szykman and Cagan, 1993): given a current design stage, a valid rule [i.e., one that has its preconditions satisfied] is selected and applied to the design. Applying a rule to the current design and transforming it into a new design corresponds to taking a step in the simulated annealing algorithm. If the step results in an improvement for the objective function, that step, and thus the new design, is accepted. If it results in a worst objective function value, it can still be accepted with some probability dictated by the cooling schedule [which will be later explained in chapter 5]. In this method, simulated annealing is thus used to control the design generation itself.

Szykman and Cagan (1993) proposed an extension of the shape annealing concept by using a three-dimensional shape annealing algorithm, which implied the use of a three-dimensional shape grammar, to generate 3D component layouts. Finding an objective function can be one of the most complex and controversial parts of these implementations. In this case, the objective function reflected a preference for more 'compact' designs.

Perhaps one of the most interesting researches using shape annealing is that of Kristina Shea (Shea, 1997; Shea and Cagan, 1998), which applies it to structural optimization. Although the concept of structural generation and optimization using shape annealing was first introduced by Reddy and Cagan (1995, a & b), Shea further developed it, turning the method into a more consistent and robust one. The design representation of structures is based on an analogy with a network, where the design objects are joints and the relations between them represent the flow of forces through the structural members. A finite-element analysis method is then used to calculate the performance of the structure. The grammars used are based on triangles, which can be divided, joined, etc. Shea et al.(1997) first proposed a planar truss grammar, and Shea and Cagan (1997) proposed a space truss grammar, which is used to generate geodesic-like domes. The selection of rules to be applied at each step is a dynamic, probabilistic one. There are also size modification rules, apart from shape modification ones [for example, the cross section of a given element can be altered]. To run the system, some initial specifications are required, such as the number and location of supports, materials properties and if symmetry is required or not. Constraints relate to factors as structural stress, Euler buckling, geometric obstacles, etc. The objective functions that can be included are several, relating to efficiency [achieving minimum mass], economy [minimum number of distinct cross-sections and distinct lengths], utility [maximum enclosure space, with minimum surface area], and even aesthetics [achieving a uniform metric or a golden ration metric]. The final objective function value is calculated as a weighted sum of the objective functions chosen, plus a weighted sum of constraints violations [since the system allows for the appearance of solutions that are not structurally viable]. For a given, user-defined loading condition, the structure is analyzed for violation of behavioral limits [stress, buckling, displacement] and these violations are summed resulting on the total violation of all loading conditions (Shea and Cagan, 1998).

Sohn et al. (1999) use Simulated Annealing and steepest descent to estimate air exchange rates and effective mixing volumes of a partitionless building with heterogeneous spatial flow conditions. The method uses simulated tracer gas measurements, which vary in quantity and quality, and a non-linear model (Sinden's first-order multicompartmental conservation of mass approach) is used to estimate air change rates based on concentrations of the tracer gas. The objective function is the difference between measured results and those estimated by SA.

2.5 Other optimization studies and methods

2.5.1 Dynamic Programming

Gero (1978) used dynamic programming to approach an optimal lighting problem using artificial lighting sources to light a room, while minimizing the cost of providing a minimum desired direct illuminance. The variables under consideration are the number of light sources, their position in the room, and their power [it is assumed that the luminous intensity of the sources is equal in all directions]. The reflected component of illumination is not taken into account. However, the problem under study does not meet the requirements of separability of the optimization process in different independent stages, what is usually necessary to apply dynamic programming methods, since a source added at a later stage will influence the quality of a decision taken at an earlier stage, and vice-versa. The problem has thus both feed forward and feedback interactions. To deal with this situation, Gero introduces the concept of a delayed decision, meaning that the stage at which the decision is taken is later than the stage for which the decision is taken. A difficulty of using this method is that the number of states which may need to be stored between the two stages can become very large, depending on the range of influence of each light source. The algorithm can thus become very computationally expensive and slow, and after a certain number of variables it may become infeasible. In the same paper, Gero also suggests that a similar approach can be used to size and position roof lights, windows and shading devices, although there is no actual implementation of those suggestions. This work by Gero will be revisited in chapter 3, section 3.4. An internal report by Radford (1978) will further develop the methods proposed by Gero to apply them to window design. Radford and Gero (1978) had already published a paper proposing the application of dynamic programming to window design, but the method presented was still at an initial, embryonic level.

Later on, Radford and Gero (1985) also suggested the use of Pareto-optimality concepts to multicriteria optimization in architectural design, a topic that will be explored in some depth in chapter 8 of this dissertation.

2.5.2 Nonlinear programming

The use of a nonlinear optimization technique has been proposed for optimal daylighting design by Law (1997). The study couples the Levenberg-Marquardt method, which is an extension and improvement of Newton's method, with a ray-tracing based lighting simulation software to achieve high quality solutions in terms of daylight use.

Nonlinear programming has also been used by Mitchell (1976) for the synthesis and optimization of small rectangular floor plans. Given certain specified requirements about spatial adjacencies, the algorithm exhaustively produces all topological arrangements that satisfy those requirements. For the dimensioning step, linear programming may be used under special conditions to find the least-cost solutions, but a more general solution requires the use of nonlinear programming methods.

Much work had been done since the 1960's on floor-plan layout optimization. Early work, like the CRAFT program (Armour and Buffa, 1963) used a square grid representation of a floor plan, and employed the quadratic assignment formulation of the floor layout problem. CRAFT used a representation of layouts based on a two-dimensional array of integers. According to Mitchell (1974), "each integer represents a square module of space of some defined dimensions (...). Letting S = {i_1, i_2,..., i_m) be the set of modules to be located and R = {j_1, j_2,..., j_m) be the set of possible locations in the array, a particular map ρ can be represented as":

$$\rho = \begin{matrix} i_1, i_2, \ldots, i_m \\ j_1, j_2, \ldots, j_m \end{matrix}$$

There will be m! such maps, and CRAFT searches among them to discover the ones that possess the desired characteristics, like particular adjacencies.

Many other floor plan synthesis programs developed in the 60's and 70's used a similar approach, with generalizations to include rectangular and other grids, and to three-dimensions too. Nonlinear programming techniques were used for this type of problems too. Nonlinear programming has been used together with dimensionless representations of floor plans to generate optimum dimensioned layouts, with respect to some cost criteria and subject to functional constraints (Mitchell, 1977). As mentioned above, floor layout problems may be part of the future work coming from this project, bringing a renewed interest to these earlier studies.

2.5.3 Linear programming

Linear programming has also been used for certain floor layout problems, which can be formulated in terms of a linear objective function [like minimization of overall plan length, width, perimeter, etc.] and subject to linear constraints. However, there are many limitations to the applicability of such method in architecture-related problems. For example, area constraints are already nonlinear, and thus cannot be incorporated in this type of formulations. Objective functions that are a function of surface area, like construction cost or heat loss, are also excluded (Mitchell, 1977).

CHAPTER 3 Development of Generative System

3.1 Introduction

This chapter describes the actual implementation of the Generative System. The system consists of two main components, the simulation module and the search | optimization module. The choices for each of these modules are described and discussed, as well as the method for incorporating them in the overall system. A description of current input and output formats is given, with some insights towards possible paths of development.

The diagram below summarizes the flow of information between the several components of the Generative System:

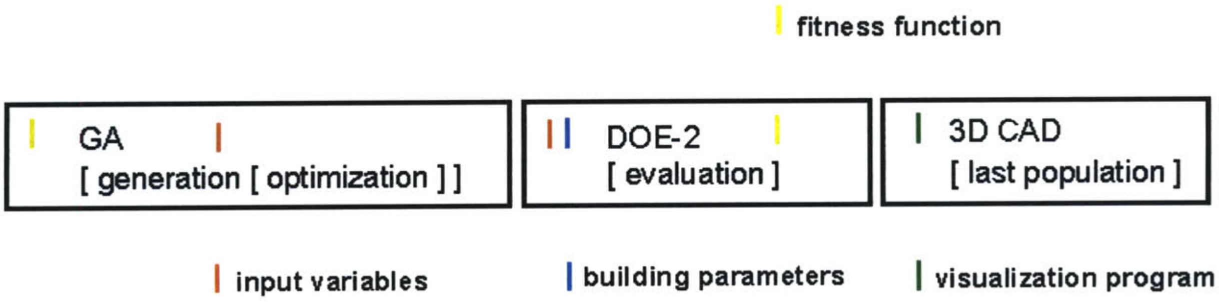


Figure 3.1 Flow of information across the Generative System components

The Genetic Algorithm [search and optimization module] first generates values for the input variables, which together with the remaining building parameters are passed into DOE2 [evaluation module]. DOE2 performs the evaluation of the resulting solution, and returns its fitness function value, which in the simplest case is just the annual energy consumption of the building. This value is fed back to the GA, which uses it to guide the search progression. When the GA reaches the last population [after the number of generations determined by the user], the corresponding results can be inspected using an existing visualization program.

3.2 The Simulation module: DOE2.1E capabilities and limitations

DOE2 has been developed since the 1980's by Lawrence Berkeley National Laboratory [LBNL], for the US Department of Energy. It is one of the more respected programs available for building simulation, and has been used in the construction or renovation of several well-known buildings, such as the Pentagon, the U.S. State Department headquarters, the U.S. Embassy in Berlin, the Monterey Bay Aquarium, the San Francisco Airport, the Ronald Reagan Library, the Intel and American Express corporate headquarters, numerous federal courthouses, and the Frank Lloyd Wright Museum in Wisconsin (LBNL, 2001).

DOE2.1E (SRG, 1993) was chosen to be the simulation engine of this Generative System because it performs both thermal and lighting calculations, and because it offers good accuracy for reasonable computational times. It was considered that, in this type of system, accounting for only lighting or thermal behavior without including the other aspect would be rather incomplete. In the future it may be possible to develop similar joint analyses by coupling programs like Computational Fluid Dynamics [CFD] and accurate lighting simulation software like Radiance ©. However, at the present time, and considering the several hundred evaluations the Generative System performs, including this type of programs would be infeasible, for in a standard personal computer one might have to wait unreasonable amounts of time to obtain results for a moderately complex building.

DOE2.1E has some major drawbacks, one being that it cannot compute air flow patterns in the spaces, thus being unable to simulate, for example, natural ventilation. For reasonably accurate simulations of air movement, one must resort to Computational Fluid Dynamics. DOE2, being a single-node system, does not allow the prediction of variations in air temperature in different points of the space, or prediction of air movement patterns. Although it is possible to specify different infiltration methods [some taking into account outside wind velocity, like the Crack Method, others just allowing the user to input the number of air changes per hour], this information is only used to calculate the thermal loads of the space. For the 'Residential System' option, it is possible to set a very simplified natural ventilation schedule, either specifying number of air changes and a possible schedule for that, or specifying the fraction of ventilation area used, and a schedule for that too [a simulation of window opening patterns]. All these methods are very simplified and do not model airflow inside the building.

Neither can DOE2 be used for performing thermal comfort predictions. The new software recently released by LNBL, Energy Plus, which incorporates the main features of both DOE2 and BLAST, and some additional ones, is able to do thermal comfort modeling since it incorporates mean radiant temperature calculations. However, at the time this Generative System was developed, Energy Plus was not available yet. Nevertheless, the system could be easily upgraded to include Energy Plus[2] instead of DOE2, in the same way it can be upgraded to include many other simulation programs.

DOE2.1E is mainly a thermal simulation program, although it includes lighting calculations, based on the daylight factor method (Winkelmann, 1983, Selkowitz et al., 1982). Departing from a weather file for the geographical location of the building, and from a detailed description of its geometry, construction materials and mechanical systems, DOE2 performs dynamic, hour-by-hour simulations of the building for the whole year. DOE2's accuracy in predicting thermal loads in buildings has been tested against actual measurements in buildings. One of these studies, comparing DOE2.1E predictions with IEA Annex 21 measurements of inside air temperature in three unconditioned test cells located in England [U.K. Building Research Establishment and DeMonfort Univeristy, 1993], shows that DOE2.1E predictions are in excellent agreement with actual measurements done, despite the fact that it performs simplified calculations [like using weighting factors instead of solving all the heat transfer calculations across the building].

Another reason DOE2 is often the choice for building simulations is its high degree of flexibility in terms of the features that can be modeled. As for the building fabric, DOE2.1E includes a vast library of opaque construction materials which can be layered in any desired way, and the possibility of adding user-defined materials. It also has a large glazing library, with photometric results extracted from the Window4© program [also developed by LBNL]. The user can add new glazing materials by running first Window4, provided by LNBL for free, and then incorporating the result file into the DOE2 library. For windows, there is the possibility of simulating fixed shading elements like overhangs and fins, and also movable shading elements [roller blinds, curtains, etc.], for which it is also possible to define a control strategy, dependent

[2] Some of the new features of Energy Plus in relation to DOE-2 are: realistic system controls, moisture absorption and desorption in building elements, interzone air flow, low temperature radiant heating/cooling, interior surface convection, thermal comfort modeling options, evaporative cooler models, steam absorption chiller, air flow sizing based on zone requirements, accurate sky illumination model for daylighting calculations, ability to read multiple interval per hour weather data files, plenums, enhanced calculation of return air heat gain from lights, flat plate exhaust air heat recovery, example heating, ventilating, and air-conditioning system and equipment input templates and user-customizable reports [http://gundog.lbl.gov]

on total incident solar radiation, space load, or other factors. There is an additional module that allows the simulation of switchable glazing [like electrochromic glass], which many researchers with an interest in this kind of material have found difficult to module. The program can also simulate special features like sunspaces, trombe walls, etc.

DOE2 makes use of a specific language to describe building geometry and materials, called the Building Description Language [BDL]. Within this language, there is a palette of elements that can be combined to define a space, and also a complex coordinate system working at three levels: the building coordinate system, the room system, and the wall system. Maintaining the internal coherence of these three coordinate systems in a building description is probably one of the most difficult parts to master in terms of inputting building geometry and space layout into DOE2. BDL is flexible enough to describe most types of building geometries, except for curved surfaces, which have to be approximated by a number of planes. Surfaces like walls, roofs or floors can have any orientation and be tilted in any direction. In previous versions of DOE2, these surfaces had to be rectangles, but in the latest versions a surface can be any kind of polygon, with the limitation of a maximum of 20 vertex points.

DOE2 also has a very extensive and detailed library of HVAC systems, heating systems, equipment types, operation modes, control strategies, etc. This includes less common, energy conservation systems like absorption chillers, heat pumps, and many others. Finally, DOE2 provides a large number of tabulated reports, with information ranging from the total annual energy consumption of the building [in the Building Energy Performance Summary (BEPS)], to very detailed information about individual spaces or environmental variables, if requested by the user.

Overall, it was considered that DOE2 provided a good compromise between reliability of results and computational speed, while allowing a quite high degree of flexibility in terms of building geometries modeled and construction and operation details.

3.3 The Search module: Genetic Algorithms - description, advantages and shortfalls

The main criteria for choosing a Genetic Algorithm for the search and optimization module was the fact that GAs provide a population of final solutions instead of a single one. This represented a move towards the adaptation paradigm that was in the genesis of the work, and was also considered to be an important factor in an application area like architecture. A system that would provide a single 'optimized' solution might have poor acceptance among architects, who might prefer to have a range of choices between different solutions, all having a high standard of performance according to the objective functions included in the search, and from where the architect could exert further choice, according to other criteria like personal preference, initial design intentions, etc. It will be seen in chapter 5 that GAs differ considerably from other methods described there, like Simulated Annealing and Tabu Search, in that it searches from a population of points instead of a single point. Using a population of solutions is thus intrinsic to the mechanics of the algorithm, apart from being a beneficial advantage in terms of its solutions.

The validation studies described in chapter 4 proved the use of GAs to be a good option as a search method. Although the solution spaces do present global minima as well as local ones, they tend to be quite flat around the global minima. Usually, the solution spaces found in an architectural design problem are not characterized by the presence of spikes [that is, having very good solutions surrounded by very poor ones]. This means that within some range, the relative impact of changing some of the variables is small. Outside of this tolerance range, however, the impact in objective function values can be quite significant. This can provide useful information to the architect about the more sensitive points in a design, and on the consequences of introducing certain changes in relation to the best solution found by the algorithm. Because this system was conceived as an open-ended tool, one with which the architect could engage in an interactive loop, by running the program, getting some results, introducing some changes, and running the system again, this was considered to be an important advantage towards the use of GAs.

Another criterion for using a genetic algorithm was that GAs use a coding of the variables during the search process, not the variables themselves. This applies mainly to binary GAs, the ones that use only binary language [0,1] in its chromosomes. The chromosomes can then be decoded into any kind of variable, numerical or non-numerical, what provides an extra degree of flexibility towards using the system for architecture applications. For example, part of a chromosome can be decoded into a material name, or into any other kind of non-numerical variable.

Before going into more details about the use of GAs in this work, a description of some important characteristics of this search method is given below. Chapter 2 provided a quick overview of the basic principles behind typical GAs. One of the aspects mentioned was the use of the three basic genetic operators: reproduction, crossover and mutation. This section will complement that initial information with a closer insight into the different operators.

The most common implementation of reproduction is that of the biased roulette wheel (Goldberg, 1989) where each current string in the population has a roulette wheel slot sized in proportion to its fitness. That means the probability of a given individual being selected for reproduction is proportional to its fitness:

$$P(x) = f(x) / \sum_{j=1}^{n} f(j)$$

where n is the number of individuals in the generation

In this way, fitter individuals have a higher probability of being selected to the next generation, although individuals with poor fitness still have some probability of doing so. There are other strategies such as elitism that can also be implemented in conjunction with reproduction. Elitism implies always passing the best solution from a generation to the next generation, in order to preserve valuable genetic information contained in that solution.

Other reproduction operators have also been proposed in the literature, such as multi-parent reproduction (Eiben, 1995), parallel steady-state reproduction (Godart, 1996) and gradient-like reproduction (Pham, 1995). The reader is referred to the literature for detailed information about these alternative methods.

Roulette wheel selection is usually combined with other methods to determine which individuals will pass to the next generation. The most common process used for this purpose is deterministic tournament selection. In this method, two strings are randomly picked, with probability proportional to the biased roulette wheel slot, and this pair will compete for presence in the next generation. The one with the highest fitness is passed. These two individuals will have the chance to be chosen again for the next tournament, that is, they are not removed from the initial population after each tournament. For a population of size n, after performing n tournaments [or n-1, if elitism is used, since in that case the best solution will not have to compete for having a copy of itself in the following generation], the parent population for the next generation is complete. After that, crossover is applied among the parent population for generation of offspring.

Crossover is the most prominent operator in GAs, being responsible for most of the diversification occurring during the search process. During crossover, parts of two randomly chosen parent chromosomes will be swapped with a given probability to create a new individual. Only the elite solution will receive an exact copy of itself without going through crossover. The role of crossover is to interchange existing information between different solutions in order to generate new points in the solution space, but it does not create new genetic information itself, It is important that the genetic pool contained in the initial random population is diverse enough to contain potential information on the entire search space. This is why the initial population must be large enough, as will be discussed below.

There are several types of crossover operators, the most common being one-point crossover, two-point crossover and uniform crossover. In one-point crossover, the algorithm randomly picks two parents selected for reproduction and randomly assigns a crossover point, after which the remaining parts of the two chromosomes will be swapped. The process for two-point crossover is similar, with the exception that two crossover points are randomly chosen instead of one. In uniform crossover, two strings x and y are combined into a string z as follows (Muhlenbein, 1997):

$Z = (z_1, \ldots, z_n), \quad z_i = x_i \text{ or } z_i = y_i$

where x_i or y_l are chosen with equal probability
n is the chromosome length

Mutation involves randomly changing an allele in a solution, with a given probability, to look for new points in the solution space. Mutation is thus an operator that acts very locally. It is responsible for introducing new genetic information in the search that was not contained in the initial population. A rule of thumb, based on parametric studies, says a mutation rate $m=1/n$, where n is the size of the chromosome, is almost optimal (Muhlenbein, 1997).

Another central concept to Genetic Algorithms is that of building blocks. Building blocks are short, low-order and highly fit schemata that tend to present in high-performance solutions. A schema is a similarity template describing a subset of strings with similarities at certain string positions (Holland, 1975). For example, over the binary alphabet {0,1}, a schema trinary alphabet is created by appending the * or *don't care* symbol. This way, a schema matches a particular string if at every location in the schema a 1 matches a 1 in the string, a 0 matches a 0, and a * matches either. Schemata are important in GAs because they introduce a new wealth of information into the search by looking at similarities between solutions, particularly those similarities that lead to high performance solutions. This information will then help guide the search process.

In general, a binary string or chromosome contains 2^l schemata, where l is the length of the chromosome. A population of size n contains between 2^l and $n{\cdot}2^l$ schemata, depending on the diversity of the population (Goldberg, 1989). The phenomenon of implicit parallelism, also described by Goldberg, states that in a population of size n, about n^3 schemata are processed by GAs. This means that the algorithm is testing in parallel hundreds of potential building blocks. This search for particular high performance schemata is implicit in the GA mechanism, as there is no formal mechanism that recognizes good schemata and stores them in memory.

To analyze the effect of reproduction in schemata, Goldberg uses the following reasoning:
Let m(H,t) be the number of occurrences of schema H in the population at time t. During reproduction a string is copied with probability $p(x)=f(x)/\Sigma f(j)$. After reproduction of a population of size n the equation for the number of schemata H is:

$$m(H,t+1)=m(H,t){\cdot}n{\cdot}f(H)/\Sigma f(j)$$

where f(H) is the average fitness of strings representing schema H at time t.

Since the average fitness of the total population may be expressed as $f_{avg}=\Sigma f(j)/n$, the reproductive schema growth equation may be rewritten as:

$m(H,t+1)=m(H,t)\cdot f(H)/ f_{avg}$

This means that a particular schema will grow as the ratio of the average fitness of the schema to the average fitness of the population, that is, schemata with fitnesses above the average of the population will receive increasing number of copies and those with fitnesses below average will receive a decreasing number of copies. This behavior happens in parallel with all the schemata present in the populations.

To try to transpose this behavior to more quantitative terms, Goldberg (1989) uses the following reasoning:

"Suppose we assume that a particular schema H remains above average an amount $c\cdot f_{avg,}$ with c a constant. Under this assumption we can rewrite the schema difference equation as follows:

$m(H,t+1)=m(H,t)\cdot(f_{avg} +c\cdot f_{avg})/ f_{avg} = m(H,t) \cdot (1+c)$

Starting at t=0 and assuming a stationary value of c, we obtain the equation

$m(H,t+1)=m(H,t)\cdot(1+c)^{t}$

(...) The effect of reproduction is now quantitatively clear: reproduction allocates exponentially increasing (decreasing) number of trials to above- (below-) average schemata."

This notion is of quite significance to GA theory and its an important basis for what was given the name of Schema Theorem, which states that short, low-order, above average schemata receive exponentially increasing trials in subsequent generations.

However, Muhlenbein (1997) points out that this theorem cannot be used to explain the search strategy of the genetic algorithm, because there cannot be a schema remaining a constant c above average since the average fitness of the population is steadily increasing. But although the Schema Theorem cannot fully explain the success of genetic algorithms in finding high performance solutions in complex problems, empirical studies have proven this to be the case. Chapter 4 presents one empirical study done with the Generative System implemented for this research, and shows that even in very large solution spaces the GA could consistently find individuals with very good performance levels.

Syswerda (1989) examines three crossover operators (one-point, two-points and uniform crossover) in terms of their effect on schema survival and on schemata combination (combining schemata together in one genome). In terms of survival rate, both one-point and two-point crossover are better than uniform crossover. One-point crossover is the best for schemata with short defining length (the distance between the first and the last specific string position that defines the schema), but two-point crossover is better for longer defining lengths. Since the author demonstrates that most of the possible schemata in a string of a particular length are of higher defining lengths, two-point crossover is considered to be better.

In terms of schemata combination, one- and two-point crossover have better combination rates when the defining lengths of the schemata are low. But as explained above, most of the possible schemata have long defining lengths, and in that region uniform crossover performs best. Since the empirical results presented by the author suggest that a higher combination rate is more important than a better survival rate, uniform crossover is pointed out as the best of the three crossover operators. Following these conclusions, for this work uniform crossover was used, although the user is given the option of choosing between it and two-point crossover.

A main alteration in relation to standard GAs used in this Generative system was the application of a Micro-GA (Krishnakumar, 1989), which differs from a typical GA mainly in the small population size used and the type of search progression.

Not many guidelines exist for setting population sizes for a typical GA experiment. Those are problem dependent settings that must be fine tuned by the user. A study has suggested that n=16 might be too small a population (Goldberg, 1989), not containing enough diversity in the initial pool of individuals to ensure that good solutions are found by the algorithm. Large population sizes eventually lead to better convergence due to the larger pool of schemata available, but have an initial large inertia and so a poorer initial performance. They require a large number of generations in order to converge to high-performance solutions, so too large population sizes tend to slow down the algorithm. Moderate size populations may be a reasonable compromise between finding good solutions and speed.

Typical population sizes for GAs range from 30 to 200, based on earlier studies such as those of De Jong (1975) and Grefenstette (1986), where suggestions for optimal population choices based on parametric studies are presented. These large populations are supposed to lead to better schema processing, lesser chance of premature convergence and better optimal results, by introducing from the beginning a pool of diverse genetic information into the search. Since all the final solutions are already contained in some way in the initial population [apart from the small changes introduced by the mutation operator], the initial population needs to be large enough to ensure the presence of that wealth of genetic information in the process. However, the implications of using large populations are both large inertia in search progression and high time penalties due to the evaluation of the fitness functions for all the individuals, generation after generation.

Micro-Gas try to address that problem by using small populations, promoting early convergence, and then randomly generating a new population while maintaining the best individual from the previous generation [elitism]. This process introduces high variety in the process by often introducing new points into the search. Since many new random populations will be generated during the search progression, it is not necessary for the initial population to contain all the genetic information needed to reach the optimized solution, thus its size can be greatly reduced. Also, premature convergence of the population is not a concern in this method.

Giving a more detailed description, micro-GAs start with a small population [in the case of this study, of only five individuals], which quickly converges to an initial solution. Convergence is measured by comparing the chromosomes of the individual solutions. If they differ by less that 5%, it is considered the population has converged. When that happens, the Micro-GA generates a new random population while carrying over the individual with best fitness from the previous generation. This way, new individuals are often brought into the search without loosing track of the one that did better until that point. Another advantage of using Micro-GAs is that the algorithm tends to perform a local search around the best solutions during the generations prior to convergence, since at that stage solutions only differ by a few alleles. This local search is important in finding local minima around good solutions, and is usually hard to implement in conventional GAs.

The steps involved in the Micro-GA process are (Krishnakumar 1989):

1. Start with a small population of randomly generated individuals [for example, 5 individuals]
2. Evaluate fitness of the population, determine best solution and copy it to the next generation [elitism].
3. Choose the remaining four strings for reproduction, with the best string also competing for reproduction, using deterministic tournament selection [strings are grouped randomly and adjacent pairs compete for presence in the next generation].
4. Apply crossover with probability =1. This allows high order of schema processing. Mutation rate is kept to 0 as enough diversity is introduced after convergence by the generation of a new random population.
5. Check for nominal convergence using bit wise convergence. If population has not converged, go to step 2. If it has converged, start a new random population while keeping the best string, and go to step 2.

Note: Applying crossover with probability = 1 in uniform crossover means that all alleles in all individuals, except for one copy of the elite one, will undergo crossover.

Micro-GAs tend to quickly direct the search to optimum or near optimum solutions, but the average performance of the final population may not be necessarily high, since it may happen that the cut-off point of the GA [maximum number of generations being reached] happens right after a new random population has been restarted, thus leading to a population containing one good solution [the elite one] while the others have just been randomly generated. In this case, one can run the program again for a few generations until convergence is reached again.

Another strategy that was implemented in this study was to add a simplified kind of 'memory' to the GA. In a standard GA, all the individuals of any population are always evaluated. The feature added makes the GA inspect a database of previously evaluated individuals each time it is presented with a new solution. If the current individual already existed in a previous generation, the GA simply retrieves from this database the fitness function value associated with it. If it is an entirely new solution, it is evaluated and the result stored in the database for future use. Since there is always some degree of overlapping between subsequent generations [at least one individual, the elite one, is always repeated, but more can exist], this simple function adds more efficiency to the algorithm, making it faster. In this study, since each building design evaluation takes a reasonable amount of time [depending on the complexity of the building, it can take from about 20 seconds to more than a minute], this feature provides some significant time savings.

As a final note, it should be added that a genetic algorithm is a heuristic procedure that offers no guarantee of finding the global optimum. This shortfall of the method was the reason behind work presented in the next chapter [chapter 4], where the GS is tested within a domain where the solution space is known, to assess the performance of the system in finding optimal solutions and the degree of confidence one could have in the results, before proceeding to solution spaces where the global optimum was not known in advance.

3.4 Previous studies applying building energy simulation softwares to optimization

Radford and Gero published in 1978 two papers addressing the problem of the optimal lighting design using dynamic programming. One of the papers focus on the design of windows (Radford and Gero, 1978), and begins to present a method to address the problem using dynamic programming, but stays at an embryonic level. The other paper (Gero and Radford, 1978) presents a detailed method for applying dynamic programming to an artificial lighting problem [although no implementation of the algorithm is mentioned], and suggests that a similar procedure can be applied to size and position roof lights, windows and shading devices. An internal report by Radford (1978) will further develop the methods proposed by Gero and Radford (1978) to apply them to window design. As mentioned in chapter 2, the dynamic programming approach presented in all these studies has a serious shortfall, since the problems considered do not meet the requirements of separability of the optimization process in different independent stages. The introduction of delayed decisions to address this problem causes a high computational burden due to the need to carry information about all possible alternatives until the stage is reached when a decision can finally been taken. This makes the method unpractical for problems of even reduced size. In a moderately complex building, where the interdependency of variables can be high, the applicability of this procedure is very limited.

At Lawrence Berkeley National Laboratory, some attempts where made in the past to use DOE-2.1E to optimize design parameters of windows. One of these studies relies on the use of indices and weighting factors (Sullivan et al. 1988) to develop a fenestration performance design tool. Five indices were defined, which when combined with user-specified weighting factors yield a single figure of merit that measures the window performance. Three of the indices are related to the effects of fenestration on building energy performance [fuel and electric use, and peak electricity demand]. The other two are related to thermal and visual comfort. The thermal comfort calculation is very simplified, based on a correlation between the magnitude of solar radiation coming through a window with the percentage of dissatisfied people. The visual comfort calculation uses the Cornell method, which is integrated in the daylighting module of DOE2, to calculate a Glare Index. This approach presents two main problems: first, the Cornell method predictions have been proved to disagree with people's assessment of built situations. Second, the implementation of Cornell method in DOE2 does not provide reliable results in terms of the method itself.

Sullivan et al. (1988) derived index values and correlations to window design parameters by creating a database consisting of a large number of building energy simulations for a prototypical office building, using DOE2. Four glazing types and two shading devices were combined in several ways so that a representative sampling of realistic fenestration systems was analyzed.

The problems with this study are many. The combination of the five indices into a single figure of merit using weighting factors is dubious as a method for multicriteria analysis [this issue will be further discussed in chapter 8, under the topic of 'Plain aggregating approaches']. Calculations of comfort indices seem to be oversimplified and unreliable. And the resulting data only apply to solutions close to those simulated, what makes them of limited use in an architectural design domain.

Sullivan et al. presented a second study in 1992, called 'A method for optimizing solar control and daylighting performance in commercial office buildings.' Here, the two comfort indices were abandoned, and only annual electricity use for cooling and lighting, and peak electricity demand are considered. Heating is not included, and other energy sources apart from electricity are excluded. Using regression analysis procedures, fenestration performance is characterized as a function of solar aperture [defined as the product of shading coefficient and window-to-wall ratio], and effective daylighting aperture [defined as the product of visible transmittance and window-to-wall ratio]. Optimum performance consists of defining the solar and effective daylighting aperture values that minimize annual electricity consumption and peak demand.

Once again, results from this study are only applicable for buildings similar to that used for the simulation [an office building, with a large core area and individual cells along the perimeter], and to the type of climate used in the study. It also does not account for interactions between different daylight sources in the same space. Glazing types are limited to those included in the study.

Both studies presented by Sullivan et al. are thus confined in their applicability, and require a large amount of previous work to create the database of simulations from which results can be derived. On the contrary, the Generative System presented in this dissertation aims at being a very general method applicable to any kind of building [with limitations imposed just by the complexity of the geometry, that is, if it can be modeled using BDL], with any type of materials and in any climate. The time burden on the user is only that of creating the initial input file, and of doing small modifications in the GA code. All the time consuming efforts of doing parametric runs and regression analysis are excluded from this GS.

Other simplified procedures like the LT method (Baker and Steemers, 2000) that use energy graphs to explore the impact of a few key building-design parameters, are unable to handle the level of detail and complexity enabled by the system presented here. Some of the previous studies described, such as that of Sullivan et al. (1992), also use energy graphs, although those are derived from the specific parametric experiments done for that study, and are not as general as those created for the LT method.

It is worth mentioning the work by Wetter (2000), which describes the use of the GenOpt software for optimization purposes, using as a search method the Simplex method of Nelder and Mead with an extension by O'Neill. The main drawback of this search method is that it works well only in small problem sizes, up to about 10 variables. GenOpt is also a flexible platform for implementing user-defined algorithms, according to the author, and can apparently be linked to any simulation program that reads and writes text files. In this sense, GenOpt seems to be the currently available software that is closest to the goals and methods of this Generative System.

3.5 Building the Generative System

The Generative System was first developed on Unix platform, running remotely from a computer based at Lawrence Berkeley National Laboratory. This implementation was rather slow, and later a PC-based system was developed, which was about 20 times faster. As the work progressed from simple test buildings and single objective functions towards complex architectural designs and multi-criteria optimization, computational power and speed became a main issue, and efforts had to be directed towards developing a faster implementation. The 'memory' function described in the previous section was developed when the algorithm was still running on Unix platform, to try to increase its execution speed, and was later adapted to the PC version too.

The basic method used was to create a link between the GA and DOE2.1-E, so that the GA would make a call to DOE2 each time it was necessary to calculate the fitness of an individual. The objective function of a solution is thus the result, in terms of annual energy consumption, of running a DOE2 thermal and lighting simulation of the building under study, with the absent building parameters determined by the individual chromosome of that solution.

Making the two programs communicate was not simple due to a number of data formatting issues, but eventually a method was developed that allows the exchange of information without errors or data corruption. Different procedures had to be used for numerical variables, like dimensions or coordinates, and for non-numerical ones, like materials.

The process of encoding non-numerical values like materials names had to be carefully designed so that the binary coding would relate in some way to the properties of the materials, to allow meaningful building blocks to emerge during the search. For example, if the GS could choose materials from a library of 16 available choices, the number of alleles necessary to code each material was 4 [$2^4 = 16$]. So, the coding for a given material could look something like 0101. Materials were ordered in the respective library so that the presence of a 0 or a 1 in a certain position would carry with it some information about the material. For example, a 0 at the beginning of the code would always mean an insulation material. Then a 1 in the second position would mean a type of insulation material with lower conductivity, while a 0 would stand for a material with higher conductivity. This way, a material starting with 01 would always be a good insulation material, and in cases where insulation was an issue, the '01' building block in that particular position would be favored by the GS. The remaining alleles could code the thickness of the material, for example.

The other main problem was how, and at what level, to incorporate architectural language constraints into the algorithm. The Building Description Language [BDL] input format used by DOE2 provided a useful platform for describing building geometry, but presented no formal way to incorporate information about relations between the variables under study, which was one of the mechanisms intended for the architects to describe and control design intentions. Given this, it is inside the GA code that the constraints encoding architectural design intentions are implemented, in the current version of the software.

3.5 Input and output formats

At the moment, there are no user interfaces specifically designed for the Generative System. The input process is all text based, and consists of two main steps. The first creates a DOE2.1E input file that completely describes the building [geometry, orientation, space layout, construction materials, etc,] and also provides information on the electrical and mechanical systems to be used and on building schedules. Absent from this file are the values for the specific variables under study, and that will be created by the genetic algorithm. Each of these variables is represented by a code that will later be replaced by a numerical value, a construction material reference, etc.

The second step involves the adaptation of the genetic algorithm source code to the particular problem under study, as well as the input files it reads during execution. As for the source code, the main alterations required relate to the 'func' routine, which does the objective function evaluation by calling DOE2. It is necessary to ensure that 'func' will pass to DOE2 the right number of arguments and in the correct order. As mentioned above, it is also at this level that some of the language constraints are implemented, by specifying desired relations between the different variables. Other minor alterations relate to some format statements at other points of the code.

Finally, the input files the GA reads need to be modified. One of these files allows the user to specify the number of variables under study, the upper and lower bounds for each variable, the number of intermediate steps between those limits and the number of binary bits required to code each variable. It is also in this file that the user defines the population size for the GA and the number of generations it will run for. The use of other strategies, such as elitism, Micro-GAs and niching, is also specified there. Probability values for both crossover and mutation are required too.

As for output formats, the first version of the algorithm had an AutoLisp routine that would draw three-dimensionally in AutoCad© the results from the GS. AutoLisp procedures allowed the results to be visually inspected, both two- and three-dimensionally. The architect could this way visualize the several steps of evolution of the process, as well as the final results. The inconvenience of this method was obvious. For each building under study the user would have to write the AutoLisp code that would represent the building. Only the values of the variables under study would then be automatically read from the GS result file.

Since this process was quite labor intensive, a commercially available program [DrawBDL©] is now used. This program generates a 3D model of the building from a DOE2 input file, which can be rotated and viewed from different directions. It can thus be used to view the complete DOE2 input files after they have been modified by the GA, and thus visualize the results from the GS. This model can be imported into AutoCad, but has the major drawback of losing three-dimensional information during that process, meaning that the AutoCad drawings obtained this way are only two-dimensional. If a 3D AutoCad model is required, it needs to be constructed manually, with the help of the 2D drawings imported from DrawBDL.

If a 3D AutoCad model is constructed, it can then be imported into programs like 3DStudio Max© for improved visualization of results, or to lighting analysis programs like Lightscape© for a better understanding of the GS solutions in terms of its lighting performance. In this case, each space can be visualized individually for different times of the day and the year, to gain a more detailed insight of the lighting patterns inside the spaces.

Much work needs to be done in terms of input and output formats if this GS is to be used by architects and practitioners. In terms of input, a graphical interface needs to be created to help the user insert necessary information, which would automatically upgrade the input files and the GA code. A more ambitious project is to create a CAD interface to the GS that would allow to extract from a 3D model of the building all the building-related information to run the program. A crucial factor is to be able to use standard graphics formats, as the ones generally used by architects in their practices. The GS should also be able to generate a new, modified 3D model of the building, based on the results achieved by the software, to allow for quick visualization of results and further development and changes from the architect.

Improving both the input and output formats of the GS would mean more than simply making it more user-friendly and thus more prone to be used in practice. By choosing the appropriate computer graphics paradigm for this interface, there may exist a potential for making this GS able to handle other class of problems not explored until today. More considerations on this issue will be provided in the 'conclusions and further work' chapter, as this represents one of the main areas of development that we envision as relevant future work regarding this Generative Design System.

CHAPTER 4 Testing the Generative System

4.1 Introduction

This chapter describes the first testing of the Generative System using computational experiments. The objective of the tests was to gain confidence in the system's performance before applying it to more complex problems, so that no 'leap of faith' would have to take place. An experimental setup was defined, with a very simplified building scheme consisting of a core area and a number of surrounding rooms facing each of the four main cardinal orientations [figure 4.1]. Each room had a single window so that no interactions between openings facing different orientations could happen. This experimental building layout has been used in several studies done at Lawrence Berkeley Lawrence Laboratory [LBNL], where the research presented in this dissertation first started.

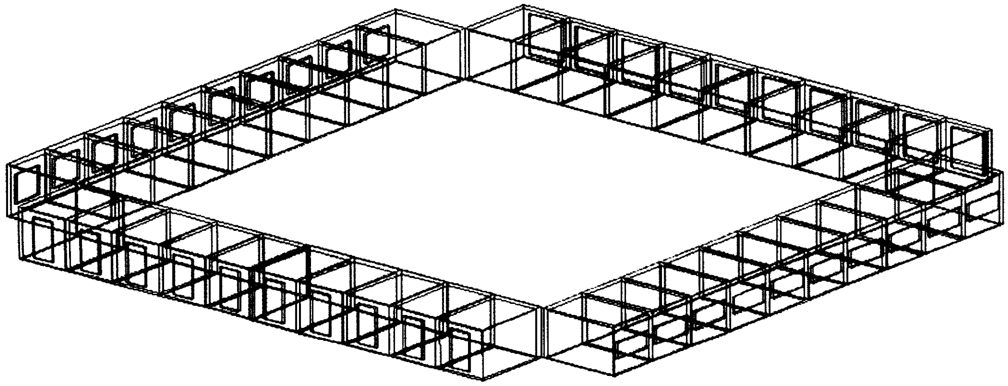


Figure 4.1 The schematic building setup used for the experiments

This configuration allowed the objective function to be separable, that is, the best solution for the overall building was a combination of the solutions for each individual orientation. This way, a very large 8-dimensional hyperspace, of more than 16 million solutions, could be generated while knowing its global optimum. This provided an excellent test bed for the effectiveness of the algorithm in locating high performance or near-optimal solutions in large solution spaces. The methods and results for those tests are described below, together with some extensions and additional experiments at the end of the chapter.

A note must be added about the impact of this particular building layout on the results observed. Although the test building allowed a very important factor to be present in the experiments, the above-mentioned separability, it also had a somewhat negative factor attached. Because the building has a very large core area (31 X 31 m), with no contact with the external envelope, the impact on overall energy consumption of changing some envelope characteristics, like fenestration dimensions and proportions, was relatively low. This happens because the energy consumption of the building is highly dominated by the core area, which has to be lit, heated and cooled always by artificial means and does not reflect the beneficial changes performed in the envelope. It will be noticed in the results of the several experiments that energy levels usually vary only by small amounts. This note alerts the reader to the reasons behind it. In later chapters, different building designs will reflect a much more significant impact of envelope modifications. From a positive side, the building used in this chapter's experiments was beneficial in that it represented a quite demanding test environment, since the Generative System would have to find points that represented solutions which performed only slightly better than alternative ones, and it succeeded quite well in this task.

4.2 Simulation method

The first step was to find a representation method within a genetic algorithm framework that would allow the encoding and manipulation of the variables we wanted to study. To make use of a Genetic Algorithm it is necessary to provide a representation of feasible solutions in terms of a binary string of bits, an objective function definition, and mechanisms for selection, crossover and mutation. It is also necessary to choose a population size and decide on the maximum number of generations the algorithm is supposed to run.

a] Representation of feasible solutions

The experimental building has four different types of windows facing different orientations, each being characterized by two dimensions [width and height]. Each window dimension could vary from 0.3 m to 2.4 m in discrete steps of 0.3 m, meaning it could only take 8 possible values. To encode 8 values in a binary string, only 3 bits are necessary, since $2^3=8$. The window dimensions can thus be encoded the following way:

Binary code	**Decoded numerical value [m]**
000	0.3
001	0.6
010	0.9
011	1.2
100	1.5
101	1.8
110	2.1
111	2.4

To encode a solution consisting of 8 window dimensions [width and height for north, south, east and west orientations] it is necessary to use a binary string of 24 bits [3 bits per variable x 8 variables], what becomes the chromosome length. A valid chromosome for this problem could look something like:

0 0 1 1 0 1 0 0 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0 1

b] Objective function

The objective function value is the result of running a thermal and lighting simulation with the window parameters chosen, and expresses total annual energy consumption in the building in MWh [megaWatthour], taken from the BEPS report [Building Energy Performance Summary] of DOE2.1E.

c] Selection mechanism

Proportional or biased roulette wheel selection was used.

d] Crossover mechanism

Uniform crossover was used. The probability of crossover in MicroGAs is usually set to 1, to ensure a high recombination rate.

e] Mutation mechanism

Mutation is used in standard GAs to introduce new points into the search space. Because MicroGAs already have a high rate of generation of new solutions, mutation becomes unnecessary and is set to 0.

f] Population size and number of maximum generations

Since a MicroGA was used, population size was kept small [n=5]. The number of generations was initially set to 100, what proved to be a good choice, since extending that number to 200 and 300 generations caused no improvement in the final solutions, while implying a large time penalty [almost doubling and tripling the simulation time, respectively].

4.3 Problem description, simulation method and test building

The particular problem used in this chapter is the sizing of windows in a building to optimize its lighting, heating and cooling performance. The exact sizing of windows occurs typically at later stages of design, but will have a significant effect on the environmental behavior of a building. The optimal sizing [in energy terms] will depend on the climate the building is located in, the glazing used, the orientation the window is facing, the type of use of the building [offices, housing, etc.], and other factors like shading elements.

Windows display a complex environmental behavior due to different requirements often being in conflict with each other, making the evaluation of their overall performance a complex one. During the heating season, as window sizes increase there is an increase in daylighting availability and thus a decrease in the need for artificial lighting, leading to less electrical energy consumption. For some orientations, there are also more useful solar gains through the window, which tends to reduce heating expenditure. On the other hand, there is an increase in heat loss through the glass, which in turn requires more spending for heating. Less use of artificial lighting will also lead to lower internal heat gains in the building, leading to an increased need of heating.

During the cooling season, as window sizes increase there is a decrease in the need for artificial lighting, which leads to less electrical energy consumption and also in cooling loads generated by the lights themselves. On the other hand, there are increased solar gains through the glass, leading to more spending for cooling.

These interactions are simulated in the following way: daylighting levels inside each of the spaces in the building are dynamically calculated at two reference points determined by the architect, for each hour of the year. The lighting levels achieved by using natural light only are then compared to the required levels [in lux] also set by the architect. The artificial lighting system, which is continuously dimmable, is set to provide just enough light to make up for the difference between available daylighting and required light levels. The electrical energy spent by the lights is quantified, and electrical lighting is also accounted as a heat load to the space.

The system then goes on to calculating internal temperatures inside the space by taking into account external climatic conditions, the characteristics of the building fabric [wall, roof, floor and window materials], solar gains through the windows, internal gains from occupants, equipment and artificial lighting, etc. Those temperatures are compared to set-points defined by the architect, and heating or cooling are provided if required to compensate for the difference between calculated internal temperatures and predefined set-points. The energy spent in heating and cooling is determined [including energy for running ventilation fans, pumps, etc.], and added to lighting energy to determine total annual energy consumption in the building.

The building simulated is assumed to be an office building, where internal conditions are usually controlled, and systems like dimmable artificial lighting are more likely to exist. The hypothetical office building consists of a square core zone, 31 meters on a side, surrounded by four identical perimeter zones, each of 31 x 4.6 m [see figure 4.1]. The module faces the four cardinal directions. Each perimeter zone is divided into ten office spaces of equal size (3.1 x 4.6 m). One reference point for daylighting calculation is located 1m from the window wall, and the other one is located 1 m away from the wall opposite to the window wall. The points are centered in the space in the other direction, and at desk height. The required horizontal illuminance is 540 lux. The window glazing was set as double pane with a layer of spectrally selective glass, with a shading coefficient of 0.34 and visual transmittance of 0.41.

The problem was studied for two locations: Phoenix, Arizona, a cooling-dominated situation; and Chicago, Illinois, a heating-dominated climate. The aim was to provide some insight on how the optimal sizing of windows varies with different climatic conditions.

4.4 Determining solution spaces for Phoenix climate

To assess if the Generative System would actually be able to locate optimal solutions for this problem, the algorithm was first tested against a hand-worked example of limited size, for which the optimal solution was found by manually sampling all possible solutions. The location chosen was Phoenix, Arizona. The solution space for each of the four cardinal directions was first determined. Eight window widths and heights were considered, from 0.3 to 2.4m at discrete steps of 0.3m, creating a solution space of 64 points for each orientation. While the dimensions for an orientation were varied, the windows facing the other orientations were kept 1.2m wide and 0.9m high. For each orientation a 100-generation GA was run.

The solution spaces for each of the four orientations are shown in figures 4.1 - 4.4. In general, the solution spaces proved to have several local minima and a global minimum for all orientations but north. There are no local minima for the north orientation. The presence of local minima makes the problem inappropriate for derivative-based search methods and makes GA a reasonable approach.

Figure 4.2 shows the solution space for south-facing windows for the Phoenix climate. It can be seen there is a relatively flat surface of configurations corresponding to low energy consumption. Within that flat surface there are however several local minima and a global minimum. Being stuck in a local minimum would not be too serious in this case since the objective function difference in relation to the global minimum is small. However, we are interested in the algorithm behavior and not only in the particular results for this example.

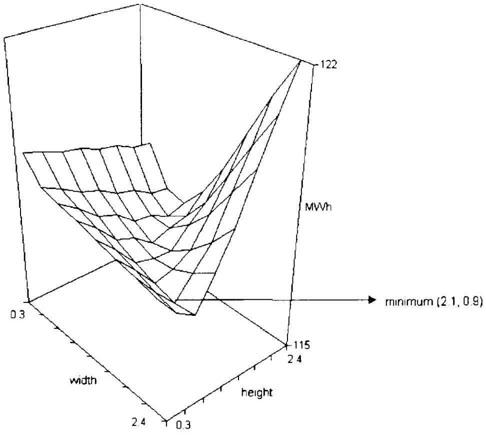


Figure 4.2 Annual energy consumption [MWh] for south window dimensions

The global minimum of 116.94 MWh is located at point (2.1, 0.9), corresponding to a window 2.1m wide and 0.9m high. There were two local minima of 117 MWh for points (1.2, 1.5) and (0.9, 2.1). The GA located the global minimum in generation 18, and the (0.9, 2.1) local minimum in generation 21. It can be seen that the global minimum and one of the local minima correspond to windows with the same area (1.9 m^2), but with different aspect ratios, what shows the effect of accounting for daylighting penetration in the space. The more horizontal window corresponds to a slightly lower energy consumption. The optimal area is 5% larger than the area of the second local minimum.

For the east orientation the global minimum was similar to the south orientation (2.1, 0.9) [see figure 4.3], and the local minima were located in points (1.8, 0.9) and (1.2, 1.5). The GA located point (1.8, 0.9) by generation 16, but did not manage to locate point (2.1, 0.9) in 100 generations. Point (1.8, 0.9) stands next to the global minimum, and has an objective function value only 0.03 MWh higher. The observed behavior is a consequence of the fact that genetic algorithms have a difficulty in succeeding in very local searches. GAs are heuristic procedures that are often able to locate the global minimum for the problem, but give no guarantee of doing so. The search can get very close to the global minimum without actually getting to it. It should be noted that the actual difference in objective function values is negligible in this case.

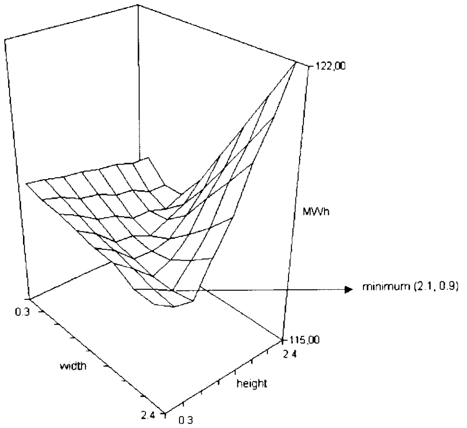


Figure 4.3 Annual energy consumption [MWh] for east window dimensions

For the west orientation [figure 4.4], the solution landscape proved to be flatter than for the south and east orientations. Although solar angles are symmetrical for the east and west orientations, temperatures and occupancies are not. The global minimum was at point (1.5, 0.9) and local minima were found at points (0.6, 2.1), (0.9, 1.5) and (2.4, 0.6). The GA located the global minimum by generation 10.

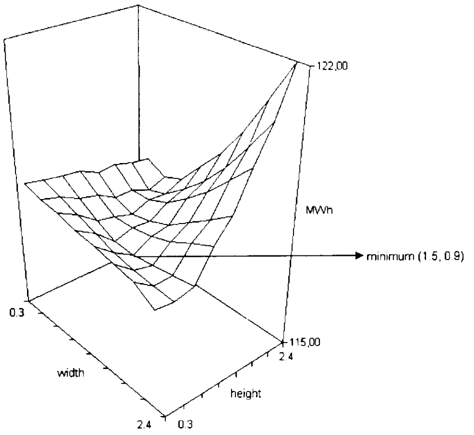


Figure 4.4 Annual energy consumption [MWh] for west window dimensions

For the north orientation [figure 4.5], there was only a global minimum at point (2.1, 2.1), with no other local minima. The GA located that global minimum very quickly, at generation 7. This good performance of the GA is thought to be due to the smooth configuration of the solution space in this case. Apart from the minimum, the GA located a series of high-performance solutions in its vicinity, all by the end of generation 8.

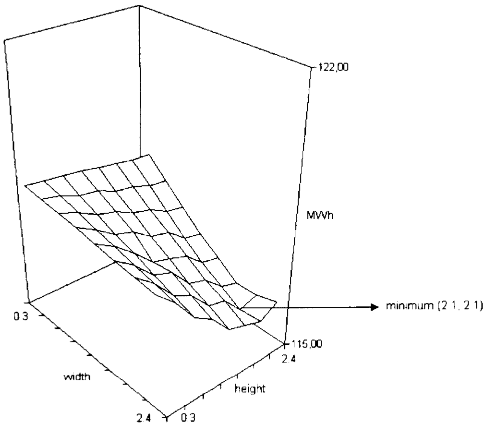


Figure 4.5 Annual energy consumption [MWh] for north window dimensions

4.5 Assessing GS performance in locating the global minimum in a larger solution space

The resort to a hand-worked example proved the GS succeeded in most cases in finding a global minimum for the problem. In the cases it did not reach the actual minimum it located points very closed to it, what provides enough confidence in the results to extend the method to larger problems for which it is not possible to calculate manually the solution space, given their dimension.

The GS was thus used to search a hyperspace of 8 dimensions for the optimal size of windows to all orientations simultaneously, resulting in a solution space of more than 16 million points [8^8]. It was only possible to know the global minimum for the eight-dimensional space by combining the individual minima for each orientation due to the above-mentioned separability, with no interactions between the different orientations either in terms of spaces, windows or building systems. However, buildings are not typically designed this way and thus it is usually not possible to optimize for each orientation and then combine the individual results. From the hand-worked example it is known that for the Phoenix climate the best result corresponds to the point (2.1, 2.1, 2.1, 0.9, 2.1, 0.9, 1.5, 0.9) [north window dimensions – width, height - then south, east and west], corresponding to an annual electrical energy consumption of 114.53 MWh. The worst point in the hyperspace is (0.3, 0.3, 2.4, 2.4, 2.4, 2.4, 2.4, 2.4), with an annual electrical energy consumption of 131.4 MWh, about 15% higher than the best possible result.

Figures 4.6 and 4.7 show the evolution for one experiment from the initial solutions to the final generation of improved performance designs. It can be seen that initially there is a high variance in the design patterns, which progressively converge into the optimized solutions.

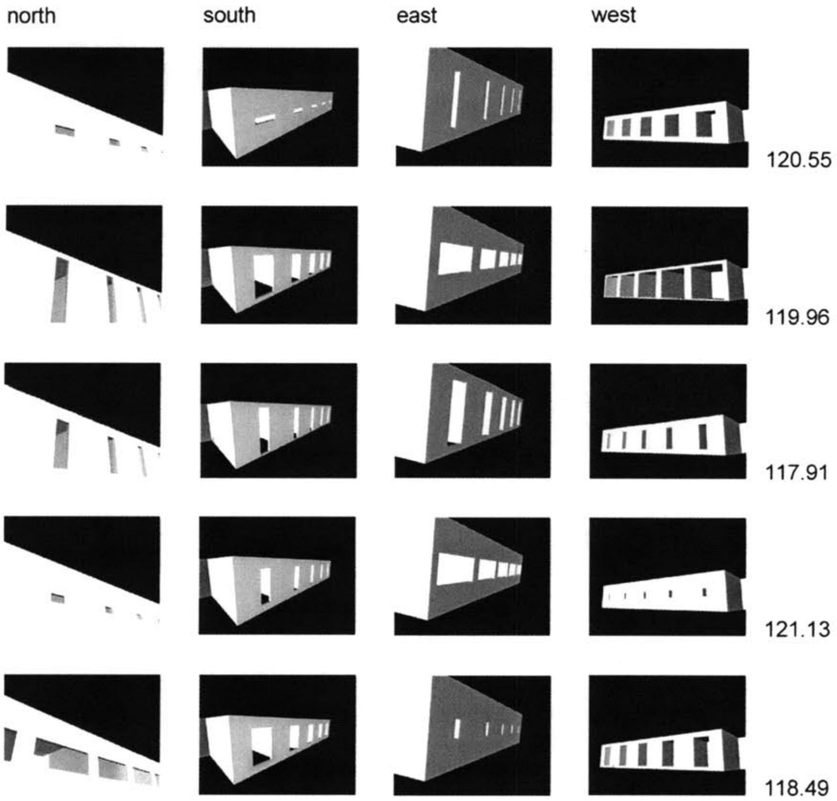


Figure 4.6 Initial solutions for Phoenix climate, experiment 3 [see table 4.1 for experiments numbering]. Each row represents a solution. The numbers on the right show its annual energy consumption in MWh.

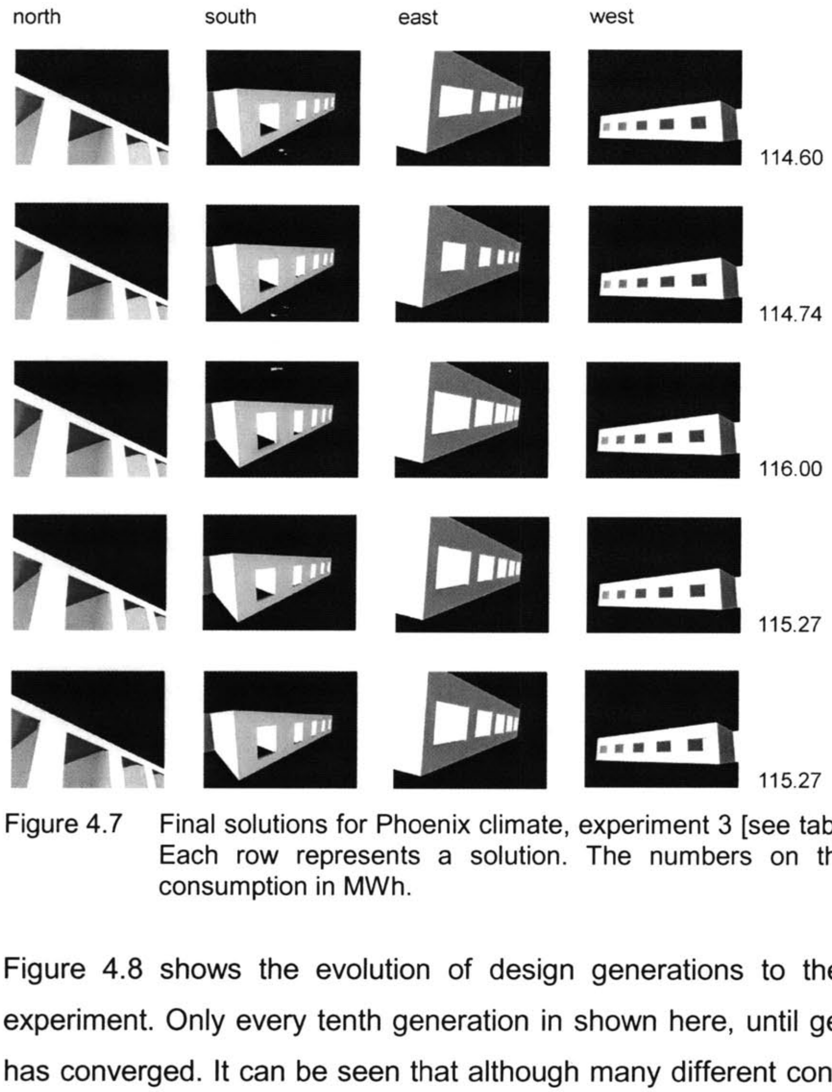


Figure 4.7 Final solutions for Phoenix climate, experiment 3 [see table 4.1 for experiments numbering]. Each row represents a solution. The numbers on the right show its annual energy consumption in MWh.

Figure 4.8 shows the evolution of design generations to the Phoenix climate for another experiment. Only every tenth generation in shown here, until generation 80, when the solution has converged. It can be seen that although many different configurations are tested along the search, the design steadily tends to the final optimized solution: large windows for north orientation (2.1 x 2.1 m), where available daylighting can be maximized without incurring on high solar heat gains in the summer. Due to the mild winter temperatures in Phoenix and to the use of double pane windows, heat losses through this large glazing area are not enough to penalize this solution when overall energy consumption is calculated.

Average size windows (1.2 x 1.5 m) are chosen for east and south orientations, showing a balance between admitting useful daylighting and winter solar gains and preventing too much solar gain in the summer. Smaller windows (1.2 x 0.9 m) are chosen for west since that is the worst orientation in terms of summer overheating: the building is already warm at the end of the afternoon and excessive west solar gains will lead to internal overheating. It must be added that these solutions are dependent on the type of glazing used: a different glazing option would lead to different optimum window sizes, as will be seen later in the chapter.

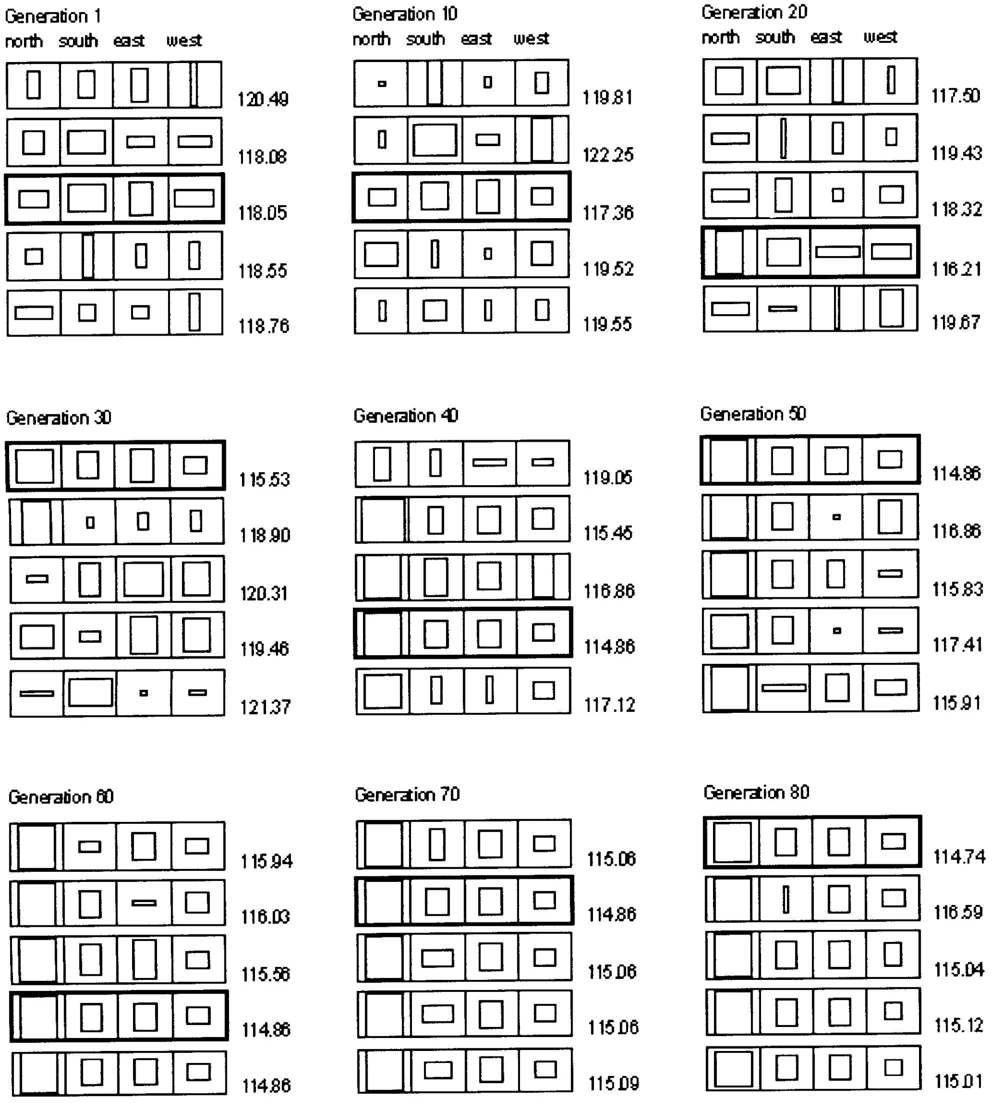


Figure 4.8 Generations 1-80 for Phoenix climate, experiment 1

In total, five experimental runs were made for the Phoenix climate, to compare results from different trials. A maximum number of 100 generations was used for each experiment. The results are summarized in table 4.1. In general, the GA performed very well, being able to find high performance solutions close to the global minimum [the highest error margin in table 4.1 is 0.03%]. The results are even more satisfactory considering that 100 generations is a relatively low number for GAs, and a population size of 5 is very small. The maximum of 500 different solutions evaluated by the GA corresponds to only 0.00003% of the solution space dimension.

Experiment	Best solution found	Objective function value (MWh)	Percentage above minimum (%)
1	2.1 / 2.1 / 1.2 / 1.5 / 1.2 / 1.5 / 1.2 / 0.9	114.74	0.018
2	2.4 / 2.1 / 2.1 / 0.9 / 2.1 / 0.9 / 0.9 / 1.5	114.65	0.01
3	2.1 / 2.1 / 1.2 / 1.5 / 2.1 / 0.9 / 1.5 / 0.9	114.60	0.006
4	1.8 / 2.1 / 1.2 / 1.8 / 0.9 / 1.5 / 1.2 / 0.9	114.89	0.03
5	2.4 / 2.1 / 2.1 / 0.9 / 1.2 / 1.5 / 1.5 / 0.9	114.65	0.01

Table 4.1 Solutions for the Phoenix climate

Results from these experiments are displayed graphically in figure 4.9. It can be seen that although total energy consumption is very similar, design solutions are somewhat different. The fact that different runs of the Generative System may lead to different solutions with similar performance may be of interest to architects, who may try more than one run of the GS to get a sense of available alternatives, and to try to understand which main features of a design contribute to a successful behavior.

4.6 Testing for Chicago climate

Five similar experiments were also done for the Chicago climate, a heating dominated situation, to assess to what extent optimum window designs may be sensitive to climatic conditions. In this case the global minimum was not known in advance and so it was not possible to determine the error margins. Results are summarized in table 4.2.

Experiment	Best solution found	Objective function value (MWh)
1	0.3 / 0.3 / 1.8 / 1.5 / 1.2 / 1.5 / 2.4 / 0.3	133.03
2	0.3 / 0.3 / 1.5 / 1.5 / 0.9 / 1.5 / 1.5 / 0.9	132.97
3	0.6 / 0.3 / 1.5 / 1.5 / 1.2 / 0.9 / 0.3 / 0.9	133.15
4	0.3 / 0.3 / 1.8 / 1.5 / 1.8 / 0.9 / 0.9 / 0.6	133.00
5	0.3 / 0.3 / 1.5 / 2.4 / 1.8 / 1.2 / 0.6 / 0.3	133.03

Table 4.2 Solutions for the Chicago climate

Figure 4.9 also depicts the solutions obtained for Chicago climate, which vary in significant ways from Phoenix solutions. Analyzing the results of the first experiment shows that due to the extreme winter conditions, north window dimensions were reduced to the minimum allowed by the constraints (0.3 x 0.3 m), since north openings are, in this cold climate, a large source of heat losses and provide little solar gains in winter. South window size increased in relation to the Phoenix solutions, to allow for more solar gains in winter (1.8 x 1.5 m), and because overheating is no longer a problem. East window size remained basically similar (1.2 x 1.5 m), and west was reduced to a horizontal strip (2.4 x 0.3 m) that allows for some daylighting but prevents excessive heat loss.

The significant differences between solutions found for each climate, and also between orientations within the same climate, suggests the use of a Generative System like the one proposed here may not only provide increased energy savings but also introduce a positive degree of diversity in the design. It also suggests that unexpected solutions that might not be obvious to the architect may be pointed out by the GS, such as the very large north-facing windows for Phoenix, or the horizontal strip windows facing west for Chicago.

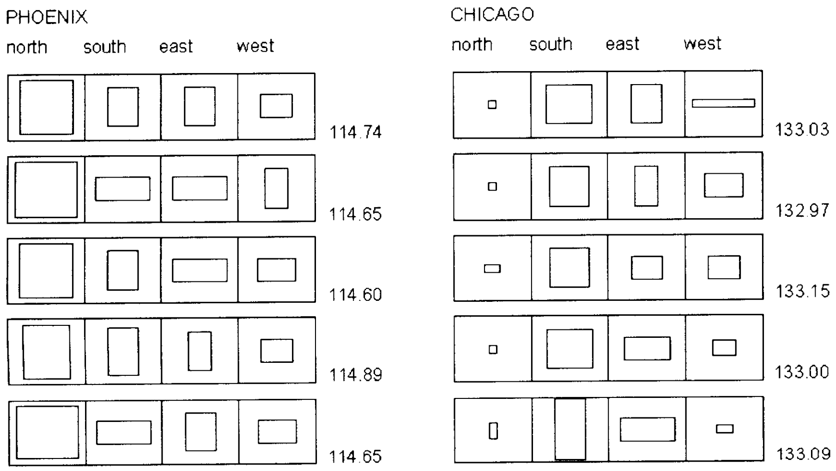


Figure 4.9 Comparative results for Phoenix and Chicago locations

Solutions optimized for environmental performance may not always represent a good behavior in terms of other criteria. Actually, that is one of the main problems of using optimization methods in architecture. For example, the north-facing windows for the Chicago climate may be too small to provide occupants with an adequate view out. In those cases, the architect may modify the results provided by the GS while making use of knowledge gained during the optimization process, such as knowing that north-facing windows should be kept as small as possible. A simple computer simulation can then be performed to determine to what extent the modified design's energy consumption differs from the optimized solution.

4.7 Testing different types of glazing

It was mentioned before that the characteristics of the glazing used would also influence the design of the windows. If a highly insulating glazing system is used in a cold climate, larger glazing areas are possible by allowing a better balance between daylighting and heat loss. To test this hypothesis an experiment was done for the Phoenix climate, where instead of the double selective glass, a single clear glass was used. Table 4.3 summarizes the characteristics of the two glazing systems:

Glazing type	Layers [mm]	Shading coefficient	Visual transmittance	U-value
Double selective	3 / 12 / 3	0.34	0.41	2.73
Single clear	3	1	0.898	6.31

Table 4.3 Characteristics of the two glazing types used

The results from the experiments are shown in figure 4.10. The most immediate perception is that window sizes decreased in general, for all orientations. That confirms the hypothesis that if a lower quality glazing solution is used, windows will represent more environmental problems to the interior of the building, and will thus be reduced. Energy values corresponding to all solutions will be shown in figure 4.11.

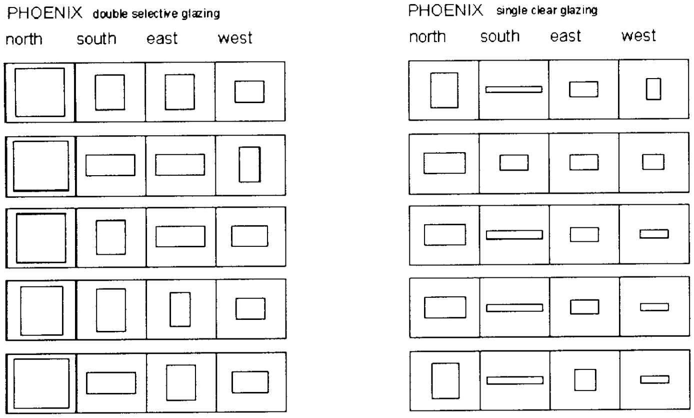

PHOENIX double selective glazing
north
south
east
west
PHOENIX single clear glazing
north
south
east
west

Figure 4.10 Results from experiments using different glazing types

North window sizes have been significantly reduced. Several factors may contribute to that. First, because the single glass has a much higher U-value, the window becomes a more important source of heat loss. Heat loss is not too prevalent in a climate like Phoenix, but is still a factor to consider during winter. Second, the single glass has a much higher shading coefficient, meaning it allows more solar gains into the space. Again, solar gains are not very significant in the north façade, but between the March and September equinoxes, north façades still receive some amount of direct solar incidence in early morning and late afternoon. Apart from that, there are the reflected and diffuse components of solar radiation that may reach the glazing. Finally, because the visual transmittance of the single glazing more than doubles that of the selective one, it is not necessary to have such large windows to admit into the space the same amount of daylighting. All these factors probably come together to cause the decrease in north window dimensions that can be seen in the figure.

For the other orientations, reducing solar heat gain was probably the main cause for considerably decreasing window sizes. Again, penalties in terms of daylighting admission are not so high because the visual transmittance of single clear glass is very high. West orientation seems to be the more problematic in terms of heat gains, showing the smallest fenestrations.

This simple example aimed at demonstrating that a given façade configuration suggested by the GS may be significantly influenced by the materials chosen. Here only glass type was used as a variable, but construction materials for walls and roofs may play an important role too. Given that, the architect may chose to run the system several times, changing the type of materials used, or may also turn the materials into variables themselves, used by the GS in search for a good solution. An extension of this latter possibility will be presented in chapter 8, where choice of construction materials is presented as a problem to the generative system.

4.8 Using shading as a variable

Even though the GS manipulated window sizes to overcome some of the environmental problems created by using a lower quality glazing, the energy consumption of the experiments using single glass was always higher than those using the double selective glazing. The next question posed was: if the GS was allowed to use fixed shading systems [overhangs and vertical fins] as complementary control strategies, could it find solutions that would be as good as those using the better glass, in terms of energy use?

The new experimental setup implied that, apart from varying fenestration dimensions, an overhang would exist in the south façade, with width equal to the width of the window, and variable depth determined by the GS. In the east window, along with the overhang, a vertical fin also exists, with height equal to the height of the window and depth equal to the overhang, and located on the left side of the window, since that is the direction from where most of the direct sun comes. The west window follows similar rules, with the exception that the fin is located on the right side. No overhangs or fins are predicted in the north façade, since direct sun is not too important there.

The results are depicted in figure 4.11. The energy levels shown in the figure very closely approximate those where double selective glazing is used, suggesting that by properly sizing fenestrations and adding shading elements, it is possible to achieve similar levels of performance even if using lower quality glazing.

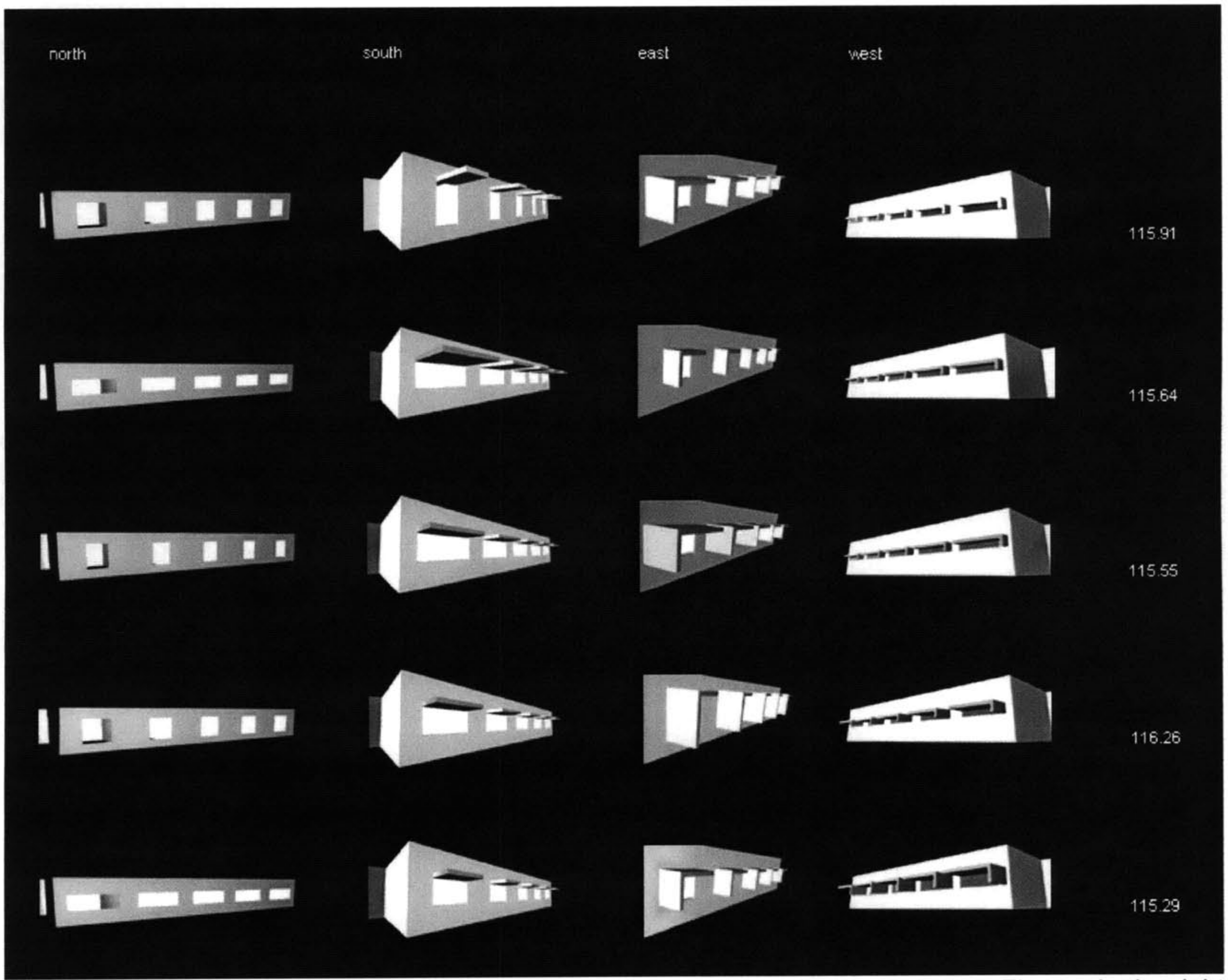


Figure 4.11 Using variable-depth fixed shading systems with single clear glazing. Figures in the right show annual energy consumption in MWh.

In the best solution [last row], south shading has moderate depth, east shading is the deepest, with moderate window sizes associated with it, and west windows are also larger and have deeper shading devices than any of the previous four experiments. In most results, west windows are small horizontal slots with shallow shading, but in the best solution, shading becomes deeper and window sizes increase.

Figure 4.11 shows the elevations and energy results for the three situations studied: double selective glazing, single clear glazing, and single clear glazing with shading, all for Phoenix.

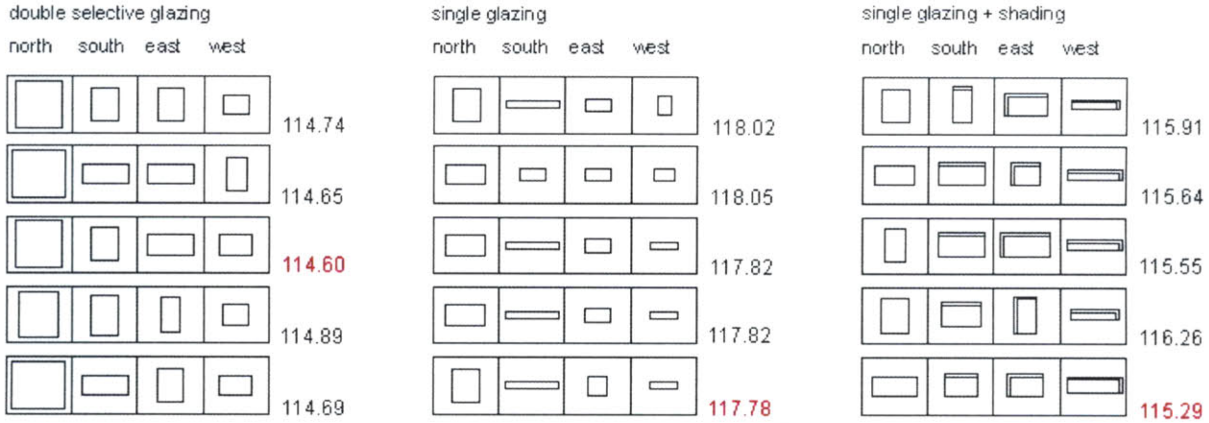


Figure 4.11 Results for the three cases, Phoenix. Figures in the right show energy consumption in MWh.

A manual experiment was performed, removing from all solutions above the east and west vertical fins. The result was that the energy performance of the building slightly decreased [see table 4.4]. This suggests that the vertical fins are more detrimental than helpful, probably because they block daylighting while providing little protection against the direct sun. Horizontal overhangs, on the other hand, proved to be good additions to the designs.

experiment	with vertical fins	without vertical fins
1	115.91	115.82
2	115.64	115.52
3	115.55	115.44
4	116.26	116.11
5	115.29	115.08
average	115.73	115.59

Table 4.4 Results with and without vertical fins. Figures in the right show annual energy consumption in MWh. Vertical fins slightly increase energy levels.

4.9 Testing crossover and mutation probabilities

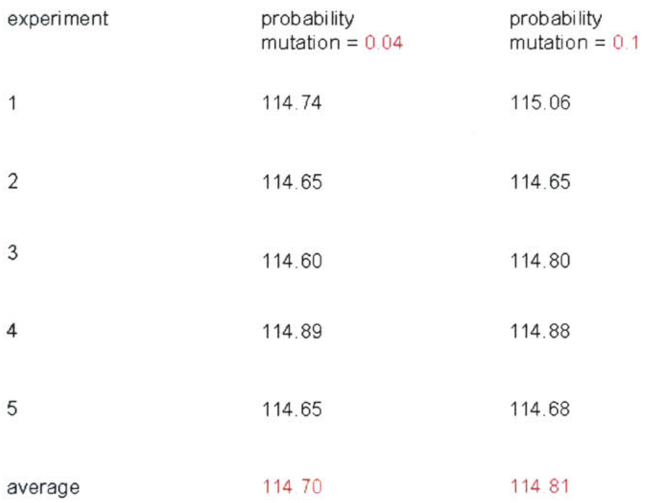
The values used for crossover and mutation probabilities throughout these experiments were taken from the literature, and are based on results from parametric studies done by others to determine values that would provide high quality results. However, and just for the sake of completeness, some parametric studies with different crossover and mutation values were performed here too.

MicroGAs usually use 1 as the value for crossover probability. Some experiments were done where this probability was reduced to 0.5 and to 0.1. The results corroborate the information found in the literature, as reducing crossover probability decreases the quality of results [table 4.5]. With the small population sizes used in MicroGAs, a high genetic combination rate favors the appearance of better results.

experiment	probability crossover = 0.1	probability crossover = 0.5	probability crossover = 1.0
1	114.77	114.74	114.80
2	114.88	114.65	114.56
3	114.83	114.60	114.59
4	115.21	114.89	114.53
5	114.53	114.65	114.77
average	114.84	114.70	114.65

Table 4.5 Results for different crossover probabilities. Figures show annual energy consumption in MWh.

The impact of changing mutation probability was assessed for a standard GA, with a population size of 30 individuals. The literature mentions that a mutation rate of 1/n, where n is the length of the chromosomes, is almost optimal (Goldberg, 1989). In this case, n=24, so 1/n=0.04. When mutation rate was increased to 0.1, and results were slightly worse, confirming the information found in the literature [table 4.6].



experiment	probability mutation = 0.04	probability mutation = 0.1
1	114.74	115.06
2	114.65	114.65
3	114.60	114.80
4	114.89	114.88
5	114.65	114.68
average	114.70	114.81

Table 4.6 Results for different mutation probabilities. Figures show annual energy consumption in MWh.

These experiments were not intended to be exhaustive, since others have done similar tests that are documented in the literature. They simply aimed at providing some evidence that the results achieved by others were also applicable to this problem setup. As mentioned in the introduction to the chapter, variations in objective function values are usually small, mainly due to the fact that the specific building layout used in these experiments is not too sensitive to changes in fenestration dimensions.

4.10 Conclusions of the chapter

The Generative System was first validated in relation to a hand-worked example for which the optimal solution could be manually calculated. Results proved highly satisfactory and provided enough confidence for the process to be extended to a larger solution space for which there would be no practical way of calculating the optimized solutions if the building would not have been designed with separable objective functions in mind. The GS was thus applied to a hyperspace of eight dimensions, corresponding to simultaneously sizing eight window dimensions in the building.

The results obtained showed that from random initial populations with high diversity the algorithm slowly converged to populations consisting of optimal or near-optimal solutions. For different runs of the problem the best solution obtained did show some particularities and variations, what demonstrates, even in this very limited and confined case, that different configurations may correspond to similar standards of environmental performance. This may constitute valuable information to the architect, who is provided with a number of alternative solutions over which he can further overlap other design criteria not included in the optimization process. This was one of the initial goals of this project, and these initial experiments are starting to confirm it.

It was also observed that for different orientations the solutions obtained varied substantially, as expected from the different requirements the glazing elements have to respond to. Also, for different climates the design solutions proposed by the algorithm differed considerably too. The same building layout needs to have different design components according to the climate it is located in order to improve its environmental performance.

Finally, the GS proved to be rather efficient and reliable in locating high performance solutions, even in the 16 million+ solution space, what provided enough confidence to pursue the experiments towards more complex designs, resembling more an actual piece of architecture than the schematic building layout used in this chapter. However, before that next step is taken, another test will still be made, comparing Genetic Algorithms to other existing search procedure for complex solution spaces, Simulated Annealing. That work will be described in the next chapter.

CHAPTER 5 Comparing GAs with Simulated Annealing and Tabu Search

5.1 Introduction

Many large-scale optimization problems cannot be solved to optimality using standard optimization techniques. For this type of problems one is forced to use approximation algorithms or heuristics, for which there is no guarantee that the solution found by the algorithm is optimal. In this chapter we present and compare three of those heuristic methods: Genetic Algorithms [GA], Simulated Annealing [SA] and Tabu Search [TS]. GA mechanisms have already been described in previous chapters. Here, SA and TS techniques are briefly described in their approaches to optimization and methods of search. Some comparative studies from the literature on the performance of the three different methods are presented. Hybrid implementations that combine features drawn from more than one technique are also described.

An implementation of Simulated Annealing working together with DOE2 was completed, and some results of computational experiments comparing the two techniques [SA and GA] are presented. Finally, there is a brief discussion of the relative merits of each of the methods and of potential applications of Tabu Search to architecture-related problems.

5.2 Description of two heuristic techniques [Simulated Annealing and Tabu Search]

5.2.1 Simulated Annealing

Simulated Annealing is a general optimization technique for solving combinatorial optimization problems. It is called a metaheuristic because it guides another heuristic technique, the descent method. It also incorporates randomization techniques.

The descent method is first described, to understand the role it plays in simulated annealing. For a minimization problem, the descent method needs a definition of feasible solutions, a cost function and a way to generate a transition from a solution to another. A neighborhood N(i) is defined for each solution i, consisting of all the solutions k that can be reached from i in one single transition. Starting at a given solution, the algorithm generates a possible transition from there to a neighbor j of it. If that neighbor has a lower cost, the current solution is replaced by it, otherwise another neighbor is selected and evaluated. The algorithm terminates when a configuration is obtained whose cost is no worse than any of its neighbors (van Laarhoven, 1987).

The descent method has the disadvantage being easily trapped in a local minimum, for which there is no information on how much it deviates from the global minimum. The local minimum location depends on the initial configuration, for which choice there are no guidelines. One of the possible ways to help prevent iterative improvement searches like this from being locally trapped is to promote a process where transitions which correspond to an increase in the cost function are also acceptable in a limited way. This is contrary to the descent method where only transitions corresponding to a decrease in the cost function are accepted, and is used by the simulated annealing algorithm.

Simulated annealing is based on the analogy between the annealing of solids and the problem of solving large combinatorial optimization problems. In condensed matter physics, annealing is a process in which a metal solid is heated up in a heat bath to a maximum temperature where it is melted, at which all the particles randomly arrange themselves in the liquid phase. This is followed by slowly lowering the temperature of the heat bath in a cooling process, during which the particles arrange themselves in the lowest-energy ground state of the corresponding lattice. The solid will achieve this lowest energy state provided the initial temperature is sufficiently high and the cooling is carried out slowly enough. If the cooling is too rapid, that is, if the solid is not allowed to reach thermal equilibrium for each temperature value, defects can get frozen into the solid and metastable amorphous structures may be reached rather than the low energy crystalline lattice structure (van Laarhoven, 1987). To avoid this, at each temperature level the crystal should be kept long enough to reach thermal equilibrium at that temperature.

Metropolis et al. (1953, quoted by van Laarhoven et al., 1987) used a Monte-Carlo method to simulate the evolution to thermal equilibrium of a crystal for a fixed value of the temperature T. It is called the Metropolis algorithm and works in the following way: Given the current state, S, of the crystal, a small displacement of a randomly chosen molecule is applied. If the new state corresponds to a lower energy level, that is, the difference in the energy level, Δ, between the current configuration S and the newly generated configuration

S' is negative, S' is accepted with probability 1. On the other hand, if $\Delta \geq 0$, that is, the new configuration corresponds to a higher energy state, S' can still be accepted with probability:

$$e^{-\Delta/k_B T}$$

where k_B is known as the Boltzmann constant. This acceptance rule is referred to as the Metropolis criterion.

At each temperature value T the solid is allowed to reach thermal equilibrium, characterized by a probability of being in a state i with energy E_i given by the Boltzmann distribution:

$$\mathbf{P\{X=i\}} = \frac{\exp(-E_i / k_B T)}{\sum_j \exp(-E_j / k_B T)}$$

As the temperature decreases, the Boltzmann distribution concentrates on the states with lowest energy and finally, when the temperature approaches zero, only the minimum energy states have a non-zero probability of occurrence [figure 5.1].

The Metropolis algorithm can also be used to generate sequences of solutions in a combinatorial optimization problem. For that we use an objective function C, the analogue of energy E, whose minimization is the goal of the procedure; and a control parameter c, the analogue of temperature T, which controls the radius of neighborhood and influences the probability of a non-improving solution being chosen.

The simulated annealing algorithm works like a series of Metropolis algorithms performed at decreasing values of the control parameter. It starts with a high value of the control parameter, corresponding to the state of high temperature mentioned for actual annealing. Given an initial solution i, a generation mechanism obtains another solution j by choosing at random an element from the neighborhood of i. This corresponds to the small displacement of a randomly chosen molecule in real annealing. The new solution j is

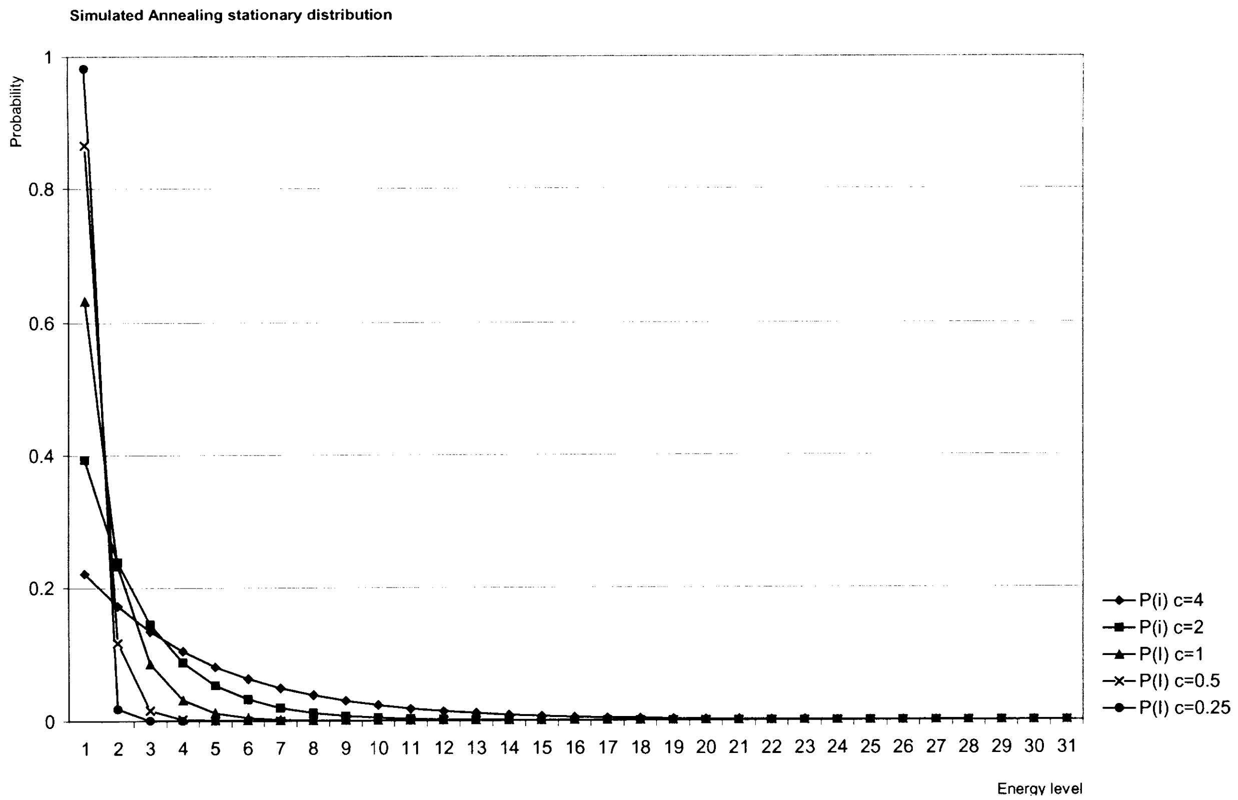


Figure 5.1 The graph shows that for decreasing temperature values, the probability of occurrence of lower-energy states increases, and it approaches 1 as the temperature control parameter approaches 0

accepted as the next configuration with probability 1 if ΔC_{ij} = C(j)-C(i) is negative or zero. If $\Delta C_{ij} > 0$, j can still be accepted with probability exp(-ΔC_{ij} /c). At each value of the control parameter this process is continued until equilibrium is reached, that is, until the probability distribution approaches the equivalent of the Boltzmann distribution.

In simulated annealing two types of algorithms can be used, homogeneous and inhomogeneous. For the homogeneous algorithm, the value of c does not vary with the iteration k. A sequence of homogeneous Markov chains of finite length is generated at decreasing values of the control parameter. For the inhomogeneous algorithm the value of c is changed between subsequent transitions.

For a homogeneous model it can be proven that the equivalent of the Boltzmann distribution for reaching thermal equilibrium is given by a stationary distribution q(c) whose components are given by (Aarts et al, 1997):

$$q_i(c) = \frac{\exp(-C(i)/c)}{\sum_j \exp(-C(j)/c)}$$

It can be proven that for the homogeneous model the simulated annealing algorithm asymptotically finds an optimal solution if we first take the limit of the homogeneous Markov chain for an infinite number of trials, and then take the limit for the control parameter to 0 (Aarts et al., 1997):

$$\lim_{c \downarrow 0} \lim_{k \to \infty} P_c\{X(k) \in S^*\} = 1$$

where *S** denotes the set of optimal solutions

This is a basic property of the simulated annealing algorithm, that it asymptotically converges to the optimum. But in any implementation of the algorithm, asymptotic convergence can only be approximated. The number of trials for each value of c_k (the control parameter) is a finite number, and $\lim_{k \to \infty} c_k=0$ can only be approximated by a finite number of values for c_k.

Usually one resorts to an implementation where a sequence of homogeneous Markov chains of finite length is generated at decreasing values of the control parameter, as mentioned above. The following elements must then be specified to make use of the Simulated Annealing algorithm (van Laarhoven, 1987):

1. A description of feasible system configurations
2. A generator of random changes in the configurations
3. An objective function C (analog of energy) whose minimization is the goal of the procedure
4. A control parameter c_k (analog of temperature) and a cooling schedule which tells how the temperature is to be lowered from high to low values. This includes after how many random changes in configuration is a downward step in c taken, and how large is that step.

For the cooling schedule the following parameters should be specified:

1. Initial values of the control parameter, c_0
2. Final value of the control parameter, c_f [stop criterion]
3. Length of Markov chains [after how many random changes in configuration is a downward step in c taken]
4. A rule for changing the current value of the control parameter, c_k, into the next one c_{k+1} [how large is the downward step in c]

There are two possible types of heuristic cooling schedules, static and dynamic (Aarts et al., 1997). In a static cooling schedule the parameters do not change during the execution of the algorithm. In a dynamic cooling schedule the parameters are adaptively changed during the execution of the algorithm.

A common example of a static cooling schedule is the geometric schedule, which is further described in section 5.4.1 and will be used in the simulated annealing implementation described there.

Aarts et al. (1997) also describe ways to implement a dynamic schedule. For example, a sufficiently large vale of c_0 may be achieved by forcing the initial acceptance ratio (number of accepted transitions at c_0) to be close to 1, typically between 0.9 and 0.99. The acceptance ratio can be calculated from a number of generated transitions, and the value of c_0 adjusted accordingly. The final value of the control parameter may be calculated adaptively by terminating the execution of the algorithm if the value of the objective function remains unchanged for a number of consecutive trials. The length of the Markov chain may be determined so that at each value of c_k a minimum number of trials is accepted. Since transitions are accepted with decreasing probability for smaller values of the control parameter, it is advisable to put an upper bound on the maximum number of trials for small values of $c_{k.}$

5.2.2 Tabu Search

Tabu Search is a metaheuristic that searches local neighborhoods for the best possible path to progress (Hertz et al., 1997). Contrarily to hill-climbing or steepest descent methods, which as mentioned before can be easily trapped in local minima, TS continues exploration without becoming confounded by an absence of improving moves, by always picking the best available move at each step even if it is a non-improving one. That is, lacking the possibility of improvement, TS will choose the least disimprovement.

However, when moves that imply a worse objective function value are allowed there is the risk of visiting again a solution that has already been evaluated and of generating cycles within the search. To help preventing falling back into a local minimum from which it has previously emerged, Tabu Search includes a form of short-term memory that keeps information on the itinerary through the last solutions visited. This information is kept on a list of size T called a tabu list and will prevent the occurrence of cycles of at most size T. The structure of the neighborhood N(i) will thus depend on the itinerary followed until i and thus on the iteration k. So, the neighborhood N(i,k) = N(i) – T(k).

In many cases using a list of solutions may be impractical if the solution description requires too much memory use. For considerations of memory conservation, Tabu Search uses instead a list of tabu moves, and often records less than the full range of attributes required to characterize a move or the solution to which it is applied (Glover, 1989). A partial range of attributes is recorded instead, which potentially may be shared by other moves or solutions. This implies that we may be giving a tabu status to solutions that were not yet visited, if moves leading to those solutions are covered by the attributes present in the tabu list.

To explain it better, in the simplest case, if move s is performed, move s^{-1} (the inverse move) is added to the tabu list to prevent cycling. Thus the list T is given by:

$T = \{ s^{-1} : s = s_h \text{ for } h > k - t\}$

where k is the iteration index and t is the size of the tabu list

T is thus the set of moves that would reverse (or 'undo') one of the moves made in the t most recent iterations of the search process.

After each iteration k the tabu list is updated by setting $T = T - s_{k-t}^{-1} + s_k^{-1}$

However, when moves are not completely characterized but instead only a partial range of attributes is used to describe them, the tabu list does not consist simply of the moves s_h^{-1} for h>k-t, but of collections C_h of moves where each C_h defines its membership by certain attributes (Glover, 1989). This cause the list T to contain s_h^{-1} and other moves that likewise satisfy these conditions, and thus a tabu status may be given to moves leading to solutions that were not already visited. To help improve on this situation, we may accept a tabu move in spite of its tabu status if it leads to a solution better than the best found so far. This means the solution passed an aspiration level, that is, it is better than a certain threshold value, in this case the best objective function value found so far.

One aspect that is evident about Tabu Search is that it is a quite greedy algorithm. This aggressive orientation relies on two considerations, according to Glover (1989): first, that many optimization problems can be solved optimally by making the best available move at each step; second, that local optimality does not represent a barrier to TS due to its procedural organization and its use of short-term memory and thus there are less compelling reasons to delay the approach to a local optimum.

Apart from short-term memory, TS also makes use of intermediate and long-term memory functions. Intermediate memory is used to implement temporary intensification of the search, what means searching more deeply around some area of the solution space because it contains some interesting solutions. Intensification is thus implemented by giving higher priority to solutions similar to the current solution. It operates by recording and comparing features from a number of best trial solutions generated during a particular period of the search. Features that are common to a majority of those solutions, such as values attributed to certain variables, are taken to be an attribute of good solutions. TS then seeks new solutions containing those features by restricting or penalizing other types of solutions during that period of regional search intensification.

Long-term memory is used to implement diversification of the search. Its principles are roughly the reverse of those for intensification, since diversification guides the process to regions that markedly contrast with those examined thus far by penalizing features that are found to be prevalent in previous executions of the search process (Glover, 1989). This process differs considerably from those methods that seek diversity by randomly generating new random points in the search, because it provides the opportunity to learn from previous steps in the search. Both intensification and diversification can be periodically introduced in the TS process and represent an important use of memory structure in Tabu Search.

5.3 Comparative studies between GAs, SA, TS and hybrid approaches

Many studies exist in the literature comparing GAs, SA and TS performance for a variety of problems. Other studies propose hybrid approaches where features from different techniques are incorporated into a new hybrid method in order to improve the overall behavior of the search. In this section some of those studies are presented.

Gallego et al. (1998) compared SA, TS and GA in terms of memory use, intensification, diversification, and problem modeling issues like neighborhood structures. Table 5.1 summarizes the key issues pointed out in the paper, which are further developed in the text.

In relation to short-term and long-term memory, Simulated Annealing is based on a statistical process (the Boltzmann distribution) and hence makes little use of memory. Configurations may be visited repeatedly during the search as a consequence of that. The only use of memory in SA is that the best solution found so far is stored, but this is usually not used to guide the search. A way to guide the search based on that information is to start searching from the best solution each time a temperature reduction is performed.

Genetic algorithms make explicit use of memory when elitism is used (always copying the best solution of a generation to the next generation), in order to preserve valuable genetic information. They also make implicit use of memory by keeping track of high-performance building blocks, thought the process of selection. The fact that the algorithm is based on sequences of generations also implies an implicit use of memory.

The strategy that makes more use of memory is Tabu Search. Here different types of memory structures are used to guide the search. Four types of memory structures are used: *Recency*-based memory is a type of short-term memory that keeps track of the most recent moves done by the algorithm and labels the reverse moves as tabu active to prevent revisiting solutions already tested in the recent past. Tabu lists are used to manage this type of memory.

	Memory	**Intensification**	**Diversification**	**Neighborhoods**
SA	Almost no use of memory Only if best solution found so far is used to restart search after each temperature reduction	Search moves less freely as it progresses, with lower probability of accepting very different solutions from the current one Neighborhood radius can also decrease	At early stages of the search, probability of accepting solutions that significantly decrease performance is higher, meaning that new regions of the search space are explored	Requires a definition of neighborhoods. Size of neighborhood is not too important, as not all of neighborhood is explored [at each step, a single solution is picked for evaluation]
GA	Some use of memory Explicit – elitism Implicit – keeps track of high-performance building blocks; sequence of generations propagates useful genetic information	Non-systematic Mutation: introduces a small change to an existing solution, meaning it is searching in its vicinity	Crossover: new points in the solution space are created by combining existing solutions MicroGAs have high diversification by often starting new random populations	No definition of neighborhood New solutions are found by the process of crossover between two existing solutions
TS	Extensive use of memory Recency-based: keeps track of recent moves to prevent reverse ones Frequency-based: keeps track of frequency of certain features in solutions already visited. Used for diversification Quality memory: stores which building blocks lead to high quality solutions. Used for path relinking Influence memory: records impact of certain moves and makes bad moves tabu active for some period of time	Systematic Stores good configurations and then explores their neighborhoods more thoroughly Path relinking: building blocks from elite solutions stored in memory can be used to produce new attractive solutions	Systematic Temporary change of rules can penalize solutions similar to the current one. Can be implemented by removing some feature of the current solution that temporarily become tabu-active, thus forcing the search to visit previously unexplored regions.	Requires a definition of neighborhoods. Neighborhood size is more critical, since it is a greedy algorithm that explores all of a single neighborhood for best path. May include some method to explore only most attractive areas of the neighborhood, like first randomly sampling neighborhood and then exploring only the best solution areas

Table 5.1 Comparison of characteristics of the 3 heuristics methods, adapted from Gallego et al.(1998).

Frequency-based memory is a type of long-term memory that can keep track of the frequencies with which certain features of a solution appeared in the solutions already visited by the algorithm. This information can then be used in the diversification phase so that configurations containing unused features are preferred. There is also a *quality* dimension to memory use in Tabu Search in a way that building blocks that lead to high-quality solutions can be stored in long-term memory in elite configurations. These building blocks are then used in the creation of new configurations using a path relinking strategy. Finally, there is *influence* memory, which accounts for the impact of performing a certain move on the quality of the solution. Moves that considerably worsen the solution performance will remain tabu-active for as long as possible.

In terms of intensification, Simulated Annealing implements it by allowing the search to move more freely only at the early stages of the process when the temperature control parameter is higher. As the temperature lowers down the algorithm is hopefully closer to the optimal solution and has a lower probability of accepting non-improving solutions, thus meaning it would not accept points too far from the current solution. Also, the radius of the neighborhood can be decreased during the search so that at later stages only the immediate neighborhood of solutions is sampled.

In GAs, intensification is carried out in a non-systematic way by the mechanism of mutation, which performs small changes in candidate solutions in a way that corresponds to searching its immediate neighborhood.

TS performs intensification in a systematic way, by storing good configurations found along the search and later exploring their neighborhoods more thoroughly to find local optimal solutions near those configurations. Intensification can also be performed using the mechanism of path relinking, where building blocks from elite solutions stored in memory can be used to produce new attractive solutions.

For diversification, Simulated Annealing uses the fact that at early stages of the search the probability acceptance test is more likely to accept solutions implying a significant decrease in objective function values, thus meaning new regions of the search space are being explored.

In typical GAs, diversification is mainly introduced by the crossover mechanism, where two high-performance solutions swap portions of their chromosomes to generate new configurations. This usually implies exploring new regions of the search space. The process of randomly choosing pairs of individuals for crossover, and randomly generating crossover points further ensures introduction of diversification. In Micro-GAs [previously described in chapter 3], diversification is much enhanced by the fact that new points are introduced in the search each time a new micro-population is generated.

Again in TS diversification is used in a more systematic way, by penalizing solutions similar to the current one. This can be implemented by temporarily changing the rules for finding new solutions, for example, removing some feature of the current solution that temporarily becomes tabu-active, thus changing the construction of the solution and forcing the search to visit previously unexplored regions.

In terms of problem modeling, the main issue apart from defining the structure of feasible configurations and an objective function is the definition of neighborhood structures. In Simulated Annealing the size of a neighborhood is usually large, but this does not represent a problem since a single solution is picked from the neighborhood for evaluation.

In GAs there is no actual definition of a neighborhood since new solutions are found by the process of crossover between two existing solutions.

In TS the definition of neighborhoods is more critical since the algorithm is more greedy and tends to look at the entire neighborhood to find the best path to follow. If the size of the neighborhood is too large to be computationally practical to evaluate all of it, there could be a way to explore only the most attractive areas of the neighborhood, like first randomly sampling the neighborhood and then exploring around better solution areas.

Gallego et al. (1998) then describe an hybrid approach between Tabu Search and Genetic Algorithms. The main algorithm is of TS type, but it incorporates GA concepts, namely using a population of solutions instead of a single one. The TS search is carried out for each solution independently. After a certain number of iterations, a new generation of configurations is formed from the current one, as in GAs. Two main mechanisms are applied to form the new generation: guided crossover and path relinking, both of which operate on high-quality building blocks. Guided crossover is used as the main mechanism for diversification, instead of the more conventional frequency-based memory applied in TS. Path relinking takes two solutions, one a current solution and other belonging to the list of elite configurations kept by the algorithm. It adds attributes of the elite solution to the current one, assuming that the elite configuration will have elements that can improve the other. In the experiments described by the author it was shown that the hybrid algorithm performed better than either standard SA or GA. GA results were nevertheless better that SA results.

Josefowska et al (1998) compared three metaheuristics (Genetic Algorithms, Simulated Annealing and Tabu Search) for problems of scheduling nonpreemtable jobs that require simultaneously a machine from a set of parallel, identical machines and a continuous, renewable resource. The optimization criterion is the schedule length. The problem is to find a sequence of jobs on machines and simultaneously to find a continuous resource allocation that minimizes the schedule length. For each feasible sequence visited in the solution space the corresponding schedule length is then calculated, and the optimal resource allocation among jobs sequences is found by solving a convex mathematical programming problem.

The three heuristics were compared by setting the stop criterion as the number of solutions visited (2000 solutions). This ensures a comparable computational effort devoted to each method. The first experiment was carried out for 2 machines and 10, 15 and 20 jobs. For n=10 and n=15 jobs, the experiments were done for 100 instances. For n=20 only 60 instances were generated due to large computational demand. The second experiment was carried out for 10 jobs and number of machines m=2, 3 and 4. For all groups 100 instances were generated.

It was observed that TS was able to find optimal solutions most often, followed by GA and finally SA. Furthermore, it was observed that the growth of the number of jobs caused a larger deterioration in the performance of SA and GA than it did for Tabu Search. Increasing the number of machines has no negative effect on TS, with the number of optimal solutions actually increasing. The performance for SA was the most affected, and that of GA was also affected although not as much as SA. In general the Genetic

Algorithm behaved better than Simulated Annealing, but was still outperformed by the Tabu Search strategy.

Chiang et al. (1998) propose an hybrid tabu search approach that combines tabu search with simulated annealing. It is basically a tabu search mechanism with a revision on the acceptance or rejection rules for candidate solutions. Consider a move (u,v) with difference in objective function from the initial solution $\Delta C(u,v)$. Move (u,v) can be classified as tabu or non-tabu. This fact is used to accept or reject the solution, according to the following rules: In the case move (u,v) is not on the tabu list, the candidate solution is accepted with probability 1 if $\Delta C \leq 0$. In case $\Delta C > 0$, the candidate solution is accepted following the Metropolis criterion ($P_{accept} = \exp(-\Delta C/T)$), where T is a control parameter like the temperature in SA. If move (u,v) is in the tabu list, $\Delta C \leq 0$, and the current cost of the solution is lower than the best cost so far (Curcost < Mincost), the solution is accepted due to the aspiration criterion. If move (u,v) is in the tabu list, $\Delta C \leq 0$, and Curcost ≥ Mincost, the solution is accepted with probability $\exp(-\Delta C/T)$. If move (u,v) is in the tabu list and $\Delta C > 0$, the solution is accepted with a further reduced Metropolis probability $\exp(-\Delta C/T)[(1-kuv)]$, where kuv is set in the range (0.5,0.9). Fine tuning is required to obtain the best value for kuv, and from the authors' experiments it seems to be 0.7. The hybrid tabu search approach proved to outperform both typical TS and SA in those experiments. Table 5.2 summarizes the several strategies used by Chiang et al. to calculate the probability of accepting a given move.

Almeida et al. (1998) propose a composite heuristic where different strategies are used at different phases of the search. The problem under study is that of generating the schedule that minimizes the weighted sum of total earliness and total tardiness of a set of jobs n to be processed in a single machine. The different types of neighborhoods used are:

$N_1(S)$ = {S': S' is obtained by swapping a pair of adjacent jobs in S}

$N_2(S)$ = {S': S' is obtained by swapping a pair of jobs that are sequenced at most q_1 positions apart in S}

where q_1 is an integer parameter set to a small positive value

$N_3(S)$ = {S': S' is obtained by reinserting one job at least q_2 positions apart from its position in S}

where q_2 is an integer parameter

Tabu condition	Change in objective function [ΔC]	Probability of accepting move
Move is not in Tabu list	$\Delta C \leq 0$	1
	$\Delta C > 0$	exp ($-\Delta C/T$)
Move is in Tabu list	$\Delta C \leq 0$ and Curcost < Mincost and Curcost ≥ Mincost	1 exp ($-\Delta C/T$)
	$\Delta C > 0$	exp ($-\Delta C/T$) [1-kuv] kuv $\in$ [0.5, 0.9] fine tuning required by user

Table 5.2 Strategies used by Chiang et al. (1998) to calculate the probability of accepting a given move.

At the first phase of the process a tabu search method is applied, using neighborhood N1. This corresponds to a short-range search. The second phase uses a simulated annealing method corresponding to a medium-range search, with neighborhood N2. If the SA search yields a new best solution, a search for a local optimum around that solution is done using steepest descent, a short-range method [phase 3]. If it does not, the search moves on to phase 4, where random selection is performed in a long-range approach.

Performing a simulated annealing search at a medium-range after completing an intensive tabu search allows for a wider knowledge over the sub-region under examination before it is abandoned in phase 4. If the medium-range flexible search does not lead to a new best solution the decision to move the search to a new sub-region is taken immediately. If it does, the sub-region is not left before looking for a local optimum.

The overall design of the composite heuristic is based on the assumption that alternating the search range and strategies would yield a better coverage of the set of feasible solutions than just performing an intensive search at short range around solutions selected during a diversification step. When compared with both Tabu Search and Simulated Annealing, the composite heuristic performed better than any of those, being able to find optimum solutions more often, and also being more efficient in terms of reducing CPU time.

Mantawy et al. (1997) propose a combination of simulated annealing and tabu search that basically consists of a SA algorithm with a TS algorithm used as a filter to prevent cycling of accepted solutions. Because SA is a memoryless technique, the use of short-term memory is supposed to improve its performance. This is achieved by applying the TS test to all trial solutions. The accepted solution by the tabu test is then tested by the SA. The TS test is mainly used to reject repeated solutions before they are tried by the SA method. The results obtained were again better than those of simple SA and TS. The basic advantages of the proposed algorithm were higher speed of convergence and higher quality of solutions.

5.4 Application of SA and TS to the test building used in chapter 4

5.4.1 Simulated Annealing

For this experiment, the same building as in chapter 4 was used. To apply the Simulated Annealing algorithm to that problem, the following approach is proposed:

1. Description of feasible configurations

A configuration is an array of eight numbers corresponding to window width and height for north, south, east and west orientations. Those dimensions have to be between 0.3 m and 2.4 m, at discrete steps of 0.3 m.

k_1 = north window width
k_2 = north window height
k_3 = south window width
k_4 = south window height
k_5 = east window width
k_6 = east window height
k_7 = west window width
k_8 = west window height

So, a description of a solution is:

$(k_1, k_2, k_3, k_4, k_5 , k_6, k_7, k_8)$

subject to $0.3 \leq k_n \leq 2.4, \quad n=1,2,..,8$

2. A generator of random changes in the configurations

A neighbor of a given configuration may be found by adding or subtracting the value of 0.3 m from one of the eight window dimensions forming a solution. To randomly generate a neighbor, the following procedure can be used: randomly choose between the values 0.3 or –0.3 (increasing or decreasing the window dimension); randomly pick one window dimension to be changed; add either 0.3 or –0.3 to the window dimension chosen; verify if the resulting solution is feasible.

Let r_1 = random (0.3 or –0.3)
Let r_2 = random (k_1, k_8)

Set $r_2 = r_1 + r_2$
Subject to $0.3 \leq r_2 \leq 2.4$

3. Objective function (analogue of energy)

The objective function value is the result of running a DOE2 simulation with the window parameters chosen, and expresses annual energy consumption in the building in MWh.

4. Cooling schedule

For this problem a static cooling schedule is applied. An example of a static cooling schedule is the geometric schedule proposed by Kirkpatrick et al (1983), and described by Aarts et al. (1997), which works the following way:

a] Initial values of the control parameter, c_0

A proposed value for c_0 is the maximum difference in objective function values between any two neighboring solutions. This is supposed to ensure a sufficiently large initial value for the control parameter. For the hand-worked example done for the Phoenix climate, it is possible to determine that the maximum difference in objective function values from two neighbor solutions is 1.2 MWh. Thus, 1.2 could be a possible value for the initial value of c for the Phoenix climate.

b] Final value for the control parameter, c_f [stop criterion]

It is proposed that the final value for the control parameter is fixed at some small value that may be related to the smallest possible difference in cost between two neighboring solutions. For the hand-worked Phoenix example, the minimum objective function difference between two neighbors is 0.006 MWh, what could thus become the final value for the control parameter.

c] Number of trials after which a downward step in c taken

The number of trials after which a downward step in c is taken is set at some number that may be related to the size of the neighborhood. In this case, the maximum size of the neighborhood is 16 (although this may include infeasible solutions in situations where one or more parameters are already at their upper or lower bounds). A number like 32 could thus be proposed.

d] A rule for changing the current value of the control parameter, c_k, into the next one c_{k+1} [how large is the downward step in c]

A possible rule for decrementing the control parameter is

$$c_{k+1} = \alpha \cdot c_k, \quad k=0,1,\ldots,$$

where α is a positive constant smaller than but close to 1. Typical values are between 0.8 and 0.99.

The graph in figure 5.2 shows the probability of accepting a non-improving solution for decreasing values of the control parameter, as it is lowered from a maximum value of 4.1 to a minimum of 0.19 [Note: These values are in MBTU, Million British Thermal Units, and correspond to the maximum and minimum differences in objective function values from two neighbor solutions, respectively 1.2 and 0.006, in MWh]. The constant by which the control parameter is decreased is α=0.9. The plots were done for three different values of cost difference: 4, 2 and 1 MBTU. It can be seen that when the cost difference is high (4 MBTU), the probability of accepting a non-improving solution is low, with a maximum value of 0.37 at the maximum value of the control parameter (4.1) and quickly tending to 0 as the control parameter value decreases. When the cost difference is small (1 MBTU), the probability of accepting that non-improving solution is quite high (0.78) at maximum values of the control parameter and slowly tends to 0 as the control parameter decreases.

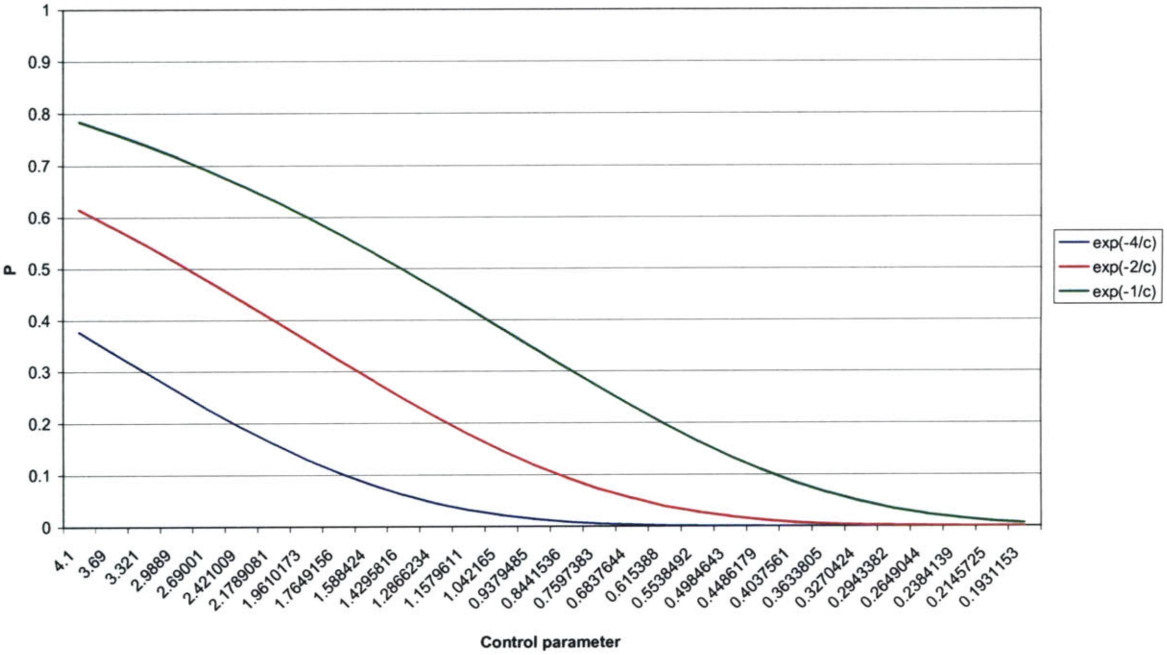


Figure 5.2 Probability of accepting a non-improving solution for decreasing values of the control parameters

5.4.2 Tabu search

To apply Tabu search to the problem under study we need a definition of feasible solutions, a definition of moves, size of the Tabu list, definition of an aspiration level, definition of the objective function, and ways to perform intensification and diversification of the search.

1. Description of feasible solutions

As in simulated annealing, a solution is an array of eight numbers corresponding to window width and height for north, south, east and west orientations. Those dimensions are between 0.3m and 2.4m feet, at discrete steps of 0.3m.

k_1 = north window width

k_2 = north window height

k_3 = south window width

k_4 = south window height

k_5 = east window width

k_6 = east window height

k_7 = west window width

k_8 = west window height

So, a description of a solution is:

$(k_1, k_2, k_3, k_4, k_5 , k_6, k_7, k_8)$

subject to $0.3 \leq k_n \leq 2.4$, $n=1,2,..,8$

2. A definition of moves

A move may be performed by adding or subtracting the value of 0.3 m from one of the eight window dimensions forming a solution. The TS algorithm will explore the entire neighborhood, that is, consider all possible moves at a certain point, and then chose the one that offers the best improvement [or the least disimprovement].

Set $r_1 = r_1 + 0.3$ or $r_1 = r_1 - 0.3$

Subject to $0.3 \leq r_1 \leq 2.4$

3. Tabu list

If a move of adding 0.3 to a dimension is performed, the move of subtracting 0.3 from that same dimension becomes tabu. The size of the list is related to its main function of preventing cycling. If the list is too short it may not prevent cycling. But if the list is too long it will create excessive restrictions and increase the mean value of visited solutions, thus slowing down the algorithm. A way of overcoming the difficulty of sizing the Tabu list is to vary its size during the search from a maximum to a minimum value (Hertz et al., 1997).

4. Aspiration level

The aspiration level most commonly used is if the Tabu move leads to a solution that is better than the best solution found so far.

5. Objective function

The objective function value is the result of running a DOE2 thermal and lighting simulation with the window parameters chosen, and expresses annual energy consumption in the building in MWh.

6. Intensification of search

To intensify the search in regions that seem to hold good solutions, the algorithm goes back to the best solutions found so far. The size of the Tabu list is then decreased for a certain number of iterations. Other way of performing intensification is to use 'long-term memory', where the components of good moves or solutions are memorized as building blocks. During the intensification phase, the moves or solutions are evaluated taking into account the building blocks they include, and moves that tend to destroy those building blocks are penalized in terms of objective function values, while those that generate them are favored. Path relinking can also be applied, making use of those building blocks.

7. Diversification of search

A possible way to diversify the search is to perform several random restarts in order to assure that all areas of the solution space are sampled. Another way is to penalize moves frequently performed and solutions visited too often or too similar to the current ones (Hertz et al, 1997), in terms of the objective function. This penalty should be set large enough to allow the search to escape from the current region. The modified objective function is then used for a certain number of iterations. Another alternative to using penalties applied to the objective function is to turn some characteristics of the solution tabu-active, so that the search would have to move away from solutions containing those characteristics.

5.5 Test procedures

The two optimization techniques tested were Genetic Algorithms and Simulated Annealing, which were compared in terms of results achieved during a limited number of function evaluations [500]. Results were analyzed in terms of proximity to the global minimum and number of evaluations necessary to achieve that result.

5.5.1 Genetic Algorithm

For the Genetic Algorithm, a MicroGA was applied as in the previous experiments, which was also was given the additional property of restricted 'memory' [automatically retrieve from memory the fitness of solutions already evaluated]. Because experiments were done in terms of number of function evaluations performed, this gave an additional edge to the GA with respect to Simulated Annealing, since it reduced the number of evaluations performed by the GA.

5.5.2 Simulated Annealing

The base code for the Simulated Annealing implementation was developed by William Goffe and is described by Goffe (Goffe et al.,1994) and Corana (Corana et al., 1987). Some modifications to the existing code were performed in order to adapt it from continuous to discrete variables, and to introduce problem-dependent neighborhood structures.

The Simulated Annealing experiments were somewhat modified in relation to what had been proposed as a cooling schedule in section 5.4.1, due to computational time constraints. The initial temperature of 1.2 was still used as the maximum possible objective function difference between two neighboring solutions, but the temperature decrease parameter was set to 0.8 to let the temperature lower sufficiently in the limit of 500 evaluations. The number of trials after which a temperature reduction was performed was 32. The stopping criterion was the maximum number of function evaluations [500].

After several experiments, the neighborhood radius chosen for the tests was 0.3, that is, a neighbor was generated by either decreasing or increasing one of the variables by the value of 0.3 [as previously described]. For out-of-bounds trials, the solution found was to move a step in the opposite direction from that causing the variable to get out of bounds. The initial points for SA search were always points that were contained in the GA initial population. That was considered to give a fair basis for comparing the two methods, since initial points were identical. After each temperature reduction in SA, the search was restarted from the best point found so far, in order to include some memory use in the SA algorithm too.

5.6 Results from computational experiments

The results of the computational experiments are summarized in tables 5.3 and 5.4. For each method [GA and SA] only ten experiments were performed due to computational time constraints [at the time these experiments were carried out, each objective function evaluation took about 3 minutes and 20 seconds, since the program was still running on Unix platform, and remotely from LNBL]. The tables list, for each experiment, the minimum objective function value found, the difference of that value to the minimum [known to be 144.53 MWh], and the number of function evaluations after which that value was found.

In general the two algorithms performed quite well, being able to find results very close to the global minimum. The Genetic Algorithm performed somehow better than the Simulated Annealing algorithm, finding on average slightly lower results, although the differences are very small. The GA needed fewer evaluations than the SA to achieve similar results [about 5% less evaluations]. The GA was also able to locate the global minimum in one of the experiments, what did not happen with Simulated Annealing.

Genetic Algorithm			
Experiment	Result [MWh]	Difference to minimum	Number of evaluations
1	114.82	0.29	399
2	114.85	0.32	456
3	114.53	0	383
4	114.67	0.14	426
5	114.64	0.11	418
6	114.74	0.21	289
7	114.65	0.12	376
8	114.60	0.07	438
9	114.89	0.36	351
10	114.65	0.12	476
Average	114.704	0.174	401

Table 5.3 Results from experiments using a Genetic Algorithm

Simulated Annealing			
Experiment	Result [MWh]	Difference to minimum	Number of evaluations
1	114.97	0.44	439
2	114.78	0.25	478
3	114.68	0.15	423
4	114.66	0.13	393
5	114.73	0.20	406
6	114.68	0.15	483
7	114.71	0.18	264
8	114.67	0.14	474
9	114.66	0.13	387
10	114.86	0.33	452
Average	114.74	0.21	420

Table 5.4 Results from experiments using Simulated Annealing

Figures 5.3 and 5.4 show the search progression for both the MicroGA and SA. The graphs clearly illustrate the differences in the corresponding search processes.

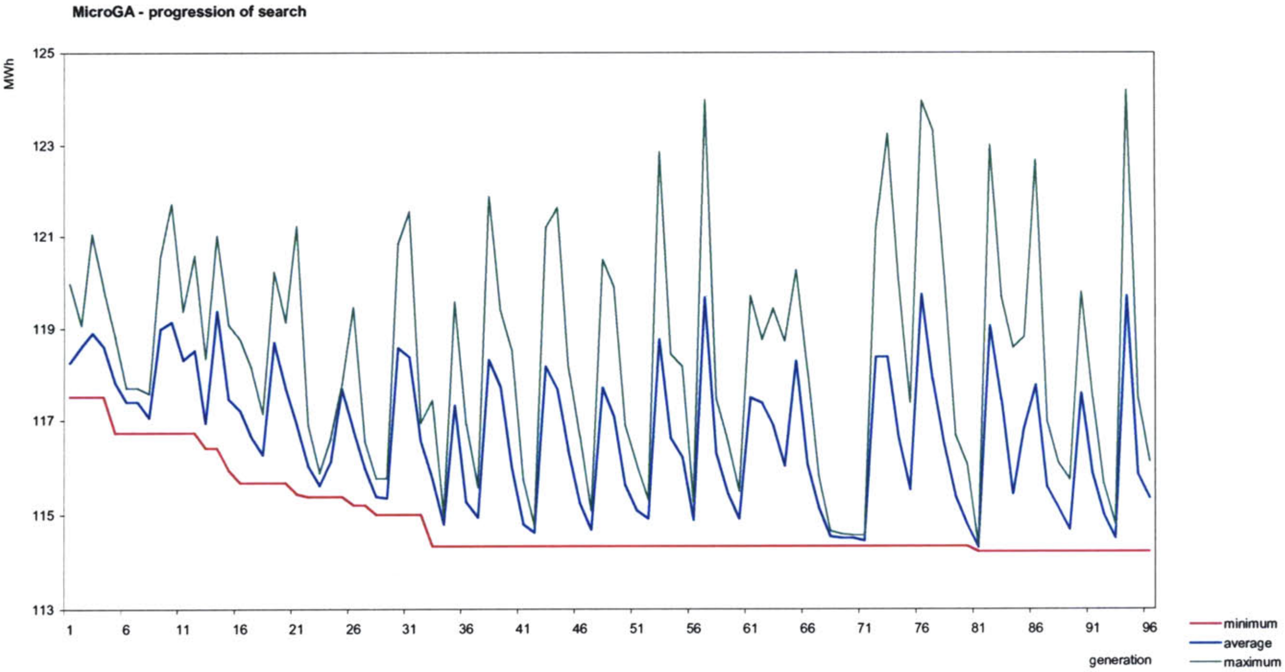


Figure 5.3 Search progression for one experiment using a MicroGA

Because the GA searches a population of individuals in parallel, three types of information are displayed: the minimum objective function value, the average objective function value of the population, and the worst individual. It is possible to gain more insight into the mechanics of the algorithm's search by looking simultaneously at all this information. From the graph it can be seen that the minimum objective function value quickly decreases during the first 30 something generations, and after that stabilizes and suffers only minor improvements. The valleys in the average fitness show points of convergence of the population, after which a peak always exists, illustrating the effect of a new random population being generated, causing the average fitness to suddenly go up. The succession of peaks and valleys is a consequence of the dynamics of the algorithm. The worst individual's performance shows that very bad solutions can be introduced by the new random generation, but those are quickly excluded and replaced by high-performance ones, although in that process they may still exchange useful genetic information with the more fit individuals.

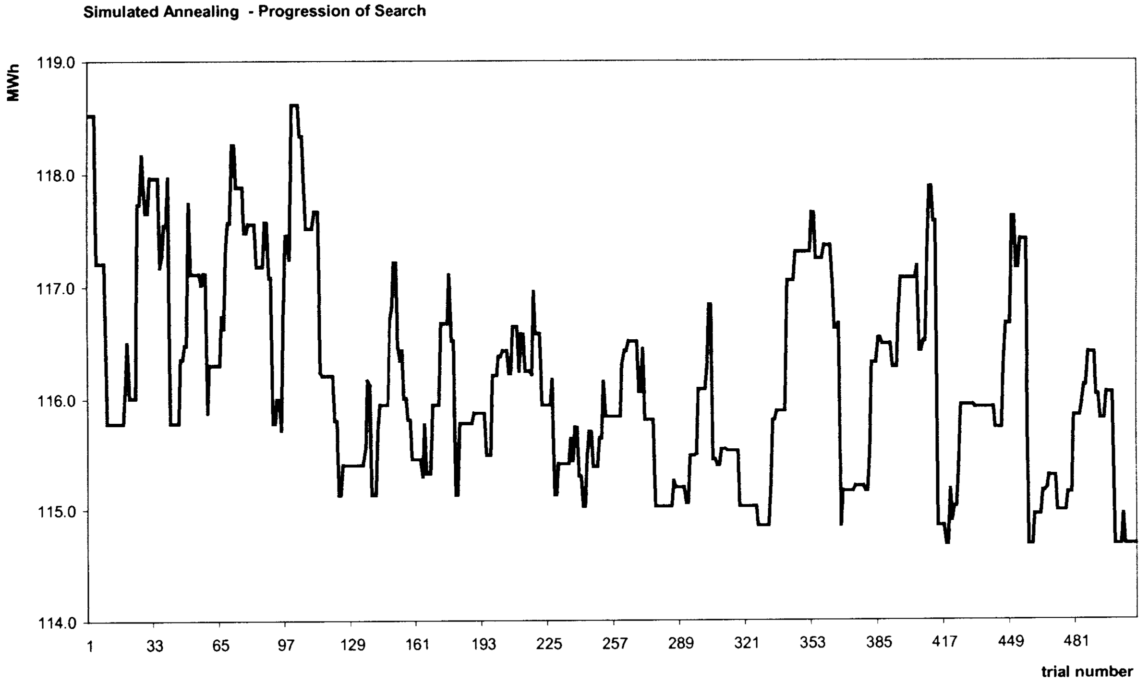


Figure 5.4 Search progression for one experiment using Simulated Annealing

Simulated Annealing, on the other hand, searches from a single point and so its progression is simply illustrated by a single line. In general terms, the search progression for SA is slower and more gradual than the MicroGA. The first few trials led to a steep decrease in the objective function value, but then the search drifts to poor performance areas. The 'memory' implementation makes the search restart from the best solution found so far each time the control parameter value is decreased, what causes the consecutive appearance of similar 'valleys' in the search. This forces the search to abandon areas of little interest and go back to the neighborhood of the best performance solution so far. Improvements on the best solution can occur right after the restart point, or after the search has been in more unpromising areas, showing that in just a few moves the objective function can steeply decrease, what suggests the presence of a few variables that may strongly condition the solution [such as north window dimensions, for example]. However, the sudden presence of a valley after a peak is often due to the restart of the search from the best solution. An interesting extension of this experiment could be to analyze the SA search progression with no memory implementation and see how it differed from this one.

As for the results, the literature review had already suggested that GAs usually outperform SA, what was confirmed by these experiments. In any case, comparing GA and SA performance for the particular problem of improving buildings environmental performance further increased our confidence in the Generative System results, since SA is another well established heuristic method for optimization problems, and the code used for this implementation had been previously tested by its author, with published results (Goffe et al.,1994). More extensive tests and fine tuning of the SA algorithm [studying different cooling schedules, for example] could be done, but the main objectives of these experiments were to provide a basis for comparing our GS results with other methods, and were considered to be achieved here.

5.7 Conclusions of the chapter

Three heuristic techniques [Genetic Algorithms, Simulated Annealing and Tabu Search] were described, and some studies presented from the literature where those methods are compared or combined into hybrid approaches. Both GA and SA where then applied to the same problem that was described in chapter 4.

Apart from the performance results, already analyzed, some considerations can be made about the application of the three search methods to architecture-related problems. Even though Simulated Annealing seemed to perform slightly worse than the GA in finding good performance solutions, that difference would not be significant enough to exclude it as the choice for the Generative System search module. However, other factors that differentiate SA from GAs are also to be considered.

One such factor is that Simulated Annealing searches for a single solution, while GAs use populations of solutions. From an architectural design point of view, the latter has the advantage of providing alternative solutions to the architect. That could also be achieved by doing more than one run of a SA-based system, but if alternative solutions are an issue, GAs seem more appropriate from the outset.

In later chapters of this thesis, work will be presented that heavily relies on the existence of a population of solutions. That work relates to the search of Pareto fronts for multicriteria problems, where a frontier of solutions that represent the best trade-offs between conflicting objective functions is sought for, instead of the best performance for a single objective function. Having that extension in mind, GAs seem again a more appropriate solution.

A word remains to be said about Tabu Search. From the literature review, TS outperformed both GAs and SA in some problems such as those studied by Josefowska. Although it can be argued that the way the problem itself was structured may have influenced the results, Tabu Search seems to be a more sophisticated algorithm than either SA or GAs, mainly due to its use of different types of 'memory,' and to its adaptive changes in search strategy along the process. No implementation of TS was carried out or adapted for the work presented in this chapter, because it is a complex algorithm to write, and the number of researchers using TS and willing to share existing implementations is reduced. Nevertheless, it remains as further work that may be explored in the future to study the adaptation of TS to architecture-related problems. If some speculations based on Josefowska's results are permitted, TS seems not to degrade its performance with very large problem sizes. This may be a quite useful feature for the type of complex problems that will be presented as future work to follow from this dissertation, in the conclusions chapter. Hybrid implementations, such as those presented in the literature review, taking advantage of the best characteristics of each search method, may also be a path for further exploration.

CHAPTER 6 Including a Formal Balance Algorithm

6.1 Introduction

To this point, the Generative System was used with only one kind of objective function [annual energy consumption] to assess the quality of solutions and guide the search process. In this chapter, the GS is combined with another algorithm, which looks at aspects such as order and diversity present in a façade's composition (Duarte, 1997). The quantification of these attributes allows the development of other types of evaluation criteria for the façade-level problems we have been studying until now. The two criteria [energy performance and compositional balance] are combined into a single number by a linear function. Linear combination is a plain aggregation approach, often making use of weighting factors. The shortfalls of such simplified methods will be later discussed in chapter 8, where several multicriteria optimization methods are reviewed. However, for the very simplified work presented in this chapter, it was considered that a plain aggregating method was suitable enough.

The work presented in this chapter is somewhat restricted to the simplest cases. More appealing developments are possible, and will be considered in the conclusions of the chapter. However, some interesting results are still observable, like analyzing the different strategies the GS used for solving conflicts between the two evaluation criteria according to the geographical location of the building.

6.2 Experimental setup

The façade balance algorithm described by Duarte (1997) has two basic components: vertical balance and horizontal balance. In this study only vertical balance was considered, which is based on an analogy between vertical balance in a composition and gravity balance.

A façade can belong to a single building or to a group of buildings, such as a street façade. In this case, we used a single building, so that it could be simulated using only one DOE2 model. Nevertheless, that building was given enough volumetric diversity, in terms of the height, width and length of the spaces, so that a variety of solutions could emerge. Figure 6.1 shows the basic geometry of the building. Note that room depths also vary, so some of the spaces may be more difficult to side-light from the façade openings than others. Room depth becomes this way another factor to promote diversity.

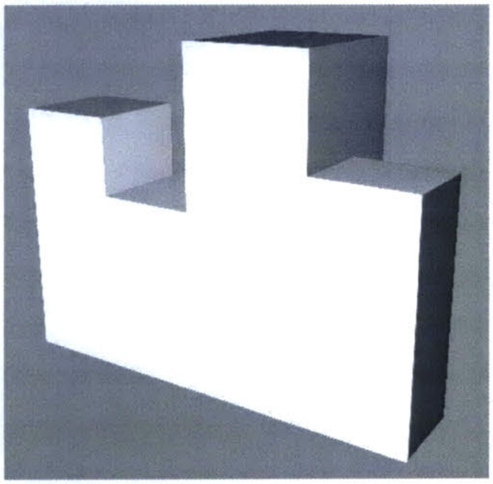


Figure 6.1 Volumetry of the building

To determine a façade's degree of unbalance, three vertical compositional axes are used. The first one, named the 'symmetry axis', is positioned at the mid point of the façade length, and is shown in figure 6.2 in blue. Its x coordinate value is equal to half of the façade length.

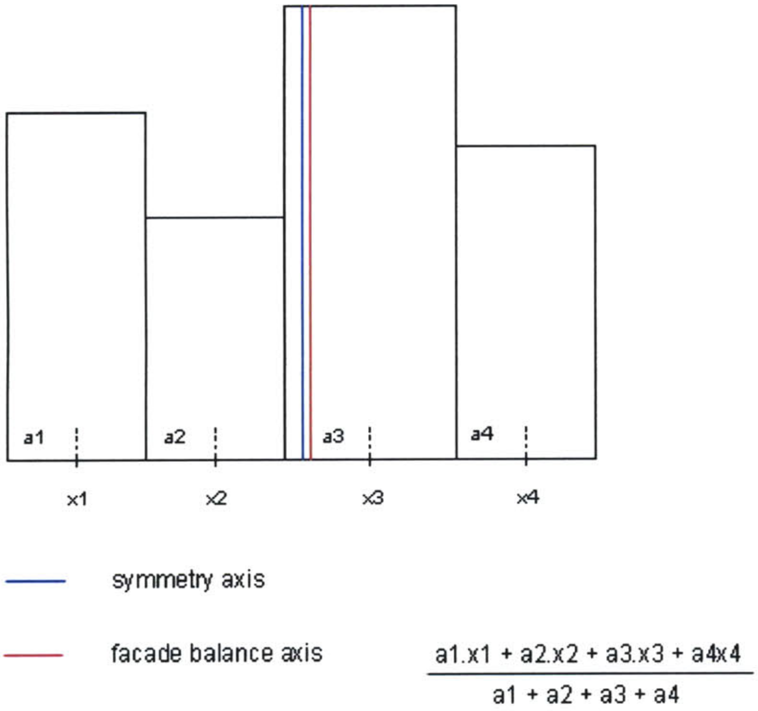


Figure 6.2 Façade profile, showing location of symmetry and façade balance axes. From left to right, we can see modules 1,2,3 and 4

The second one, named the 'façade balance axis', and shown in red in figure 6.2, has its x coordinate calculated using the formula illustrated in the picture. That formula relates the areas of each façade component [a_1, a_2, a_3, and a_4] with the x coordinate of the middle point of that component [x_1, x_2, x_3, and x_4; the x coordinate origin is the left lower corner of the façade]. A façade profile will be in perfect balance if the two axes precisely coincide. In the case presented in the figure, the façade layout is only slightly unbalanced towards the right. The intention was to depart from a mostly balanced façade, and then move to further experiments where the façade would be more out of balance.

Finally, the last axis relates to window balance. It is calculated in much the same way as the façade balance axis, but with the difference that window areas and window mid-point x coordinates are used. If the profile of the façade is already in balance, the objective is to make the window balance axis coincide with the other two. If the façade profile is unbalanced, the windows axis should be used to restore the balance of the whole composition. So, if the façade axis is displaced to the right, as in this case, the window axis should be displaced by an equal amount to the left, so that the sum of the two errors equals to zero. Because windows are centered in the walls, their x value is fixed. So, the mechanism for creating window balance is by manipulating window area. In very general terms, to displace the window axis to the left, larger windows would have to exist towards the left side of the façade. Also, according to Duarte [1997], distance from the symmetry axis increases the weight of a certain mass, a conclusion that can be extracted from the formula used. This means the same window area will have more 'weight' the further away it is from the center of the façade.

This mechanism of using window area to create or restore a façade's compositional balance may generate possible conflicts between the two objective functions. The balance algorithm may push a solution towards a larger window area in some specific location, for compositional purposes. However, that increase in window area may be detrimental in terms of environmental performance. Or the opposite may also be true. The objective of these experiments was thus to test to what degree the two evaluation functions would be in conflict with each other, or if instead it was possible to find satisfactory compromises between the two.

A note about the number of windows the algorithm can manipulate to achieve better performance levels: module 1 has four floors and thus four windows. Module 2 has two floors and two windows. Module 3 has in the ground level a double-height space, but with a single window, and two more floors, totaling three windows. Module 4 has three floors and three windows. All the twelve windows are centered in the respective walls.

As a final modeling issue, all the perimeter walls apart from the façade ones were modeled as adiabatic surfaces, so that heat loss or gain through those walls would not influence the façade design.

The combined objective function or fitness of the solution was obtained by linear combination of the two objective functions, giving equal weights to each of them. For a minimization problem, the applied formula was:
Fitness = (energy consumption) + (degree of unbalance)

A standard GA implementation was used in these experiments, as one of the aims was to gain a sense for a large number of design alternatives. MicroGAs also provide several design alternatives via successive generations or consecutive runs. However, a standard GA gives a more immediate sense of many different solutions by looking only at the final population. A population size of 30 individuals was used. Elitism was also included in this standard GA.

6.3 Experiments with a balanced façade profile

Experiments were carried out for facades facing south and west, for a hot climate [Phoenix]. Figure 6.3 shows the final population of a 100-generations run, for south orientation. The best solution found is in the extreme right of the middle row.

A large variety of solutions can be observed in this final population. However, not all of them represent high-performance solutions. For example, one can predict from the problem setup that the bottom window in the third module should be reasonably large: the space is double height, it is quite deep in relation to the other spaces, and since the façade is facing south, heat gain is not so much a problem as it could be for a west-facing elevation. Nevertheless, we can see that several solutions present very small openings at this particular location. That fact suggests the population has not yet converged to a group of good solutions.

Plotting the search progression of this standard GA confirms this hypothesis [figure 6.4]. The population is not converging, even though the best solution in progressively improving due to

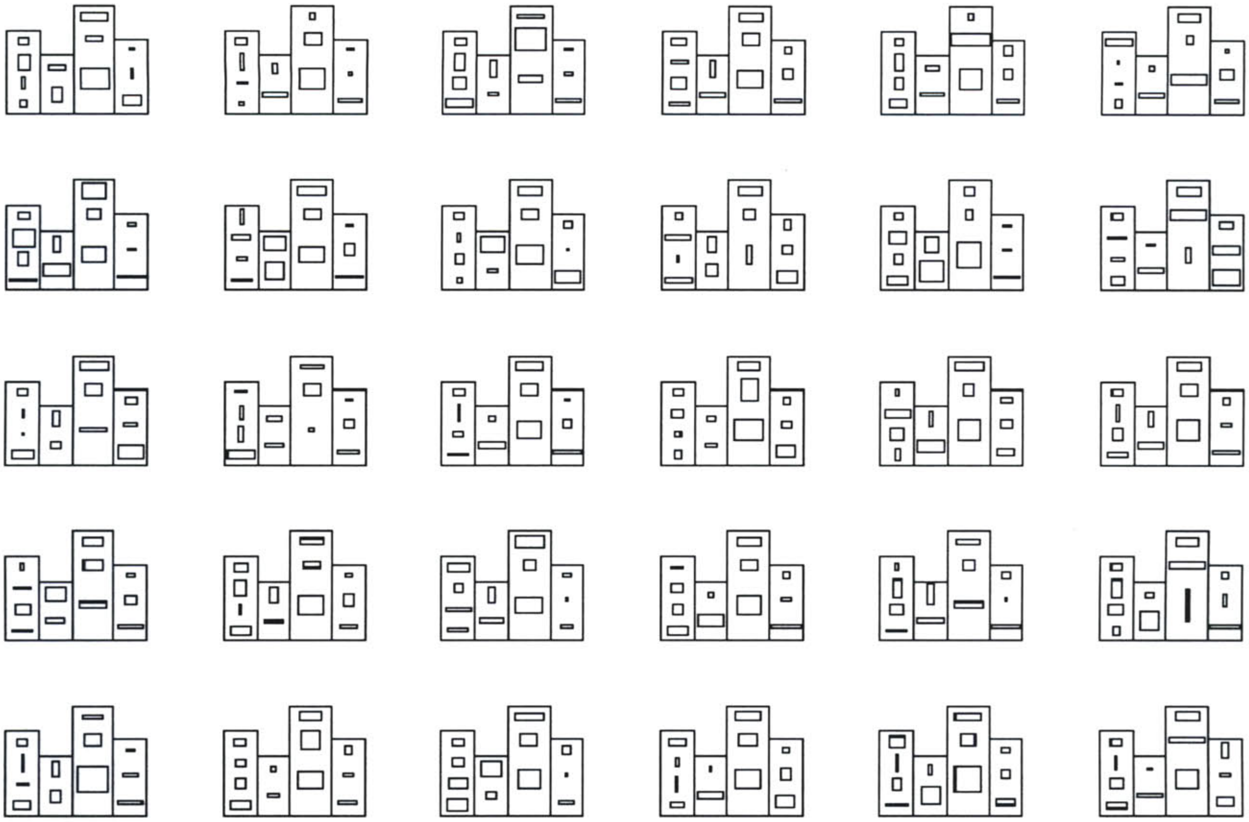


Figure 6.3 Results from a 100-generations run, south orientation. Best solution is in the extreme right of the middle row.

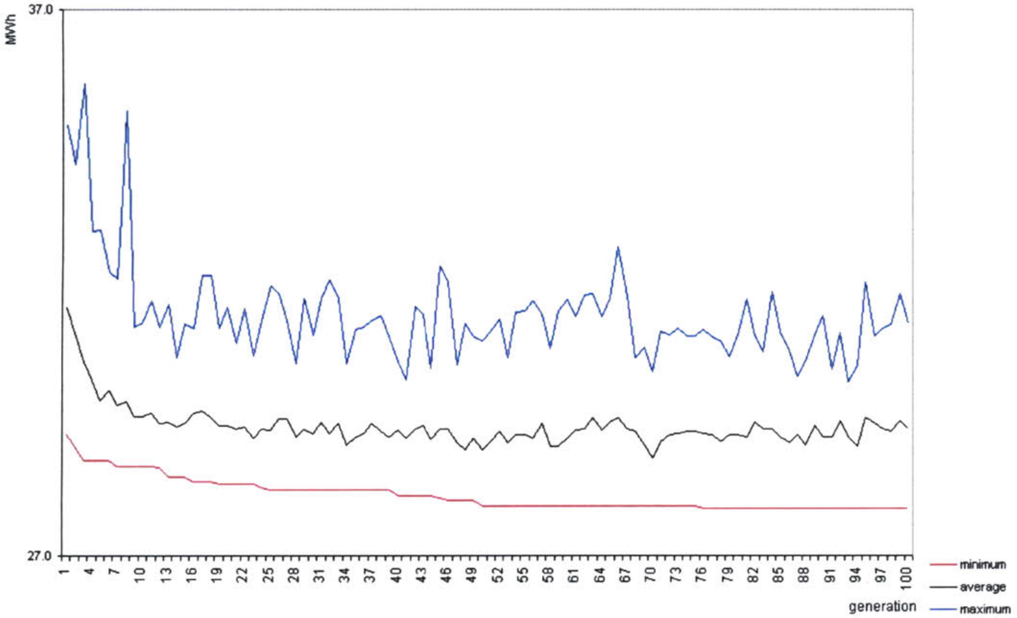


Figure 6.4 Search progression for the standard GA. Although the minimum is steadily decreasing, the average of the population has stabilized after a quick decrease during the first generations.

the elitism strategy. This lack of convergence may be promoted by the fact that the fitness evaluation is based on a linear combination of two quite different measures, which might confound the search mechanism, as solutions that have a good performance in terms of one of the objective functions may have a poor performance when considered from the viewpoint of the other evaluation criterion. Since genetic algorithms use building blocks as a way to collect meaningful information to guide the search, the fact that the same building blocks may imply significant variations in performance according to each objective function may be detrimental to the effectiveness of the search. Another factor may be that the larger population size brings an extra inertia to the search, so a larger number of generations may be required to achieve better convergence.

A related information display is shown in figure 6.5, which plots all the individual points tested by the GS during the course of the search, totalizing 3000 different solutions. In that image is also evident that although the best solutions are saved from generation to generation by elitism, and further improved through crossover with other individuals and mutation, the majority of the remaining points seem to be detached from this minimum trend line.

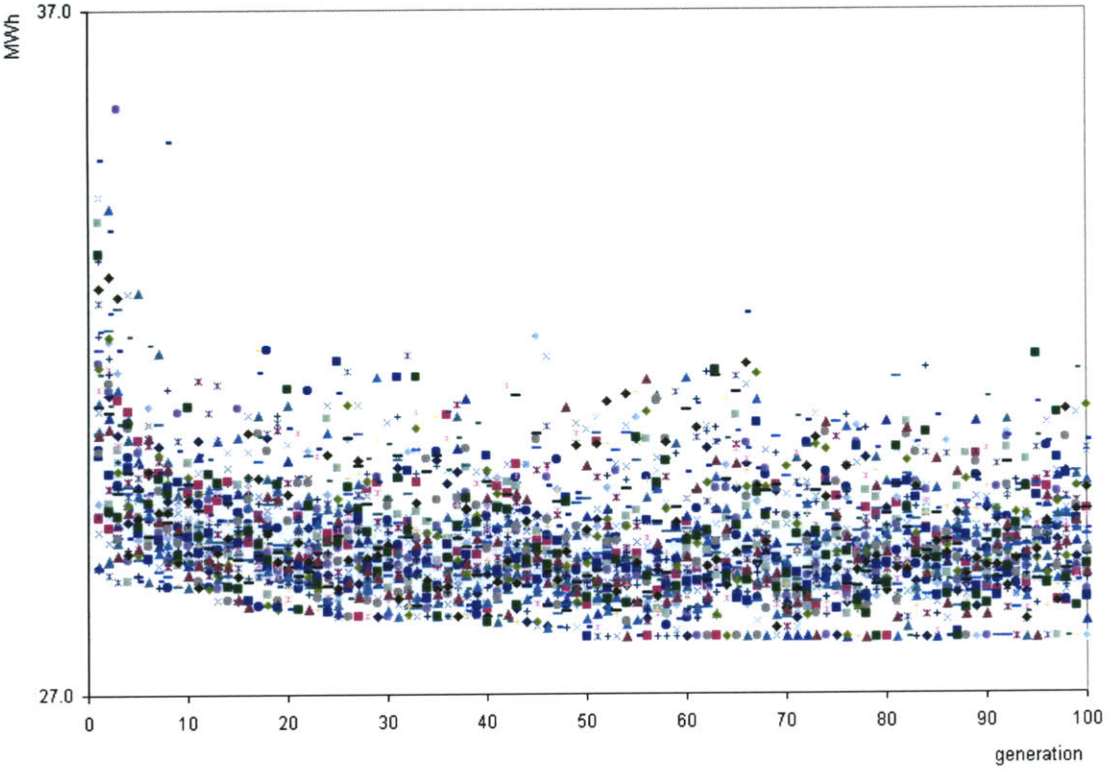
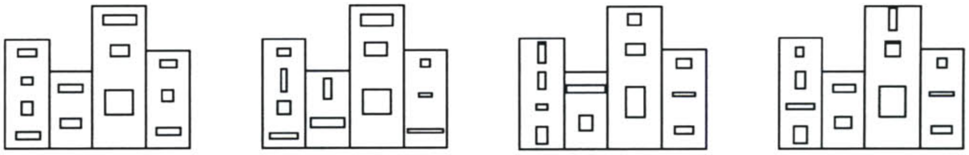


Figure 6.5 All points tested by the GS during the search [3000]

Figure 6.6 shows the results from the experiments done for south orientation. The following procedure was used: first, the system was run using energy consumption as the only objective function [for easier reference, this was called the 'functional' algorithm, and is represented in the images by the letter F]. Then, the degree of unbalanced was calculated using the formula provided in figure 6.2 plus the formula for the window compositional balance, and the resulting F+B value calculated [B refers to the balance algorithm]. Those results are shown in the first column of figure 6.6. The numbers in red always represents the values used as objective functions in that experiment, and the numbers in black represent values manually calculated.



	F first	F+B over results of F	F+B first	F over results of F+B
F : FUNCTIONAL	27.85	28.06	28.13	27.90
B : BALANCE	1.66	0.42	0.11	0.05
F + B	29.51	28.48	28.24	27.95
MAXIMUM	42.19	41.28	44.15	42.09
% > MINIMUM	51%	47%	57%	51%

Figure 6.6 South experiments, showing different sequences of F and F+B experiments.

The GS, using energy consumption alone, generated a solution that had a relatively high degree of unbalance, 1.66 feet, corresponding in meters to 0.5m, in a façade about 20m long. The numbers on the bottom show the maximum function value using energy alone, showing that in this case there is a high variability of objective function values with façade design, since no core areas exist. The worst solution between all those searched had an energy consumption about 51% higher than the best found.

The second column shows the results of running the combined objective function algorithm [F+B] over the results obtained by F only. This was done by using as the initial population for the second experiment the final population from the first experiment. The combined results improved over the previous one, mostly by lowering the degree of unbalance, even though energy performance became slightly higher. The more successful strategy seemed to be to run first F+B, and then F only to further improve on the energy performance aspects. This yielded the best results of these first experiments, with a result that has low consumption and is almost in perfect balance.

In order to assess to what degree could the two objective functions actually be in conflict, a linear programming solver was used to find the design solution that would produce the largest degree of unbalance. The balance objective function is linear, since the x values are fixed, so it was possible to use linear programming [Excel© solver] to get this information. The corresponding façade design solution was then fed into DOE2, to determine its energy performance. Figure 6.7 shows that although the energy performance of the worst balance solution is high, it is by no means the worst possible case. Interestingly, the worst energy-wise result found by the GS is actually almost in balance, only slightly worst than the result from the first experiment. This shows there is no direct relation between good energy performance and a balanced façade design, what makes the results found by the GS more interesting, since it was able to find good compromises between the two objective functions.

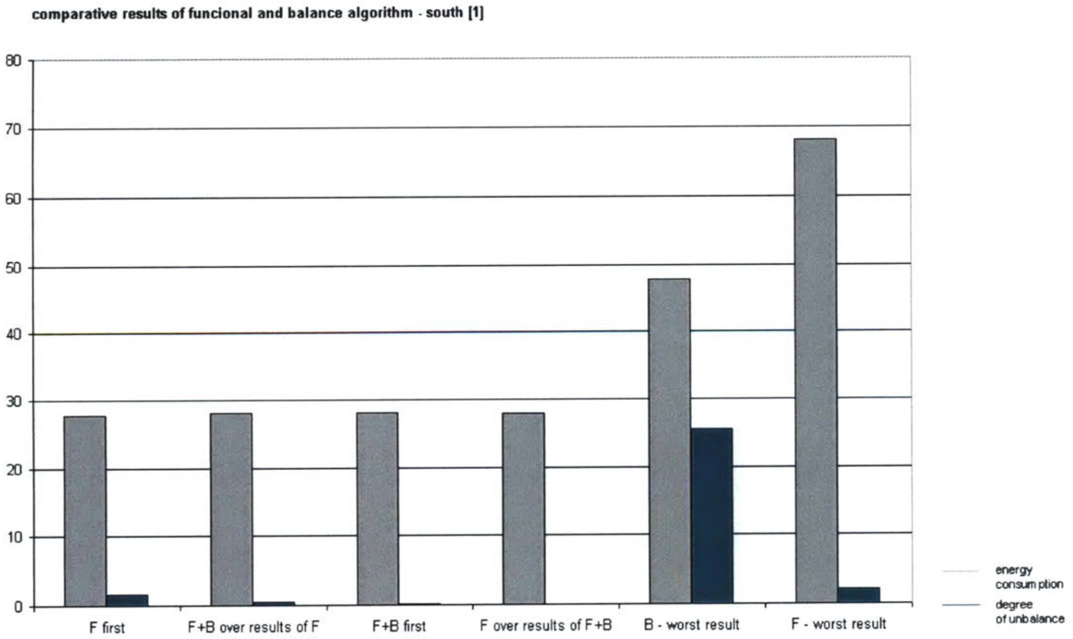
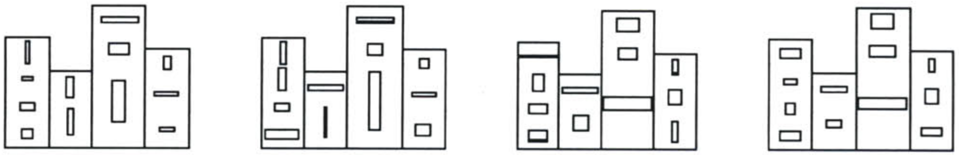


Figure 6.7 Comparing energy consumption and degree of unbalance to several south façade solutions, including worst-case scenarios [F is the functional algorithm, B is the balance algorithm]. Y axis coordinates are dimensionless, since they combine different units.

Figures 6.8 and 6.9 show a second, similar experiment done for the south orientation. Here, the results from the first experiment about the best sequence of objective function to use are inverted, which leads to the conclusion that the order used to run the algorithms seems to be unimportant, and probably just running the two objective functions simultaneously is satisfactory enough.



	F first	F+B over results of F	F+B first	F over results of F+B
F : FUNCTIONAL	28.13	28.11	28.11	27.99
B: BALANCE	1.74	0.56	0.62	2.50
F + B	29.87	28.67	28.73	30.49
MAXIMUM	43.53	42.19	43.73	41.37
% > MINIMUM	55%	50%	56%	48%

Figure 6.8 South experiments [2], showing different sequences of F and F+B experiments.

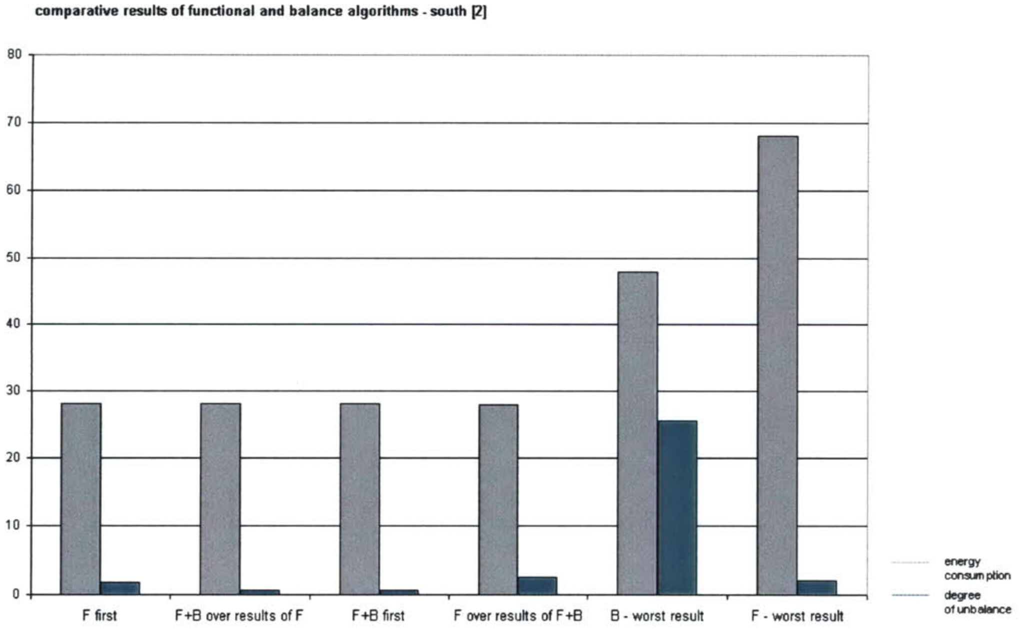
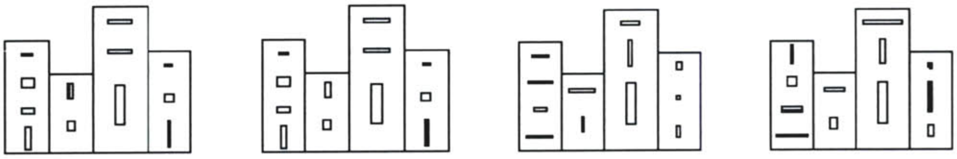


Figure 6.9 Comparing energy consumption and degree of unbalance to several south façade solutions, including worst-case scenarios [F is the functional algorithm, B is the balance algorithm]. Y axis coordinates are dimensionless, since they combine different units.

Finally, figures 6.10 and 6.11 show similar experiments but now for west orientation. The main difference to be noted is that, in general, window sizes are all smaller, as expected in such a rigorous climate in summer. Energy consumption levels are also higher than for south, as expected. Finally, a façade with the lowest energy performance and perfect balance was found during the F+B run. In general, the overall degree of unbalance is close to zero.



	F first	F+B over results of F	F+B first	F over results of F+B
F : FUNCTIONAL	31.24	31.24	31.39	31.24
B : BALANCE	0.00	0.00	0.06	0.15
F + B	31.24	31.24	31.45	31.39
MAXIMUM	46.74	49.30	49.15	46.11
% > MINIMUM	49%	58%	56%	47%

Figure 6.10 West experiments, showing different sequences of F and F+B experiments.

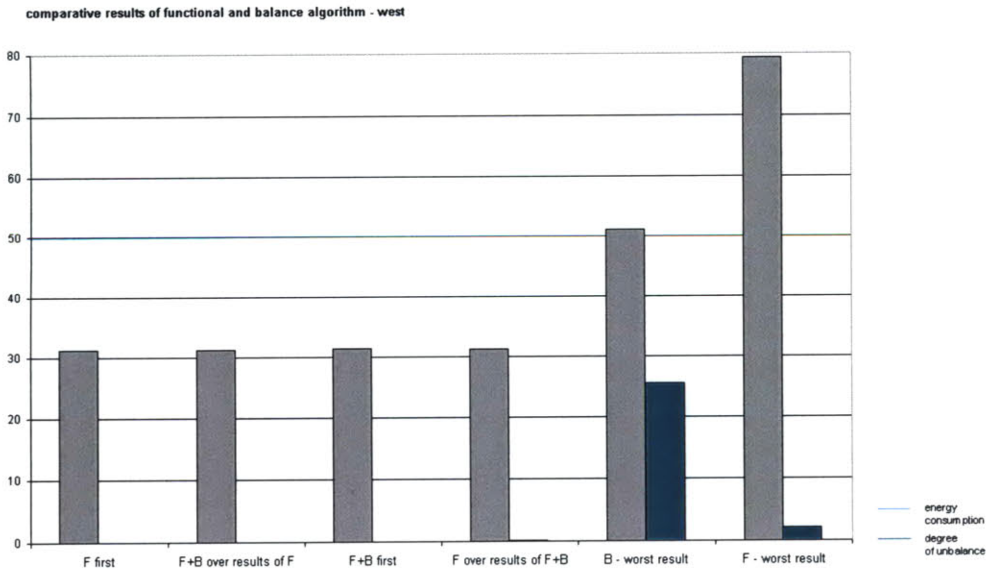


Figure 6.11 Comparing energy consumption and degree of unbalance to several west façade solutions, including worst-case scenarios [F is the functional algorithm, B is the balance algorithm]. Y axis coordinates are dimensionless, since they combine different units.

6.4 Experiments with an unbalanced façade profile

An unbalanced façade was created by removing the two top floors from module 4. The façade surface area toward the right side of the symmetry axis was thus reduced, and the façade balance axis moved towards the left of the symmetry axis. To solve that unbalance in the composition, the GS would have to place larger window areas toward the right side of the central axis, so that the larger windows would counteract the reduced façade area.

This situation was tested for two climates, Phoenix and Chicago [figure 6.12]. As expected, the GS did use larger window sizes towards the right side of the symmetry axis. However, for the hot climate the strategy seems to be to use smaller window sizes, making those on the left side of the building much smaller than in the previous experiments [which were also done for Phoenix]. The window sizes to the right of the building closely resemble those of the balanced façade experiments, which suggests that the GS did not increase window sizes towards the right but instead decreased them on the left side, probably due to the large heat gain penalties incurred if large openings were used.

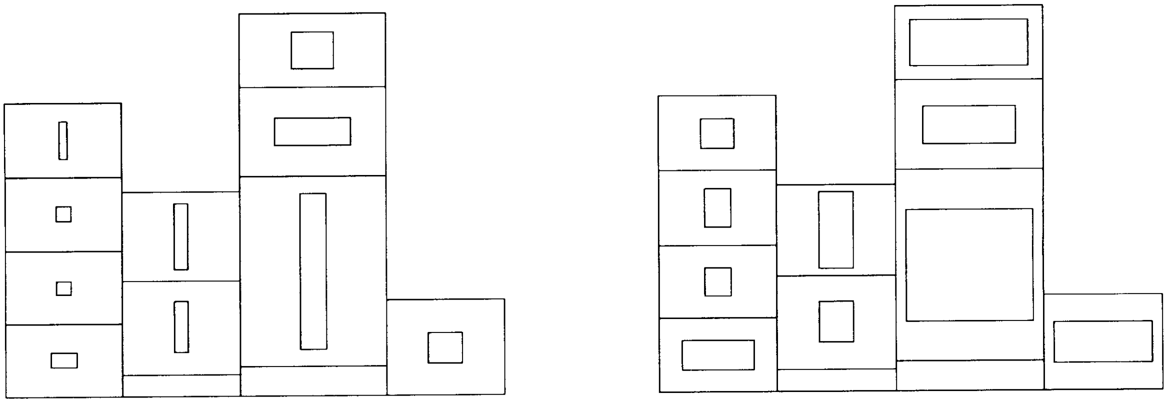


Figure 6.12 Unbalanced façade results, Phoenix [right] and Chicago [left]

On the contrary, for Chicago the GS did use quite large fenestration sizes towards the right side of the building, since in this geographical location south-facing windows mainly represent useful solar gains during the winter. Left-side openings are smaller but still generously sized.

6.5 Possible extensions of the work

Some potentially interesting extensions of this work have been identified. One would be having the GS generate façade profiles instead of generating only the fenestration elements for a previously defined façade. Façade balance could be one of the objective functions given to the algorithm. Other possibility would be that the GS would generate random unbalanced profiles, and would then need to use window area as a posterior mechanism to achieve balance.

Another possible extension would be the generation of windows that are not centered in the walls. However, this would require the use of dynamic constraints, which have not yet been implemented in the GS. A detailed discussion of the implications of using dynamic constraints will be provided in chapter 9, which deals with shape generation.

Finally, Duarte's work [1997] describes the inclusion of color as an additional element to take into account when predicting façade compositional balance. This is achieved by including a Color Weight Index in the original vertical balance equation, which would then read:

$$x = \frac{a_1.x_1.w_1 + a_2..x_2.w_2 + \ldots + a_n.x_n.w_n}{a_1.w_1 + a_2.w_2 + \ldots + a_n.\ w_n}$$

where

x is the x coordinate of the compositional axis

a_n is the area of shape n

x_n is the distance of shape n to the origin of the façade [left lower corner of façade]

w_n is the Color Weight Index

The Color Weight Index measures the visual weight of shape n's color against the color of its background, and is the difference between the weight index of both colors against white. The Color Weight Index of a color against white is given by:

$$W_b = \frac{65535 - y}{65535}$$

where y is obtained from an equation (Pennebaker, 1993) that transforms colors into a grey tone by calculating their degrees of whiteness according to their RGB values:

$$y = 0.299\ R + 0.587\ G + 0.114\ B$$

This framework for including color into the balance algorithm was developed by Duarte, and since no experiments have yet been performed, we cannot ensure the meaningfulness of any outcomes resulting from its application. Nevertheless, it is suggested as a potentially interesting extension because façade color would also have an impact on the wall's solar absorptivity, and subsequently on the building's energy consumption. It represents thus another variable that would be meaningful in terms of the two objective functions dealt with in this chapter, and could add another appealing dimension to the work.

CHAPTER 7 Generative System applied to Siza's School of Architecture at Oporto

7.1 Introduction and objectives

This chapter approaches the integration of architectural design intentions into the Generative System. In previous chapters, the GS was tested and run in very simple building layouts, almost schematic, so that results could be easily interpreted. In this chapter, the aim was to place the GS in the realm of an actual piece of high-quality architecture, with all its complexities and interaction of elements. Álvaro Siza's School of Architecture at Oporto was used as a test bed, since its clear but complex composition rules provided an excellent framework to work upon, in order to achieve the proposed objectives. However, due to the large dimension of the project, the study focused solely on one of the studio towers [tower H, see figures 7.1 and 7.2], which proved nonetheless to be diverse and rich enough to pose challenging problems to the GS and to provide interesting results from an architectural perspective.

The objectives of this chapter are thus twofold: first, to study the incorporation of language constraints into the generative design system, so that solutions generated are within certain design intentions. Second, to examine the generative system results from the perspective of the existing design by Álvaro Siza, an architect well known for his control of daylight, and to analyze to what extent the inclusion of factors other than light [like the thermal behavior of the building] could make solutions follow a different path.

Some further experiments were also performed, placing the existing building in different geographical locations, to test the GS capability to adapt solutions to distinct climatic conditions, within similar language constraints. A note should be added that the exercise of placing the building in different locations is purely academic, since Álvaro Siza would never use the same approach for designing a building in such different geographical and cultural environments. While the experiments for Oporto illustrate a process that could have taken place in Siza's design, the other examples are a test of the generative system but do not represent an appropriate architectural design process.

7.2 Description of the existing buildings

The School of Architecture at Oporto, Portugal, designed and constructed from 1984 to 1996 by Álvaro Siza, encompasses a period of time when the project faced several changes in programmatic needs due to the establishment of a new curriculum at the school. During this period, Siza's sketches and concepts evolved in order to create a diversity of spatial configurations. These architectural relations can be seen in the final organization of the buildings, such as long corridors that unite, under the entrance level, the four towers where the architecture studios are housed, or the exterior communal space that visually and physically relates the towers with the auditorium, library and administrative services.

Besides these major architectural intentions that are clearly expressed in the overall plan of the school, other elements acted as anchor or seed points to the development of the project. These include the extremely rich morphological condition of the site, the southern part of the Quinta da Póvoa, with a steep slope facing the Douro River and the delicate landscape of Porto's neighboring town, Vila Nova de Gaia; and the necessity of connecting the new complex with a former building also designed by Siza, the Carlos Ramos Pavilion, situated northeast of the new agora. These factors, together with the distribution of academic activities among different architectural units [studios and faculty rooms in towers E, F, G and H; and the library, auditoriums and administrative services in the northernmost buildings] informed a cultural process that enabled Siza to create of a unique piece of architecture. For a more detailed analysis, see Testa (1994, 1999).

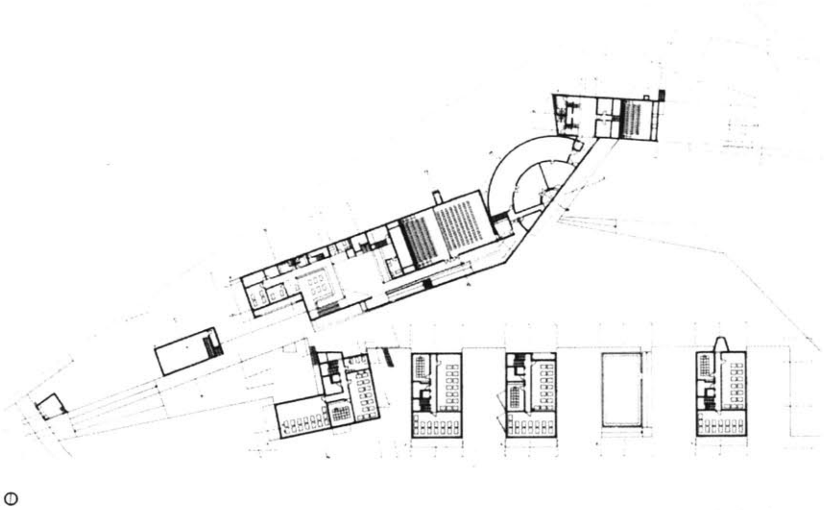


Figure 7.1 School of Architecture plan. Tower H is on the right lower corner [orientation: 8° east of north]

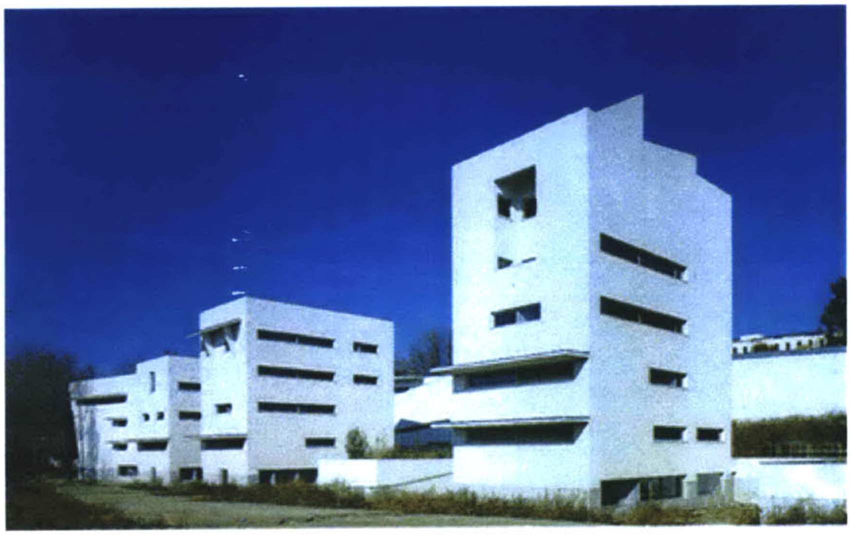


Figure 7.2 Southeast view of the four studio towers. Tower H is on the foreground

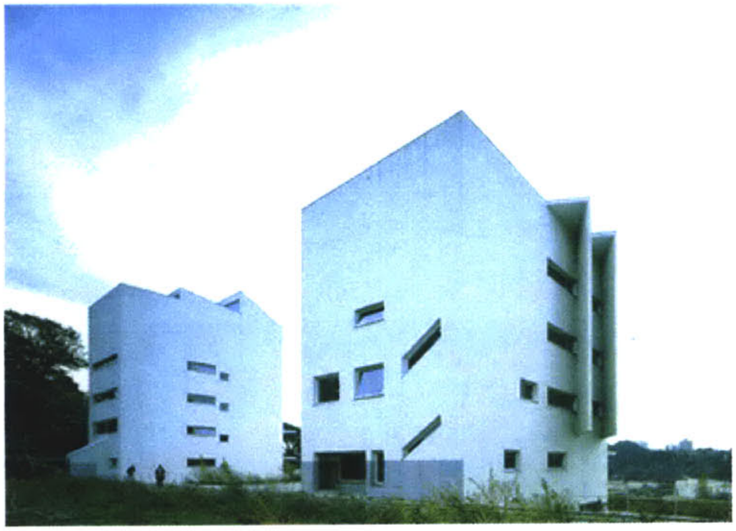


Figure 7.3 Northwest view of towers H and G. Tower H is in the background. Tower G is in the foreground, showing the large vertical fins that protect the west-facing studio windows from direct sun.

7.3 Analysis of studio rooms orientations

It is interesting to note that although Siza had a similar program to the 4 studio towers [towers H, G and F all have the same dimensions in plan, 10.88 x 17.88 m], their internal spaces diverge and give rise to rich and unexpected spatial configurations.

A preliminary analysis integrates the four towers in terms of their spatial distribution, mainly of the studio teaching rooms. As an initial synthesis of the programmatic distributions within all the towers, the majority of studio rooms are oriented towards east [14 studios rooms]. Seven studios are oriented towards north, six to south and only three face west. Figures 7.4 and 7.5 show this spatial distribution analysis, both by building and by orientation.

Except for Tower G, there are no studio rooms facing west, an orientation that in all towers has very short façade openings and mainly houses functional spaces such as stairs, elevators, and services. In tower G, Siza used large vertical fins that effectively block direct sun into the studios [see figure 7.3] but have the disadvantage of blocking the view out too.

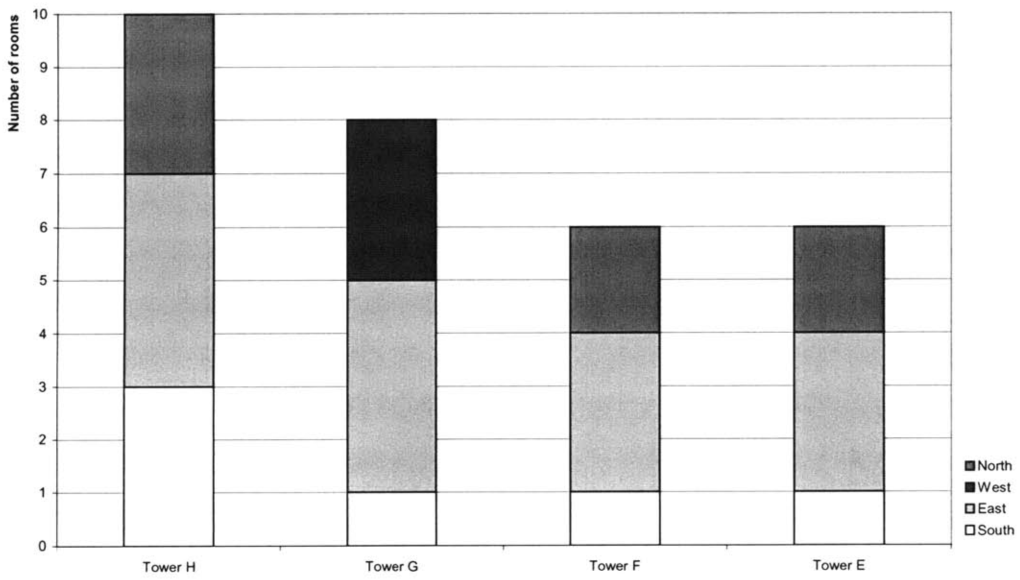


Figure 7.4 Studio rooms orientations according to building

7.4 Justification of the choice of tower H for this study

Tower H was chosen for its rich spatial configurations and use of a variety of architectural light sources: fenestrations of different proportions and sizes facing distinct orientations [some including overhangs], zenithal light as roof monitors in the top floor, and a loggia in the south façade.

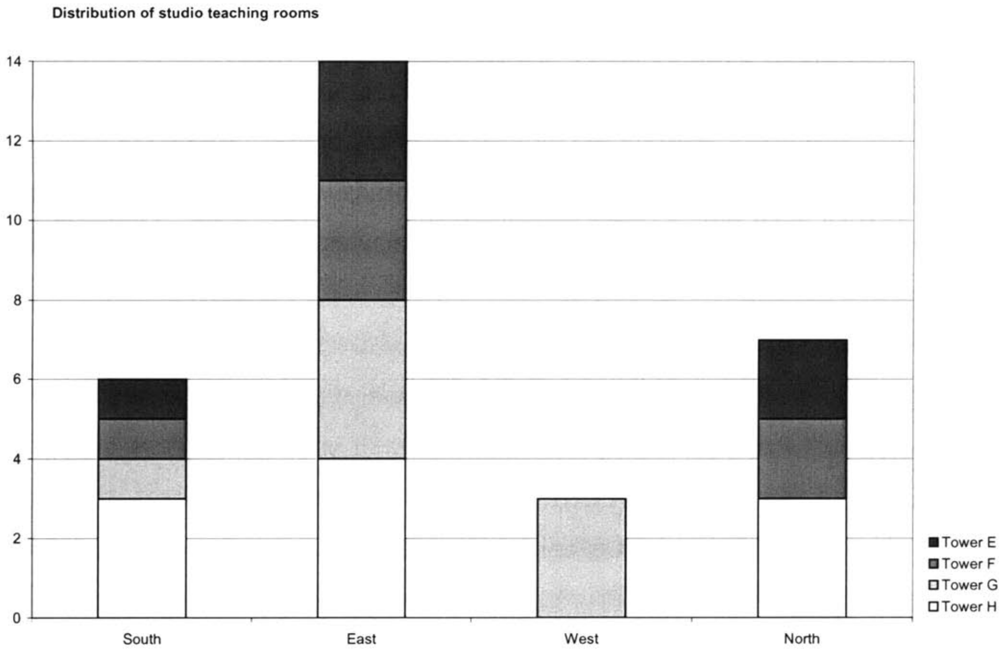


Figure 7.5 Distribution of studio rooms according to orientation. East clearly predominates.

From a computational perspective, tower H also presents some challenging features. The internal relations between the different spaces and their light sources give rise to a multiplicity of interactions that are hard to predict and make the resort to computational analysis an interesting option. Some examples are the interactions between the roof monitors and the loggia in the 6th floor, or cases where studio rooms have more than a window, facing different orientations.

The fact that Tower H mainly houses studio teaching rooms makes a strong case for the careful control of natural light in order to maintain adequate daylighting levels for drawing and model making tasks while precluding direct sun over the drafting tables and excessive solar gains in the rooms

Figure 7.6 shows several floor plans of tower H. Studio rooms face south, east or north, and vary their spatial disposition from floor to floor. Services are on the west side of the building. Two aspects are worthwhile noticing in these plans. One is the recurrent use of an uncommon detail, where internal walls perpendicularly hit an external window. This can be seen in the 1st floor plan, towards the east, where an apparently continuous large strip window is interiorly divided between two different rooms. The same detail occurs in the 4th floor, in the south façade, and in the 5th floor in the south loggia window. It also happens in other less relevant situations, like restrooms.

This particular architectural detail will prove to be important at later stages of the work, when the compositional rules and constraints for the facades are investigated. Some windows that appear to be a single, continuous element when viewed from the outside, are actually two different openings when viewed from the inside. This introduces new constraints into the design that have to be considered when encoding the existing rules.

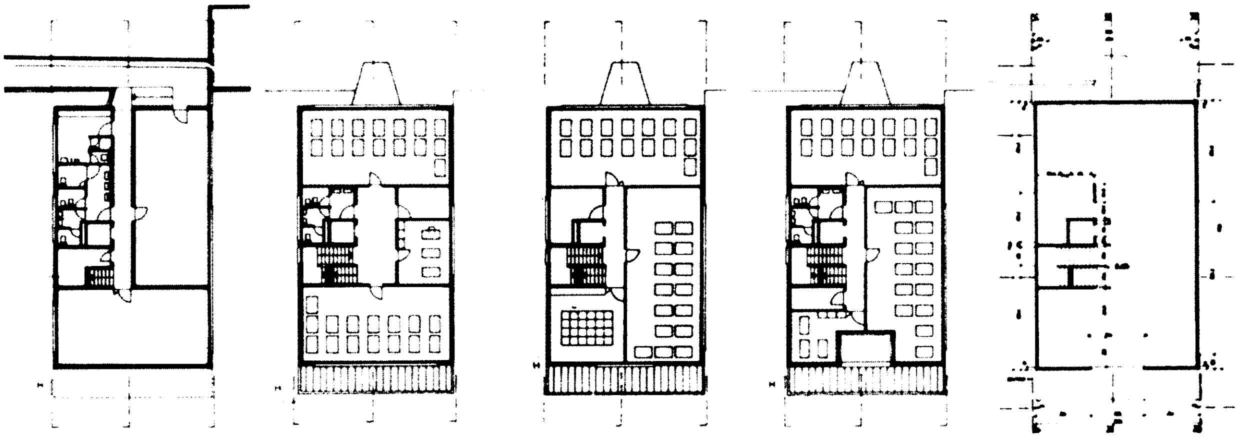


Figure 7.6 Some of the floor plans of Tower H [from left to right, floors number 1, 3, 4, 5, and 6 (top)]

The other important point to notice is that the top floor [6th floor] has a very different configuration in relation to all the others. It is basically a single space that occupies the entire floor, which has blank walls towards all orientations but south, where it is lit by the loggia only [even though the loggia itself has both south, east and west openings]. Daylight enters this space mainly through roof-top monitors that take advantage of the tilted, sawtooth roofs, to create top-level, north-facing openings. The roof has different inclinations, so the size of the openings is variable too. The larger roof-top monitor is closer to the south façade, and the small one is placed towards the north end of the space [see figures 7.2 and 7.3 for a visualization of the roof configurations].

7.5 Simulation method for tower H

For this study, building geometry, space layout and construction materials were left as in Siza's original project, and the algorithm acted solely upon the elevations design. For the existing building layout, the software generates a population of façade solutions that take into account the use of daylighting in the space, the subsequent use of artificial lighting, and the energy consumed to heat and cool the building. Solutions that make maximum use of natural lighting are preferred, but the control of heat gains and losses introduces a balance point to be achieved. If maximum amounts of daylighting were the single criterion, maximum opening sizes allowed within language constraints would always be the best solution. However, above a certain level more natural light will bring no benefits and will carry with it disadvantages such as high thermal gains or losses that will need to be offset by mechanical systems. It is this elusive balance point that the Generative System tries to locate.

For each individual space in Tower H, virtual photocells were placed at chosen locations [typically the furthest points from the windows where a certain light level is to be achieved] and desired illuminances values were specified according to the type of occupation and tasks performed. Generally, 500 lux were used for studios and other working spaces and 300 or 150 lux for service areas. The remaining steps are as described in previous chapters.

7.6 Description of existing design | encoding rules

Due to the need of finding elements that would lead to the development of a method to understand and encode Siza's design intentions [rules], a visit to the School of Architecture took place in January 2000. The analysis of the drawings and the visit to the building allowed inferring design rules that we consider to be applicable to the existing elevations. Those rules relate both to compositional axes of the facades and to general proportions of the openings. In tower H, different rules seem to apply to each elevation, while maintaining a strong coherence in the overall design of the building and in the relations with internal spaces [for example, long horizontal windows are always used in the architecture studios].

For each of the existing elevations, a brief description of its main elements and rules of composition is provided below, together with the existing elevation drawings.

7.6.1 South elevation

The south façade is dominated by a vertical symmetry axis. It is blank in the first floor, and has long strip horizontal windows shaded by 2 meter deep overhangs in floors 2 and 3, where south-facing studios exist. Although in the first floor there is also a south-facing room behind the façade, it is only illuminated by east and west facing openings in its ends. The 4th floor exhibits very small windows, the right side one working as a top-end opening to a east facing studio illuminated mainly through the east, and the other, symmetrical to this one, being the only opening in a small meeting room [see plan in figure 7.6]. This symmetry is based on the fact that an internal wall divides internally the window, as described in the previous section. The 5th and 6th floors contain a double-height loggia recessed into the façade plane. In the 5th floor, there are glass doors in all three sides of the loggia, all with the same height, but varying in width. The south door is again divided internally by a wall, which separates the studio from the circulation corridor. In the 6th floor there are only regular windows, vertically aligned with those in the 5th floor [that is, width is constrained to be the same], and all with the same height.

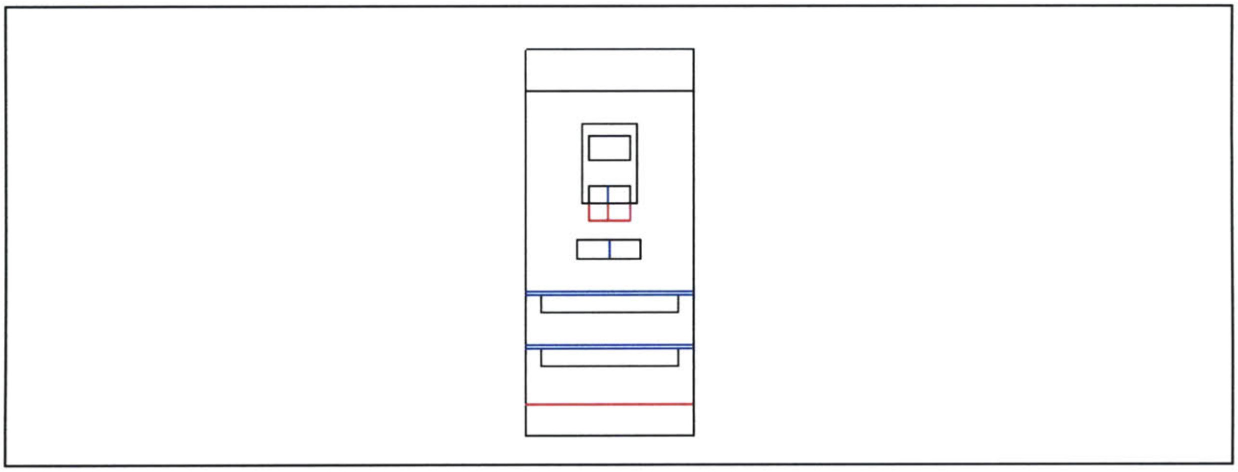


Figure 7.7 Existing south elevation

7.6.2 East elevation

The east façade has two main vertical composition axes. The first axis constitutes the vertical alignment of the right ends of the windows from floors 1 to 5. The second axis is the vertical alignment of windows in floors 1 and 2 on their leftmost ends. While windows in floors 2, 3, 4 and 5 are long strip windows, the windows in the first floor are taller and with a less elongated proportion that relates to the stone embasement. The sixth floor façade is blank.

Some points worth noticing are that the blue vertical lines on some of the fenestration elements represent internal walls that hit them internally. The two windows in the 2nd floor belong to a studio, which is the only one that does not have a single, horizontal strip window. This room is the only east-facing studio that is lit solely from the east. The long strip windows in the 4th and

5th floors are part of studio rooms too, but which have the particularity of also being lit from other directions. Thus, the 4th floor room has also a south facing window, which is 'shared' with the adjoining small meeting room. The 5th floor room has also some loggia windows, which are in this case subject to all the compositional rules of the loggia. This analysis starts to show a complex architectural scheme, where elements cannot viewed as independent entities, but instead they are related by a complex set of rules that relate both to elevation design and also to the floor plans and to the relations between the different spaces, shedding some light into the sophisticated architecture design process created by Siza. The apparent simplicity of the final building hides in it a complex network of interactions that need to be understood so that they can eventually be successfully encoded into the Generative System. The vertical compositional axes and window proportion rules by themselves would be unable to account for all these other interactions present in Siza's design.

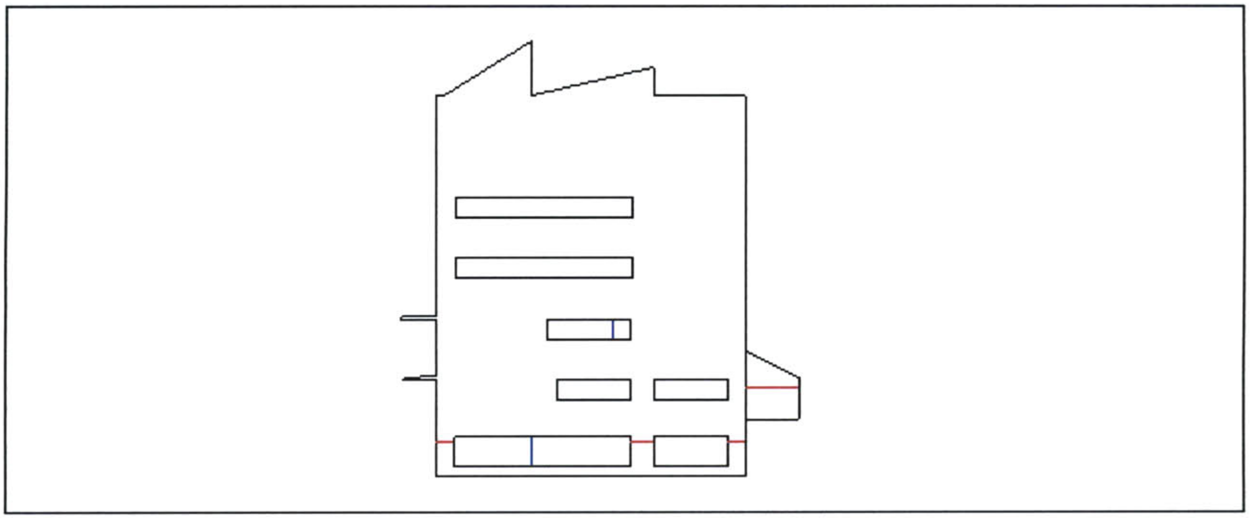


Figure 7.8 Existing east elevation

7.6.3 North elevation

The north façade is again dominated by a vertical symmetry axis. It is the only elevation with just five floors, since it starts in the second floor due to the connecting tunnel between the several studio towers. It has an attached module that constitutes the entrance to the building at the second floor. The only asymmetric element in the composition is a small window in the second floor that illuminates a small corner room nest to the entrance. The other windows are long strips that illuminate the studios. The two roof monitors can also be seen in the north elevation drawing below. The taller roof monitor, in the southern end on the roof, appears in the drawing behind the smaller one.

The north elevation does not establish direct interactions with the other elevations. All the studios facing north are absolutely similar in size and have only a north-facing opening. However, and probably to generate some diversity in the elevation, Siza created a melodic variation in the height of the three north studio openings, as can be seen in figure 7.9. The roof monitors, on the contrary, already establish relations to other elements of the façade design, not explicitly by composition axes or alignments, but implicitly, by interacting with the loggia openings in attempting to provide adequate daylight to the top floor. Since the design of the loggias' 6th floor has an impact on the loggias' 5th floor, and that an impact on the 5th floor east façade, and that an impact of the 4th floor east façade, and that an impact of the 4th floor south façade, etc., this forms a long stream of interactions where the design solution for one specific element is not autonomous, but instead impinges on a large number of other elements.

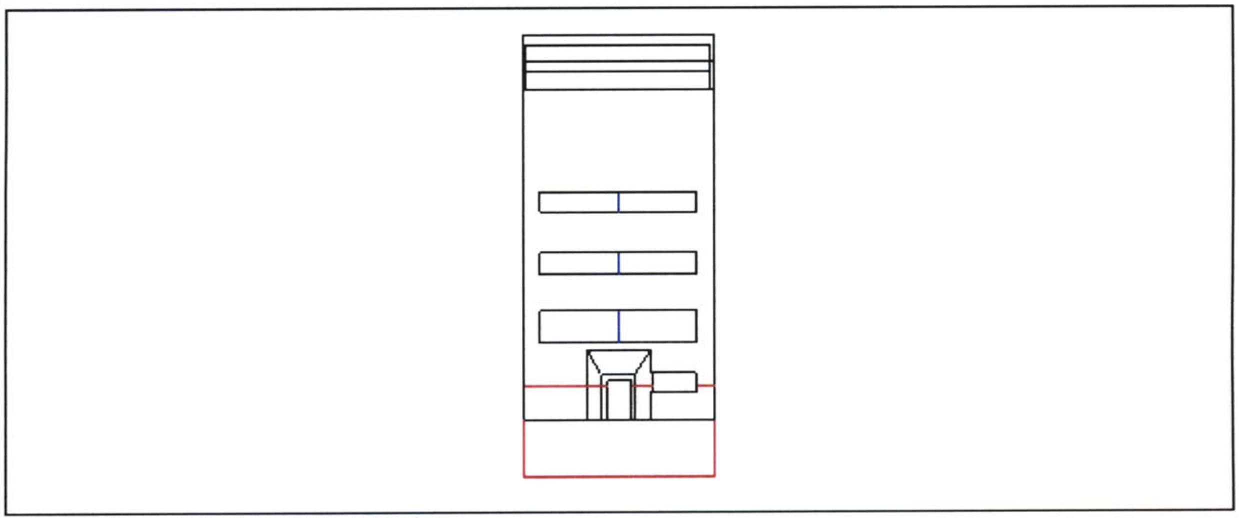


Figure 7.9 Existing north elevation

7.6.4 West orientation

The west façade is mostly blank and has small openings from the second to the fifth floor. The square windows illuminate the stairs, and horizontal ones the restrooms. The two are placed at different levels, due to the stair footing. The first floor has larger windows that open to different types of spaces, like meeting rooms and model-making spaces, and that follow the proportions of the 1st floor windows in the east façade.

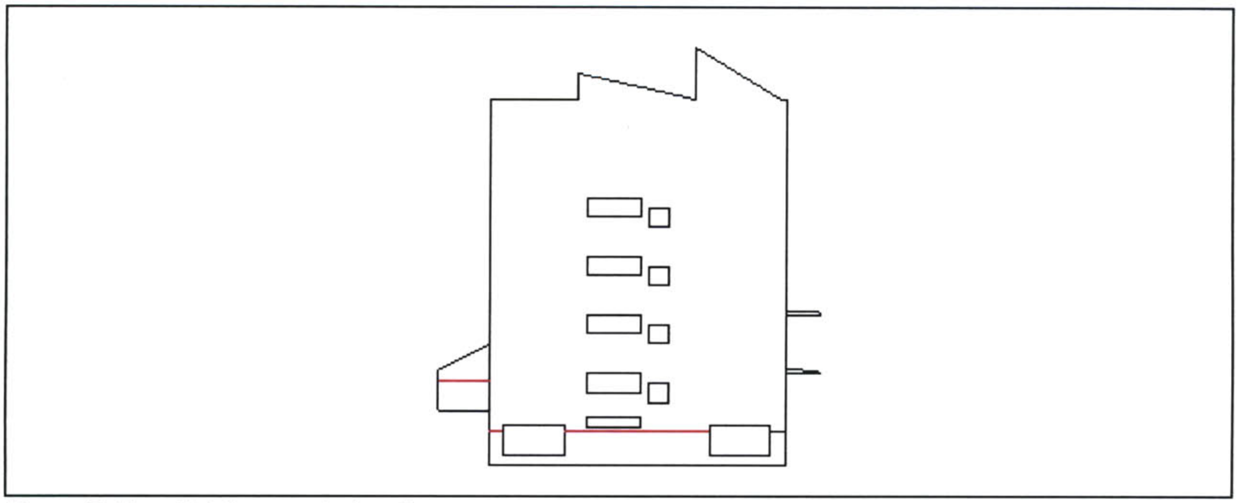


Figure 7.10 Existing west elevation

7.7 Generation of constraints

This interpretation of existing design rules was followed by the determination of areas of search for the generative system, implemented both as constraints to the algorithm and as prescriptive relations between different types of variables. The search areas are bounded by maximum and minimum dimensions the openings can assume, and those limits were made distant enough to allow for a significant search space that could promote the emergence of a rich variety of solutions. Other constraints implement the compositional axes determined during the analysis stage.

In figures 7.11, 7.12, 7.13 and 7.14, the left image represents the constraints applied, and the right image the existing design. Compositional axes are represented by the red lines on the constraints image. For each opening, the smaller area represents the lower bounds to the algorithm, and the larger area the upper bounds. These set of constraints are proposed as being able to control the generation of solutions within certain architectural intentions that we relate to Siza's design. It should be noted that the constraints were always applied in a way that the existing design was included in their search space. That is, in all the cases, Siza's design could emerge from the GS solutions.

7.7.1 South elevation constraints

For the south façade, the basic symmetry axis was respected. For the strip windows in the 2nd and 3rd floor, the windows were constrained to be centered in the wall, and the maximum window height was set equal to the minimum window width, so that the appearance of vertical windows would be excluded, as that was considered not to comply with Siza's design intentions. This way only horizontal windows are allowed, with the extreme solution being a square. The maximum window width is equal to the wall width, minus the thickness of the exterior walls. Finally, The 2nd and 3rd floor windows are constrained to be similar.

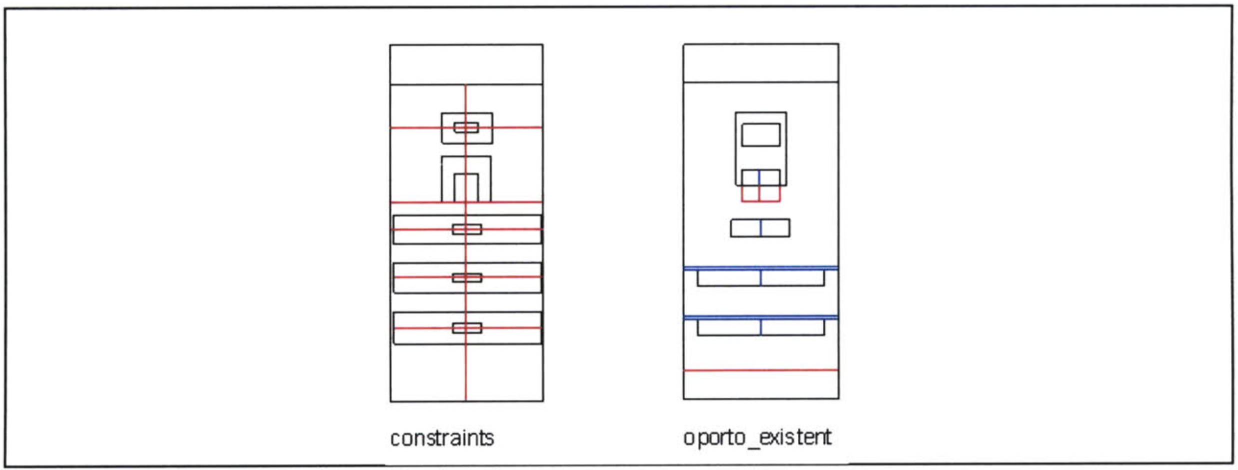


Figure 7.11 Constraints for south elevation

For the 4th floor, an extra requirement had to be dealt with. Since the window is shared by two adjacent rooms, and separated by an internal wall as explained before, the constraints impose that the two halves of the window are exactly the same, even if they serve quite different spaces [see plans in figure 7.6]. The windows insertion points [their left lower corner] would also have to be calculated in such a way that the two halves would meet at the central symmetry axis. The overall dimensions are similar to those in the lower floors.

For the loggia, a different set of constraints is used. The 5th floor opening are doors, so minimum constraints are defined by functional requirements, that is, minimum height and width for a door. The height of all openings was constrained to be the same, following the existing design, and the width of the east and west openings was also the same, as they mirror each other. Maximum dimensions are constrained by the respective wall sizes. The y coordinate is always 0, so that the openings always start at ground level. As an internal wall also exists behind the south wall, the same case as exists in the forth floor, in terms of correctly calculating the x coordinate. The 6th floor loggia windows were constrained to have the same width as those of the 5th floor, again following Siza's existing design. Their height could vary within the constraints set, but had to be similar for all the three windows.

It should be noted that these constraints that make some variables dependent on the characteristics of other elements do not imply any kind of explicitly hierarchy. Since the GS performs a whole building simulation for each configuration, what happens is that all these dependencies are being dynamically and simultaneously evaluated, and the most favorable combination of variables will emerge from this dynamic process. Any type of dominance of one element over the other will only appear if that causes a better overall performance in terms of whole building appraisal. We will see during results analysis that some variables do indeed 'dominate' others, but that is not enforced by the constraints described here.

7.7.2 East elevation constraints

For the east façade, other aspects were included in addition to the two compositional axes mentioned before. In the first floor, some doubts arose about the rules that should apply to the middle window. Its right end is conditioned to align by the vertical axis of the façade. However, its left end was originally also conditioned, since it should meet the internal wall that separates it from the 1st window on the right, as when viewed from the outside the two windows appear as a single one. This would make the length of this windows fixed in advance, and thus it could only vary in height. It was decided that a larger degree of freedom was important for this element, so one of the constraints was relaxed, the one that forced the window to meet the internal wall. So, the prevalent rule became the maintenance of the vertical axis, and the first window from the left in the existing design could actually be split in two. However, to ensure that Siza's solution would also be encoded in these rules, the first window from the left has to start at the internal wall that separates it from the second window. Constraints drawn in figure 7.12 can make this explanation more clear.

Although the first two windows from the 1st floor were constrained to have the same height, all the other pairs of windows [the ones that were side by side in the elevation] could have different heights.

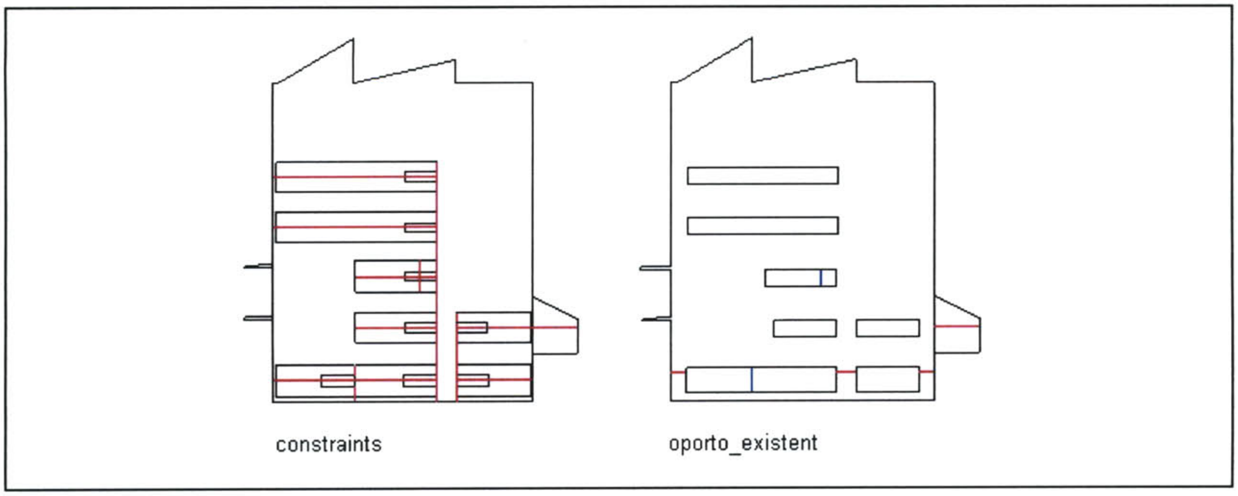


Figure 7.12 Constraints for east elevation

The 4th and 5th floor openings were constrained to be similar, as that was considered to be part of Siza's intentions. However, since the spaces they serve are slightly different, in terms of south elevation openings, that introduced another level of interactions the GS would have to account for. Finally, in the 3rd floor there is a very small space [probably a storage room] that is served by a small window whose width is constrained by the vertical compositional axis on the right, and an internal wall in the left. That width was thus kept as a constant, and only the height could vary, even though it had to be similar to the adjacent window's height.

7.7.3 North elevation constraints

Constraints for the north studio windows were similar to those set to the south façade. As for the asymmetrical element close to the entrance, it did not follow a 'centered window' logic. Instead, its left side had to be adjacent to the entrance module, and its top level was also conditioned to align with the entrance door top, as in Siza's design.

As for the roof monitors, their maximum height was conditioned by the tilt of the roof, and the minimum height was 30 cm. They also had to be centered in the respective wall, and the appearance of vertical openings was prevented using a previously described rule.

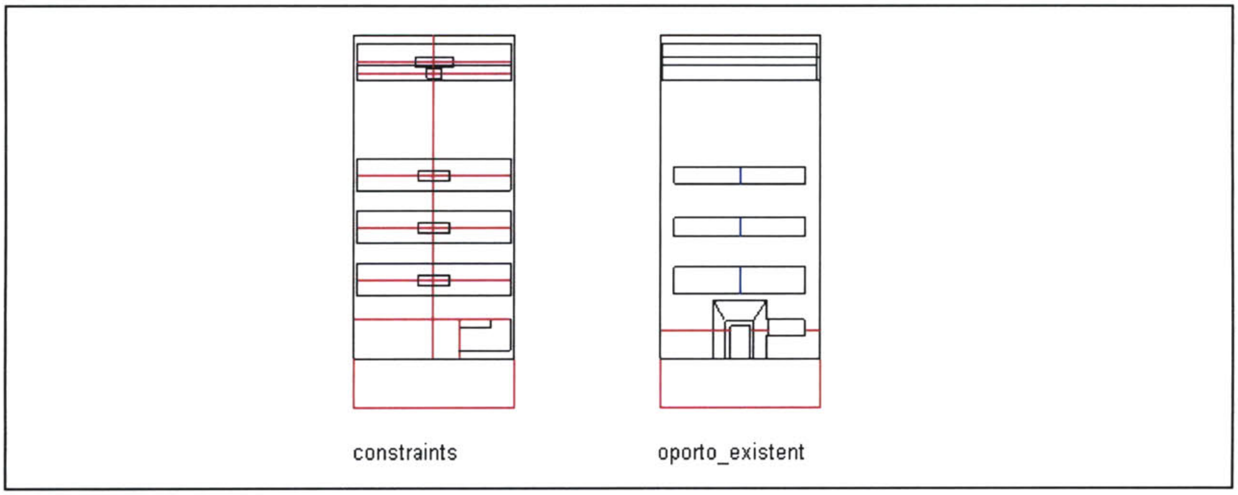


Figure 7.13 Constraints for north elevation

7.7.4 West elevation constraints

The west elevation obeys quite different rules from the other ones. From the 2nd floor to the 5th, the similar pairs of windows correspond to restrooms [on the left] and stairs [on the right]. The small distance between the two windows is basically the width of the concrete wall separating the two spaces. Those vertical axes were taken as the main compositional forces. The restroom windows were constrained to have a minimum sill height, similar to that used by Siza, for functional reasons. The windows could then grow up from there. On the contrary, the stairs openings were restricted in the maximum top height by the intermediate stair footing above them. The windows could then grow down from there. The windows were constrained to be similar for all four floors.

In the first floor, maximum heights were similar to those used in the east façade, and widths corresponded to either centered windows [as the one in the right], or windows growing from the meeting point with an interior wall [notice that in the left 1st floor window, there is also an internal breaking up the window in two, with part of it belonging to the adjacent restroom].

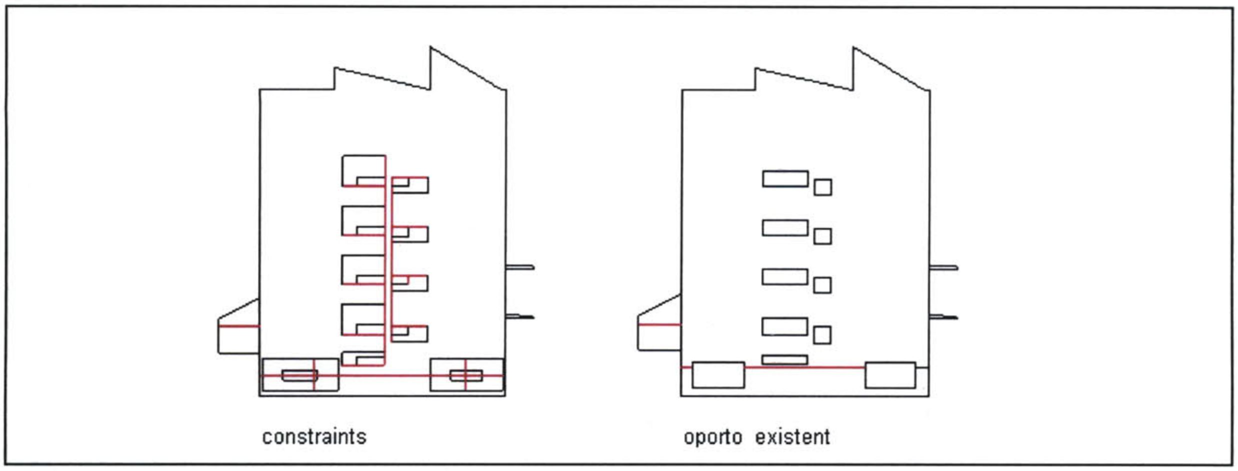


Figure 7.14 Constraints for west elevation

These are thus the constraints and compositional rules used for the GS to generate new elevation solutions for the existing building. Changing the constraints would allow for the exploration of many different design solutions, a path that was not pursued in this chapter's work. If the entire external wall area in each room was given to the algorithm as the search space for each opening, any type of openings with any type of proportions, located at any place in the façade could emerge, if that was a good solution both in terms of daylighting and thermal performance. However, there would be a total lack of control over architectural intentions. In this chapter, one of the specific objectives was to research the encoding of architectural design intentions, so using 'maximum freedom' constraints was not an option followed at this stage.

7.8 Climatic characterization of the three locations studied

The algorithm was run for three climates with distinct characteristics, to test its capability of adapting architectural design solutions to different environmental requirements while subject to the same language constraints. Apart from Oporto, Portugal, where the existing building is located, the other climates chosen were Phoenix [Arizona] and Chicago [Illinois], both in the USA, representing respectively a very hot and a very cold climate.

Oporto's climate is mild, with average monthly temperatures in the coldest months [December and January] around 10ºC. The warmest months [July and August] have average monthly temperatures of 20ºC. The lowest temperature registered in the weather file used was 2.8ºC in December, and the highest was 30.4ºC in August.

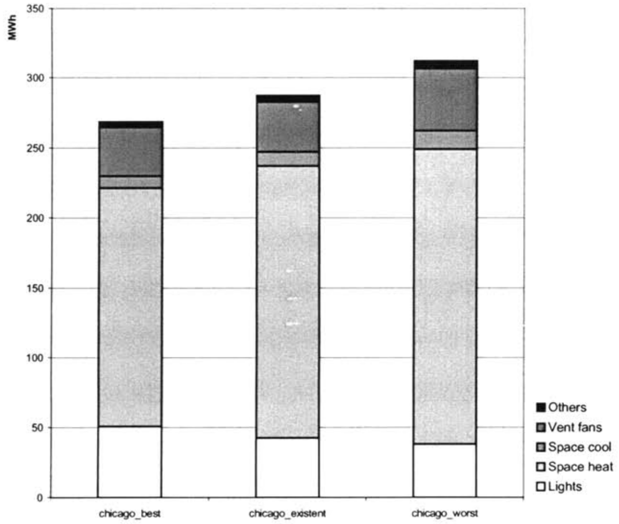


Figure 7.15 Average monthly temperatures for the three climates

Phoenix's climate is much hotter and dryer, with temperatures peaking at 44.4°C in June. Average monthly temperatures in July reach almost 34°C. Average monthly temperatures in the coldest months [December and January] are similar to Oporto's case, around 11°C. In this case, we used a WYEC weather file.

Chicago's climate is characterized by extremely low temperatures in the winter. The minimum temperature registered in the file used [TMY type] was -22.8°C in January. Average monthly temperatures are -2.2°C in December and -3.3°C in January and February. In summer [July] they are situated around 24°C.

7.9 Results

7.9.1 Oporto

Results from the generative system [GS] ranged from an almost exact coincidence with Siza's solutions to some radical departures from the existing design. In figure 7.16, three-dimensional models of both Siza's and GS solutions are displayed. The two images on the left show east and north elevations, with Siza's on the left and GS on the right. The two images on the right show west and south elevations, now with Siza's on the right and GS on the left; when the camera moved around the 3D models, their relative position was changed.

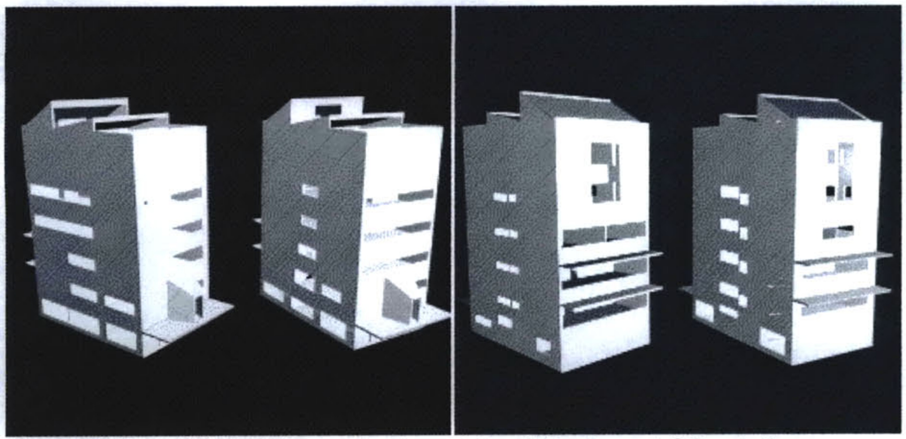


Figure 7.16 Three-dimensional models of Siza's and GS solutions

In the north façade [figure 7.17], the large horizontal stripes generated by the GS very approximately resemble those created by Siza [except for the melodic variations in height in the original design], denoting that in Oporto's mild climate the use of natural light in the studios clearly offsets the heat losses through the large glazing areas, as Siza may have predicted. Heat gains are not a significant issue in this orientation. It can be observed in fig. 7.17 that as the quality of solutions decreases [oporto_best, oporto_average, oporto_worst], window sizes decrease too. This may contradict the common supposition that reduced glazing areas should be used in north facades, at least in climates that are not too severe in winter. But it is also very interesting to observe that the GS best solution does not maximize window areas within constraint limits, instead it keeps them at an intermediate size that almost exactly coincides with Siza's design. Larger windows would probably already cause too much heat losses, disrupting the previous balance point.

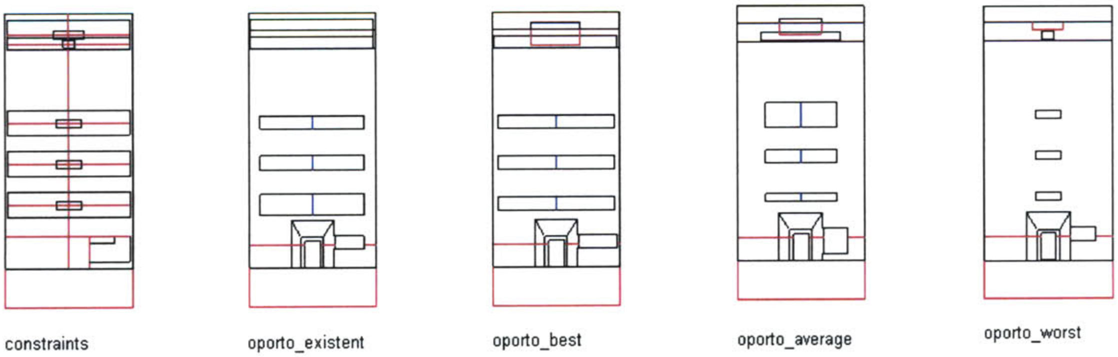


Figure 7.17 North elevation results

Towards the west [figure 7.18], the algorithm used small window sizes as Siza did, even further reducing them. This was due to the lower illuminance levels that the service areas [stairs and restrooms] require and to the reduced size of the spaces. It can be observed in figure 7.18 that as the openings get larger, the quality of the solutions decrease. The west orientation is particularly dangerous regarding heat gains, thus Siza never placed studios facing west except in tower G, where he used large vertical fins to protect the openings from the sun [see figure 7.3].

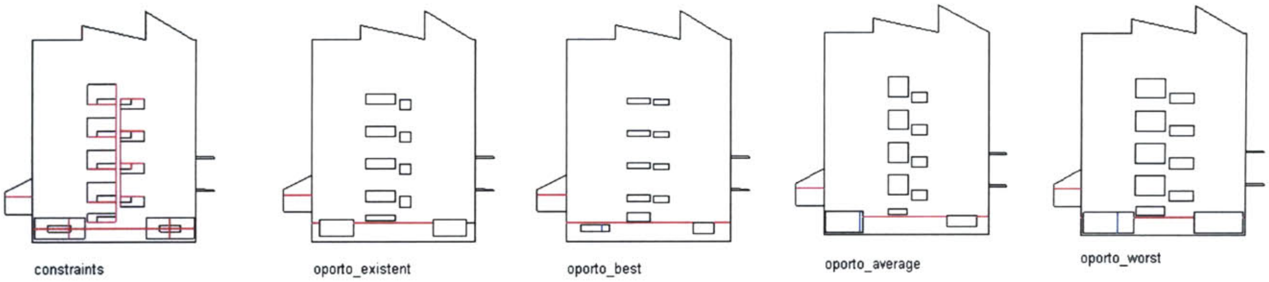


Figure 7.18 West elevation results

In the south orientation [figure 7.19] the generative system solutions present more significant modifications in relation to the existent. In Siza's design, the 2^{nd} and 3^{rd} floors have south-facing studios with long horizontal windows shaded by 2-meter deep overhangs. The algorithm solutions tend to suggest these overhangs may be too deep. When the overhang depth is kept as 2 meters [oporto_best], window sizes assume the largest dimensions allowed by the constraints. The deep overhangs block the admittance of daylight into the room, and to counteract that effect the algorithm increases the opening size. It will be seen later that when overhang depth is a variable [oporto_shading], results differ quite considerably from these.

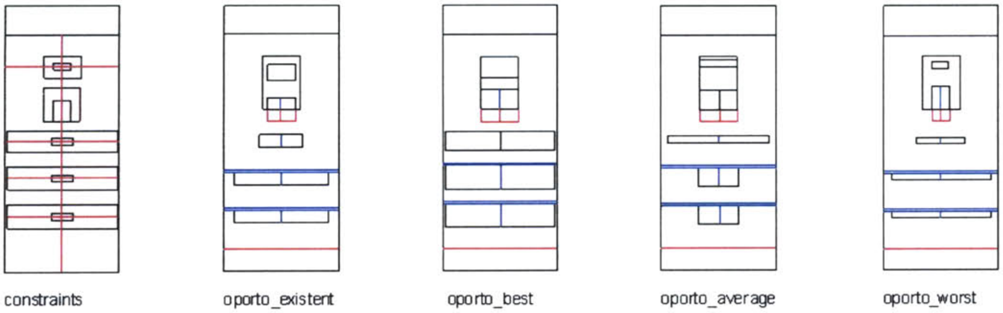


Figure 7.19 South elevation results

In the 6th floor, the solution from the GS for the south-facing loggia has to be understood in conjunction with the roof monitors solutions [which can be seen in figure 7.17 or in the 3-D images of figure 7.16]. As mentioned before, the 6th floor is basically occupied by a single space, lit from above by two roof monitors, from the south by a loggia window, and with blank walls in all other directions. The larger rooftop monitor is positioned in such at way that it lights the same area of the space as the loggia does [see figures 7.20 and 7.21]. On the other hand, the back of the room [north end] has as its only light source the smaller roof monitor, and becomes a much darker area [figure 7.21].

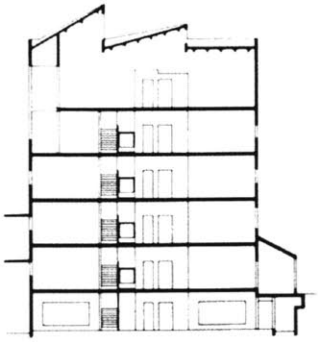


Figure 7.20 Section through tower H. The top floor section shows the loggia and the larger roof monitor light the same area of the room.

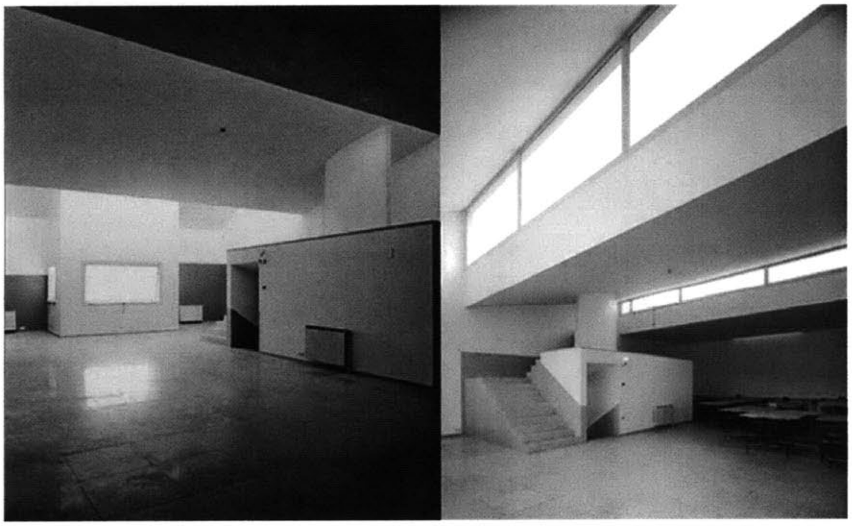


Figure 7.21 The photo on the left shows the loggia window, and the differences in light levels between the front and the back of the room. The photo on the right shows the two roof monitors.

The GS increases the south-facing loggia window to the maximum allowed by the constraints, and reduces the glazed area of the roof monitor that lights the space closer to the loggia [figure 7.22]. The roof monitor faces north and is a large source of heat losses in winter, particularly because warm air rises to the glazed areas. Increasing the south opening is a good option because it faces a favorable orientation and is shaded, since it is recessed into the façade, and permits reducing the roof monitor without losing too much daylight in the studios. On the other hand, the second roof monitor assumes the largest dimensions possible in the GS solution [as in Siza's design], since the back of the room has no other light source. This result suggests the tilt of the roof could be varied to allow for a larger roof monitor in that location.

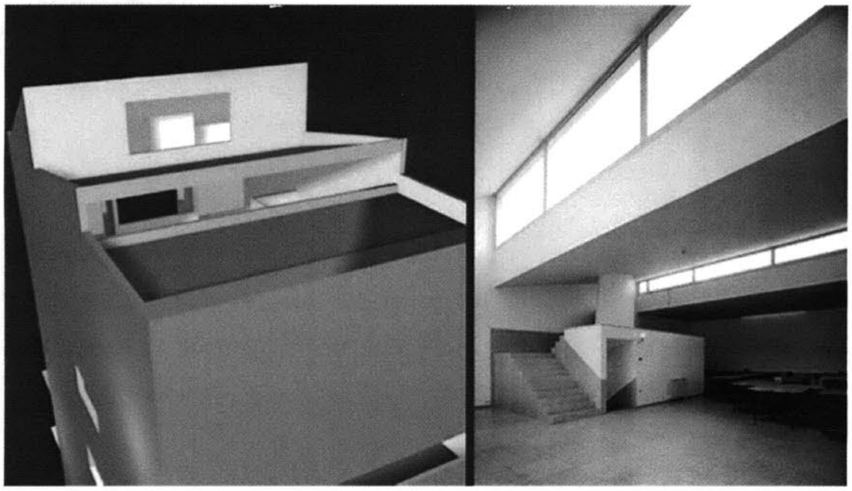


Figure 7.22 The rendering on the left shows the roof monitors proposed by the GS. The photo on the right is repeated for comparison purposes.

A final note should be added about the GS proposal for the loggia's openings, as they reflect the relative dominance of some variables over others, as mentioned earlier on the chapter. The south-facing windows in the loggia are driven to the maximum dimensions allowed by the constraints, because they allow good daylight throughout the day and the year and are also shaded against excessive solar gains. The east and west loggia windows are constrained by the rules implemented to have the same height as the south ones, which makes them become quite high. Because large east and west windows are detrimental, as can be seen from other results, their width is made the minimum possible. One can imagine that probably the east/west opening were attempting to drive the window height to a small value too, while the south opening was pushing it towards larger dimensions. The south influence seems to have prevailed, and the resulting loggia design can be seen in figure 7.23, where the 6th floor openings become closer to doors than to windows, due to the play of forces described here.

The 4th and 5th floor south solutions have to be analyzed together with east results [figure 7.20], since in those floors the studios share both south and east openings. The GS increases south-facing windows in relation to the existing design, and simultaneously reduces east-facing ones. The east orientation is unfavorable due to high solar gains during the morning in summer months, and to reduced daylighting levels during the afternoon for most of the year. South-facing openings perform better both in terms of natural light admission and control of heat gains. When the GS has the possibility of trading between the two options, it consistently favors south.

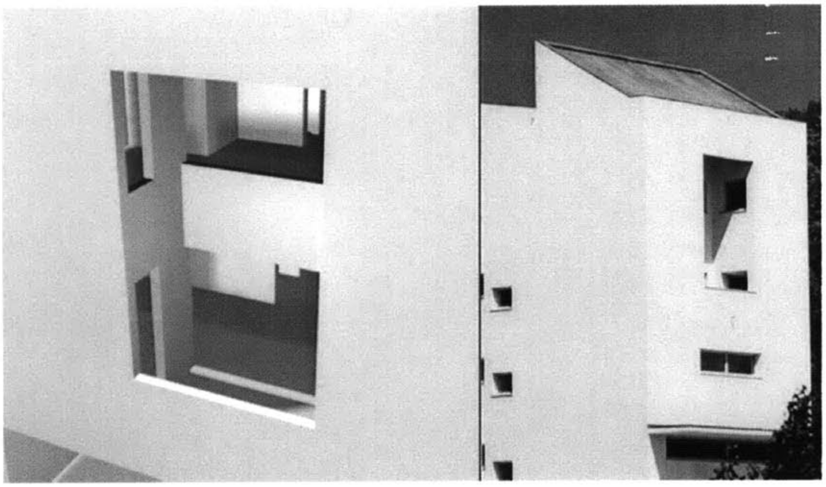


Figure 7.23 The loggia design proposed by the GS [left] and the existing one by Siza [right]. Increasing the south openings lead to the overall configuration seen on the left.

This happens because the GS dynamically evaluates for each hour of the year the interactions between the several elements of the solution, like east- and south-facing windows in the same room. To gain a better insight on the reasons behind the GS solution, we exported the three dimensional models of both Siza's and GS solutions to a detailed lighting simulation program that combines radiosity and ray-tracing methods to achieve a good degree of accuracy in daylighting predictions. The 4th floor studio room was chosen for these simulations, since it has both south and east-facing openings and thus generates trade-offs between the two openings.

Figure 7.24 shows the daylight patterns in the room during the morning for the summer solstice [June 21]. From left to right, there is Siza's solution at 9am, the GS solution at 9am, then Siza's solution at 10am and finally the GS solution at 10am. In Siza's design, it can be seen that during morning hours the low sun enters deeply into the studio space. This is a source of overheating in the room, and also generates problems in terms of visual and thermal comfort in the studio, as it could be observed during the visit to the building. Students using this room had taped large pieces of paper into the east-facing window to try to overcome these problems.

The GS proposes instead a large south-facing window that causes little direct solar penetration into the room, even without using overhangs. The east side window is made just large enough to illuminate the far end of the studio, which would otherwise become too dark. The comfort-related problems of direct sun penetration are thus reduced to this area of the space, and the amount of solar heat gain also greatly reduced.

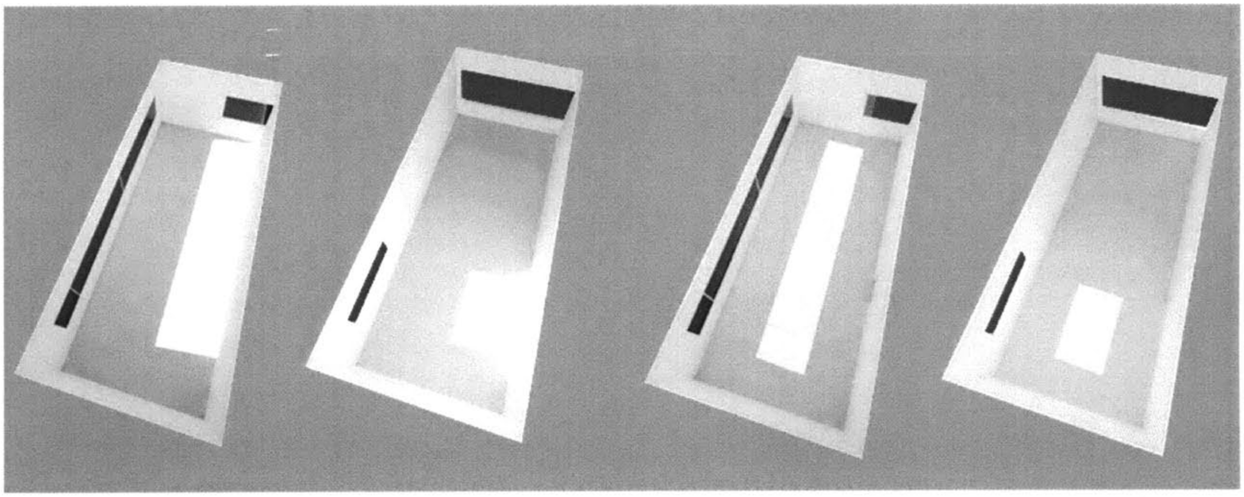


Figure 7.24 Existing and GS results for 4th floor studio with both east and south windows. The long side of the room is oriented towards east. From left to right, [Siza 9am], [GS 9am], [Siza 10am], [GS 10am].

Figure 7.25 shows the daylight patterns in the room during the afternoon for the summer solstice [June 21]. From left to right, there is Siza's solution at 12pm, the GS solution at 12pm, then Siza's solution at 3pm and finally the GS solution at 3pm. In Siza's design, there is still some direct sun entering the room at noon, because the building is oriented 8 degrees east of north, and thus the east façade is not yet in shade at noon. In the GS solution, there is little sun penetration through the south-facing window, because the sun is quite high in the sky.

Finally, at 3pm, the existing design becomes rather dark, because the east façade is in the shade, and the south opening is too small to lit such a large space. The light levels in the GS solution seem to be much higher, even though direct sun remains not a problem.

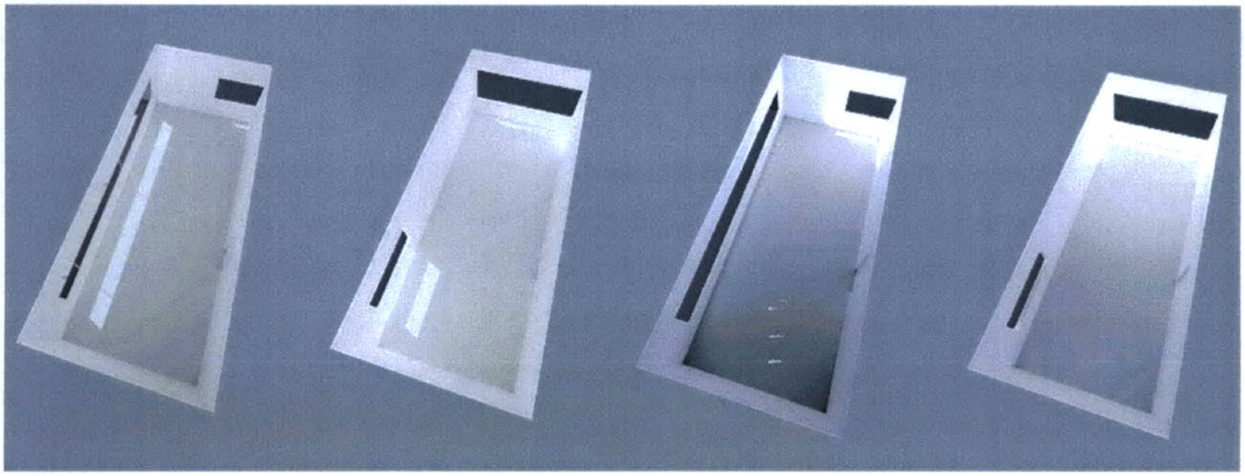


Figure 7.25 Existing and GS results for 4th floor studio with both east and south windows. The long side of the room is oriented towards east. From left to right, [Siza 12pm], [GS 12pm], [Siza 3pm], [GS 3pm].

To further investigate daylight patterns in the space during the afternoon [3pm, summer solstice], images showing illuminance level contours [in lux] were produced, and are shown in figure 7.26. In Siza's solution, the daylighting levels never achieve the specified setpoint of 500 lux. In the furthest corner from the windows, the light levels are around 200 lux only. The same location, in the GS solution, has a daylight illuminance level about 4 times higher. In general, the GS solution achieves quite high luminance levels throughout the space. Close to the south-facing windows, illuminance levels may be too high, but this could be solved by placing a shallow overhang over the south window. Placement and sizing of overhang elements will be dealt with in a later section of this chapter.

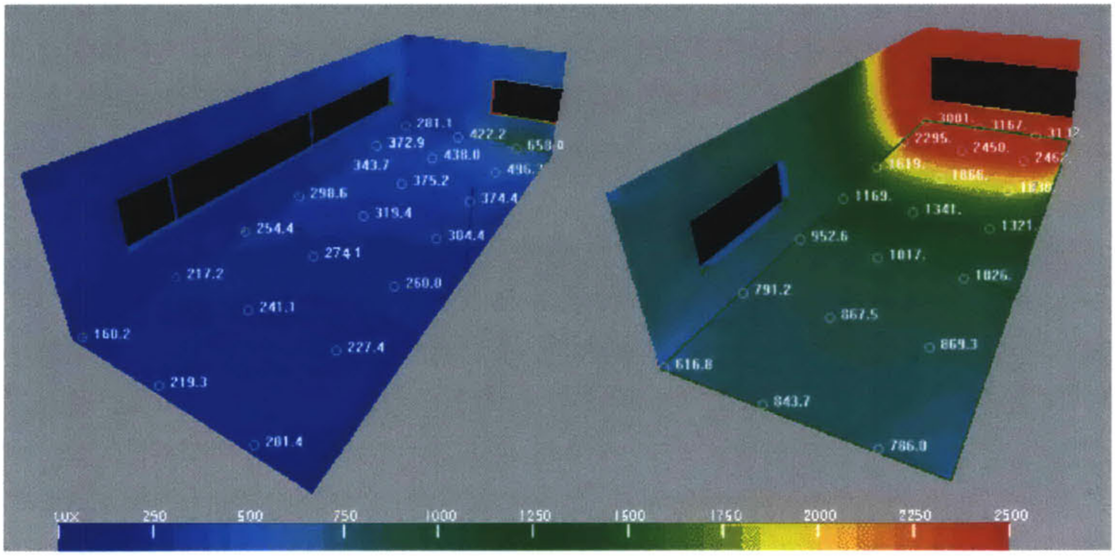


Figure 7.26 Comparison of existing [left] and GS [right] daylighting levels in 4th floor studio at 3pm, June 21.

An additional experiment was carried out [not shown in the images], where the algorithm could not change the south window from the existing design, but could use an overhang in the east window. It was very interesting to observe that, in that situation, the GS proposed a large east-facing window very similar to that used by Siza, but with a deep overhang over it to preclude direct solar gains.

Figure 7.27 shows, for the GS solution, the pattern of solar penetration into the studio for different times of the year. Apart from the summer solstice, which was already discussed above, it shows that in winter the south-facing window allows for useful solar gains throughout most of the day, although in the morning the east window receives almost no sun. In the equinoxes, there is direct sun through both of the windows, but as temperatures at that time of the year are not too high, that does not represent a serious disadvantage.

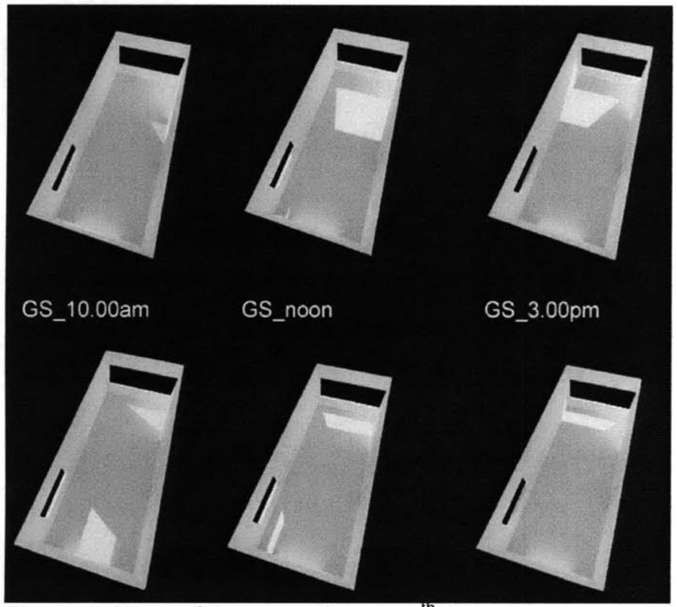


Figure 7.27 GS results for the 4th floor studio room. Top row is for the window solstice. Bottom row is for the equinoxes

East façade results [figure 7.28] have already been partially analyzed in conjunction with south results. In general, all openings decreased their dimensions in relation to the existing design, due to the problems associated with this orientation. The figure shows that as the size of east facing windows increase, the quality of solutions decrease. The studios that have only east facing fenestration maintained average sized openings, as a compromise between good daylighting and thermal control. The studios where east openings could be traded against south ones [4th and 5th floors], favored south. The 5th floor opening was not only constrained to be equal to the 4th floor [explicit interaction], but it has implicit interactions with the loggia solutions, since they both coexist in the same space. The 5th floor loggia is related to the 6th floor loggia, and that to the roof monitors, etc, creating the long chain of implicit, performance-based interactions that take place in the design, and are concomitant with those explicit enforced by the rules and constraints applied.

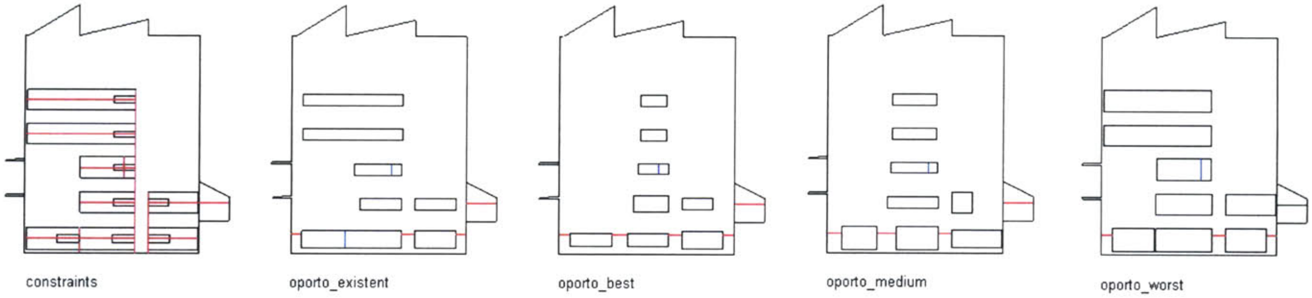


Figure 7.28 East elevation results

Finally, figure 7.29 shows a series of renders of the 3D models of both Siza's and GS solutions.

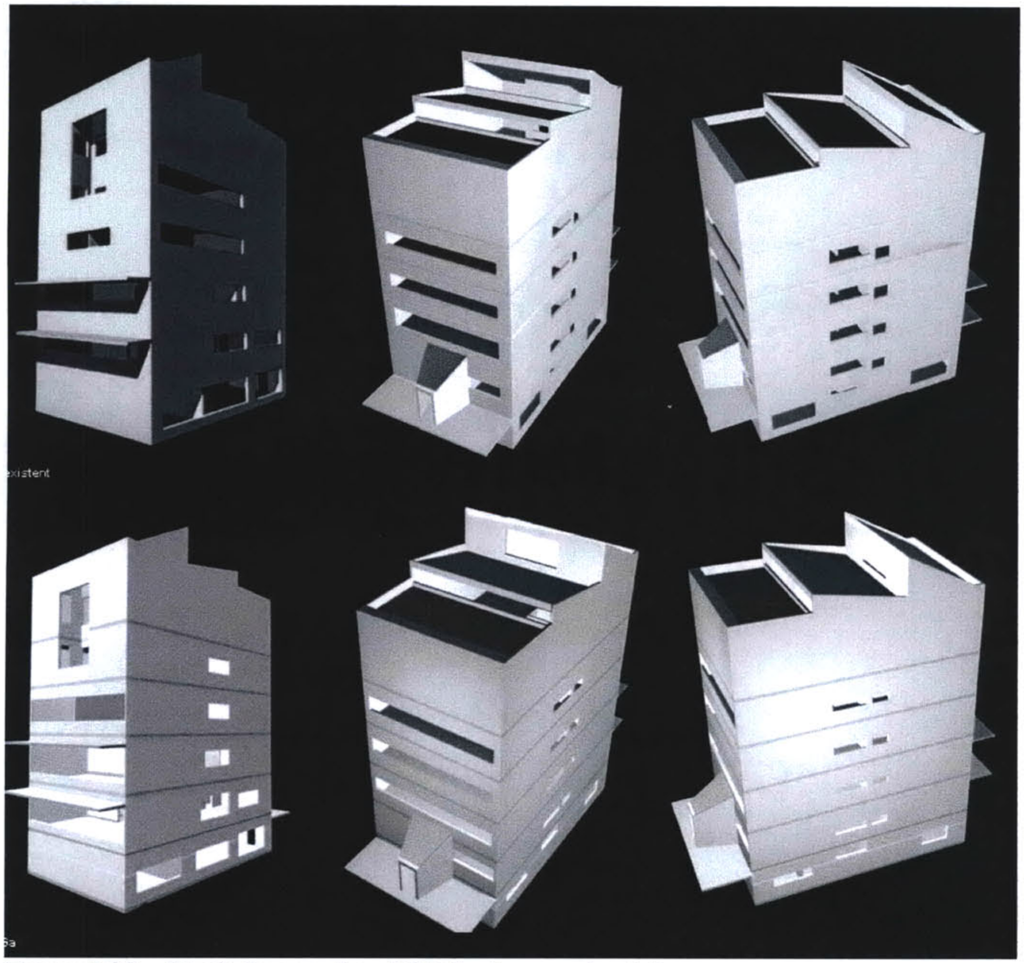


Figure 7.29 Siza's [top row] and GS [bottom row] solutions

7.9.2 Using shading as a variable in Oporto

The shading studies implied using overhang presence and depth as variables too, together with fenestration sizes. It will be seen that overhang dimensions will affect the sizing of openings, as expected. The overhang is always positioned directly on top of the window, with no distance between the two elements. Shading studies were only performed for south and east orientations, since towards the north direct sun does not play an important role in the performance of the glazing elements, and towards the west opening sizes were generally so small [due to the small service areas they serve] that it was considered it would not be a particularly interesting exercise to play with shading elements in this orientation.

Towards the south, when overhang depth is a variable [porto_shading], the GS reduces it from the original 2m depth to 0.5m, and also reduces window sizes to a dimension closer to that used by Siza, a departure from the large window sizes generated in oporto_best. The shallower overhangs allow more daylighting into the studios while still blocking direct sun and high solar gains, since in the hottest months the sun is high in the south quadrant and can be controlled with shallow overhangs. On cold winter months, when the sun is lower in the sky, useful solar gains are still admitted into the rooms, reducing the need for heating. This type of shallow overhangs the GS suggested could also control the potential glare problems identified in the previous section.

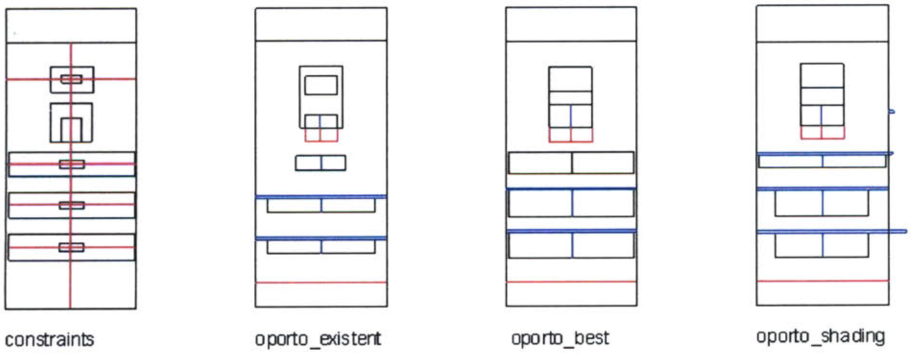


Figure 7.30 South elevations, including shading results

When the algorithm was allowed to place overhangs in the east façade too, it significantly increased window sizes in the 2nd floor, while placing quite deep overhangs to shade the low morning sun. It should be noted that the studio in the 2nd floor has only east facing windows. For the studios on the 4th and 5th floors, which have both east and south facing windows, the GS kept east openings small [although slightly larger than in the unshaded case] with shallow overhangs, and privileged south facing openings again. Overhangs were not allowed by the constraints on the 1st floor, since it was considered it would be out of Siza's language [in all of the four studio towers, Siza never used overhangs in the 1st floor].

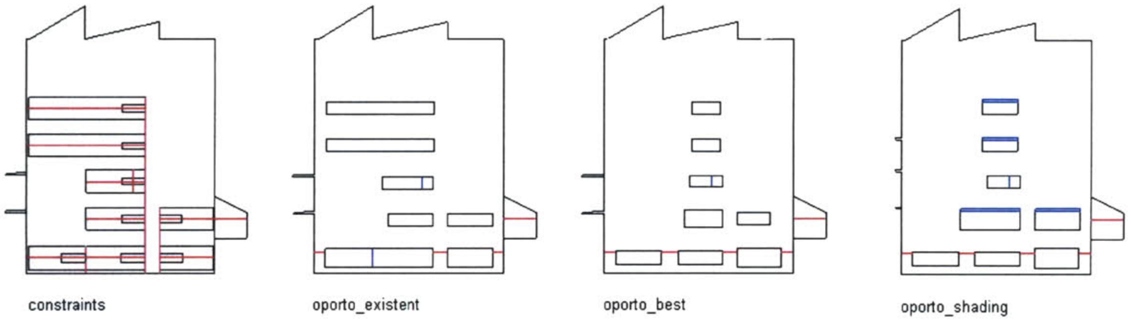


Figure 7.31 East elevation results, including shading

7.9.3 Phoenix

For both Phoenix and Chicago only the best solution found by the GS is shown. Under the hot Phoenix climate, the main differences in GS solutions relatively to Oporto were [figure 7.32]: in the south, unshaded windows [4th floor] were significantly reduced, even though shaded ones remained quite large. East windows were made much smaller too. This reflects the effect of high heat gains in that geographical location, and suggests shading could be a useful design addition in this case too. West windows remained small, as they did for Oporto, to avoid heat gains too. The north façade suffered almost no alteration, since it is only marginally affected by direct solar gains.

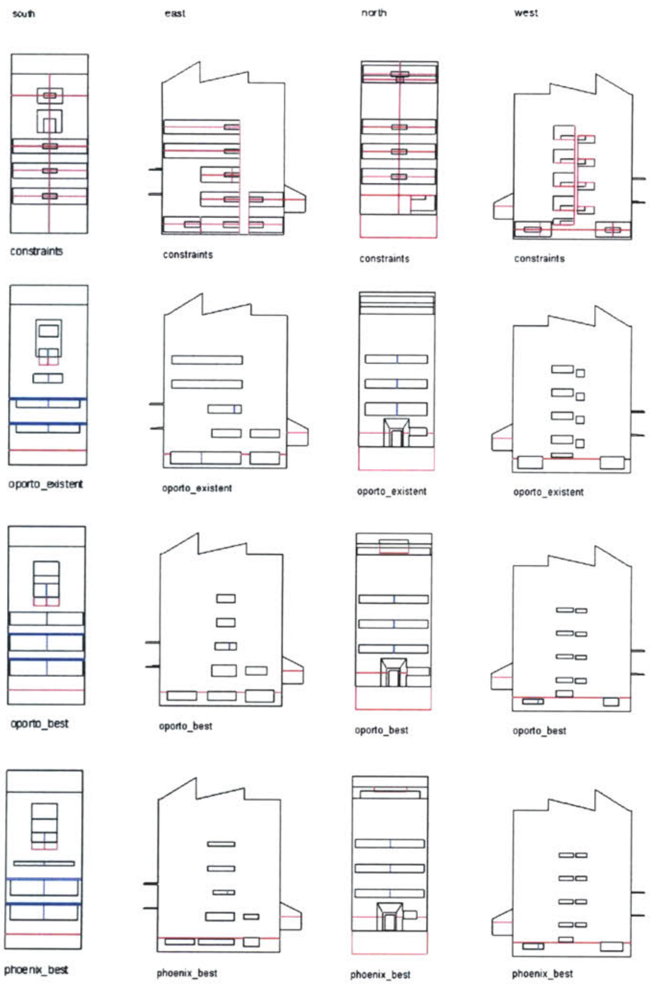


Figure 7.32 Phoenix results are in the bottom row

7.9.4 Chicago

The extremely cold Chicago climate originated some interesting changes in relation to Oporto and Phoenix solutions [figure 7.33]. For north-facing studios, the windows were reduced to the minimum dimensions allowed by the constraints, due to high heat losses through the glazing and to the absence of solar gains, which would be beneficial in Chicago's cold climate. This façade-level solution may allow for an extrapolation in terms of spatial organization, suggesting that north-facing studios should be avoided in this type of climate, since obviously the reduced window sizes suggested by the GS are not acceptable from a users point of view, for whom factors as a view out and natural lighting will certainly be primordial. It is also interesting to observe that the best solution for Chicago is very similar to the worst solution for Oporto.

Towards the south, unshaded windows were made quite large since they couple daylight admission with useful solar gains. However, shaded windows were reduced to the minimum dimensions possible, as both natural light and solar gains are blocked, and heat losses prevail. When overhang depth was used as a variable, the algorithm reduced it to the minimum allowed, and simultaneously increased window sizes [this result is not shown in the images]. It might be concluded that south shading may be undesirable in this climate.

Towards the east, rooms that have only east-facing windows received average-sized openings [1st and 2nd floors], a compromise between positive factors like daylight admission and morning solar gains, and negative ones like high heat losses through the glazing. For studios with both east and south facing windows, east ones were made quite small since once again south was preferred. The west fenestration received minimum dimensions allowed.

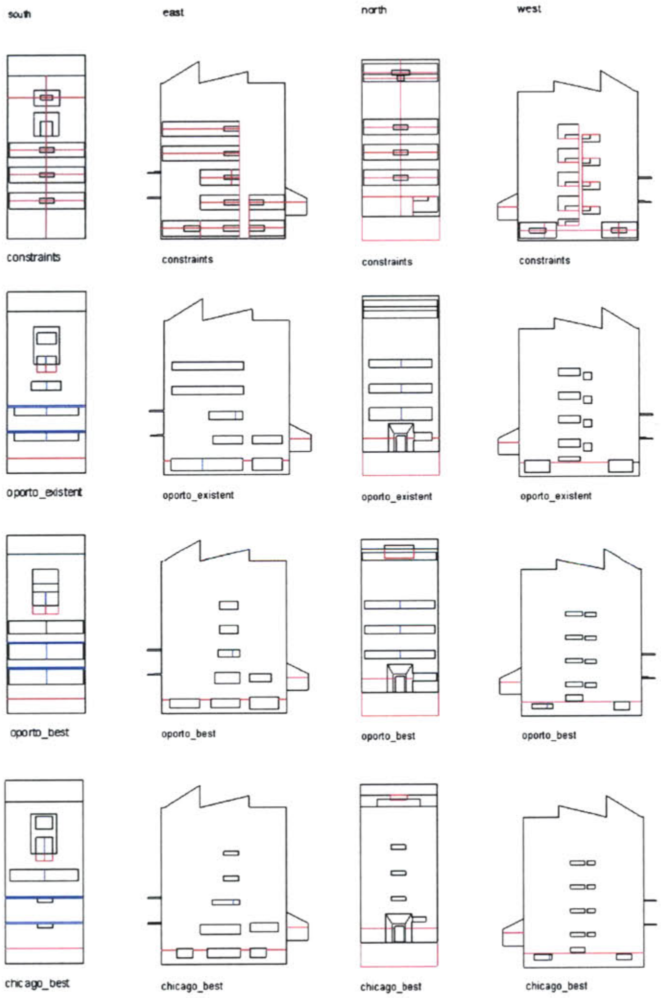


Figure 7.33 Chicago results are in the bottom row

7.10 Annual energy consumption results

7.10.1 Oporto

Annual energy consumption for the different Oporto solutions is:

Solution	**MWh**
oporto_shading	87.58
oporto_best	89.99
oporto_average	96.22
oporto_existent	96.45
oporto_worst	110.55

Table 7.1 Annual energy consumption for Oporto solutions

For Oporto's climate, the worst solution found by the GS has about 26% higher energy consumption than the best solution with shading as a variable. Oporto_shading has a slightly lower energy consumption than oporto_best. Siza's design consumes about 10% more energy than the best GS solution.

The graph in figure 7.34 shows annual energy consumption for Oporto solutions.

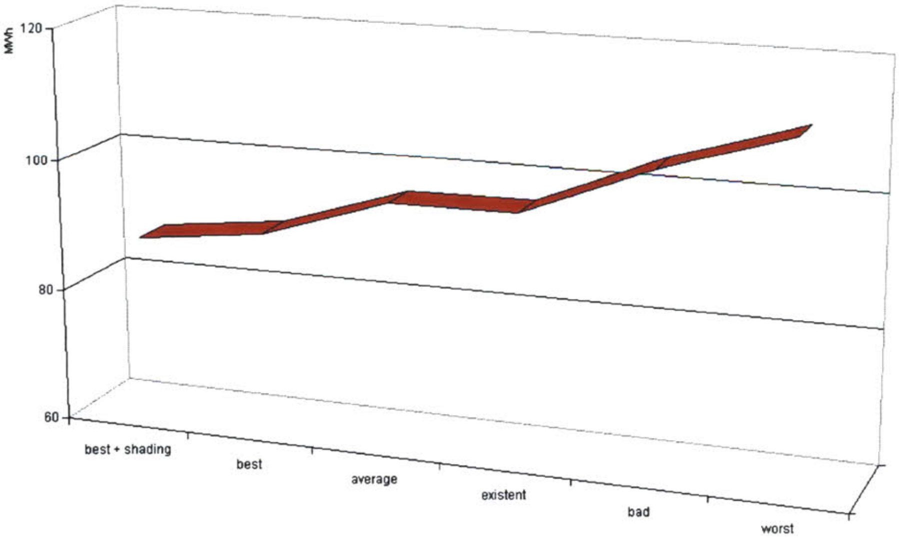


Figure 7.34 Annual energy consumption for Oporto solutions

Examining results broken down by energy end-use [figures 7.35 and 7.36], it is possible to see that the existing solution by Siza performs almost as well as the best solution from the GS in terms of natural lighting use [meaning that artificial light consumption is low]. From the previous description of results, we could see that the main differences between the two solutions were in south and east facades, as well as the roof monitors. Although the GS performed many changes in the individual spaces, which may therefore have a more balanced use of daylighting, the overall artificial lighting consumption of the building did not change much, showing that Siza's control of daylighting is actually very sophisticated.

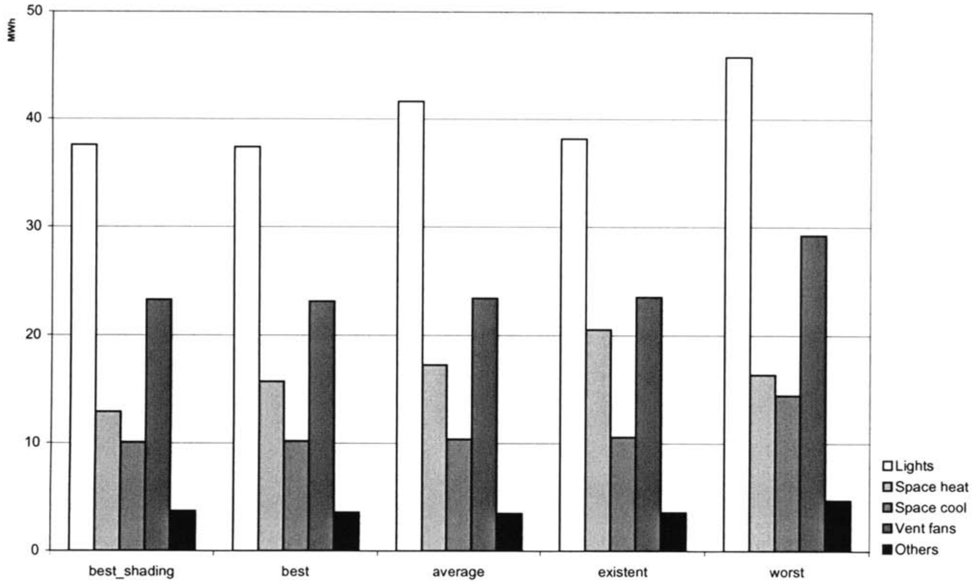


Figure 7.35 Energy end use for Oporto solutions

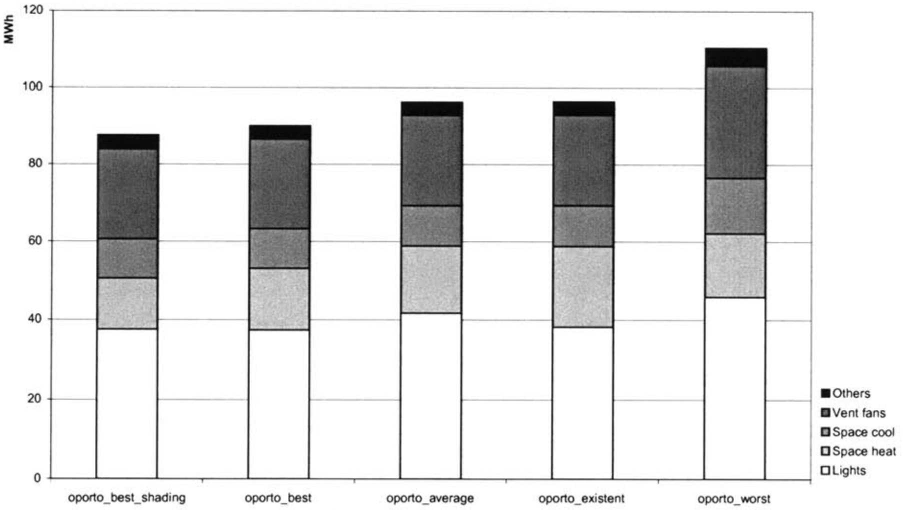


Figure 7.36 Energy end use for Oporto solutions [aggregated]

Artificial-lighting consumption levels increased about 22% from the best solution with shading to the worst solution. This number could probably be higher if some of the lighting reference points were not placed so deep into the room [about 1.5 m from the wall most distant from a window]. The increase in space cooling in the worst solution is probably due to the large east- and west-facing windows. The existing solution does not differ much from the best one with shading. Even though some east windows decreased in size, others increased but had overhangs added. South windows had their overhang depth reduced for increased natural light, but that increased heat gains, too. For space heating, the best GS solution performs considerably better than the existing solution, as it allows more useful south solar gains in the winter, and reduces heat loss through the north-facing monitor and west and east openings.

7.10.2 Phoenix

Annual energy consumption for the Phoenix solutions is:

Solution	**MWh**
phoenix_best	119.02
phoenix_existent	131.09
phoenix_worst	162.74

Table 7.2 Annual energy consumption for Phoenix solutions

In terms of energy end-use, the main differences between the several solutions occur, as expected, in space-cooling energy and associated ventilation-fan energy [figures 7.37 and 7.38]. If the existing design would theoretically be placed in Phoenix, it would consume about 20% more energy in space cooling and fans than the best solution the GS found [without using shading as a variable]. This happens because in the GS solution all unshaded east, south and west windows were greatly reduced, while shaded windows and north-facing ones were kept large for increasing daylighting use. Once again, the differences in artificial lighting use are not as significant as it might be expected, probably due to the extreme location of the lighting reference points.

The worst solution the GS found for Phoenix consumes about 38% more energy in space cooling and ventilation fans than the best one, within the same language constraints.

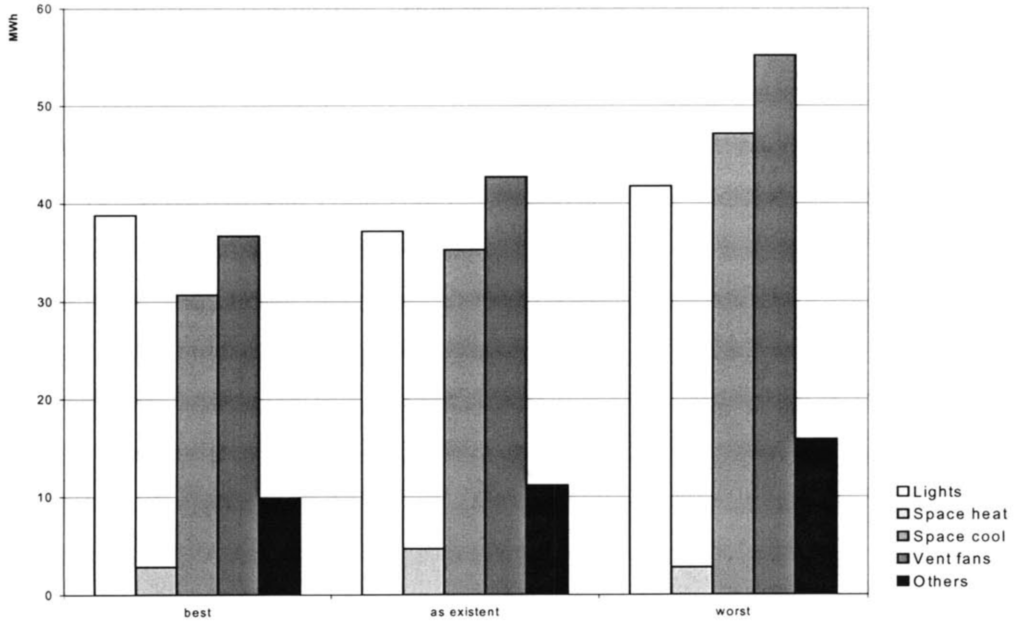


Figure 7.37 Energy end use for Phoenix solutions

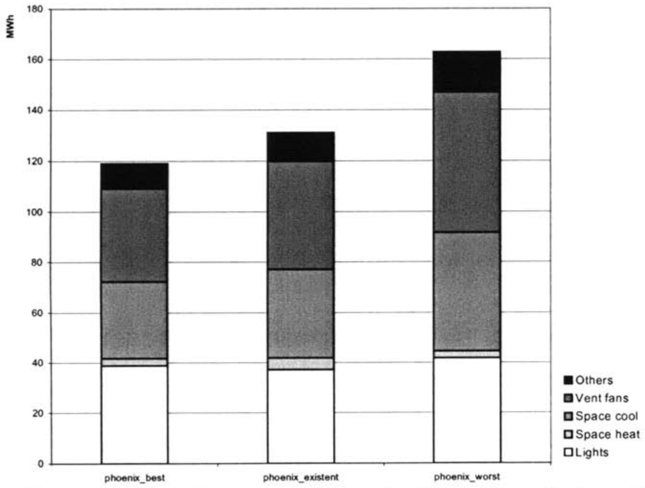


Figure 7.38 Energy end use for Phoenix solutions [aggregated]

7.10.3 Chicago

Annual energy consumption for Chicago solutions is:

Solution	MWh
chicago_best	268.72
chicago_existent	287.36
chicago _worst	312.24

Table 7.3 Annual energy consumption for Chicago solutions

In terms of energy end use, space heat dominates Chicago solutions, as expected [figures 7.39 and 7.40]. The high energy-consumption levels could be reduced using such strategies as properly insulating the building and using better glazing types, which are not variables considered in this study. But even using only opening sizes as variables, the existing design consumes about 14% more heating energy than the best GS solution, and the worst solution consumes 23% more heating energy. Because those savings in heating energy imply reducing opening size and thus reducing natural light availability, the overall performance between the different solutions is that the existing design consumes about 7% more energy than the best GS solution, and the worst design consumes about 16% more energy annually.

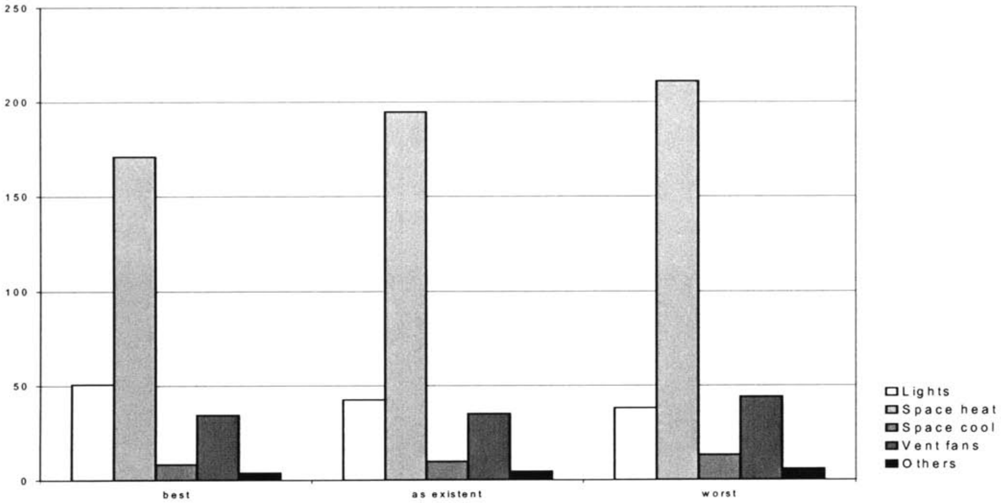


Figure 7.39 Energy end use for Chicago solutions

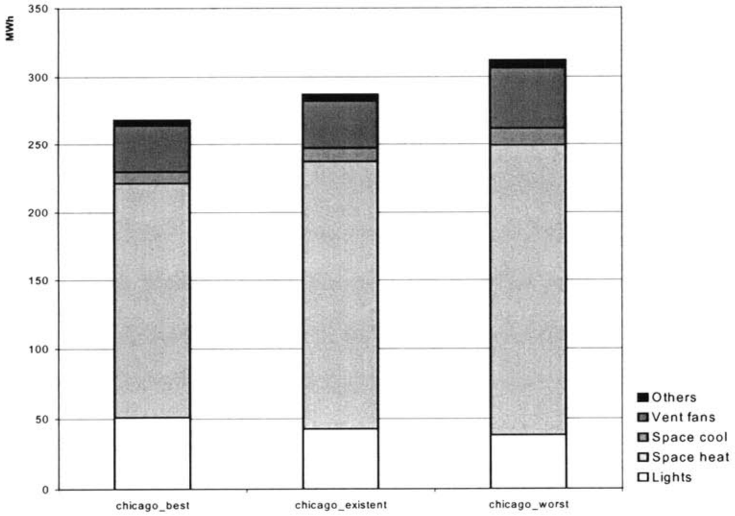


Figure 7.40 Energy end use for Chicago solutions [aggregated]

7.11 Genetic Algorithm search progression

A MicroGA was used in this study too. The search points tested during the execution of the program are displayed in figure 7.41. The use of elitism allowed for the maintenance of the best individual at all stages of the search, while the GA kept looking for new solutions. It can be seen that individuals tend to concentrate towards lower energy consumption levels, but simultaneously new areas of the search space are always being considered.

The minimum energy solution steeply decreases during the first generations, and then tends to stabilize. Further progress is then achieved in small steps and requires several generations. The maximum number of generations used was 100.

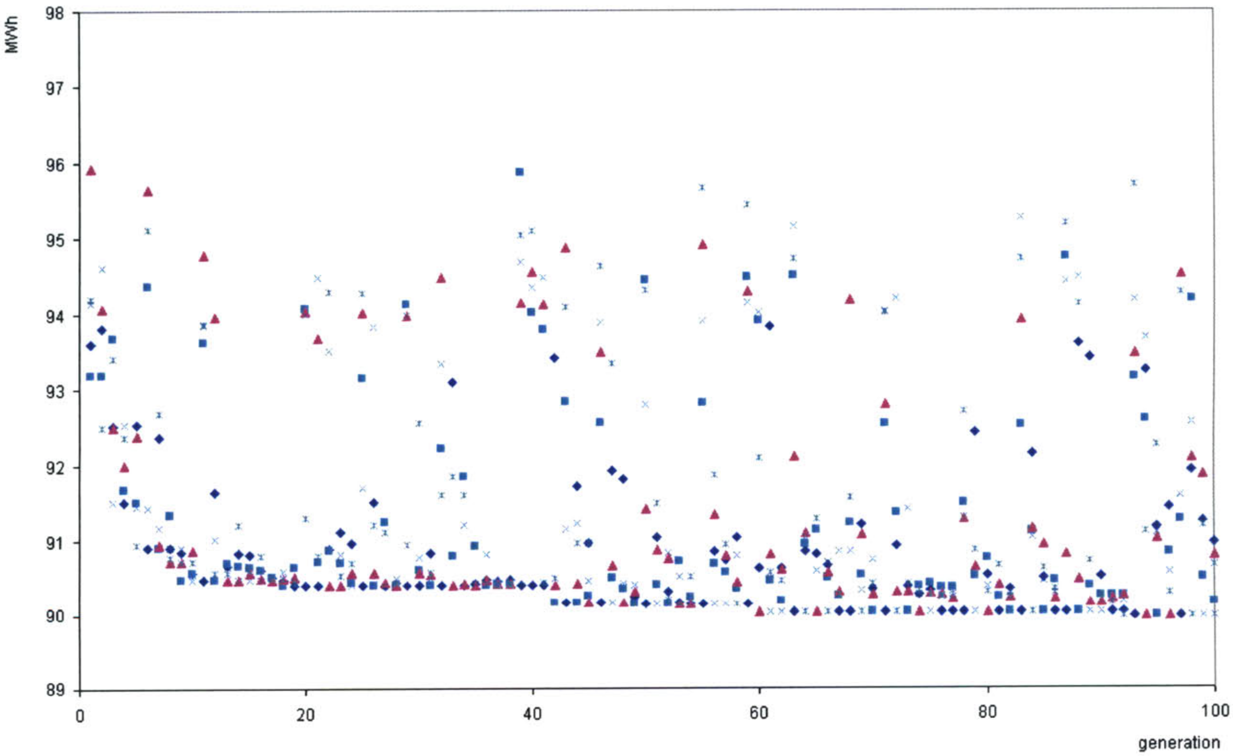


Figure 7.41 Each point in the graph represents a solution tested by the algorithm

Figure 7.42 represents a number of random individuals picked during the search process, and illustrates the intermediate, hybrid solutions the algorithm goes through during its search for high performance individuals.

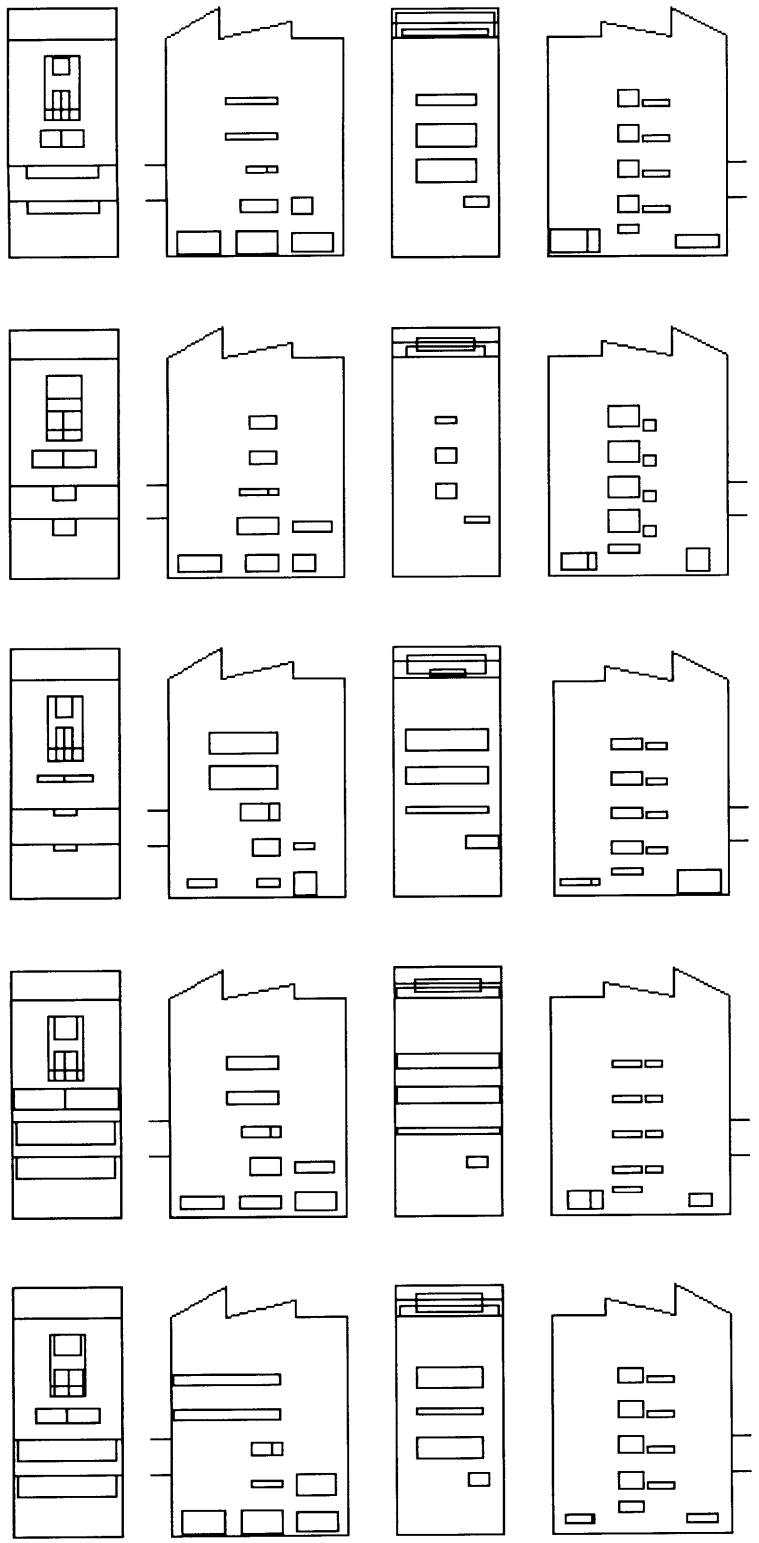


Figure 7.42 Five random individual solutions

Finally, figure 7.33 illustrates the progression of the average fitness of the population during the 100 generations, together with the fitness values of the best and worst individual at each step.

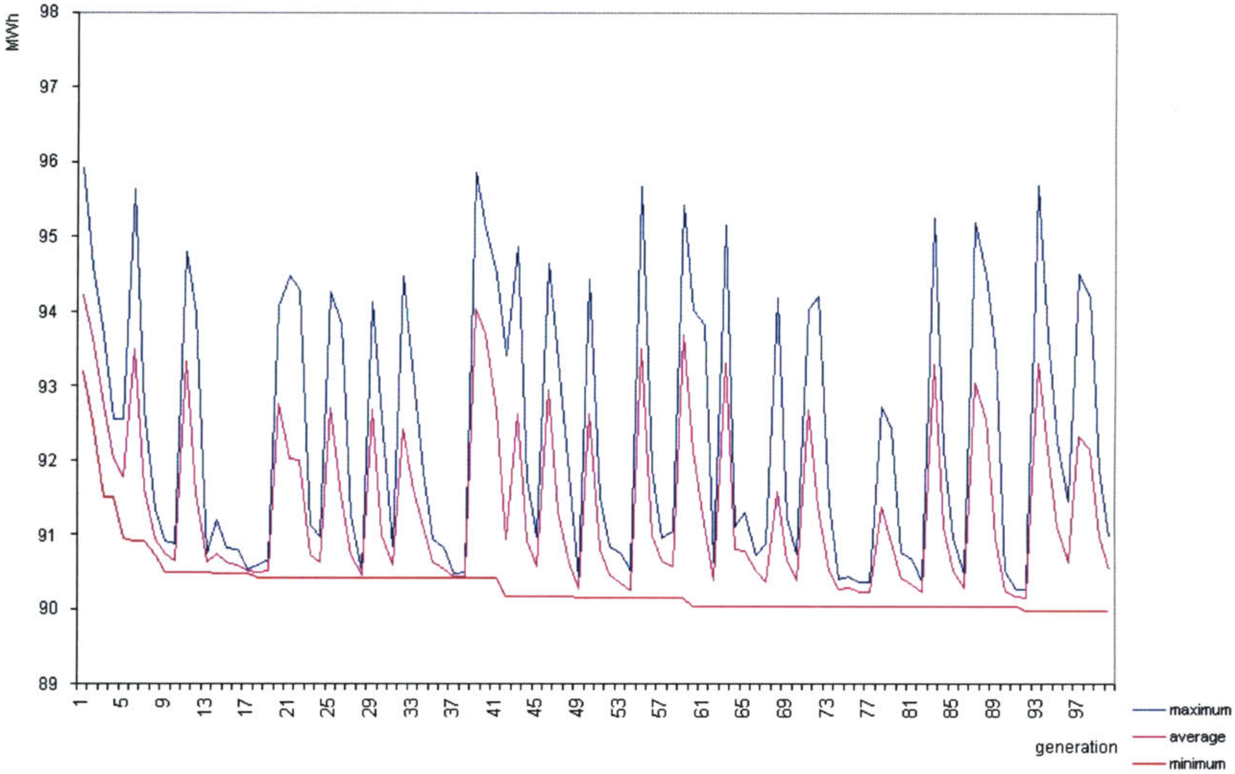


Figure 7.43 Progression of GA search for 100 generations

7.12 3D-Shape manipulation

A first attempt to introduce shape manipulation into the generative system is described in this section. Following the results analysis from the previous sections, it was considered that varying the tilt of the different roof sections could improve daylighting admission into the top floor, by permitting a more flexible sizing of the roof monitors. The GS was allowed to vary the tilt of the roofs and thus control the size of the roof monitors. Our guess was that the larger roof monitor should probably not be the one closer to the loggia, in the south side, but the one in the northern part of the room, since that area had no other light sources. To simplify the experiments, it was assumed that the rooftop monitors would always cover the entire width of the building, as in Siza's original design. The height would be determined by the tilt of the corresponding roof, as the glazed opening would have the same height as the wall it was located on, which would be a consequence of the roof inclination. Roof tilt was allowed to vary between 10º and 45º. The northernmost rooftop had to be at least 2 meters set back from the north façade so that it could not be read as part of the elevation, for compositional purposes.

Figure 7.44 represents some random solutions arbitrarily extracted from those first generated by the GS. It is possible to observe a wide variety of possible solutions, even within these apparently limiting constraints.

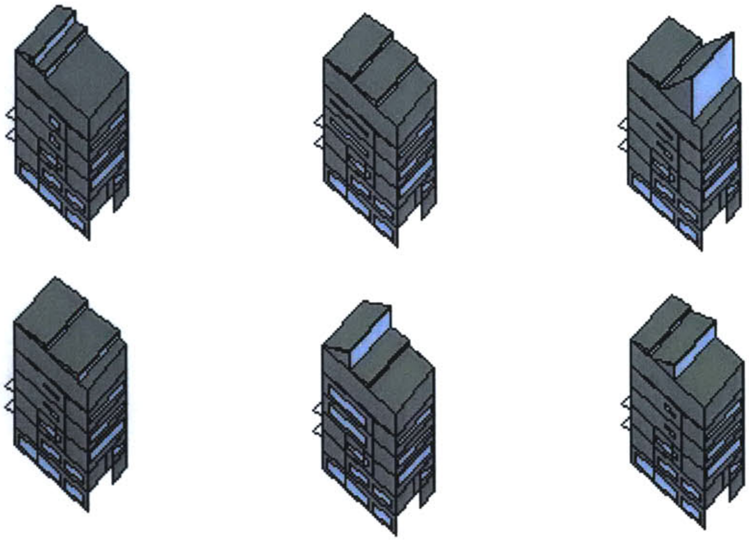


Figure 7.44 Random roof tilt solutions generated by the GS.

Figure 7.45 shows the best solution found by the GS on the left, and also our guess about what could be the best solution for the roof monitors [on the right]. The difference between that guess and the actual solution shows how difficult it is to predict the interaction of all variables without the use of computer simulations, even using information available from previous results.

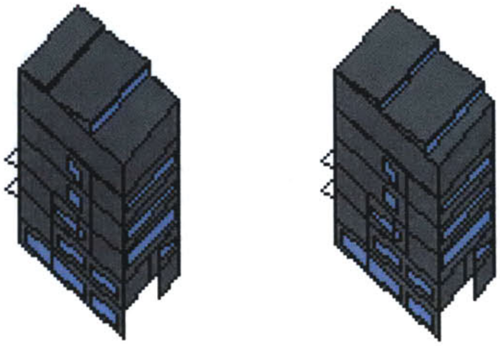


Figure 7.45 Best roof tilt solution generated by the GA [left] and initial guess of what might be the best solution [right].

Our guess performed slightly better in terms of lighting, but the heating and cooling loads were higher than for the GS solution, due to the larger roof monitor on the south, so the overall energy consumption of the latter was lower. It should be said, however, that the differences are negligible.

7.13 Rapid Prototyping

Rapid Prototyping [RP] technologies were used at the final stages of this study to allow for a more detailed observation of the GS solution. Architectural models have always been one of the best ways for architects to assess and relate all the components of a design. Using physical models permitted an in-depth understanding of all the design changes proposed by the GS, particularly in more intricate spaces as the ones containing the loggia and the zenithal light sources. The models allowed an immediate assessment of the relation between fenestration solutions and the spaces where they were located. It also made possible a more immediate comparison between the existing design by Siza and the GS solution.

For that purpose, a 3D physical model of each level was built [figure 7.46]. The 6 levels could be all put together to form the complete building model [figure 7.47]. A two-axes Stratasys FDM [Fuse Deposition Modeling] 3D Modeler was used, which deposits high-resolution lines of plastic to form the physical model, based on a 3D CAD model.

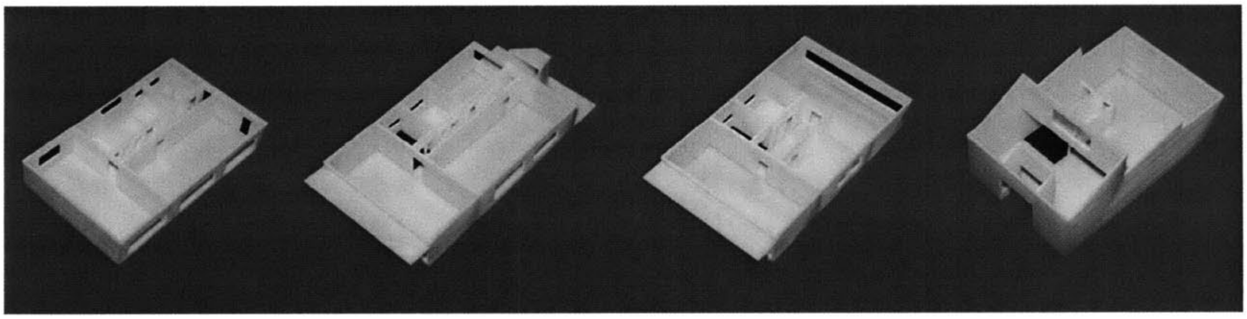


Figure 7.46 FDM models of several floors of tower H [from left to right, floors 1, 2, 3 and 6].

The impact of using RP technologies coupled with a generative system like the one presented in this chapter becomes more obvious when one considers that the GS generates not only a single 'best' solution, but an entire population of high performance solutions. The relative merit of those solutions from an aesthetics point of view can better be evaluated by the architect with the help of fast production of the several models. Introducing other variables in the GS like overhangs and shape manipulation [like the roof tilts] makes the different solutions harder to compare just with the help of computer generated images. Using RP, it is possible to produce quick models of the generated designs as a visual help to the evaluation process.

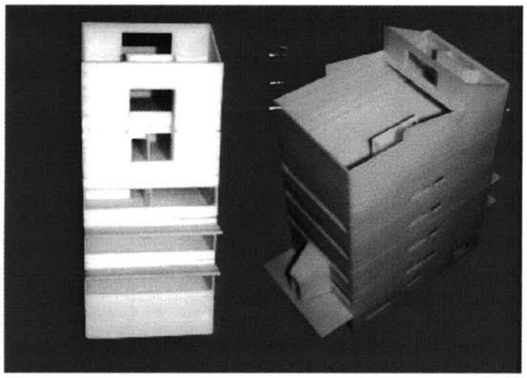


Figure 7.47 FDM model of the complete building

Apart from the RP method used in this study, other rapid fabrication technologies can be applied. Two axis laser-cutters enable the cutting of material such as wood, cardboard, or plastic, that can then be assembled into the final 3D form; water jet cutters are mainly used to cut wood, rubber or metal pieces under high pressure of water jets. This technique enables the unfolding or flattening of the individuals pieces that are water jet cut, which can later be bent to achieve the original shape. 3D Sterolithographic laser printing eliminates layers of a resin block, allowing fast execution of 3D models. Computer Numerically Controlled [CNC] machines, with multi-axis milling capabilities, can translate three-dimensional data into construction models, and constitute technological facilities that could also be used to create formwork for casting of final design solutions, in more complex design solutions.

Fabrication and concept in this case can be seen as a holistic paradigm, where theoretical premises are embedded in the genesis of the object as much as in the technical apparatus that permits its creation. By relating this new Generative System to Rapid Prototyping techniques it is not only possible to generate and evaluate computationally generated forms, but also to evaluate spatially qualities of the several GA solutions, enabling thus a full integration of technologic processes in the search of an appropriate architectural solution.

7.14 Conclusions of the chapter

The Generative System proved to be flexible enough to incorporate constraints that allow the user to manipulate certain architectural design intentions. The close coincidence between GS and Siza's solutions in some situations was of particular interest. On the other hand, the departures from the existing design proposed by the GS suggests that this generative system may be a useful tool in exploring multiple paths during the design process.

Another interesting dimension of the GS is its capability to account for interactions between different elements of the building, and to make the design for each specific element dependent on its integrated role on the architectural whole. The relations between the solutions for the loggia and the roof monitors, or between south and east facing windows in some of the studios, are a demonstration of that capability. The possibility of extrapolating from the algorithm's results to other dimensions like building geometry or spatial organization suggested new directions for further work where these aspects may also be manipulated by the GS.

The GS was able to generate, within language constraints, solutions that have low energy consumption levels. Results suggest this GS may be a useful tool to architects during the design process, by identifying potentially problematic areas and suggesting ways to approach them. It can act both as a generative tool and simultaneously as a diagnosis mechanism for problems occurring in a design that is being considered by the architect. The range of solutions the GS offered to the different geographical locations showed that the system is able to adapt the architectural design to the climate where it is located, even within the same language constraints.

The first attempts to perform shape manipulation using this GS proved that it is possible to use the system to alter building geometry in order to make it more adapted to its environment. Further work was developed to expand this interesting area of research, and will be presented in chapter 9.

Finally, in relation to RP applications, different kinds of technologies are constantly shaping the relationships between design concepts and decision-making. CAD/CAM technologies can play a major role in the evaluation steps of the design process, especially when the latter makes use of generative systems, where many solutions are developed at the design stage.

CHAPTER 8 Multicriteria Optimization using a Pareto-Front Method

8.1 Introduction

This chapter will be devoted to the study of multicriteria optimization methods using Genetic Algorithms, including applications to architecture-related problems. Most real world problems, and in fact most architecture design problems, are multicriteria in nature, in that there is often more than one objective the architect is trying to achieve while creating a particular solution. Even more, those objectives are often in conflict with each other, as a design feature that may increase the solution's quality from one point of view may have a detrimental effect on other performance criteria. This is particularly true for environmental problems, as we have discusses in previous chapters.

As a consequence, there is often no 'best' solution to be found that will simultaneously satisfy all the criteria. Instead, there will be solutions that perform well in some aspects, but worse in others. There will also be some middling solutions, that perform reasonably well in terms of several criteria, although they are not particularly good in any of them. So, when speaking of multiple criteria, we are often talking about compromises, trade-offs, and decision-making.

Multicriteria approaches are included in this chapter because they are seen as a major problem that decision-makers often face. Until now, the work presented remained in the realm of single-criteria studies, aggregating several factors into a single number that 'quantifies' the quality of a solution. In this chapter, we will further zoom into the several interacting factors, for example, decomposing a solution's behavior in terms of daylighting and of space conditioning [heating and cooling], and seeing which solutions perform better for each criteria. Other factors will also be introduced, such as the construction costs of a particular solution versus the energy savings achieved over the year, or the greenhouse gases emissions [GGE] embedded in the construction materials used versus the reduction in GGE due to less energy spent in the building. These are all trade-offs that designers often do not have enough information to deal with in an objective fashion.

In the first part of this chapter, several multicriteria methods described in the literature are reviewed. The most common methods are plain aggregating approaches, where the user defines weighting factors to attribute to the different criteria, achieving again a single figure of merit attached to the solution. Those methods are quite unsatisfactory, for reasons that will be discussed later in the chapter.

Instead, we propose another type of approach, where the search mechanism looks for the best possible trade-offs between the several criteria the designer has included in the Generative System, and the final decision is left for the architect to make. What the GS does is providing the architect with valuable and hard-to-find information on what solutions would provide the best trade-offs, but the GS will not automatically 'decide' on a solution. These frontiers of best trade-offs are called Pareto frontiers, and will be thoroughly discussed in this chapter.

Before the Pareto approaches, we will also discuss other population-based methods made possible by the use of GAs, but that do not formally use the concept of Pareto optimality. As previously discussed in chapter 5, when GAs were compared to other methods like Simulated Annealing, the fact that GAs use a population of solutions simultaneously makes them a particularly suitable method for multiobjective studies. Multiobjective GAs are usually known as MOGAs.

In this chapter we will also implement a Pareto GA, and will perform several experiments using different objective functions. In the next chapter, that deals with shape generation, the Pareto front concept will be pushed even further towards deeper insights into what factors play an important role in shaping different aspects of the environmental performance of buildings.

8.2 The concept of Pareto optimality

Genetic Algorithms are methods for approximate optimization which have been successfully applied to several problems difficult to solve exactly by conventional optimization methods. Although GA's have been commonly used in single-criterion optimization problems, a large number of real world problems are multi-criteria in nature, and these multiple criteria should be optimized simultaneously. In certain cases, objective functions may be optimized separately from each other to gain insight into each performance objective. However, suitable solutions to the overall problem can seldom be found in this way. Optimal performance according to one objective often implies unacceptably low performance in one or more of the other objective dimensions, creating the need for a compromise to be reached.

The Multi-objective optimization problem seeks to find the point x = (x_1, ... x_n) which minimizes [or maximizes, as the problem setting may be] the values of a set of objective functions f = (f_1,... f_n) within the feasible region S. Unlike single objective optimization problems, the optimal solution to this problem may not exist because of trade-off characteristics among the objectives. Hence the concept of the Pareto-optimal set, a family of points which is optimal in the sense that no improvement can be achieved in any objective without degradation in others, is introduced as solutions to Multiobjective Optimization Problems [MOP].

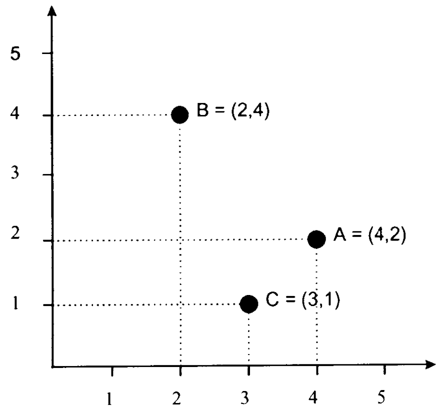
Pareto optimality makes use of the concept of dominated and non-dominated solutions. To explain the idea of a dominated solution, x dominates y if x is better than y for at least one objective function, and at least as good on all the others (Tamaki, 1996). A solution is Pareto optimal if it is not dominated by any other solution.

More formally, Pareto optimality can be written as follows:

In a maximization problem, given two solutions x,y $\in$ S, x dominates y if $\exists i \in \{1,2,3,....z\}$ such that f_i (x) > f_i (y) and that

$\forall j \in \{1,2,3,.....z\}$, $f_j(x) \geq f_j(y)$.

Figure 8.1 graphically illustrates the concept of a dominated solution.

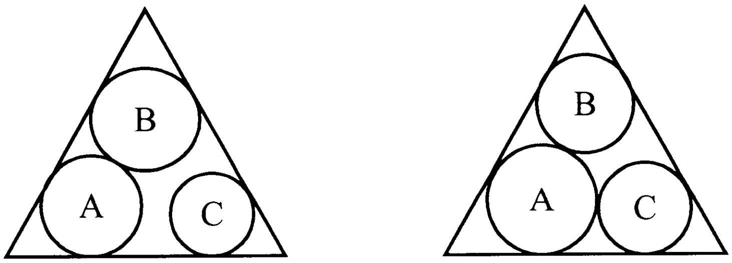


8.1 Dominated and non-dominated [Pareto] solutions for a maximization problem. C is dominated by A. Both A and B are non-dominated or Pareto solutions

The points represent feasible solutions to a multi-objective maximization problem, where each x and y value corresponds to the solutions to each individual objective function. Point A (4,2) dominates point C (3,1) since it has both higher x and y values. An hypothetical point D (4,1) would also be dominated by point A, since D would be as good as A in the x value, and worst than A is the y value. Points A (4,2) and B (2,4) are not dominated. They represent good trade-offs between the 2 objective functions. Point A performs better than B in terms of the x values, but the inverse is true for y values. This means no point dominates the other, and they are both Pareto-optimal [or non-dominated] solutions.

Stated more formally:
x_0 is Pareto-optimal if there doesn't exist any $x \in F$ such that x dominates x_0.

Another way of understanding the concept of Pareto optimality is, in simple terms, that Pareto optimality describes a solution for multiple objectives where no part of the solution can be improved without making some other part worse. This concept has been graphically depicted by Petrie et al., (1995) and is shown in the figure below. The circles represent objectives that are best satisfied when the area of the circle is maximized. The triangle represents the feasible region and the constraints require that the circles cannot overlap and must remain within the triangle.



Not Pareto Optimal
C can increase without
reducing A or B

Pareto Optimal
C cannot increase without
reducing A or B

Figure 8.2 Another illustration of Pareto optimal solutions

The left triangle represents a solution which is not Pareto optimal. That is, the area of circle C can increase without reducing the area of circles A or B. Conversely, the right triangle illustrates a Pareto optimal solution, where the area of circle C cannot increase without reducing the area of circles A or B.

Since any Pareto-optimal solution is a rational solution to a MOP, it can be said that the goal of solving a MOP is to obtain the Pareto-optimal set, or at least to sample solutions from that set as uniformly as possible, a goal that may be achieved by the use of Genetic Algorithms. This chapter reviews current Genetic Algorithm-based approaches to solving multiobjective problems, by discussing their similarities and differences.

8.2 GAs in multiobjective optimization problems [MOPs]

The search process of GA's uses a population of solutions in parallel instead of searching from a single point, as most optimization procedures do. This makes them particularly suitable for application to multi-objective problems using Pareto optimal sets, since in this approach the user requires as a final output a number of solutions [Pareto-optimal front] instead of a single solution. A GA can search for these solutions in parallel, and present to the decision-maker a set of solutions that represent the best trade-offs between the several objective functions. As it was mentioned before, in this type of MOPs there is usually not an absolute optimum, since most of the times the different objectives are in conflict with each other. It is up to the decision-maker to determine where in the Pareto set frontier the chosen solution is to be located, that is, what trade-offs is the decision-maker willing to make. The search procedure will only provide valuable information about the best trade-offs, but will not substitute the decision- maker.

There are, however, several approaches in the literature for applying GAs to MOPs. In the simplest case, a GA can be applied to a MOP by simply transforming it into a single objective optimization problem. This is done by combining the several objective functions using weighting factors, based on some knowledge of the problem, and eventually reaching a single figure of merit that supposedly reflects the overall performance of the solution according to all the different objectives [plain aggregation methods]. Then GA's can be applied repetitively, if necessary, changing the values of the weighting factors at each run, to gain some insight into the relative importance of each of the objectives. This approach, however, does not reveal the attractive properties of GA's for MOP, which are exactly their capacity to simultaneously search a population of solutions.

Apart from these plain aggregation methods, this chapter will review population-based approaches using GAs to generate solutions for MOPs but without explicitly using Pareto optimization methods, and finally the different methods described in the literature for using GAs in Pareto optimization.

One of the main factors that distinguishes these different approaches is the way preference is incorporated into the decision process. According to how much preference information is incorporated in the fitness function, approaches range from complete preference information given [plain aggregating approaches], to no preference information given, as with Pareto-based ranking approaches.

8.3 Various MOGA approaches

8.3.1 Plain Aggregating Methods

In these techniques, typically the objective functions are combined into a 'weighted sum' to use conventional GA techniques to solve the combined function. This is done by applying user-defined weighting factors to each of the objective functions under consideration. After the assignment of weighting factors, no further interaction with the analyst is required for the GA to look for the 'optimal' answer.

The main problem with this approach is that the final solution is heavily dependent on the weighting factors used, thus changing the weighting factors can lead to dramatically different solutions. Also, because all the performance objectives are mixed together into a single 'figure-of-merit,' it is hard for the analyst to understand in what aspects does a certain solution perform better than another, and what are its main advantages and weaknesses. Another handicap related to this approach is that for the analyst to define the weighting factors he must have a very good insight into all the factors involved in decision-making, what may represent a serious problem since many times the analyst is not the decision-maker. A possible approach to overcome this problem is to run the optimization program several times using different weighting factors. This turns the analysis into another kind of parametric study, where the parameters are not only the variables of the problem but also the values of the weighing factors. As typically decision-makers want a range of solutions to choose from and some understanding of alternative solutions, the analyst must resort to this approach to provide different alternatives. Nevertheless, this is never an exhaustive procedure, with only a few cases being evaluated, what brings into the realm of an optimization process [GAs] some of the disadvantages related to simulation methods that were stated before.

There are a number of techniques cited as alternative plain aggregating approaches to the weighted sum discussed above. These include:

target vector optimization	[a minimization of distance in objective space]
goal attainment	[minimization of the weighted difference between goals]

An example of a study where weighting factors are combined with a GA is by Syswerda (1997), who applies genetic algorithms to resource scheduling, specifically to resources assigned to use flight simulator equipment for development support of the F14 jet fighter. The implementation of a scheduling method isolated the GA from all problem specific details; the GA only had to work with permutations of a list of things. This isolation was due to a 'front-end' schedule builder and schedule evaluator, that worked as 'black boxes' in relation to the GA. The builder created valid but not optimized schedules from a list of tasks initially assigned on a first input, first scheduled basis. The built schedule is then evaluated by the schedule evaluator. The evaluator only has to consider preference information input [weighting factors] by the user or constraints associated with the site [an example of a site constraint is the maintenance of the equipment]. The task of the GA was to optimize the final schedule.

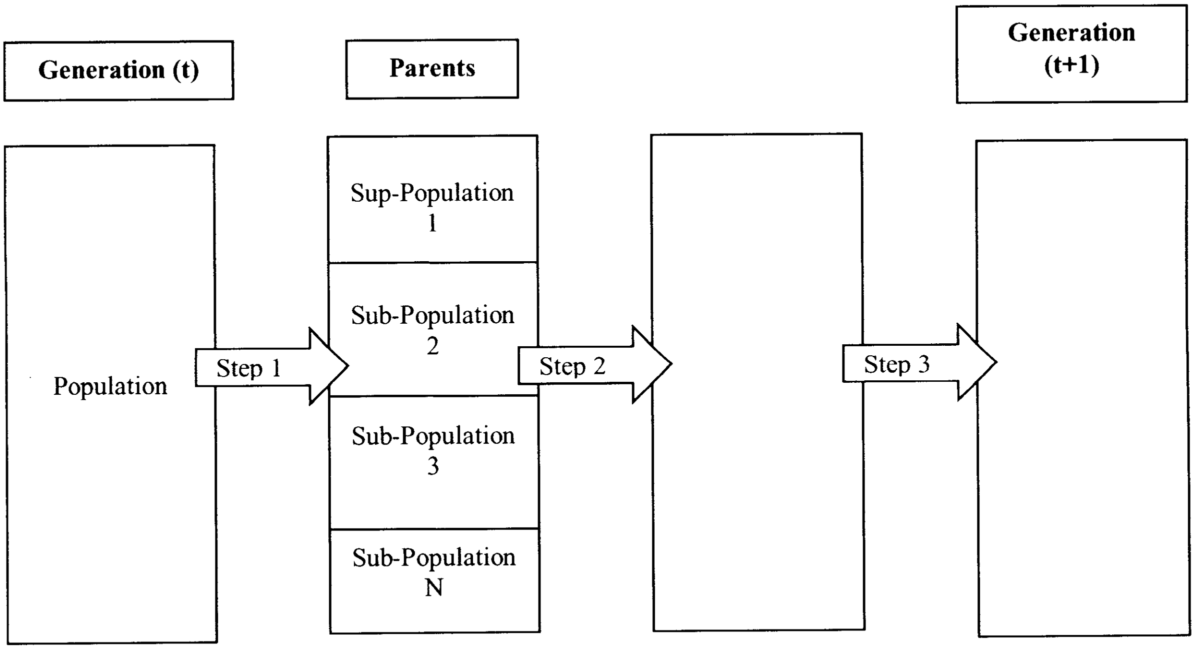
8.3.2 Population-Based Non-Pareto Approaches

Another class of multiobjective genetic algorithms is referred to as non-Pareto approaches. These methods, according to Fonseca (1995), attempt to "promote the generation of multiple nondominated solutions, a goal at which they reportedly achieved a reasonable degree of success. However, none make direct use of the actual definition of Pareto optimality." The definition of Pareto optimality was discussed in section 8.2. While in Pareto approaches these techniques monitor for nondominated solutions, in population-based approaches the GA operates on the solution set without the Pareto ranking and assignment of "Pareto fitness" used by the Pareto techniques.

The first pioneering technique of this type was developed by Schaffer (1985) and is referred to as VEGA (Vector Evaluated Genetic Algorithm). Schaffer characterizes VEGA as being capable of handling vector, not scalar solutions. He specifically distinguishes his technique from the plain aggregating approaches cited in section 8.3.1, as being able to handle dimensions that are not commensurable.

In general, the algorithm cycles in a two step process. The first step is a selection or survival-of-the-fittest step and the second is a mating or recombination step. Tamaki (1996) indicates that "sub-populations of the next generation are reproduced from the current population according to each of the objectives, separately (in a parallel selection method)." This first step results in a set of solutions that tends to be very strong in one objective and relatively weak with respect to other objectives. In the second step, crossover occurs between solutions favoring different objectives via a shuffle step as shown in figure 8.3 [adapted from Schaffer (1985)].

Where :



Step 1 is select n subgroups using each dimension of performance in turn
Step 2 is shuffle.
Step 3 is apply genetic operators

Figure 8.3 Schematic process of VEGA

The "shuffle" step is a critical one and it is where most of the discussion about the technique appears to reside. Fonseca (1995) indicates that "Schaffer used proportional fitness assignments, in turn, proportional to the objectives themselves. The resulting overall fitness corresponded, therefore, to a linear function of the objectives where the weights depended on the distribution of the population at each generation. [...] As a consequence, different non-dominated individuals were generally assigned different fitness values, in contrast with what the definition of nondominance would suggest." This means that the fitness value was a function of not only its relative fitness but also the distribution of the subpopulation at the time it was generated. This might be clearer if shown algebraically as:

Fitness = Objective1 (weight) + Objective 2 (weight) + ...

where: weight is a function of the distribution

Because of this, and the linear combination of the objectives implicit to VEGA, the population tends to split into species particularly strong in each of the objective, particularly in the case of concave trade-off surfaces. Concerns about the concave shape of the edge of a feasible region were expressed by Schaffer (1985), due to the effect of pressure favoring extreme performance on at least one dimension. The result of VEGA is that when determining the population of solutions on the edge of a feasible region, the technique will tend to find solutions at extremes of performance for given objectives. This is because of the linear combination of results associated with each of these objectives individually.

Schaffer was thus concerned about missing what he calls the "middling" performance or compromise solutions. He was aware that VEGA tended to emphasize solutions towards the extremes, what was most problematic in the case of concave situations where the middling solution may not be close to the solutions at the extremes.[3] This can be visualized in figure 8.4 below.

[3] Subsequent work by Fleming and Pahkevich (1985) cited by Fonseca (1995) demonstrated that points in concave regions of a trade-off surface couldn't be found by optimizing a linear combinations of the objectives, for any set of weights.

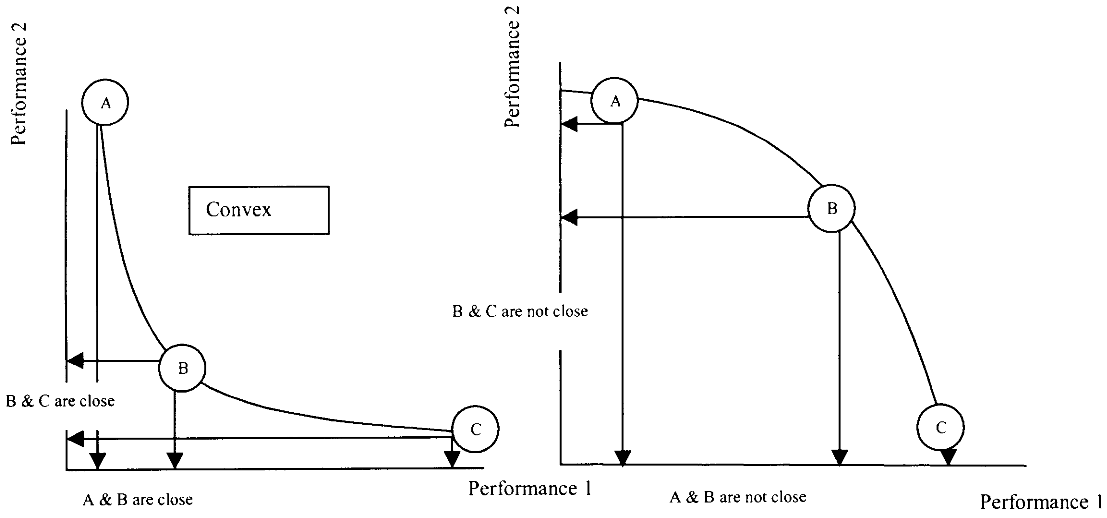


Figure 8.4 Convex (left) and Concave (right) Pareto Optimal Front

While VEGA may miss the middling solution for both convex and concave solutions spaces, the damage is more severe for concave solution spaces. VEGA will result in finding solutions A and C while the middling B is left out. In the convex case, that problem is relatively small compared to the concave case, as B still results in performance relatively close to A and C. However, in the concave case this is not true. In a maximization problem, point B may be very attractive and VEGA will miss it altogether.

In order to preserve more middling performers Schaffer tried to implement two heuristics to improve performance. The first of these was to give an extra selection preference to locally non-dominated individuals. However, this accelerated solutions to premature convergence because if there were too non-dominated solutions other individuals died off quickly and the solution converged prematurely. This is because there was not enough genetic diversity to continue to generate alternative solutions. A second heuristic Schaffer tried was to randomly select the first parent and then the second parent that was at maximum “distance” from the first parent. In test conditions, Schaffer cited that this proved no better than random mating.

A second non-Pareto technique was cited by Fourman (1985). Selection was performed by comparing pairs of individuals, each pair according to one of the objectives. Fourman tried two implementations of this. The first was having user-defined priorities on the objectives and comparisons were made based on the objective with the highest priority. The second implementation consisted of randomly selecting the objective to be used in each comparison. This is similar to VEGA, in weighting the probability of each objective chosen to decide the comparison. However, the pair wise comparisons prevent linear combinations of the objectives, because scale information is ignored.

The differences between VEGA and Forman's implementation can be highlighted using an example adapted from Fonseca (1995), who also provides a more complete description of these functions and figures. Two functions, representing objective value functions to be minimized, can be represented by:

$F1(x_1, x_2) = 1 - \exp\ (- (x_1 - 1)^2 - (x_2 + 1)^2)$

$F1(x_1, x_2) = 1 - \exp\ (- (x_1 + 1)^2 - (x_2 - 1)^2)$

A graph of function F1 space is shown in figure 8.5 (the inverse of the function is plotted for ease of viewing).

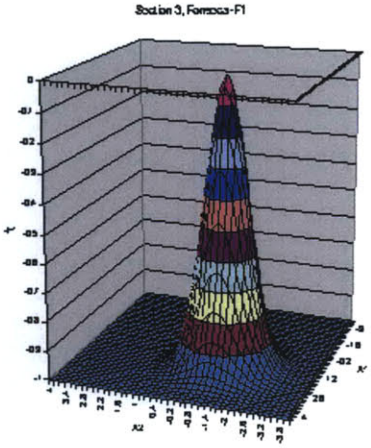


Figure 8.5 Function F1

A view of the search space as seen by VEGA can then be created by summing the solution to the functions and then ranking each solution of x_1 and x_2 relative to each other. The result is referred to as a cost space and is shown in figure 8.6, and that has two peaks. These peaks will be identified by VEGA and the answers given by VEGA will reside on these peaks resulting in "speciation" or convergence on 'best' by objective function solutions. This "speciation" develops as VEGA attempts to distribute itself on these peaks in a balanced way.

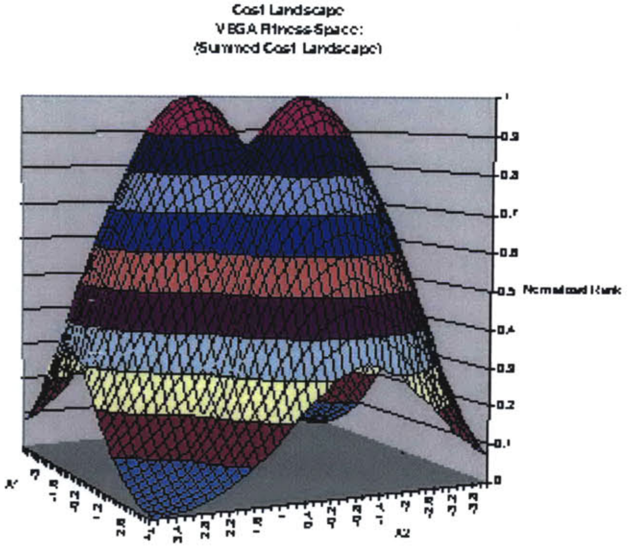


Figure 8.6 Cost function of VEGA with functions F1 and F2

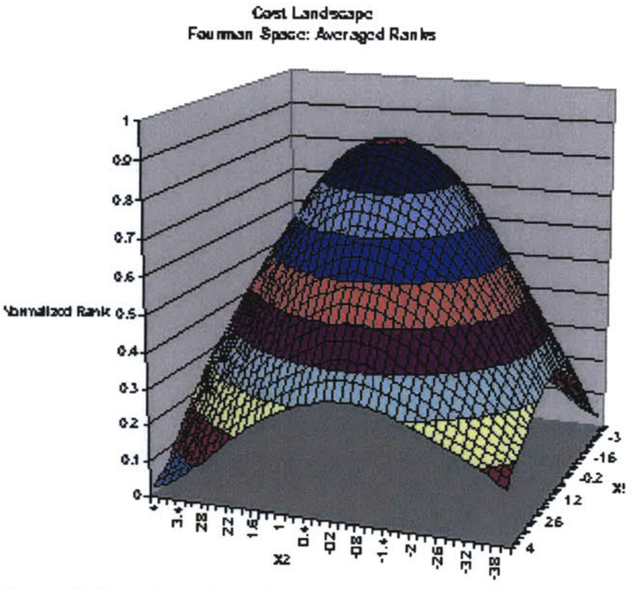


Figure 8.7 Cost function using Fourman's implementation

Similarly, Fourman's implementation can be plotted by averaging the ranks of the solutions for F_1 and F_2 for each x_1 and x_2. This cost function is shown Figure 8.7. This cost function is quite different, with a single peak between the two peaks in the VEGA space. The concavity of the trade-off surface is no longer apparent and there is a single peak near the Pareto optimal set.

8.3.3 Pareto-Based approaches using Genetic Algorithms

One of the ways to extend GAs to multiobjective optimization has been to try to overcome the limitations imposed by the single figure-of-merit objective function. The main question then becomes what happens when there are different objectives that need to be somehow compared. Several approaches have been developed which base selection and reproduction of individuals on both the objective values themselves and their corresponding dominance property. These are known as Pareto-based approaches. In Pareto-based approaches, no preference information is given. Search results provide the best possible trade-offs or compromises between the different and often conflicting objective functions by establishing the rank of any given solution in relation to other feasible alternatives. This provides a relative versus an absolute measure of how 'fit' a solution is.

The first Pareto-based approach, developed by Goldberg (1989), provides a method of ranking and fitness assignment based on the Pareto optimality of an individual in the population. A rank of 1 is assigned to non-dominated individuals in the population and these individuals are subsequently removed from the current population. A new set of non-dominated individuals is identified in the modified population (i.e. the original population without the rank 1 individuals), assigned a rank of 2, and then removed from the population. This process continues until all individuals in the population have been ranked. Using this method, individuals with rank 1 are considered most 'fit' and are selected for survival.

A second approach, developed by Fonseca and Fleming (1993), uses a different method for ranking individuals in the current population. In their proposed method, the rank of an individual corresponds to the number of individuals in the current population by which it is dominated. Consider an individual x_i at generation t which is dominated by $p_i^{(t)}$ individuals in the current population. The rank of the individual can be defined by the following equation:

$$\text{rank}(x_i, t) = 1 + p_i^{(t)}$$

Using the above equation, a rank of 1 is assigned to the non-dominated (i.e. $p_i^{(t)} = 0$) individuals in a population and all non-dominated individuals in a population will be assigned the same rank. Dominated individuals in the population are penalized based on the number of individuals in the population which dominate them. The algorithm then sorts the population according to the multi-objective ranks previously determined. A fitness value is assigned to individuals in the population and represents a measure of performance of a member of the population. The fitness of an individual is a single number in a given scale and within certain boundaries to provide an absolute measure which the genetic algorithm uses to cull the population in a certain generation to allow more fit members of the population to remain and produce offspring. Fitness is assigned to individuals by interpolating from the best rank to the worst using a linear function, in most cases. The fitness values of individuals with the same rank are averaged, meaning that these individuals will be sampled from the population at the same rate. Figure 8.8, adapted from Tamaki et al. (1996), illustrates differences in rank for a population of individuals (A-H) using the Pareto-based ranking methods proposed by Goldberg versus Fonseca and Fleming. The x and y axes represent two separate objective functions, f_1 and f_2, respectively.

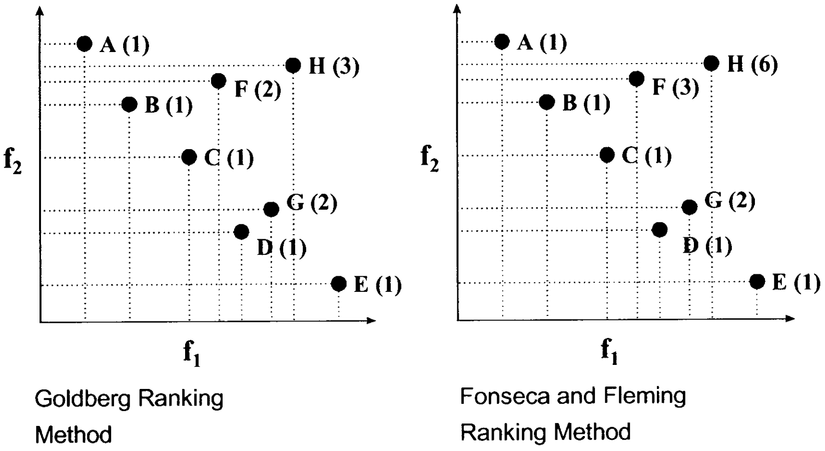


Figure 8.8 Goldberg's and Fonseca's ranking methods

A third Pareto-based approach, developed by Horn, Nafpliotis, and Goldberg (1994), is known as tournament selection and uses dominance properties of individuals in the population as the criteria for selection. This is one of the most widely used selection techniques for genetic algorithms. In tournament selection, a set of individuals is randomly chosen from the current population and the best of this subset is placed in the next population. By adjusting the size of the tournament, selection (or domination) pressure can be increased or decreased and speed of convergence of the model can be controlled. The tournament selection method described here maintains multiple Pareto-optimal solutions and avoids convergence to a uniform solution. Tournament selection consists of randomly selecting two candidates from the population which compete with each other in a 'tournament.' A comparison set of individuals is also randomly selected from the population and is used to increase domination pressure. Each of the candidates selected for the tournament is then compared against each individual in the comparison set. If the first competitor in the tournament dominates all individuals in the comparison set and the other competitor is dominated by at least one individual in the comparison set, the first competitor wins the tournament and is selected for survival and reproduction. Figure 8.9 is a graphical illustration of Pareto-based tournament selection, adapted from Tamaki et al. (1996):

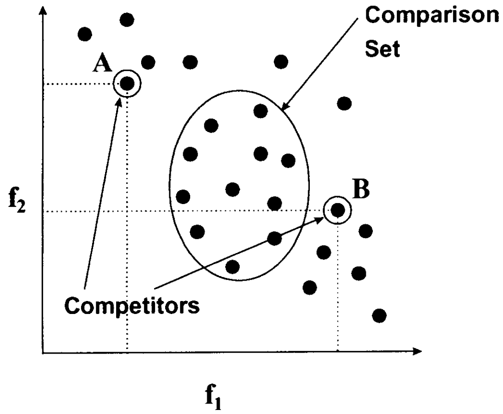


Figure 8.9 Pareto-based tournament selection

Referring to figure 8.9, f_1 and f_2 represent two different objective functions being minimized. Two individuals, A and B, have been selected to compete in the tournament. Individual A is not dominated by any individuals in the comparison set. Conversely, individual B is dominated by some of the individuals in the comparison set. Therefore, individual A will be selected for survival.

In the case where both competitors are either dominated or non-dominated, the outcome of the tournament is determined using a fitness-sharing technique. The goal of fitness sharing is to degrade an individual's objective fitness (f_i) if too many other individuals are in the same niche [that is, are similar solutions], in order to promote diversity in the final population. This is done using a niche count (m_i) calculated for that individual. The degradation in the individual's objective fitness value is obtained by dividing the objective fitness by the niche count to obtain shared fitness as follows:

Shared fitness = f_i/m_i

The niche count (m_i) is an estimate of how crowded the neighborhood of individual i is and is calculated over all individuals in the population, using the following formula:

$m_i = \sum$ Sh[d[i,j]]
$j\in$ Population

where: d[i,j] = distance between individuals i and j
Sh[d] = Sharing function; a decreasing function of d[i,j] such that Sh[0]=1 and Sh[d $\geq$ σshare] = 0.

The quantity σ_{share} is known as the niche radius, which is fixed by the user and is an estimate of minimum separation desired or expected between the goal solutions. Stated differently, the niche radius is the biggest distance at which two individuals in the population will be considered to be within the same niche. Individuals within σ_{share} distance of each other, where d[i,j] $\leq$ $\sigma_{share,}$ will degrade each other's fitness since they are in the same niche. Individuals spaced further apart than σ_{share}, where d[i,j] $\geq\sigma_{share,}$ will not degrade each other's fitness since Sh[d $\geq$ σ_{share}] equals zero. The fitness-sharing technique creates a condition where convergence will occur within a niche, but convergence of the full population is avoided. As one niche becomes full, the niche count value will increase resulting in a lower shared fitness value in comparison to other niches. This way, it is expected that the final population will sample individuals all across the Pareto frontier, as it will be better discussed in section 8.4.

A fourth Pareto-based approach, proposed by Tamaki, Mori and Araki (1995), is known as the Pareto reservation strategy. Using this method, non-dominated individuals in a population at each generation are all reserved for the next generation. If the number of non-dominated individuals in the current population is less than the population size of the next generation, the remaining individuals in the next generation are selected using the parallel selection method of the Vector Evaluated Genetic Algorithm (VEGA). Later revisions of the Pareto reservation strategy incorporate fitness sharing techniques for selection of individuals for the next generation from the set of non-dominated individuals. In VEGA, separate sub-populations (1,2,3,...n) are reproduced from the current population according to each of the objectives. This is commonly referred to as the parallel-selection method. Figure 8.10 below, adapted from Tamaki et al. (1996), is a schematic representation illustrating the Pareto reservation strategy.

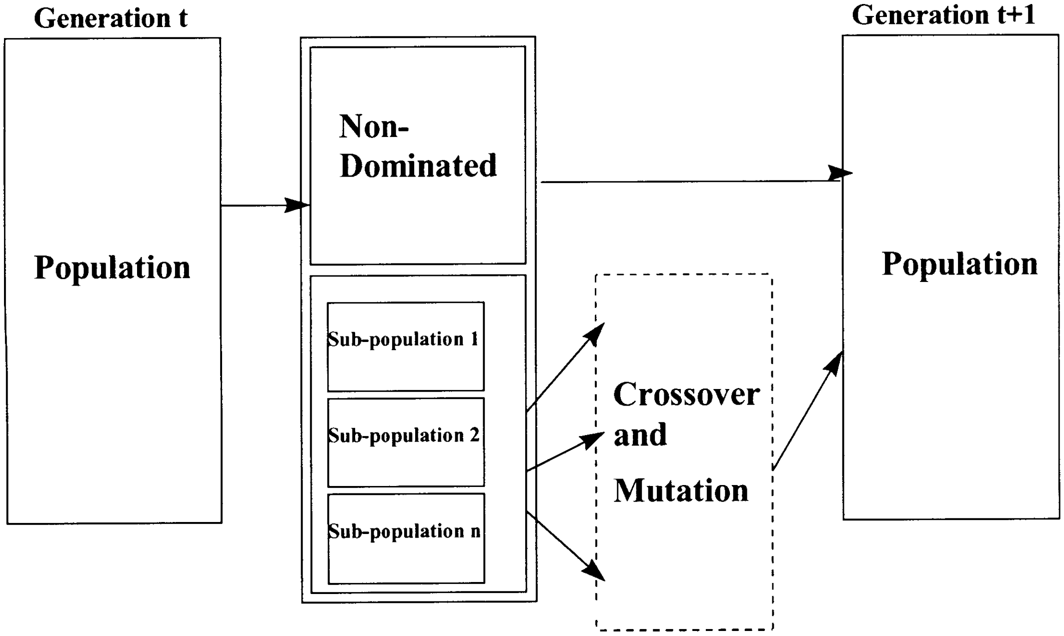


Figure 8.10 Pareto reservation strategy

A fifth Pareto-based approach, proposed by Tanaka, Yamakura and Kobayashi (1996 b), is known as the Pareto-optimal selection method. Using this method, all non-dominated individuals in a given population are retained and all dominated individuals are discarded. Duplication of individuals is not allowed to make the search process more efficient. The process consists of applying the selection operator after many individuals have been produced using mutation and recombination operators. After these operators are applied, non-dominated individuals are selected for survival from the population. Both parent individuals and individuals produced by recombination and mutation can be included in the next generation. Application of mutation and recombination operators will result in variation in population size over generations. In some cases, a "combined" selection method is applied, where non-dominated individuals and also individuals with a favorable Pareto-based rank are selected for survival. Figure 8.11, adapted from Tamaki et al. (1996), graphically illustrates the Pareto-optimal selection method.

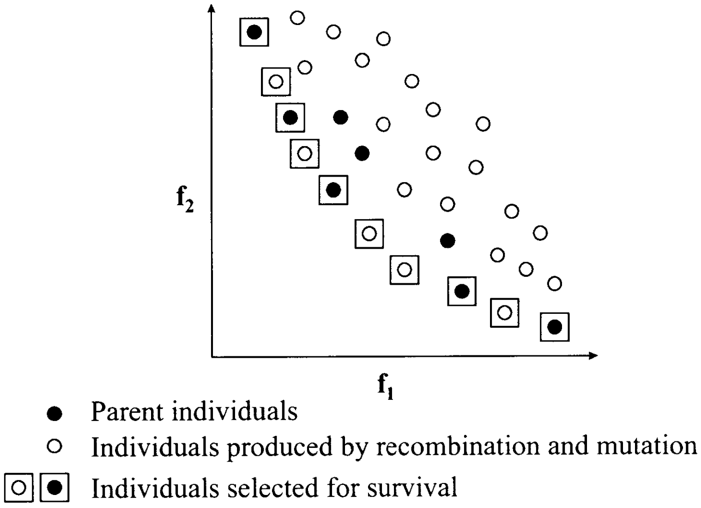

f₂
f₁
● Parent individuals
○ Individuals produced by recombination and mutation
Individuals selected for survival

Figure 8.11 Outline of Pareto-optimal selection

8.4 Niche induction strategies in Pareto optimality

Niche induction techniques, modeled after the biological idea of niching, represent yet one more way that Genetic Algorithms have been manipulated to address different problems encountered in solving complex multiobjective systems.

"In nature, a *niche* is viewed as an organism's task in the environment, and a *species* is a collection of organisms with similar features. The subdivision of the environment on the basis of an organism's role reduces inter-species competition for environmental resources, and this reduction in competition helps stable subpopulations to form around different niches in the environment" (Deb et al., 1989).

Niche induction techniques are used for several reasons. First, while Pareto-based ranking correctly assigns all non-dominated individuals the same fitness, it does not guarantee that the Pareto set will be uniformly sampled. When multiple equivalent optima are present, finite populations tend to converge to only one of these solutions, due to stochastic errors in the selection process. Once the algorithm has converged to this solution, the size of the gene pool is reduced to the size of the solution(s) gene pool. This phenomenon, known as *genetic drift*, has been observed in natural as well as in artificial evolution, and can also occur in Pareto-based evolutionary optimization. Additionally, recombination and mutation may be less likely to produce individuals in certain regions of the trade-off surface [for example, the extremes] than in others, causing the population to cover only a small part of the surface. The concept of *niching* [reducing competition to promote stable subpopulations], prevents the GA from converging to a single point on the non-dominated frontier. Using niching induction techniques to solve multiobjective GA improves over the previous methods by promoting the sampling of the entire Pareto set by the population.

Niching techniques are also used to find and maintain a diverse 'Pareto optimal population' (Horn 1994). Niching mechanisms, such as sharing and crowding, allow the GA to maintain individuals all along the non-dominated frontier. The maintenance of diversity is important since diversity along the non-dominated frontier helps in the search for new and improved trade-offs, thus extending the frontier. The fact that most crosses of parents on or near the front yield offspring also on or near the front, is evidence that "Pareto *diversity* helps Pareto *search*", and thus diversity is an important quality in a method of solving GAs (Horn et al.,1994).

Niche induction techniques find the Pareto optimal set and then apply a niching pressure to spread the population of the optimal set out along the Pareto optimal trade-off surface. Niched GAs have the ability to both find and maintain a diverse "Pareto optimal population" (Horn et al., 1994). Two main techniques using this niching idea have been implemented on GAs.

The first, crowding, was first developed in 1975. *Crowding* reduces the number of solutions in a "crowded" area by dictating that only a certain proportion of the population be permitted to reproduce each generation and removing similar solutions with matching alleles. A small set of parents is selected from the population, with each child replacing an individual from this set which is most similar to itself. In this method, however, stochastic replacement errors prevent the algorithm from finding all niches since all parents do not participate.

In applying crowding as a niche induction technique, two new parameters are needed. These are the Generation gap (G) and the Crowding Factor (CF). The Generation gap (G), "dictates the use of an overlapping population model in which only a proportion G of the population is permitted to reproduce each generation. To induce niching in the population, the following approach is adopted. When selecting an individual to die, CF individuals are picked at random from the population, and the one which is most similar to the new individual is chosen to be replaced, where similarity is defined in terms of the number of matching alleles. The new individual [chosen by usual selection methods] then replaces this chosen individual in the population" (Deb et al., 1989).

The second technique involves the use of a *sharing* scheme. A *sharing parameter*, σ_{share}, is introduced to control the extent of sharing by limiting the distance between solutions. Here σ_{share} is the niche radius, which is fixed by the user at some estimate of the minimal separation desired or expected between the goal solutions. "The parameter σ_{share} depends on the number of peaks and the upper and lower bounds of the solution space" (Deb et al., 1989). Sharing is generally used for maintaining stable subpopulations when the solution includes multiple niches and has been shown to be more effective at maintaining diversity than the crowding technique described above. As figure 8.12 indicates, a GA with sharing is able to converge and distribute trials at all the peaks of the functions by dividing the population into different niches, while a GA with crowding is unable to maintain subpopulations at all the peaks.

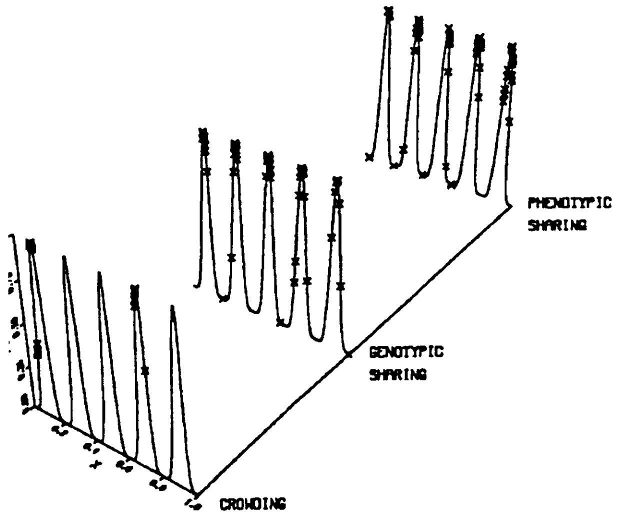


Figure 8.12 GAs with sharing distribute trials to all five peaks in this function, while a GA with crowding converges only to two peaks. (Deb et al., 1989).

Three methods of sharing will be discussed, including fitness sharing and phenotypic and genotypic sharing, as well as a means to improve on these methods, by implementing a mating restriction.

The first method, *fitness sharing*, aims to maintain the genetic diversity in multiobjective optimization. The goal of fitness sharing is to "distribute the population over a number of different peaks in the search space, with each peak receiving a fraction of the population in proportion to the height of that peak" (Horn et al., 1994). In this method, the fitness value of each individual is reduced if other individuals exist in its neighborhood. Individuals within σ_{share} distance of each other degrade each other's fitness, since they are in the same niche. As one niche 'fills up,' its niche count increases to the point that its shared fitness is lower than that of other niches. Therefore, convergence will occur within a niche, but the entire population will not converge. An individual located in a more crowded area will produce less offspring, and we will obtain a population distributed more uniformly over the Pareto-optimal set.

In *phenotypic sharing*, the parameter σ_{share} is used to set the distance between strings in the niche space. To better understand this parameter, "imagine that each niche is enclosed in a p-dimensional hyperspace of radius σ_{share} such that each sphere encloses 1/q of the volume of the space, where q is the number of peaks in the solution space" (Deb et al., 1989). This allows the optimization to perform sharing in a phenotypic environment [along the phenotypic Pareto optimal front], that is, the decoded parameter space.

Genotypic sharing, the use of sharing based on genetic proximity, or the genetic closeness of two individuals, uses a different solution space, and so the parameter σ_{share} must be defined slightly differently. It is defined as "the maximum number of different bits allowed between the strings to form separate niches in the population" (Deb et al., 1989). This different definition allows the GA to optimize using the entire genotype.

There are, however, drawbacks to using genotypic sharing. Unfortunately, this method is sometimes unable to maintain stable subpopulations at the peaks of relatively low value, especially in those functions with peaks of unequal value. This is shown in figure 8.13 (adapted from Deb et al., 1989).

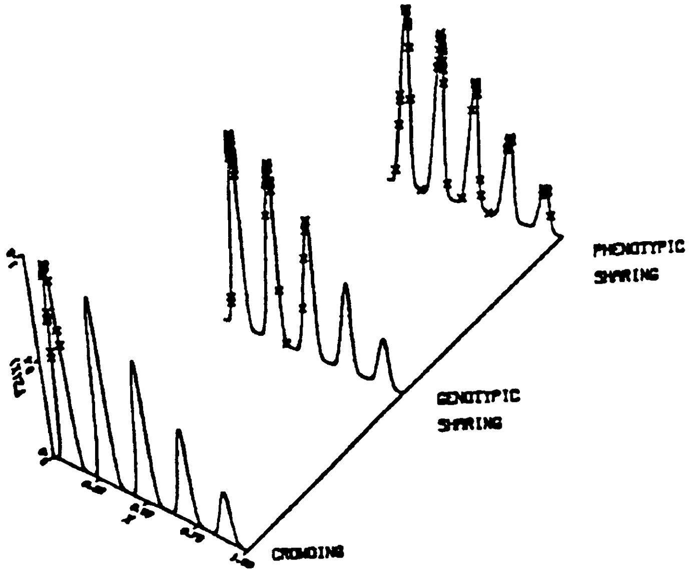


Figure 8.13 A GA with phenotypic sharing distributes trials to all five peaks, a GA with genotypic sharing to three peaks, and a GA with crowding to a single peak.

Adding a *mating restriction*, or biasing the way in which individuals are paired for combination, can greatly improve the performance of most of the sharing techniques previously discussed. “Once the sharing scheme clusters subpopulations of trials at the peaks, crossover between strings on different peaks may produce new strings that do not represent any peak. The presence of these lethal strings in the population degrades the on-line performance of the process. Therefore, to achieve better performance on multimodal functions, crossover between strings of different peaks must be reduced. In nature, this is achieved by creating separate species (or subpopulations) corresponding to each niche (or peak) in the solution space and restricting the mating between species. It has been shown above that a population can be divided into as many subpopulations as there are peaks by suitably selecting a parameter σ_{share}. A similar parameter can also be defined to create subpopulations that constitute a species. When the distance metric considered in both these cases is measured in the decoded parameter space, the so-called phenotypic mating restriction, the individuals with the corresponding distance metric less than σ_{share} are shared and the individuals with distance metric less than σ_{mating} are mated. To keep the analysis simple, the parameters σ_{share} and σ_{mating} are set equal. The mating restriction method imposed on the individuals is as follows: To find a mating companion of an individual, if an individual within a distance of σ_{mating} is found, mating is performed, otherwise another individual is tried. If no such individual is found in the population, a random individual is chosen" (Deb et al., 1989).

From figures 8.14 and 8.15 below, also adapted from Deb (1989) it is clear that phenotypic sharing is much more effective when a mating restriction is used. The distribution of offspring among the various peaks is more uniform and the quality of the offspring is consistently higher.

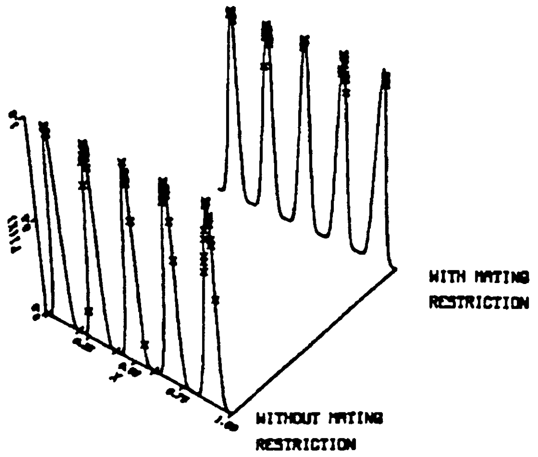
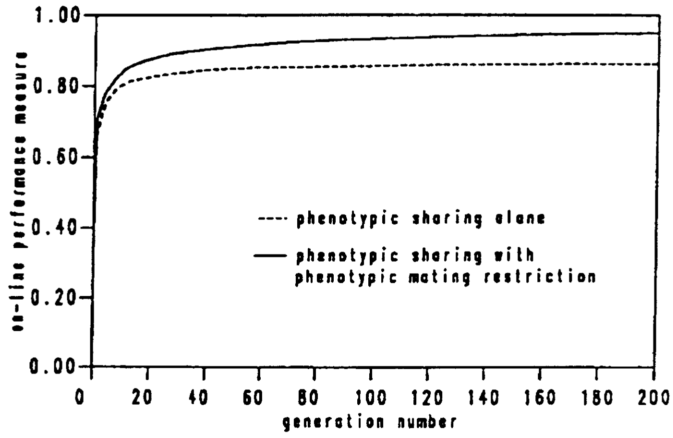


Figure 8.14 Distribution of trials on function F1 with phenotypic sharing alone and phenotypic sharing with mating restriction. Sharing with mating restriction reduces the number of lethal trials.



8.15 Comparison of on-line performance with phenotypic sharing alone and phenotypic sharing with phenotypic mating restriction

A final method incorporating the niche induction technique is known as *Nondominated Sorting Genetic Algorithms* (NSGA). "The idea behind the nondominated sorting procedure is that a ranking selection method is used to emphasize good points and a niche method is used to maintain stable subpopulations of good points. (...) The ranking classification is performed according to the nondominance of the individuals in the population, and a distribution of the nondominated points is maintained using a niche formation technique. Both aspects cause the distinct nondominated points to be found in the population" (Srinivas 1995).

This method is able to solve any number of objectives and has been shown to outperform several MOGAs [Multi-Objective GAs] and VEGA (Srinivas et al., 1995). The flow chart shown in figure 8.16, adapted from Srinivas et al. (1995), illustrates the process of using a nondominated sorting genetic algorithm.

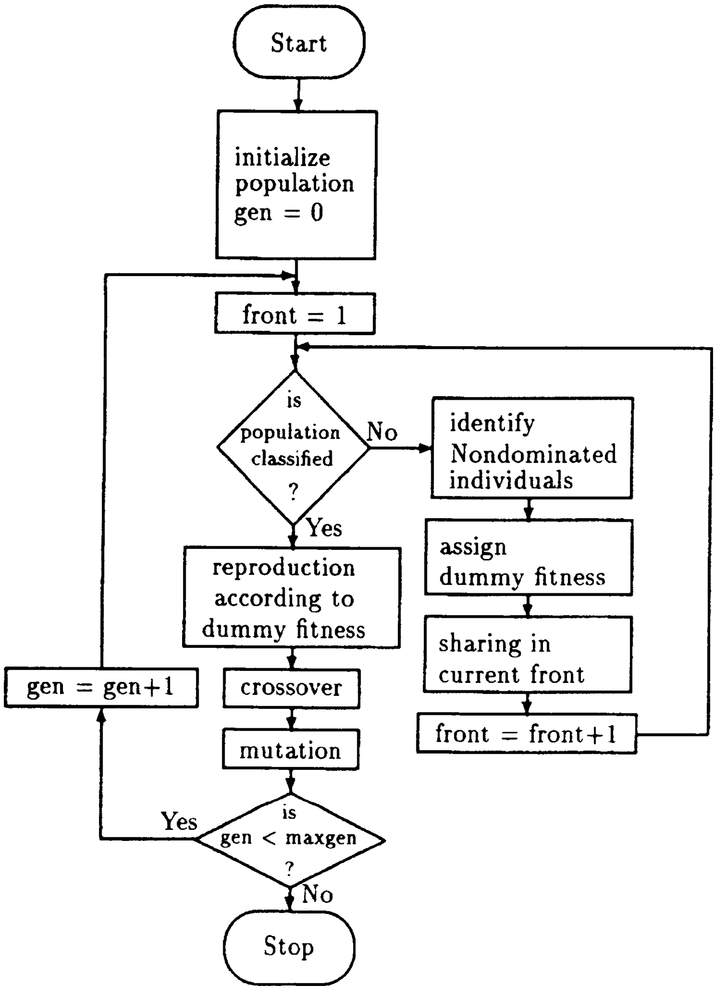


Figure 8.16 Flow chart of Non-Dominated Sorting Genetic Algorithm [NSGA]

The niche induction techniques described above are more complicated to implement than other strategies. However, these methods overcome the drawbacks of the previous ones by effectively finding the optimal solution along the Pareto front while maintaining the most diversity in the population.

8.5 Case studies and applications found in the literature

Few real world applications of MOGAs have been found in the literature, probably due to the fact this is a recent field of research. Pareto GAs are the most commonly found approaches, usually combined with some sharing/crowding strategy to ensure that most of the Pareto front will be sampled.

Tamaki et al. (1996) apply the Pareto reservation strategy [discussed in section 8.3.3] to a scheduling problem in a hot rolling process of a steel factory. The process contained three furnaces and one rolling unit, with slabs being provided from the preceding process. Each slab needed to be heated to a certain temperature in one of the furnaces, and then pressed in the rolling unit.

The scheduling problem was defined as to find the assignment of slabs to the furnaces and the pressing order of the slabs in the rolling mill.

The objectives to minimize were:

- the total quantity of fuel required for heating all slabs at all furnaces
- the total time for heating all slabs
- a complexity index determined by the difference of width, thickness and hardness between each pair of subsequently pressed slabs [related to changes required in the equipment when different slabs are processed after each other].

The existing constraints related to the quantity of fuel supply per unit time, and constrains both on the absolute position of the order in pressing for each slab, and on the relative order in pressing for each pair of slabs.

The chromosomes representing a solution for the problem were designed in the following way: the initial problem was divided into two sub-problems: an assignment problem (assigning slabs to furnaces) and an ordering problem (determining the pressing order). A chromosome was then formed by putting together two sub-strings α and β, each of length equal to the number of slabs.

Each sub-string was formed as follows:

α_i - the i-th position in string α - represented the index of the furnace to which the i-th slab was assigned [could be either 1, 2 or 3]
β_i - the i-th position in string β - represented the index of the slab that would be pressed in the i-th position [could vary between 1 and the total number of slabs]

The crossover operator used was a ten-point crossover. The mutation operator used was quite sophisticated and problem-specific: within the sub-string α, a randomly chosen allele, i.e., an index of a furnace, would be mutated with a prescribed probability. Within sub-string β, the order of two slabs selected randomly would be exchanged with a certain probability, as long as the resultant order would satisfy both the absolute and the relative constraints in the ordering problem.

This problem also used a local search method (hill-climbing) to improve the search results of GAs. If a better schedule was obtained by the local search method, it would substitute the original genotype. The local search was also defined differently for the assignment and the ordering problems, using different ways of determining neighborhoods in both cases:
1 – The neighborhood of the furnace assignment was defined as the set of schedules whose assignment was different in one slab
2 – The neighborhood of the pressing order was defined as the set of schedules whose pressing order is different in two positions, i.e., one pair of slabs.

This is a case where three different objectives, not easily comparable, were chosen: minimizing quantity of fuel, heating time for the slabs, and a complexity index. Although there is no way to know whether an absolute optimal solution was achieved, or if the entire Pareto front was sampled [since GAs are heuristic methods], search results were quite positive according to the authors, who reported that “all objectives have been improved almost uniformly.”

Weile et al. (1996) compared three types of Pareto GAs applied to the study of trade-offs in the design of microwave absorbers. Multilayer microwave absorptive coatings are used for reducing the radar cross section of objects in a variety of applications ranging from stealth to anechoic chambers. The coatings are applied in several layers over a specific backing, and must not only suppress reflection over a wide band of frequencies, but also need to be thin to be practical and economical (Weile et al., 96). These two goals – thinness and high absorptance – are very often in conflict, in the sense that the 'optimal' design with respect to absorption may be too thick and expensive to be practical. In this paper, Pareto GAs are used to choose the material for each layer from a database of available materials and their respective thickness.

The objectives here are to minimize thickness of the overall coating, and reflectance at the frequencies across which the absorber is to operate. The constraints are the availability of materials [can only pick materials from an existing database], the band of frequencies under which the absorber will be evaluated, and the number of incident angles where reflection is to be suppressed. Because the materials have arbitrarily frequency dependent properties, the Pareto surface is unlikely to be continuous, so the search procedure will generate a discrete approximation to the Pareto set.

The three MOGA compared in this study are:
CTPGA – Crowded Tournament Pareto GA, using nondomination ranking, crowding, and tournament selection. The nondomination ranking as proposed by Goldberg is described in section 8.3.3 as the first Pareto approach. Crowding is described in section 8.4
NPGA – Niched Pareto GA, described in section 8.3.3 as the third Pareto approach.
NSGA – Nondominated Sorting GA, which combine nondomination ranking and sharing.

In the NSGA, the solutions ranked 1 are given an initial fitness value F_{R1}, and then their niche counts n_i are calculated. Their shared fitness is then calculated in the usual way $F_{shared,i} = F_{R1} / n_i$.

The solutions ranked 2 are then assigned an initial fitness value F_{R2}, less than the lowest shared objective function value of those ranked 1, and then undergo sharing themselves. This process is repeated until all members of the population are assigned a fitness value, and then roulette wheel selection is used to create the next population.

This study shows that the best performing method was the NSGA, followed by the CTPGA, and last by the NPGA. The CTPGA was able to keep diversity over the Pareto curve, as well as a relatively dense curve, while the NPGA was unable to keep such compromise. The NSGA clearly outperformed the two other approaches [using the same niche radius as the NPGA], and after only 10 generations was able to locate most of the Pareto front, while in the final generations the curve only becomes more well-defined and slightly more diverse (Weile, 96).

This last point is important because although the NSGA performed the best, it is also the most computationally intensive approach of the three, followed by the CTPGA. This is easy to conclude when one looks at the algorithms themselves. Both NSGA and CTPGA involve the ranking of the entire population, while the NPGA does not [individuals are only compared with a random subset of the population]. Additionally, sharing is more computationally expensive than crowding, as it was shown in section 8.4, what makes NSGA the most expensive algorithm of the three. However, since it has been shown that the NSGA quickly locates most of the Pareto front in the first generations, and then spends most of its computational effort making it more dense and only slightly more diverse, it is possible to significantly reduce the computational time of NSGA.

A third example described application of a Pareto genetic algorithms to an object recognition system (Aherne et al, 1997). In most cases, the parameters used for object recognition have to be specified by the user, based on rules-of-thumb and trial-and-error procedures. Using a Pareto genetic algorithm, it was possible to have the system automatically determine the parameters of importance for recognition of a certain object in a large database. This represents a significant advancement in automating object recognition systems.

Finally, a multi-objective genetic algorithm was applied to the design of the control system for a gas turbine engine [an Advanced Short Take-Off, Vertical Landing aero-engine] (Chipperfield et al., 1996). In this case, a number of design parameters had to be optimized so that the system could respond to very stringent time-response design specifications while minimizing the complex interactions between the system loops.

8.6 Implementation and initial experiments

For the work described in the remaining of this chapter, a Nondominated Sorting Genetic Algorithm [NSGA] was implemented, since from the literature review it was suggested this was the best approach to MOGA applications. However, not only is NSGA a computationally expensive algorithm, the coding for implementation is not simple to develop.

The first experiments using an initial implementation [figure 8.17] demonstrated that even though the population of solutions were slightly increasing their performance levels, as shown by the trend lines in the graph, an actual front of Pareto points was not discernible. In the graph, blue represents the initial population, red an intermediate one, and green the final population.

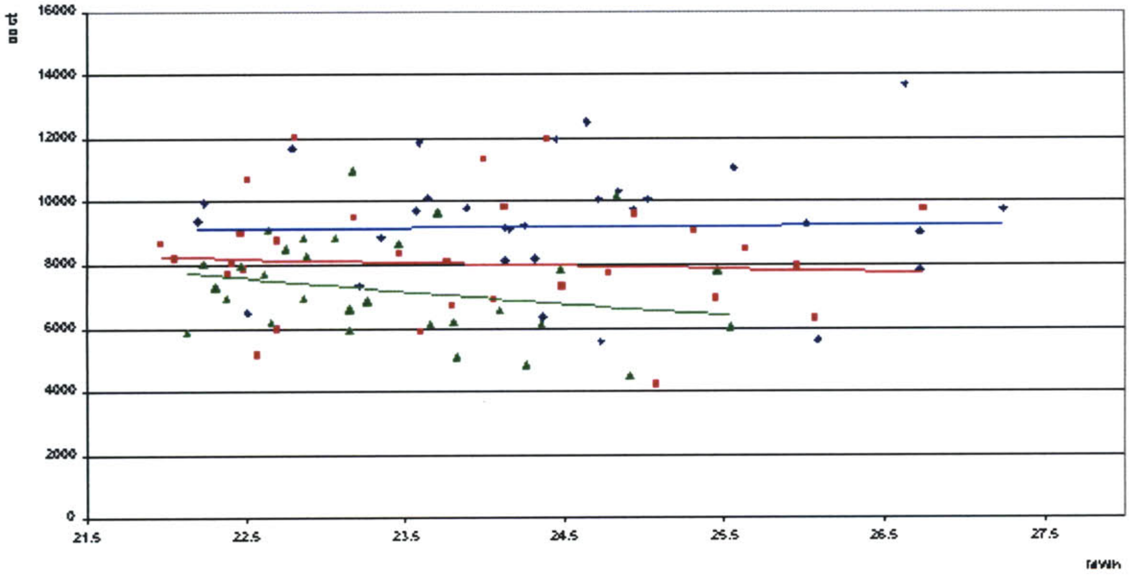


Figure 8.17 Experiments with the initial NSGA implementation showed solutions were not converging to a Pareto front, even though they were improving the overall population performance. Further code refinements were necessary.

After further code refinements, another experiment was performed, this time using the building by Álvaro Siza presented in chapter 7. Two objective functions were used to create a Pareto front, namely energy spent for lighting, and energy spent for space conditioning. This separation came from the concept that lighting and thermal issues have different qualitative impacts on the indoor environment. Daylighting is more related to the spatial quality of the architecture itself. Furthermore, building occupants usually prefer daylit spaces to artificially lit ones. There is also research being done trying to relate daylight to productivity (Norris et al, 1997). On the other hand, a comfortable thermal environment can be successfully delivered by mechanical means, as long as the system is designed and operated according to best practice standards. For that reason, it was considered that it might be pertinent to separate the annual energy consumption of the building in two components, energy used for lighting [which reflects daylighting quality] and energy use for heating and cooling, since they reflected different qualitative measures, not only quantitative ones.

Figure 8.18 shows a graph plotting the last population of a Pareto run with 20 individuals. Although many points are still not in the frontier, it is already possible to determine some kind of Pareto front, which is made more explicit in figure 8.19, where only the best Pareto solutions are represented. A trend line was added, that further helps visualizing the shape of the frontier.

#	South-W	South-H	South-A	East-W	East-H	East-A	North-W	North-H	North-A	West-W	West-H	West-A	Total-A	Mat 1	Mat 2	Mat 3	MWh	Kg CO2 eq
1	7,4	2,4	17,7	4,7	1,8	8,4	4,7	1,2	5,6	3,8	1,8	6,8	38,5	13	6	4	22,5	2617
2	7,4	2,4	17,7	4,7	1,8	8,4	5,6	1,2	6,7	3,8	2,4	9,0	41,8	5	5	4	22,5	2072
3	7,4	2,4	17,7	5,1	1,8	9,2	5,6	0,6	3,4	2,9	2,4	6,9	37,2	4	1	4	22,8	1564
4	7,4	2,4	17,7	5,6	1,8	10,0	5,1	0,6	3,1	3,3	1,8	6,0	36,8	4	1	4	23,0	1550
5	7,4	2,4	17,7	4,2	1,8	7,6	1,5	0,6	0,9	3,8	1,8	6,8	33,0	4	1	4	23,1	1405
6	7,4	2,4	17,7	5,1	0,6	3,1	5,1	1,2	6,2	2,9	1,8	5,2	32,1	4	3	4	23,2	1373
7	7,4	2,4	17,7	4,2	1,8	7,6	1,1	0,6	0,6	2,9	1,8	5,2	31,1	4	1	4	23,3	1332
8	7,4	2,4	17,7	6,0	0,6	3,7	4,2	1,2	5,1	3,3	0,6	2,0	28,5	4	1	4	23,6	1233
9	7,4	2,4	17,7	5,1	1,2	6,2	1,5	1,2	1,8	2,9	0,6	1,8	27,4	4	1	4	23,8	1193
10	7,4	2,4	17,7	1,5	1,8	2,7	1,5	0,6	0,9	2,9	1,8	5,2	26,5	4	3	4	23,8	1157
11	7,4	2,4	17,7	4,2	0,6	2,6	3,3	0,6	2,0	2,0	1,8	3,5	25,8	4	1	4	23,8	1132
12	7,4	2,4	17,7	4,2	0,6	2,6	3,3	0,6	2,0	3,8	0,6	2,3	24,6	4	1	4	24,0	1085
13	7,4	2,4	17,7	5,1	0,6	3,1	1,5	0,6	0,9	3,8	0,6	2,3	24,0	4	1	4	24,0	1064
14	7,4	2,4	17,7	1,1	1,8	1,9	1,5	0,6	0,9	3,8	0,6	2,3	22,8	4	1	4	24,3	1018
15	7,4	2,4	17,7	1,1	1,8	1,9	0,6	0,6	0,4	3,8	0,6	2,3	22,3	4	1	4	24,3	997
16	7,4	2,4	17,7	0,6	1,8	1,1	1,5	0,6	0,9	3,3	0,6	2,0	21,7	4	1	4	24,6	977
17	5,6	2,4	13,4	5,1	1,2	6,2	0,6	0,6	0,4	2,9	0,6	1,8	21,7	4	1	4	24,6	974
18	5,6	2,4	13,4	4,2	0,6	2,6	0,6	0,6	0,4	6,5	0,6	4,0	20,3	4	2	4	24,6	921
19	5,6	2,4	13,4	5,1	0,6	3,1	0,6	1,2	0,7	3,8	0,6	2,3	19,5	4	1	4	24,8	893
20	5,6	2,4	13,4	5,1	0,6	3,1	0,6	0,6	0,4	3,8	0,6	2,3	19,2	4	1	4	24,8	879
21	5,6	2,4	13,4	5,1	0,6	3,1	0,6	1,2	0,7	2,9	0,6	1,8	19,0	4	2	4	24,9	872
22	3,8	2,4	9,0	1,5	1,8	2,7	1,5	1,8	2,7	2,4	0,6	1,5	15,9	4	1	4	25,7	757
23	3,8	2,4	9,0	4,2	0,6	2,6	2,0	1,2	2,4	2,4	0,6	1,5	15,4	4	1	4	25,7	738
24	3,8	2,4	9,0	1,5	1,8	2,7	0,6	0,6	0,4	3,3	0,6	2,0	14,1	4	2	4	25,8	689
25	3,8	2,4	9,0	0,6	1,8	1,1	1,5	0,6	0,9	3,3	0,6	2,0	13,1	4	2	4	26,2	648
26	3,8	1,2	4,5	1,1	1,8	1,9	2,0	0,6	1,2	3,3	1,2	4,0	11,7	4	1	4	26,9	594
27	4,7	1,2	5,6	1,5	1,2	1,8	1,5	1,2	1,8	2,9	0,6	1,8	11,0	4	1	4	26,9	570
28	3,8	1,2	4,5	1,5	1,2	1,8	1,5	0,6	0,9	5,1	0,6	3,1	10,4	4	1	4	27,0	547
29	1,1	2,4	2,5	5,1	0,6	3,1	0,6	0,6	0,4	6,0	0,6	3,7	9,7	4	2	4	27,4	520
30	1,1	2,4	2,5	5,1	0,6	3,1	0,6	0,6	0,4	2,4	0,6	1,5	7,5	4	1	4	27,9	436

Figure 8.18 Last population of a Pareto run with 20 individuals, applied to Siza's School of Architecture, using energy spent for lighting and space conditioning as objective functions.

In figure 8.20, the top row shows the solution that performs better in terms of space conditioning [heating and cooling, represented in the legend as h|c], and performs worst in terms of lighting. The bottom row represents the best solution for lighting and worst for space conditioning.

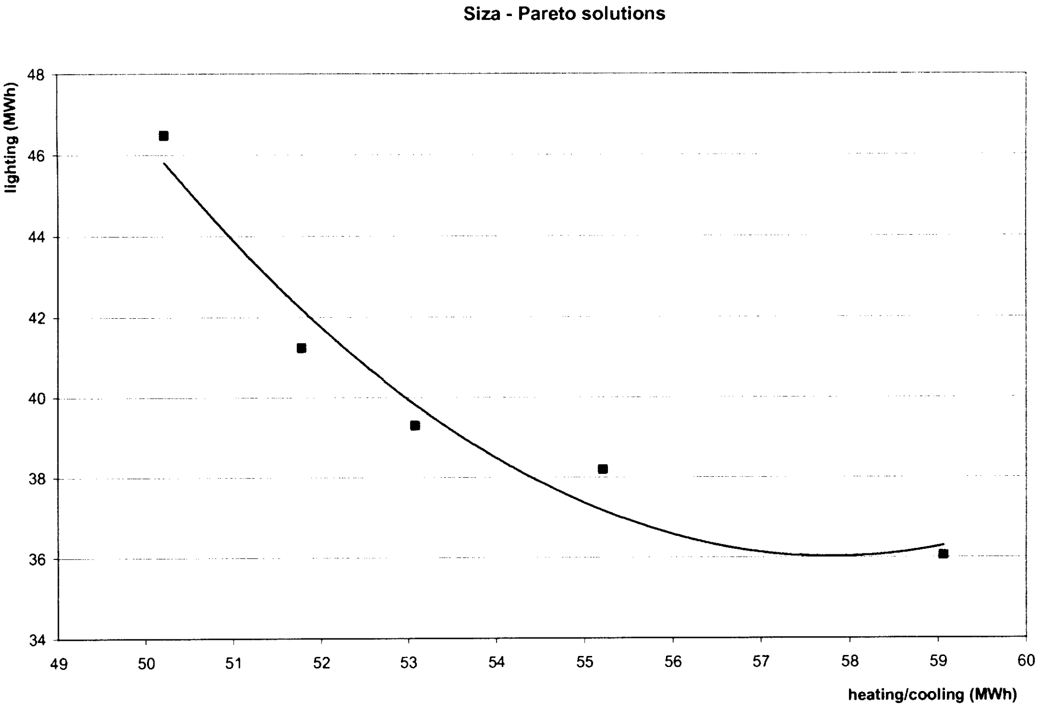


Figure 8.19 Five points in the Pareto front from figure 8.18, with trend line added. The façade solutions corresponding to those points are visualized in figure 8.20

Façade designs follow a somehow expected pattern, in that the best solutions for space conditioning have smaller fenestration sizes, and the best ones for lighting have larger openings. However, some points are interesting to observe. For example, in the west façade, for the best lighting solutions windows do not assume large dimensions, staying much smaller than what the constraints allow. This suggests that these window dimensions are enough to provide the lighting levels set for those spaces [which are lower than for the studio rooms], so increasing window size would bring no benefits in terms of lighting and would cause poorer thermal performance. The best trade-off seems to be close to the one depicted in the image. For the best lighting solution, east facing windows in the 4th and 5th floors are also not driven to very large dimensions, suggesting again that the combination of larger south facing openings and smaller east facing ones provides enough daylighting to satisfy the spaces requirements.

The third row in figure 8.20 represents the 'intermediate' or middling solution, where more compromises are made between lighting and thermal requirements. It resembles in many aspects the oporto_best solution from chapter 7, but has an overall higher energy consumption than the one found in that chapter's experiments. For that reason, and also due to the fact that

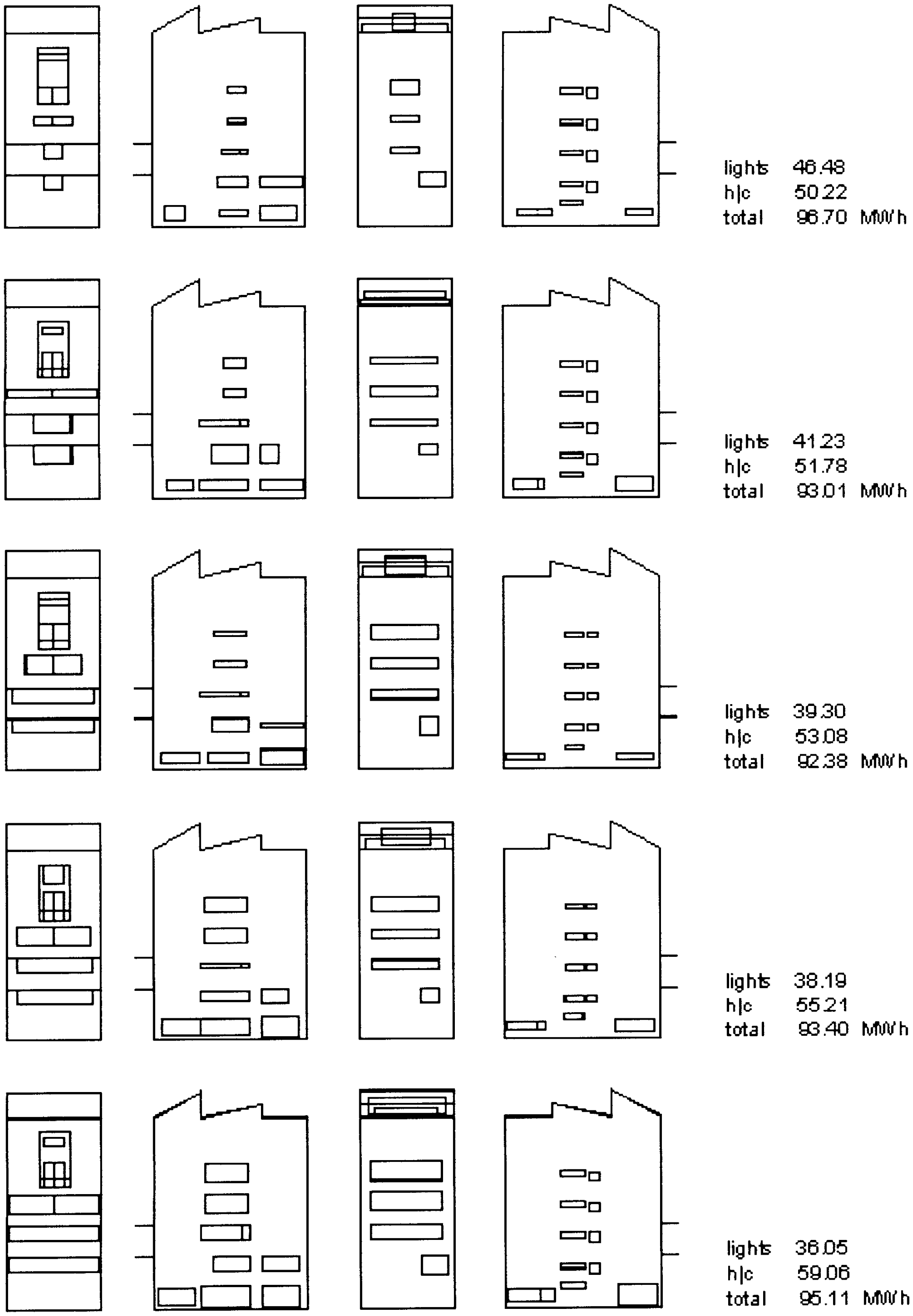


Figure 8.20 Five façade solutions corresponding to the Pareto front from figure 8.19.

the Pareto front was still not well defined in figure 8.18, it could be concluded at this stage that the Pareto code still required some improvements to provide more reliable solutions, and to be able to better define the Pareto frontier, thus offering the user more information about possible trade-offs.

Improvements to the code implied the refinement of the ranking method, changing of the elitism strategies, and upgrading the niching procedures so that the points would spread along the Pareto front instead of staying in clusters. The next experiments show that the resulting version of the NSGA code quite successfully locates Pareto fronts, spreading most of the individuals among the frontier. Few points are left in regions of poorer performance. This implementation was the basis for all the remaining experiments presented in this chapter.

8.7 Experiments relating building energy consumption with construction materials costs

The experiments descried in this section are based on Pareto search methods, and look for the best trade-offs between reductions in energy consumption during building operation, and initial costs due to construction materials. A simple, schematic building layout was created for these experiments, with independent rooms facing each cardinal direction, with just one window per room [in the longest exterior wall], but avoiding the large core area that existed in the building scheme used in chapter 4. Figure 8.21 shows the building layout.

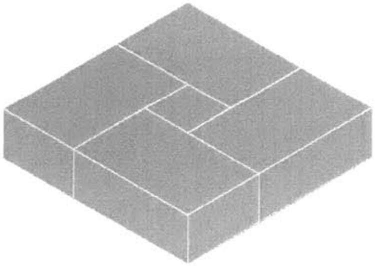


Figure 8.21 Schematic building layout for Pareto experiments

Apart from varying the window dimensions, as in chapter 4, the GS could also manipulate the construction materials used for the external walls. All walls were supposed to be built in a similar way, regardless of their orientation. The interior finish was gypsum board, and the outside had a 2.5 cm mortar layer. What the GS could manipulate were the interior materials only. Three internal layers were predicted. Material 1 represented the material used for the layer closest to the external wall surface. Material 2 was the innermost layer, and material 3 the one closest to the inside surface. The GS was allowed to choose from a list of 16 possible materials, which are listed in Table 8.1. Since some of those options are air layers, there was an additional constraint that prevented materials 1 and 3 to be air layers, due to buildability issues. Materials were selected from the DOE2 materials library, as well as their respective physical properties. It should be added that DOE2 allows the user to create new materials and insert them in the existing library.

#	DOE2 code	Name	Thickness [cm]	Density [kg/m3]	Conductivity [W/m.k]
1	AL11	Air layer	1.9 or less	-	R=0.1586
2	AL21	Air layer	1.9 to 10.2	-	R=0.1568
3	AL31	Air layer	10.2 or more	-	R=0.1621
4	IN33	Expanded Polystyrene	2.54	28.83	0.0346
5	IN35	Expanded Polystyrene	5.08	28.83	0.0346
6	IN36	Expanded Polystyrene	7.62	28.83	0.0346
7	IN43	Expanded Polyurethane	2.54	24.03	0.023
8	IN45	Expanded Polyurethane	5.08	24.03	0.023
9	CB21	Concrete block, medium weight, hollow	10.2	1217.41	0.5197
10	CB23	Concrete block, medium weight, perlite filled	10.2	1249.44	0.2617
11	CB26	Concrete block, medium weight, hollow	15.2	1041.2	0.618
12	CB28	Concrete block, medium weight, perlite filled	15.2	1073.24	0.2018
13	CB41	Concrete block, light weight, hollow	10.2	1041.2	0.3846
14	CB43	Concrete block, light weight, perlite filled	10.2	1073.24	0.22
15	CB46	Concrete block, light weight, hollow	15.2	881.02	0.4806
16	CB48	Concrete block, light weight, perlite filled	15.2	913.05	0.1705

Table 8.1 Construction materials and their properties, for the Pareto experiments

The glass used was fixed for all the experiments, and was double clear glazing [DOE2 code 2003], with the following properties:

Glazing type	Layers [mm]	Shading coefficient	Visual transmittance	U-value
Double clear	6 / 6 / 6	0.81	0.781	3.16

Table 8.2 Glazing properties used for the Pareto experiments

Costs for the different materials were obtained by telephoning several retailers in the US, and averaging the different prices provided by distinct retailers. Those values are not to be considered too accurate, but they give a relative measure of the comparative costs of the different materials. In a real world application, care should be taken in obtaining accurate values. If quantity discounts exist, they can be handled by if-then type of rules, of the kind 'if material x area is greater than y, then cost of material x equals z, else if ... '.

In general, costs per unit area of insulation materials were lower than those for concrete block. Expanded polyurethane was more expensive than expanded polystyrene. Air layer costs were set to zero. Window costs were much higher per unit area than any of the other materials.

8.7.2 Results

Some results of experiments using this setup are now presented. Experiments were done both for Phoenix and Chicago climates.

8.7.2.1 Phoenix

Figure 8.22 shows the progression of the search for the Pareto front for the Phoenix climate, from the initial random population [gen 1, in white], to generation 100 [in blue], where the Pareto front is already clearly visible. Running another 100 generations [gen 200, in red] made some small improvements in the front, but also implied the loss of some previously found points.

Figure 8.23 shows the progression of the search from generation 1 to generation 100, in steps of 10 generations. It can be seen that by generation 10 solutions have started to move towards the Pareto front, and by generation 20, most of the front is already defined. The subsequent generations work mostly in refining and making small improvements in the front, what confirms the information found in the literature.

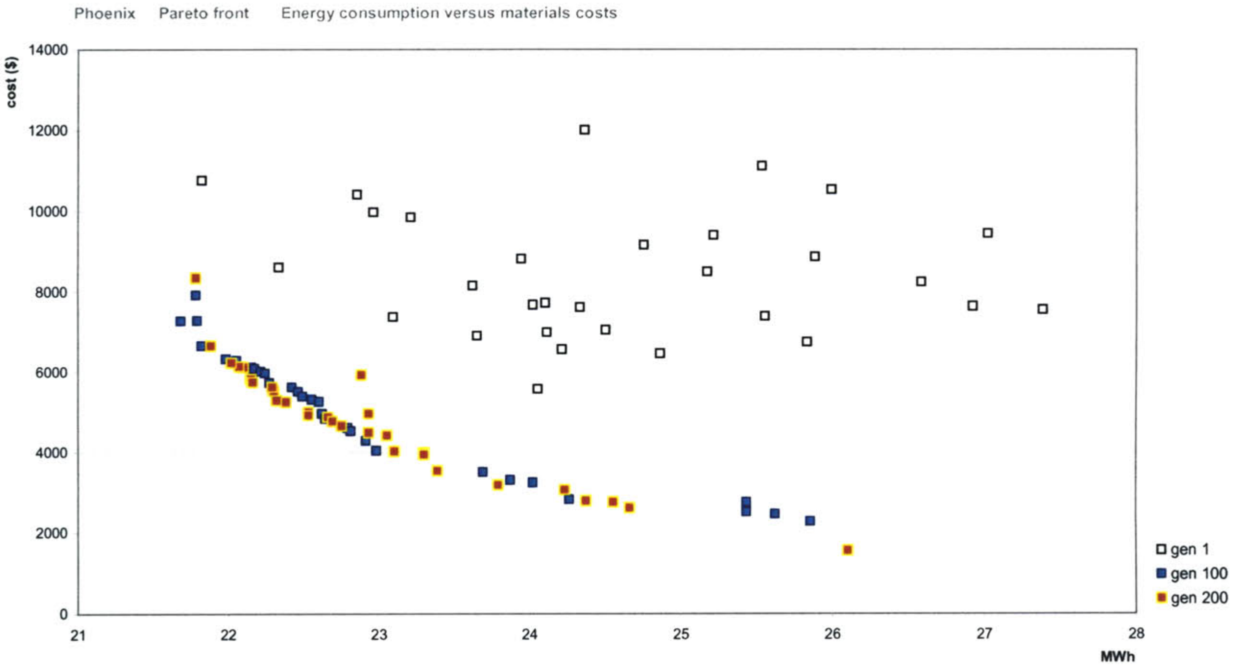


Figure 8.22 Pareto front for Phoenix climate

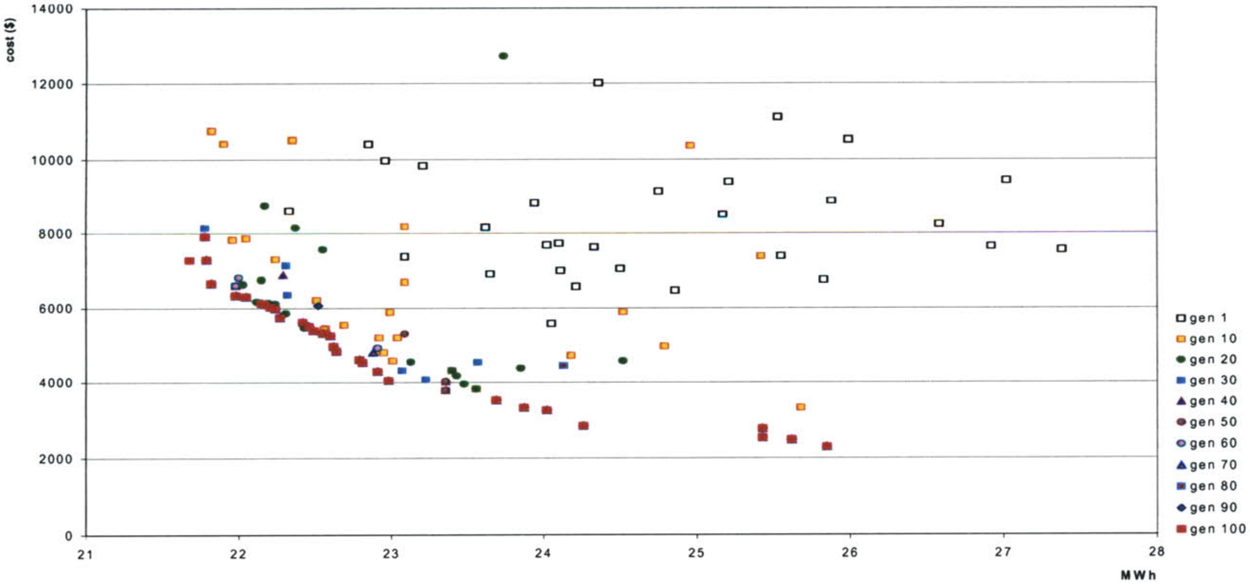


Figure 8.23 Search progression for the Pareto front for Phoenix climate

The following step was to analyze what design configurations and combinations of construction materials led to the appearance of this frontier of best trade-offs. Table 8.3 shows the initial random values given to each variable by the GS in the first generation, and Table 8.4 shows the variable values of the solutions belonging to the Pareto front.

For each window orientation, its width and height are given [for example, South-W is south window width in meters]. The area of each window [shown in blue] and the total window area for the building [in red], all in square meters, were calculated afterwards for the purpose of results interpretation.

The materials chosen for each wall layer are also shown, using the codes from the left column of Table 8.1. Mat1 corresponds to the outermost layer, Mat2 to the intermediate layer, and Mat 3 to the layer closer to the building's interior. Both tables are displayed in ascending order according to energy consumption [MWh column], with the lowest energy levels showing first.

#	South-W	South-H	South-A	East-W	East-H	East-A	North-W	North-H	North-A	West-W	West-H	West-A	Total-A	Mat 1	Mat 2	Mat 3	MWh	Cost [$]
1	7,4	1,8	13	5,1	1,2	6	5,1	5,1	26	6,0	1,2	7	53	16	6	10	21,8	10762
2	6,9	2,4	17	3,3	1,8	6	3,3	3,3	11	1,5	0,6	1	35	8	7	6	22,3	8607
3	6,0	1,2	7	5,6	0,6	3	5,6	5,6	31	3,8	2,4	9	51	6	12	7	22,9	10405
4	6,0	1,2	7	2,0	1,8	4	2,0	2,0	4	1,1	1,2	1	16	10	12	16	23,0	9970
5	2,4	1,8	4	1,5	1,8	3	1,5	1,5	2	1,5	0,6	1	10	16	2	16	23,1	7375
6	6,9	1,8	12	6,0	0,6	4	6,0	6,0	36	5,1	2,4	12	65	16	13	8	23,2	9836
7	6,9	1,2	8	1,5	0,6	1	1,5	1,5	2	2,4	1,8	4	16	15	1	12	23,6	8151
8	1,5	2,4	4	3,3	1,8	6	3,3	3,3	11	2,9	0,6	2	22	8	7	15	23,7	6903
9	6,0	1,2	7	1,5	1,2	2	1,5	1,5	2	4,2	0,6	3	14	13	9	15	23,9	8813
10	5,6	0,6	3	4,7	1,2	6	4,7	4,7	22	3,8	0,6	2	33	12	14	4	24,0	7675
11	2,0	1,8	4	2,4	1,8	4	2,4	2,4	6	1,1	2,4	3	16	12	3	5	24,1	5572
12	2,4	1,2	3	5,1	0,6	3	5,1	5,1	26	2,4	1,8	4	37	13	5	16	24,1	7724
13	2,4	1,8	4	5,6	0,6	3	5,6	5,6	31	6,9	1,2	8	47	6	2	5	24,1	6995
14	1,1	1,2	1	1,5	1,8	3	1,5	1,5	2	4,2	0,6	3	9	8	10	8	24,2	6566
15	6,9	1,2	8	1,1	0,6	1	1,1	1,1	1	5,1	1,2	6	16	11	1	14	24,3	7622
16	2,4	1,2	3	4,2	1,8	8	4,2	4,2	18	3,3	1,8	6	34	16	5	12	24,4	12008
17	3,3	0,6	2	4,2	1,2	5	4,2	4,2	18	2,0	0,6	1	26	5	10	9	24,5	7050
18	2,0	1,2	2	1,5	1,8	3	1,5	1,5	2	4,2	0,6	3	10	10	10	10	24,8	9149
19	1,5	0,6	1	7,4	0,6	5	7,4	7,4	55	2,4	1,8	4	64	6	11	7	24,9	6464
20	1,1	1,8	2	1,1	2,4	3	1,1	1,1	1	3,8	1,2	5	10	12	10	9	25,2	8494
21	6,9	0,6	4	1,1	1,8	2	1,1	1,1	1	5,6	2,4	13	21	13	6	12	25,2	9396
22	0,6	1,2	1	3,3	1,2	4	3,3	3,3	11	0,6	1,2	1	17	12	16	15	25,5	11106
23	3,3	0,6	2	5,6	2,4	13	5,6	5,6	31	4,7	1,2	6	52	16	9	6	25,6	7389
24	3,8	0,6	2	1,5	1,8	3	1,5	1,5	2	1,1	0,6	1	8	11	3	13	25,8	6746
25	0,6	1,8	1	1,1	0,6	1	1,1	1,1	1	1,5	0,6	1	4	16	2	16	25,9	8862
26	2,0	0,6	1	5,1	1,8	9	5,1	5,1	26	0,6	1,8	1	38	16	2	12	26,0	10524
27	2,9	0,6	2	7,4	1,8	13	7,4	7,4	55	0,6	1,8	1	71	16	1	16	26,6	8241
28	0,6	1,2	1	0,6	2,4	1	0,6	0,6	0	1,5	0,6	1	3	16	1	15	26,9	7635
29	3,8	0,6	2	0,6	2,4	1	0,6	0,6	0	6,5	2,4	15	20	16	3	16	27,0	9437
30	2,0	0,6	1	2,0	2,4	5	2,0	2,0	4	6,9	2,4	17	26	5	3	14	27,4	7550

Table 8.3 Initial random values given to each variable by the GS in generation 1

#	South-W	South-H	South-A	East-W	East-H	East-A	North-W	North-H	North-A	West-W	West-H	West-A	Total-A	Mat 1	Mat 2	Mat 3	MWh	Cost [$]
1	4,7	1,8	8	7,4	0,6	5	6,0	1,2	7	4,2	0,6	3	23	8	6	6	21,7	7276
2	7,4	1,8	13	7,4	0,6	5	4,2	1,2	5	5,1	0,6	3	26	8	6	6	21,8	7917
3	6,5	1,8	12	7,4	0,6	5	6,5	1,2	8	4,2	0,6	3	27	8	3	6	21,8	7286
4	4,7	1,8	8	5,6	0,6	3	6,0	1,2	7	5,1	0,6	3	22	8	4	6	21,8	6653
5	6,0	1,8	11	5,1	0,6	3	4,2	1,2	5	3,3	0,6	2	21	4	6	6	22,0	6331
6	6,0	1,8	11	6,5	0,6	4	2,9	1,2	3	4,2	0,6	3	21	5	5	6	22,1	6294
7	6,5	1,8	12	5,6	0,6	3	6,5	0,6	4	3,8	0,6	2	21	4	5	6	22,2	6127
8	4,2	1,8	8	5,1	0,6	3	2,4	1,2	3	5,1	0,6	3	17	8	6	6	22,2	6093
9	4,2	1,8	8	4,2	1,2	5	2,9	1,2	3	5,6	0,6	3	20	4	6	6	22,2	6019
10	6,5	1,8	12	4,7	0,6	3	6,0	0,6	4	2,0	1,2	2	21	4	5	6	22,2	5971
11	6,0	1,8	11	5,6	0,6	3	4,7	0,6	3	3,8	0,6	2	19	4	5	6	22,3	5736
12	6,0	1,8	11	5,6	0,6	3	0,6	1,2	1	4,2	0,6	3	18	4	6	6	22,4	5619
13	4,2	1,8	8	4,7	0,6	3	0,6	1,2	1	5,1	0,6	3	14	6	6	6	22,5	5507
14	4,2	1,8	8	6,9	0,6	4	0,6	2,4	1	5,1	0,6	3	16	6	5	7	22,5	5388
15	4,2	1,8	8	6,9	0,6	4	0,6	2,4	1	4,7	0,6	3	16	6	7	7	22,6	5318
16	6,0	1,8	11	2,0	1,2	2	2,4	1,2	3	1,5	0,6	1	17	4	5	6	23	5256
17	4,2	1,8	8	5,6	0,6	3	2,0	0,6	1	5,6	0,6	3	16	4	5	6	22,6	4962
18	4,2	1,8	8	5,6	0,6	3	1,1	0,6	1	3,3	0,6	2	14	4	6	6	22,6	4834
19	4,2	1,8	8	4,2	0,6	3	2,4	1,2	3	5,6	0,6	3	16	4	5	4	22,8	4606
20	4,2	1,8	8	2,0	1,2	2	1,1	1,2	1	3,8	0,6	2	14	4	7	6	22,8	4527
21	4,2	1,8	8	3,8	0,6	2	0,6	2,4	1	1,5	0,6	1	12	4	5	6	22,9	4284
22	4,2	1,8	8	4,2	0,6	3	1,1	1,2	1	3,8	0,6	2	14	4	3	6	23,0	4038
23	2,4	2,4	6	2,0	0,6	1	0,6	0,6	0	2,0	0,6	1	9	4	7	6	23,7	3509
24	2,4	1,8	4	5,6	0,6	3	0,6	1,2	1	1,1	1,2	1	10	8	3	4	23,9	3315
25	2,4	1,8	4	2,0	0,6	1	0,6	0,6	0	1,1	1,2	1	7	6	5	4	24,0	3248
26	2,4	2,4	6	2,0	0,6	1	0,6	0,6	0	1,1	0,6	1	8	4	7	4	24,3	2829
27	0,6	1,8	1	5,6	0,6	3	1,1	1,2	1	1,1	0,6	1	6	6	1	4	25,4	2517
28	0,6	1,8	1	2,0	0,6	1	0,6	1,2	1	2,0	0,6	1	4	8	3	6	25,4	2752
29	0,6	1,8	1	1,5	0,6	1	0,6	1,2	1	5,6	0,6	3	6	4	3	6	25,6	2461
30	0,6	1,8	1	1,5	0,6	1	1,5	1,2	2	4,7	0,6	3	7	4	3	7	25,9	2278

Table 8.4 Final values given to each variable by the GS in generation 100 [most points are Pareto-optimal]

One important feature about this problem is that window costs dominate the overall cost of the building. For that reason, one of the main strategies the GS would use for reducing materials costs would be to decrease window sizes. This is illustrated in figure 8.24, where the green bar lengths are proportional to the overall window area in the building. It is apparent that the highest cost solutions [cost is represented in the x axis] have larger window areas. However, those solutions also correspond to the ones showing better energy performance [solutions are ranked in the y axis, 1 being the best].

With this information in mind, it may become easier to interpret the numerical information present in Table 8.4. In general, larger window sizes, particularly towards the south and north, lead to better performance solutions, but imply higher costs too. There seems to be a stong correlation between reducing fenestration sizes and a decrease of the building's performance. West seems to be an exception, since even in the best solutions the window sizes are rather small. East windows are somewhat larger, showing that west orientation is indeed the most problematic regarding glazing areas.

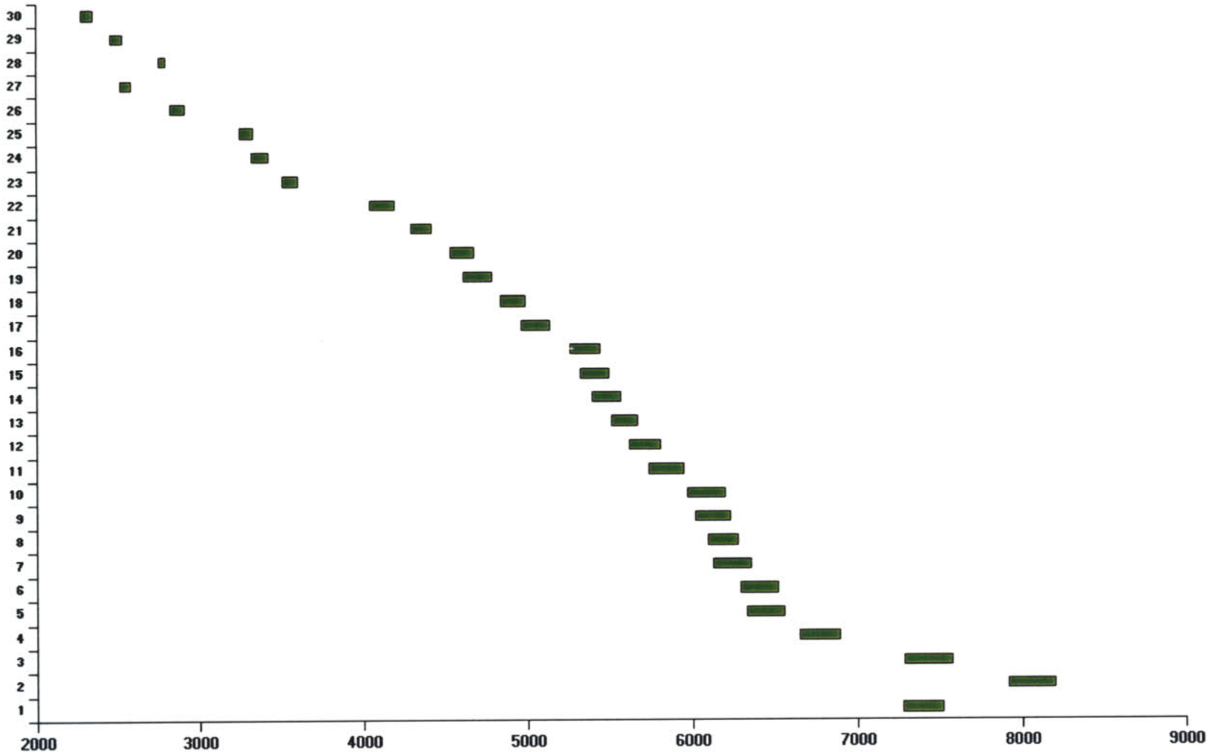


Figure 8.24 Generation 100: Relation of overall window area [green bars] with materials cost in dollars [x axis] and energy performance [best solution is ranked 1 in the y axis].

In terms of wall materials, which are the main difference introduced in these experiments, it is interesting to note that even though in the first random solutions many instances of heavyweight, masonry elements appear, in the final Pareto solutions only lightweight, insulation materials are used. Recalling from Table 8.1, material numbers 1 to 3 are air layers, 4 to 8 are insulation materials, and 9 to 16 are different types of concrete blocks, some filled with perlite, which is a type of insulation too.

Looking at Table 8.4, it is possible to see that all material numbers are smaller or equal to eight, meaning that only insulation materials and air layers are applied. This may be due to several facts. One is that the cost per unit area of insulation materials is lower. The other is that the GS may be using the thermal mass from the concrete floors and roofs to dampen internal temperatures, as heat storage during the day to avoid large peak loads. It was predicted that in a hot climate like Phoenix some thermal mass would be applied in the walls. However, the massive floors and ceilings seem to be able to take that role, and the walls can become lightweight, highly insulated elements. Finally, it is possible to see that the lowest energy solutions make use of better insulation materials, like #6 [7.6 cm expanded polystyrene, the thickest allowed], and #8 [5cm expanded polyurethane, also the thickest]. To reduce construction costs, the GS starts using large air layers and lower quality insulation materials, combining that with a reduction in window sizes.

It is interesting to note that in generation 1, one of the randomly generated solutions already performed almost as well as the best Pareto solution in terms of energy, but its construction costs were about 33% higher. This demonstrates the usefulness of applying a method like the Pareto front search. This way, it was possible to achieve a similar performance solution while saving about 33% in the costs. In terms of the average performance of the populations, the energy level reduction from gen 1 to gen 100 was only 6% on average [from 24.7 MWh to 23 MWh], but reduction in costs were about 41% [from $8,434 to $4,965], a much more significant figure, suggesting that including materials costs in the Pareto front studies may be an effective measure for achieving similar performance levels at lower costs.

Finally, we include in figure 8.25 a combination of the best Pareto solutions found at generation 100 and 200. The above analysis was only performed for results up to generation 100, but we

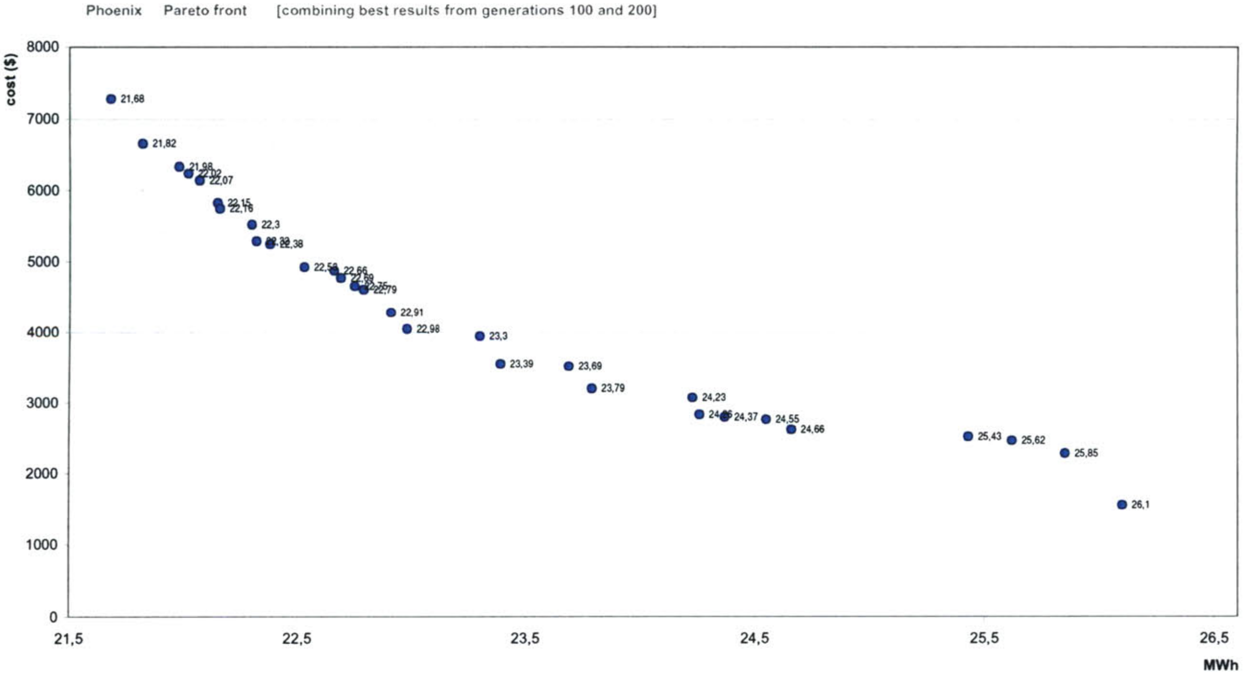


Figure 8.25 Combined Pareto front points of generations 100 and 200

could see from previous figures that running the algorithm for another 100 generations brought new points into the frontier, which are plotted, together with the corresponding energy values, in figure 8.25.

8.7.2.2 Chicago

A similar set of experiments was done for Chicago. Figure 8.26 shows the first random points, and generation 100. No further runs were made for this climate. The Pareto frontier is not as well defined as for Phoenix, but is still quite satisfactory. The few outliers that can be observed (like solution 28 in Table 8.5, which shows the Chicago results for generation 100] are also informative about unsuccessful features of a solution. In general, we can observe that solution costs for Chicago are lower than those for Phoenix. This happens even for the best energy performing solutions, mainly because window sizes were reduced to minimum dimensions, with the exception of south openings, and windows dominate cost. The smaller openings are usually those facing north. Questions can be raised about this small sized openings in terms of user satisfaction, so may be other runs should be done using glazing type as a variable, to consider better performance glazing for getting larger window sizes. For the best energy solutions, the GS could still choose the best insulation materials while keeping solutions at very reasonable costs, due to reduced spending in windows.

Finally, we call the attention for solution 22 in Table 8.5. It achieves extremely low costs with not too much energy penalties by radically reducing all windows [only the south one remains at reasonable size], and thus it can still use low quality, cheap insulation materials by creating this ‘bunker’ type of building.

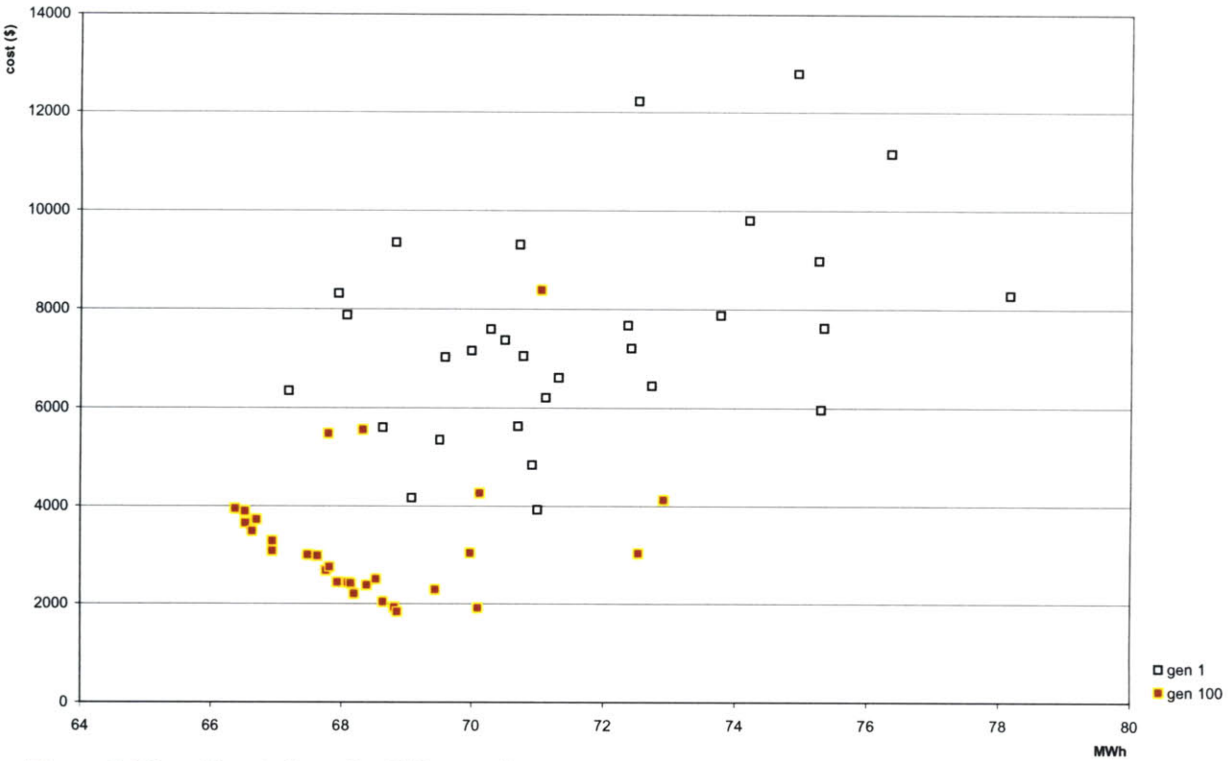


Figure 8.26 Pareto front for Chicago climate

#	South-W	South-H	South-A	East-W	East-H	East-A	North-W	North-H	North-A	West-W	West-H	West-A	Total-A	Mat 1	Mat 2	Mat 3	MWh	Cost [$]
1	2,9	1,8	5,2	0,6	1,8	1,1	0,6	0,6	0,4	1,1	0,6	0,6	7,3	8	5	6	66,4	3942
2	2,9	1,8	5,2	1,5	1,8	2,7	0,6	0,6	0,4	0,6	1,2	0,7	9,0	6	4	6	66,5	3888
3	2,4	1,8	4,3	0,6	1,8	1,1	0,6	0,6	0,4	2,4	0,6	1,5	7,3	8	4	6	66,5	3652
4	2,9	1,8	5,2	0,6	1,8	1,1	0,6	0,6	0,4	0,6	0,6	0,4	7,0	6	4	6	66,6	3486
5	5,1	1,2	6,2	4,2	0,6	2,6	0,6	0,6	0,4	0,6	0,6	0,4	9,5	6	3	6	66,7	3717
6	3,3	1,2	4,0	0,6	1,8	1,1	0,6	0,6	0,4	0,6	0,6	0,4	5,8	8	1	6	66,9	3079
7	2,4	1,8	4,3	0,6	0,6	0,4	0,6	0,6	0,4	0,6	0,6	0,4	5,5	8	4	6	66,9	3284
8	1,1	1,8	1,9	0,6	1,8	1,1	0,6	0,6	0,4	2,0	0,6	1,2	4,6	6	4	6	67,5	2997
9	6,0	1,2	7,3	1,1	0,6	0,6	0,6	0,6	0,4	0,6	0,6	0,4	8,7	4	1	6	67,6	2979
10	3,3	0,6	2,0	0,6	1,8	1,1	0,6	0,6	0,4	0,6	0,6	0,4	3,9	8	3	6	67,8	2679
11	2,9	0,6	1,8	1,5	1,8	2,7	0,6	0,6	0,4	0,6	0,6	0,4	5,2	6	12	6	67,8	5473
12	2,9	0,6	1,8	1,1	1,8	1,9	0,6	0,6	0,4	0,6	1,2	0,7	4,8	6	3	6	67,8	2752
13	2,9	0,6	1,8	0,6	1,2	0,7	0,6	0,6	0,4	0,6	0,6	0,4	3,2	6	1	6	67,9	2439
14	2,9	1,2	3,5	0,6	2,4	1,5	0,6	0,6	0,4	0,6	1,2	0,7	6,0	4	3	6	68,1	2432
15	1,1	1,2	1,3	0,6	1,2	0,7	0,6	0,6	0,4	0,6	1,2	0,7	3,1	6	3	6	68,1	2417
16	2,4	1,2	2,9	1,1	1,2	1,3	0,6	0,6	0,4	0,6	0,6	0,4	4,9	6	1	4	68,2	2206
17	2,9	1,8	5,2	4,2	1,8	7,6	1,1	0,6	0,6	1,1	0,6	0,6	14,1	8	6	6	68,3	5558
18	2,9	0,6	1,8	1,1	1,8	1,9	0,6	0,6	0,4	0,6	0,6	0,4	4,4	4	4	6	68,4	2380
19	2,9	1,2	3,5	0,6	2,4	1,5	0,6	1,8	1,1	0,6	0,6	0,4	6,4	4	3	6	68,5	2507
20	3,8	0,6	2,3	0,6	1,8	1,1	0,6	0,6	0,4	0,6	0,6	0,4	4,1	4	3	6	68,6	2042
21	2,9	0,6	1,8	0,6	1,8	1,1	0,6	0,6	0,4	0,6	0,6	0,4	3,6	4	3	6	68,8	1928
22	2,9	0,6	1,8	1,1	0,6	0,6	0,6	0,6	0,4	0,6	0,6	0,4	3,1	4	1	6	68,9	1835
23	2,9	0,6	1,8	1,1	0,6	0,6	0,6	0,6	0,4	0,6	0,6	0,4	3,1	4	1	6	68,9	1835
24	3,3	0,6	2,0	0,6	2,4	1,5	2,4	0,6	1,5	0,6	0,6	0,4	5,3	6	1	4	69,4	2289
25	1,1	0,6	0,6	4,2	0,6	2,6	4,2	0,6	2,6	0,6	0,6	0,4	6,2	6	1	6	70,0	3040
26	2,4	1,2	2,9	1,1	0,6	0,6	1,1	0,6	0,6	0,6	1,2	0,7	4,9	4	1	7	70,1	1910
27	2,9	0,6	1,8	1,1	1,8	1,9	2,4	1,8	4,3	4,7	0,6	2,9	10,9	6	4	6	70,1	4266
28	1,1	0,6	0,6	6,0	1,8	10,9	3,3	0,6	2,0	2,4	0,6	1,5	15,0	16	10	6	71,0	8387
29	1,5	1,2	1,8	0,6	0,6	0,4	0,6	0,6	0,4	0,6	2,4	1,5	4,0	4	3	10	72,5	3031
30	3,8	0,6	2,3	4,7	1,2	5,6	0,6	0,6	0,4	0,6	0,6	0,4	8,7	4	1	14	72,9	4124

Table 8.5 Generation 100 for Chicago, Pareto solutions

8.8 Experiments using materials greenhouse gas emissions as objective function

In this section, we expand the work on Parteo frontiers to include greenhouse gases emissions into the search for best trade-offs. Two objective functions are considered: annual energy spent for building operation [MWh], and GGE emissions to extract, produce and transport the materials used in the building [kg CO_2 eq.]. GGE's emissions are also an indicator of the embodied energy in the material. This section relates to the issues described in the introduction on reducing GGE at European and world scale. Finding the best compromise between using energy locally [in the building] or at a larger scale [in terms of industrial or transportation systems] is also a complex issue, to which Pareto methods could be successfully applied.

The data used for GGE emissions were supplied by Ospelt [2001], and were appropriate for Switzerland. It was considered acceptable to apply these data in the context of this thesis. In terms of the materials considered, those are mostly the same as for the previous experiments with costs, with a significant difference, though. Material #4 was changed, from 2.54 cm expanded polystyrene to 8.9 cm cellulose fill. This change was due to the fact that cellulose fill is considered to be an environmental-friendly insulation material, so including it in this experiment could possibly yield interesting results.

For the concrete blocks, only the GGE associated with cement were included. We assumed that for the lightweight blocks, the ratio of cement-to-concrete would be 0.042kg cement/kg concrete, and for the medium-weight block, that ration would increase to 0.0625 kg cement/kg concrete. These assumptions may be conservative, but they allowed the use of the data available for Switzerland. As for the perlite-filled blocks, our calculation from the DOE2 materials library was that perlite's density was 32.03 kg / m3, and we calculated for each block type the volume of perlite necessary to fill it. The modified materials properties table used for this experiment is shown in Table 8.6.

Finally, the glass used was the same type as in the previous experiments [DOE2 code 2003]. Glass GGE emissions were taken as 1.3 kg CO_2 eq. / kg glass, and glass density was taken as 2500 kg / m^3 (Incropera, 1996).

Experiments were done for the same test building, for Oporto, Phoenix and Chicago cases.

#	DOE2 code	Name	Thickness [cm]	Density [kg/m3]	GGE [Kg CO2 eq.]
1	AL11	Air layer	1.9 or less	-	0
2	AL21	Air layer	1.9 to 10.2	-	0
3	AL31	Air layer	10.2 or more	-	0
4	IN13	Cellulose Fill	8.9	48.06	0.113
5	IN35	Expanded Polystyrene	5.08	28.83	2.31
6	IN36	Expanded Polystyrene	7.62	28.83	2.31
7	IN43	Expanded Polyurethane	2.54	24.03	13.75
8	IN45	Expanded Polyurethane	5.08	24.03	13.75
9	CB21	Concrete block, medium weight, hollow	10.2	1217.41	Cement 0.0737
10	CB23	Concrete block, medium weight, perlite filled	10.2	1249.44	Cement 0.0737 Perlite 0.67
11	CB26	Concrete block, medium weight, hollow	15.2	1041.2	Cement 0.0737
12	CB28	Concrete block, medium weight, perlite filled	15.2	1073.24	Cement 0.0737 Perlite 0.67
13	CB41	Concrete block, light weight, hollow	10.2	1041.2	Cement 0.053
14	CB43	Concrete block, light weight, perlite filled	10.2	1073.24	Cement 0.053 Perlite 0.67
15	CB46	Concrete block, light weight, hollow	15.2	881.02	Cement 0.053
16	CB48	Concrete block, light weight, perlite filled	15.2	913.05	Cement 0.053 Perlite 0.67

Table 8.6 Construction materials and their properties used for the Pareto experiments, including GGE emissions [Global Warming Potential in 100 years].

8.8.1 Results

Oporto

It can be seen from figure 8.27 that the emissions associated with materials used in the Pareto-front solutions are very low. Looking at Table 8.7, it is possible to understand why. The GS uses mainly cellulose insulation as the construction material for the external walls, due to its very low GGE rate. The two solutions with lower energy levels use other materials, as lightweight concrete blocks and expanded polystyrene, but the corresponding decrease in energy use is so small it becomes negligible. The GS usually applies 2 layers of cellulose fill on the outside with an inner air layer, which could pose some buildability problems in a actual building construction. In fact, most wall configurations the GS proposed might have structural problems. The all-insulation walls could probably be built as a kind of 'box,' with rigid panes to the inside and the outside. However, even if they are not load bearing walls, they would need some kind of internal supporting structure, whose cost should probably be added to the insulation cost for a more fair comparison with the masonry walls, which are self-supporting.

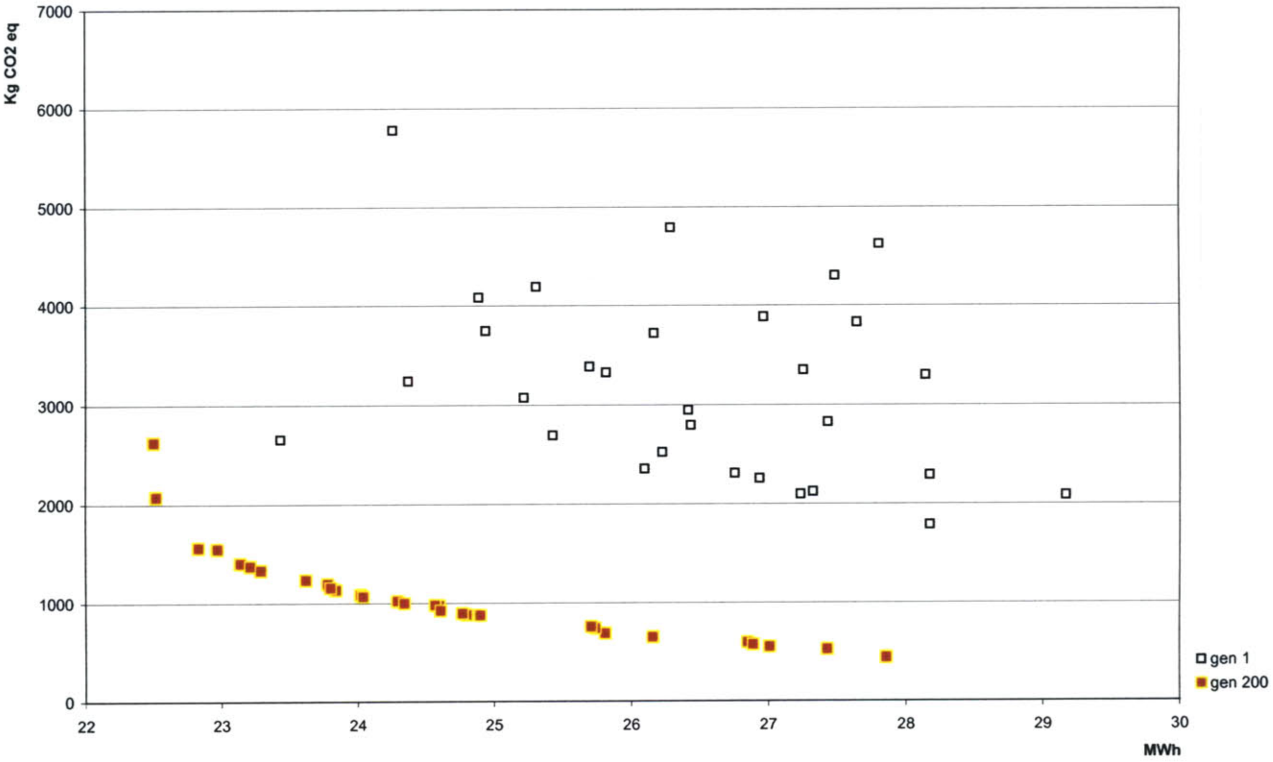


Figure 8.27 Pareto front for Oporto climate, relating energy consumption with materials GGE emissions

As for window areas, and because glass has a quite high GGE level, the strategy adopted by the GS is illustrated in figure 8.28. The GS privileges south windows [white bars in the image] to achieve low energy consumption levels. Its initial strategy to reduce GGE is to cut on the other window areas. Only when minimum areas are reached toward the other directions does the GS make a reduction in south openings, shown as the first 'step' down in image 8.28. This strategy is repeated until minimum areas are reached in all orientations simultaneously.

#	South-W	South-H	South-A	East-W	East-H	East-A	North-W	North-H	North-A	West-W	West-H	West-A	Total-A	Mat 1	Mat 2	Mat 3	MWh	Kg CO2 eq
1	7,4	2,4	17,7	4,7	1,8	8,4	4,7	1,2	5,6	3,8	1,8	6,8	38,5	13	6	4	22,5	2617
2	7,4	2,4	17,7	4,7	1,8	8,4	5,6	1,2	6,7	3,8	2,4	9,0	41,8	5	5	4	22,5	2072
3	7,4	2,4	17,7	5,1	1,8	9,2	5,6	0,6	3,4	2,9	2,4	6,9	37,2	4	1	4	22,8	1564
4	7,4	2,4	17,7	5,6	1,8	10,0	5,1	0,6	3,1	3,3	1,8	6,0	36,8	4	1	4	23,0	1550
5	7,4	2,4	17,7	4,2	1,8	7,6	1,5	0,6	0,9	3,8	1,8	6,8	33,0	4	1	4	23,1	1405
6	7,4	2,4	17,7	5,1	0,6	3,1	5,1	1,2	6,2	2,9	1,8	5,2	32,1	4	3	4	23,2	1373
7	7,4	2,4	17,7	4,2	1,8	7,6	1,1	0,6	0,6	2,9	1,8	5,2	31,1	4	1	4	23,3	1332
8	7,4	2,4	17,7	6,0	0,6	3,7	4,2	1,2	5,1	3,3	0,6	2,0	28,5	4	1	4	23,6	1233
9	7,4	2,4	17,7	5,1	1,2	6,2	1,5	1,2	1,8	2,9	0,6	1,8	27,4	4	1	4	23,8	1193
10	7,4	2,4	17,7	1,5	1,8	2,7	1,5	0,6	0,9	2,9	1,8	5,2	26,5	4	3	4	23,8	1157
11	7,4	2,4	17,7	4,2	0,6	2,6	3,3	0,6	2,0	2,0	1,8	3,5	25,8	4	1	4	23,8	1132
12	7,4	2,4	17,7	4,2	0,6	2,6	3,3	0,6	2,0	3,8	0,6	2,3	24,6	4	1	4	24,0	1085
13	7,4	2,4	17,7	5,1	0,6	3,1	1,5	0,6	0,9	3,8	0,6	2,3	24,0	4	1	4	24,0	1064
14	7,4	2,4	17,7	1,1	1,8	1,9	1,5	0,6	0,9	3,8	0,6	2,3	22,8	4	1	4	24,3	1018
15	7,4	2,4	17,7	1,1	1,8	1,9	0,6	0,6	0,4	3,8	0,6	2,3	22,3	4	1	4	24,3	997
16	7,4	2,4	17,7	0,6	1,8	1,1	1,5	0,6	0,9	3,3	0,6	2,0	21,7	4	1	4	24,6	977
17	5,6	2,4	13,4	5,1	1,2	6,2	0,6	0,6	0,4	2,9	0,6	1,8	21,7	4	1	4	24,6	974
18	5,6	2,4	13,4	4,2	0,6	2,6	0,6	0,6	0,4	6,5	0,6	4,0	20,3	4	2	4	24,6	921
19	5,6	2,4	13,4	5,1	0,6	3,1	0,6	1,2	0,7	3,8	0,6	2,3	19,5	4	1	4	24,8	893
20	5,6	2,4	13,4	5,1	0,6	3,1	0,6	0,6	0,4	3,8	0,6	2,3	19,2	4	1	4	24,8	879
21	5,6	2,4	13,4	5,1	0,6	3,1	0,6	1,2	0,7	2,9	0,6	1,8	19,0	4	2	4	24,9	872
22	3,8	2,4	9,0	1,5	1,8	2,7	1,5	1,8	2,7	2,4	0,6	1,5	15,9	4	1	4	25,7	757
23	3,8	2,4	9,0	4,2	0,6	2,6	2,0	1,2	2,4	2,4	0,6	1,5	15,4	4	1	4	25,7	738
24	3,8	2,4	9,0	1,5	1,8	2,7	0,6	0,6	0,4	3,3	0,6	2,0	14,1	4	2	4	25,8	689
25	3,8	2,4	9,0	0,6	1,8	1,1	1,5	0,6	0,9	3,3	0,6	2,0	13,1	4	2	4	26,2	648
26	3,8	1,2	4,5	1,1	1,8	1,9	2,0	0,6	1,2	3,3	1,2	4,0	11,7	4	1	4	26,9	594
27	4,7	1,2	5,6	1,5	1,2	1,8	1,5	1,2	1,8	2,9	0,6	1,8	11,0	4	1	4	26,9	570
28	3,8	1,2	4,5	1,5	1,2	1,8	1,5	0,6	0,9	5,1	0,6	3,1	10,4	4	1	4	27,0	547
29	1,1	2,4	2,5	5,1	0,6	3,1	0,6	0,6	0,4	6,0	0,6	3,7	9,7	4	2	4	27,4	520
30	1,1	2,4	2,5	5,1	0,6	3,1	0,6	0,6	0,4	2,4	0,6	1,5	7,5	4	1	4	27,9	436

Table 8.6 Pareto solutions for Oporto, including GGE as an objective function

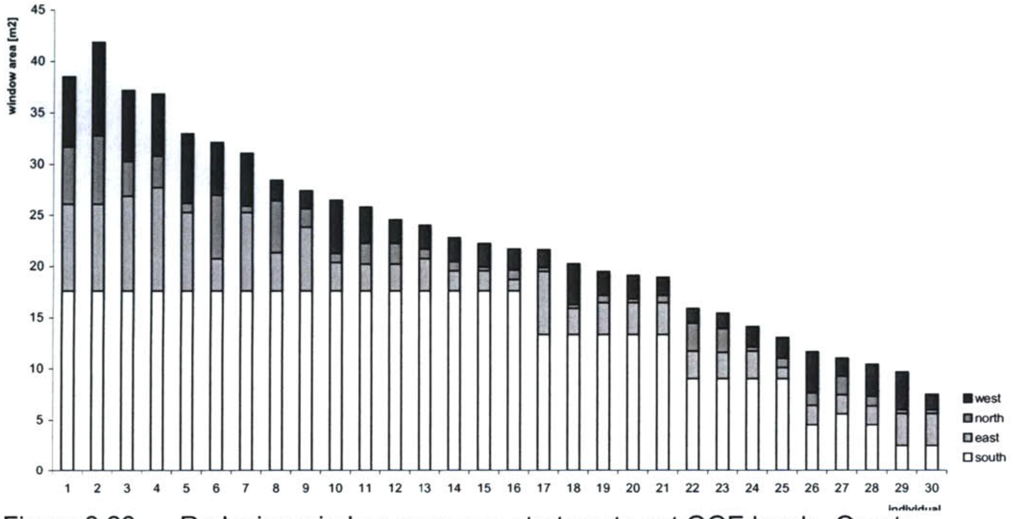


Figure 8.28 Reducing window area as a strategy to cut GGE levels, Oporto

The experiments for Phoenix [figure 8.19] and Chicago [figure 8.30] show similar results, with Chicago again showing the lowest GGE emissions due to very reduced window areas, even though it uses three layers of cellulose fill, in a total of about 27cm of cellulose for the lowest energy solutions. In Phoenix, as in Oporto, the GS cuts east and west facing windows, but tries to keep other directions larger, mainly south. In Phoenix, larger air layers [#3] are also applied between the cellulose ones.

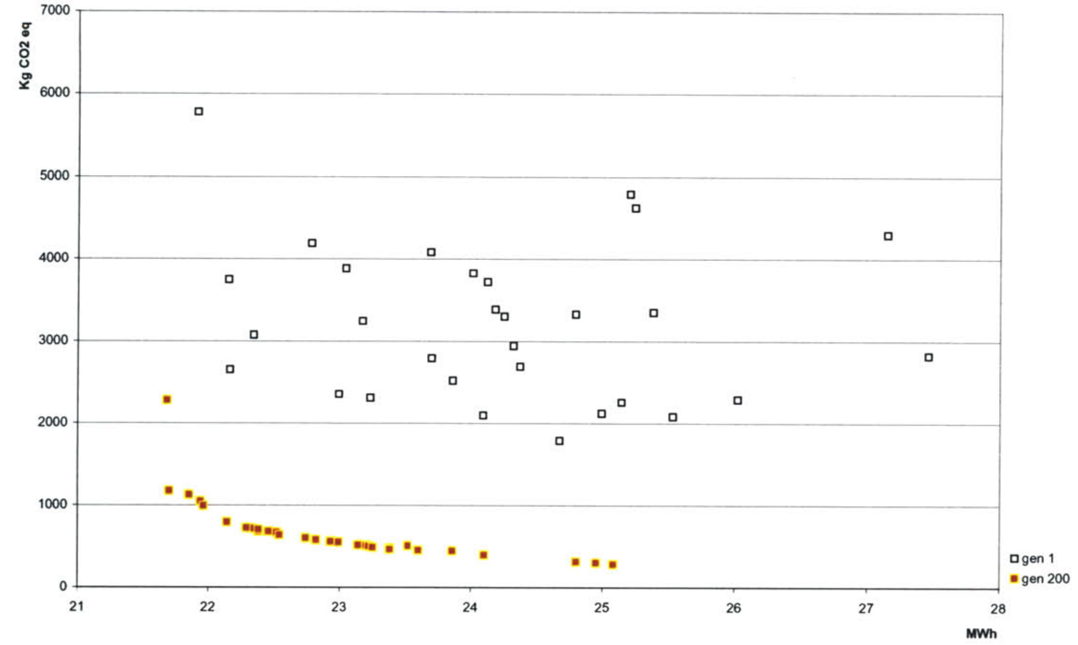


Figure 8.19 Pareto front for Phoenix climate, relating energy consumption with materials GGE

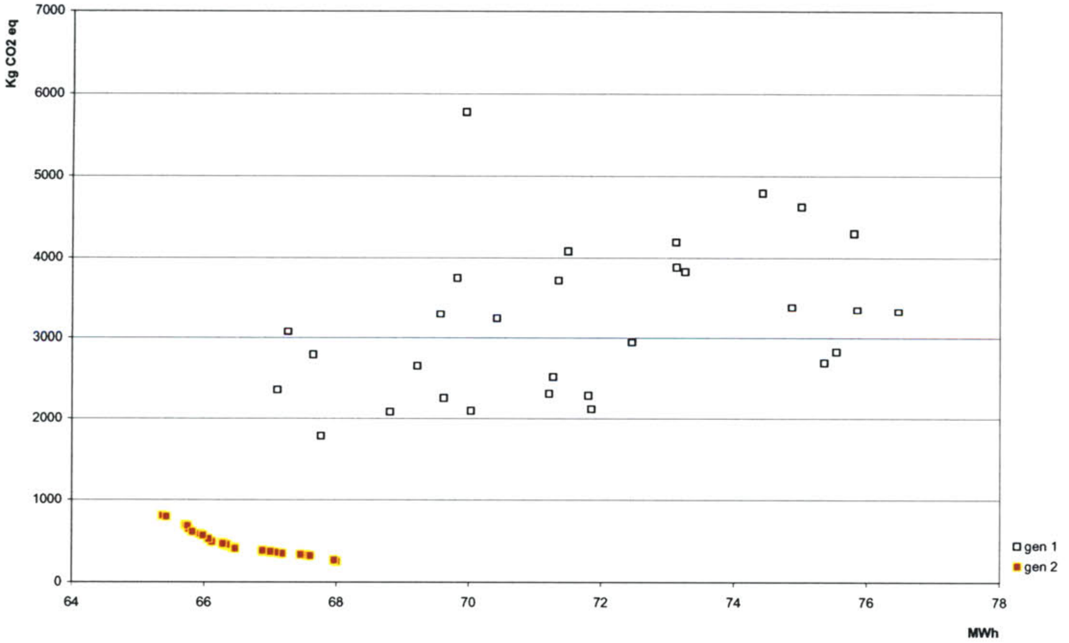


Figure 8.20 Pareto front for Chicago climate, relating energy consumption with materials GGE

8.9 Including wall solar absorptivity and glazing types

The last set of experiments relate to a simple test box, 10 x 7 x 3 m, that was used to represent a hypothetical apartment in Beijing. The shorter sides of the apartment were exposed to the exterior, and faced south and north. The other walls, floor and ceiling were modeled as adiabatic surfaces, assuming the apartment was part of a larger building.

Because there was not an available weather file for Beijing in a format that could be used by the Generative System, Chicago climate was used once again. As it was mentioned before, the two climates are very similar, what can be demonstrated by observing figure 8.21.

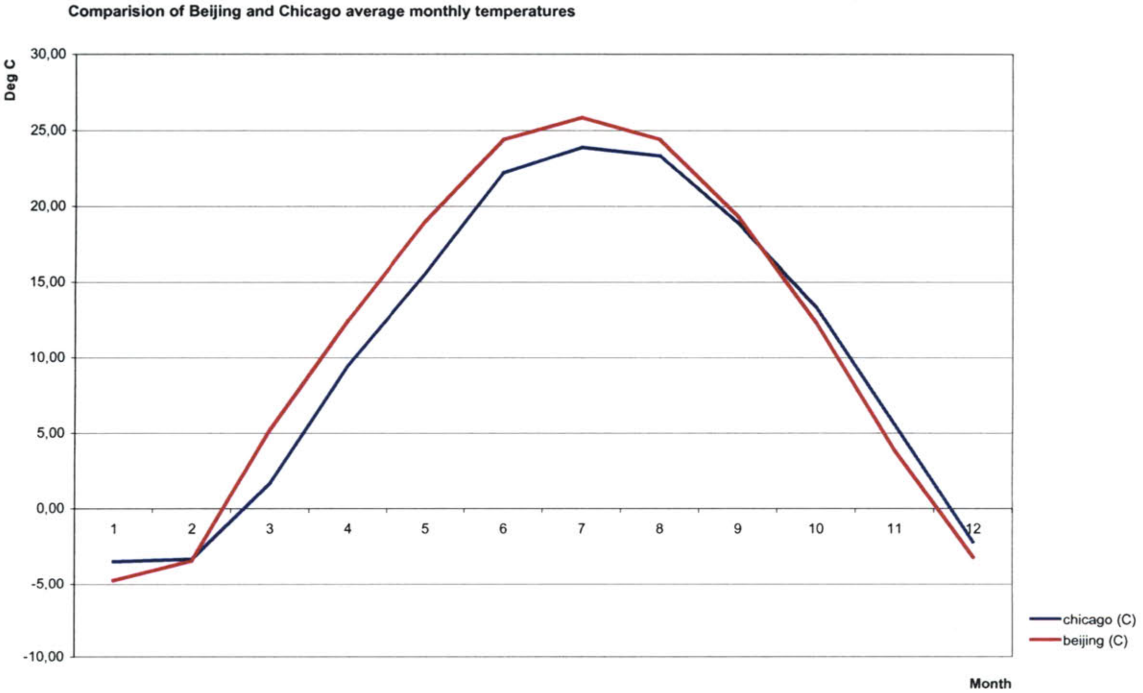


Figure 8.21 Comparison of Chicago and Beijing's monthly average temperatures

In this problem, other variables were introduced, such as wall solar absorptivity and different glazing types. The construction materials for south and north facing walls were also allowed to be different, to test if that would have an impact on the overall energy performance of the building. The two objective functions used were annual energy consumption and materials costs [including windows].

The materials used for this study are those presented in Table 8.3. Wall absorptivity could vary between 0.2 and 0.8, in 0.2 step sizes. The glazing options are described below:

Glazing types

1 - DOE2 code: 1000	Single clear, 3mm Shading coefficient = 1 Lighting transmission = 0.898 U-Value = 6.31
2 - DOE2 code: 2001	Double clear IG, 3/12/3 mm Shading coefficient = 0.89 Lighting transmission = 0.812 U-Value = 2.79

8.9.1 Results, South | North facing case

Figure 8.22 shows the first initial random generation, and the final Pareto front.

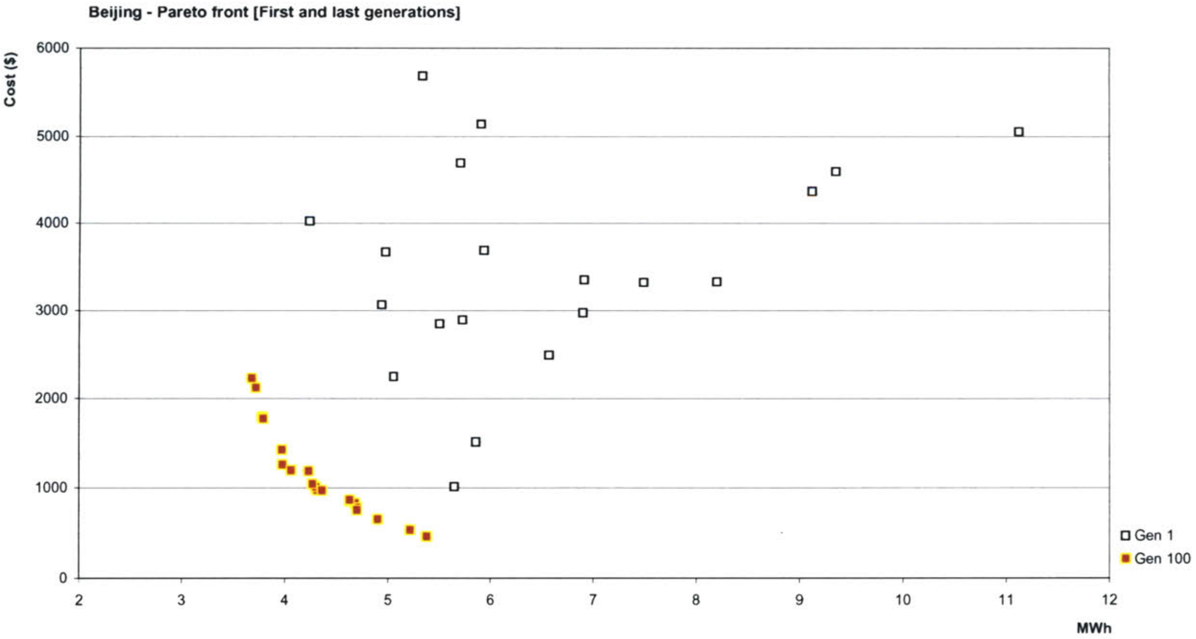


Figure 8.22 Pareto front for south | north apartment [1st and last generations]

Figure 8.23 shows the progression of generations, from generation 1 to generation 100, in steps of 10 generations. By generation 30, most of the front had been located.

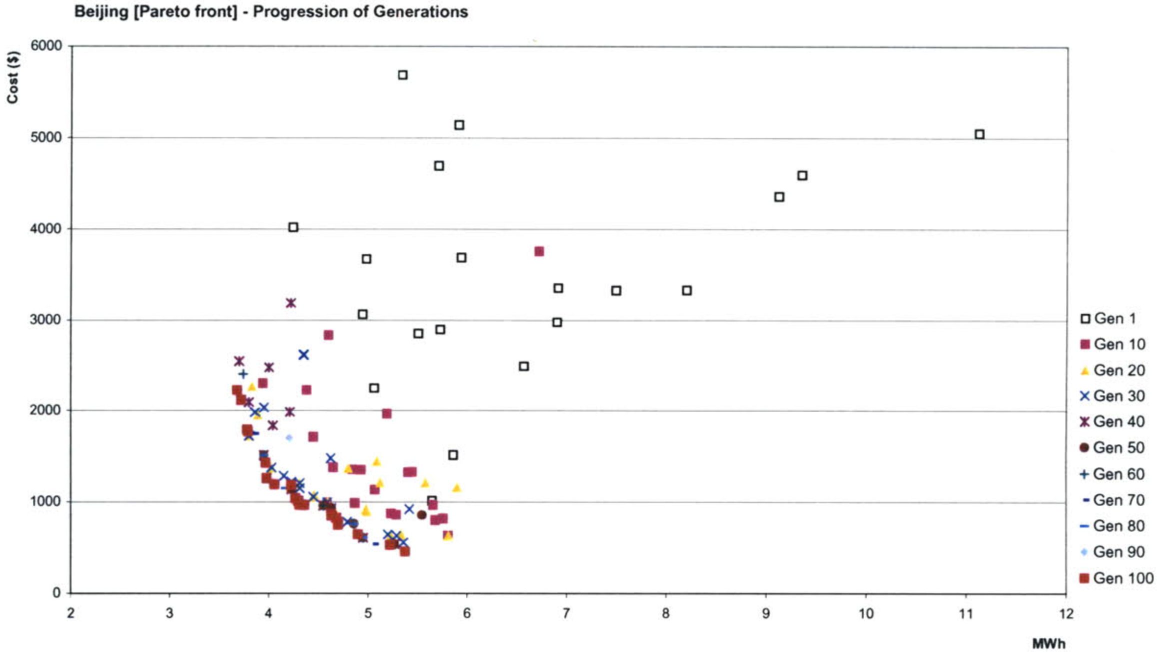


Figure 8.23 Progression of generations, from generation 1 to generation 100

Table 8.7 shows the results for each of the variables under study, for the first random generation and Table 8.8 shows the final, Pareto-front solutions. The average energy consumption of the random population is 6.5 MWh, and the average cost is about $3500.

#	South-W	South-H	South-A	North-W	North-H	North-A	Total-A	Glass	S.Absor.	N.Absor.	S-mat1	S-mat2	S-mat3	N-mat1	N-mat2	N-mat3	MWh	Cost
1	4,1	1,5	6,2	0,6	2,7	1,7	7,9	1	0,4	0,6	16	5	8	12	2	14	6,9	2973
2	5,8	1,2	7,1	0,6	0,6	0,4	7,5	1	0,2	0,6	6	10	4	12	6	13	5,1	2244
3	0,6	0,6	0,4	5,0	2,1	10,6	11,0	1	0,6	0,6	14	11	8	8	5	4	8,2	3328
4	4,1	0,6	2,5	3,2	1,8	5,9	8,4	1	0,2	0,6	6	9	6	12	4	7	6,6	2489
5	4,1	2,4	10,0	2,4	0,6	1,4	11,4	1	0,2	0,8	13	3	5	6	7	12	5,7	2893
6	5,0	1,5	7,6	6,7	2,1	14,3	21,9	2	0,4	0,6	8	3	16	4	12	9	5,9	5140
7	0,6	1,5	0,9	1,5	0,6	0,9	1,8	1	0,2	0,2	4	9	5	16	7	5	5,7	1010
8	4,1	1,2	5,0	4,1	1,8	7,5	12,5	2	0,8	0,6	15	5	12	4	12	16	5,0	3668
9	4,1	1,8	7,5	5,8	2,1	12,4	19,9	2	0,2	0,2	8	2	9	10	5	13	5,7	4696
10	4,1	1,2	5,0	4,1	1,8	7,5	12,5	1	0,2	0,6	16	1	12	13	10	5	7,5	3323
11	5,0	2,4	12,1	5,0	0,6	3,0	15,1	2	0,2	0,6	8	11	9	14	10	16	4,2	4018
12	2,4	1,5	3,6	1,5	2,7	4,1	7,6	2	0,2	0,2	13	11	13	12	14	15	5,5	2847
13	4,1	1,5	6,2	6,7	1,8	12,3	18,5	1	0,6	0,2	15	10	4	11	10	16	9,4	4598
14	4,1	0,6	2,5	5,0	2,4	12,1	14,6	2	0,2	0,2	6	11	16	13	1	6	5,9	3684
15	5,0	1,2	6,1	5,8	2,1	12,4	18,5	1	0,8	0,6	13	3	8	13	6	9	9,1	4361
16	5,0	2,7	13,6	6,7	1,5	10,2	23,8	2	0,2	0,2	16	10	13	10	6	16	5,3	5685
17	0,6	1,2	0,7	0,6	1,8	1,1	1,9	2	0,2	0,6	14	11	4	6	1	14	5,9	1510
18	0,6	2,4	1,5	6,7	2,7	18,4	19,9	1	0,8	0,2	9	9	16	6	13	7	11,1	5053
19	5,0	1,2	6,1	3,2	1,8	5,9	11,9	2	0,4	0,6	7	10	4	16	4	13	4,9	3061
20	4,1	1,2	5,0	3,2	1,8	5,9	10,9	1	0,2	0,8	15	11	6	16	11	5	6,9	3351

Table 8.7 Values assigned to variables for the 1st random population

In Table 8.7, it is possible to see all types of different materials, glazing types, window dimensions and wall absorptivity values. On the contrary, the Pareto solutions shown in Table 8.8 show a much greater convergence of variables values. Solutions are sorted in ascending order, according to energy use.

#	South-W	South-H	South-A	North-W	North-H	North-A	Total-A	Glass	S.Absor.	N.Absor.	S-mat1	S-mat2	S-mat3	N-mat1	N-mat2	N-mat3	MWh	Cost
1	3,2	2,4	7,9	0,6	1,2	0,7	8,6	2	0,8	0,6	4	5	8	6	6	6	3,7	2223
2	5,8	1,2	7,1	0,6	0,6	0,4	7,5	2	0,6	0,6	6	7	8	7	5	8	3,7	2115
3	2,4	2,4	5,7	0,6	1,2	0,7	6,5	2	0,8	0,6	4	7	8	5	5	6	3,8	1788
4	2,4	2,4	5,7	0,6	1,2	0,7	6,5	2	0,6	0,6	6	3	6	5	7	8	3,8	1768
5	3,2	1,2	3,9	0,6	1,2	0,7	4,7	2	0,8	0,6	6	1	8	6	6	4	4,0	1425
6	3,2	1,2	3,9	0,6	0,6	0,4	4,3	2	0,8	0,8	7	3	6	6	8	8	4,0	1255
7	3,2	1,2	3,9	0,6	0,6	0,4	4,3	2	0,6	0,6	4	3	6	7	8	8	4,1	1190
8	2,4	1,2	2,9	0,6	1,2	0,7	3,6	2	0,6	0,6	6	3	6	5	16	8	4,2	1183
9	2,4	1,2	2,9	0,6	0,6	0,4	3,2	2	0,6	0,8	7	3	6	7	8	8	4,3	1037
10	2,4	1,2	2,9	0,6	0,9	0,6	3,4	2	0,6	0,6	4	3	6	7	8	8	4,3	1008
11	2,4	1,2	2,9	0,6	0,6	0,4	3,2	2	0,6	0,6	6	3	4	7	8	8	4,3	970
12	2,4	1,2	2,9	0,6	0,6	0,4	3,2	2	0,6	0,4	7	3	7	6	8	8	4,4	966
13	1,5	1,2	1,8	0,6	0,6	0,4	2,2	2	0,6	0,8	5	3	8	7	8	8	4,6	849
14	1,5	1,2	1,8	1,5	0,6	0,9	2,7	2	0,8	0,2	4	3	6	6	7	8	4,6	859
15	1,5	1,2	1,8	0,6	1,2	0,7	2,5	2	0,8	0,2	4	1	6	6	8	4	4,7	826
16	1,5	1,2	1,8	0,6	0,6	0,4	2,2	2	0,8	0,6	4	3	8	6	6	4	4,7	776
17	1,5	1,2	1,8	0,6	0,6	0,4	2,2	2	0,6	0,6	5	3	7	6	8	8	4,7	749
18	1,5	1,2	1,8	0,6	0,9	0,6	2,4	2	0,8	0,6	4	3	4	6	6	8	4,9	646
19	0,6	1,2	0,7	0,6	0,6	0,4	1,1	2	0,8	0,2	5	3	7	6	8	4	5,2	529
20	0,6	1,2	0,7	0,6	0,6	0,4	1,1	2	0,6	0,4	7	3	4	4	8	8	5,4	454

Table 8.8 Values assigned to variables for the final Pareto solutions

All windows are double glazed. North window dimensions are in general very small. South window dimensions are average sized for the best energy performance solutions [still quite smaller than the maximum constrains allow], and quickly decrease for the worst performance solutions. Decreasing south window size is the best strategy to reduce construction costs, but has detrimental effects both in terms of daylighting and useful solar gains. The south wall absorptivity in always at high values, either 0.8 or 0.6. North window absorptivity has more random values, since solar gains are not too significant in that direction so benefits from increasing its absorptivity are not too evident, but values tend in any case toward the higher bounds.

Finally, in terms of wall materials, north walls are basically all filled with insulation, while in the south walls insulation exists in the outer layers, but inner layers tend to become air spaces in the more cost-savings solutions [with a subsequent decrease in energy performance]. Remembering that air layers have code numbers from 1 to 3, and insulation from 4 to 8 [with those values closer to 8 representing better insulation materials], it is apparent that in order to save costs, worst insulation materials are used in the south wall, together with the introduction of air layers, but the north wall remain highly insulated for most cases, only decreasing towards the worst performance solutions.

The average energy consumption level of the population decreased by 33% in relation to the initial random population, while the costs were reduced by 68%, a very significant value. Masonry use was completely excluded, since insulation materials are less expensive and thermal mass is not an issue in this cold climate. In any case, the floor slabs can serve as thermal mass to store the heat entering through the south facing windows.

8.9.2 Results, East | West facing case

Another set of experiments was done rotating the building by 90°, so that it would face E/W instead. Figure 8.24 shows the Pareto front results.

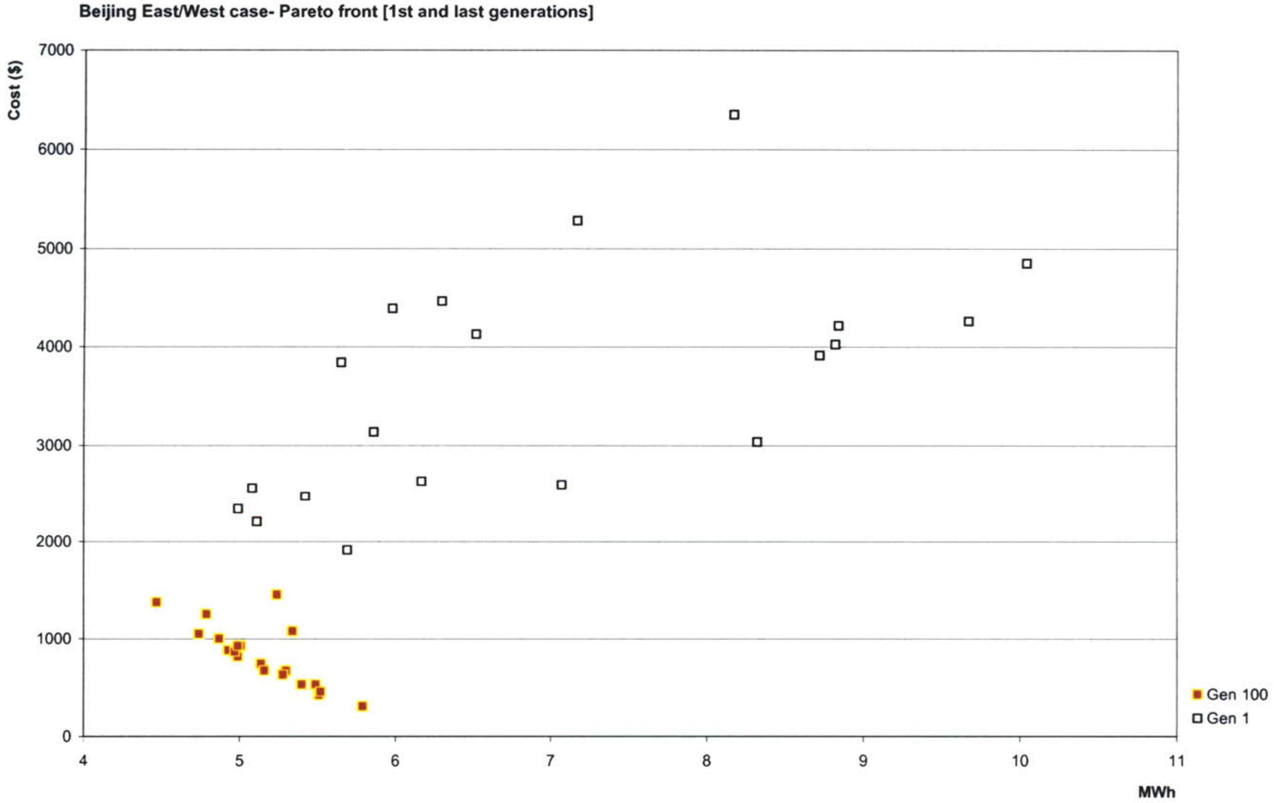


Figure 8.24 Pareto front results, East / West case

From figure 8.24, one can see that the energy consumption levels are always higher for this orientation than for S/W, as expected, since E/W is a more unfavorable orientation.

However, costs remain lower than for S/N. To try to understand why, one must look at Table 8.9, with the values for the Pareto front points.

#	West-W	West-H	West-A	East-W	East-H	East-A	Total -A	Glass	W-absorp	E-aborp	W-mat1	W-mat2	W-mat3	E-mat1	E-mat2	E-mat3	MWh	Cost ($)
1	3,2	0,6	2,0	1,5	1,2	1,8	3,8	2	0,2	0,6	5	8	6	8	8	8	4,5	1377
2	3,2	0,6	2,0	1,5	0,9	1,4	3,3	2	0,6	0,6	4	7	7	8	8	8	4,7	1050
3	3,2	0,9	2,9	1,5	0,9	1,4	4,3	2	0,4	0,6	4	7	7	12	7	8	4,8	1252
4	1,5	0,6	0,9	1,5	1,2	1,8	2,7	2	0,2	0,6	4	5	6	8	7	16	4,9	999
5	2,4	0,6	1,4	1,5	0,6	0,9	2,3	2	0,6	0,6	8	1	7	8	8	16	4,9	878
6	2,4	0,9	2,1	1,5	0,6	0,9	3,1	2	0,4	0,6	5	3	4	4	8	8	5,0	863
7	2,4	0,6	1,4	0,6	0,6	0,4	1,8	2	0,6	0,6	6	3	6	8	4	8	5,0	816
8	1,5	0,6	0,9	1,5	1,2	1,8	2,7	2	0,2	0,6	4	7	7	8	11	8	5,0	924
9	2,4	0,6	1,4	1,5	0,6	0,9	2,3	2	0,2	0,6	4	5	6	8	3	16	5,0	924
10	1,5	0,6	0,9	1,5	0,6	0,9	1,8	2	0,6	0,6	4	7	7	8	4	8	5,1	738
11	2,4	0,6	1,4	0,6	0,6	0,4	1,8	2	0,6	0,6	4	3	6	8	7	16	5,2	672
12	3,2	0,6	2,0	1,5	0,6	0,9	2,9	2	0,2	0,8	4	15	6	16	3	16	5,2	1455
13	0,6	0,6	0,4	1,5	0,6	0,9	1,3	2	0,8	0,6	4	7	7	8	7	8	5,3	629
14	0,6	0,9	0,6	1,5	0,6	0,9	1,5	2	0,2	0,6	4	7	7	8	3	8	5,3	667
15	2,4	0,6	1,4	1,5	0,6	0,9	2,3	2	0,2	0,8	4	13	4	8	7	16	5,3	1077
16	2,4	0,6	1,4	0,6	0,6	0,4	1,8	2	0,8	0,6	4	1	4	8	8	16	5,4	528
17	0,6	0,9	0,6	1,5	0,6	0,9	1,5	2	0,8	0,6	4	3	7	8	7	12	5,5	526
18	0,6	0,9	0,6	0,6	0,6	0,4	0,9	2	0,4	0,6	4	3	5	8	11	8	5,5	418
19	0,6	0,6	0,4	0,6	0,6	0,4	0,7	2	0,6	0,6	6	3	4	8	15	16	5,5	451
20	0,6	0,6	0,4	0,6	0,6	0,4	0,7	2	0,8	0,6	4	3	4	8	7	16	5,8	304

Table 8.9 Pareto front results, East / West case

In general, the reason for the lower costs is that window sizes are very reduced, even in the best energy performance cases. On average, west windows are still a bit larger than east ones, what could be a possible explanation of the appearance of concrete block in the innermost layer of some of the east walls. The low afternoon sun entering through the west windows may hit the east walls, and so having some mass there would help energy storage. The other possible explanation for the use of masonry in this E/W is that the GS can assign a little higher 'budget' to the construction on the walls because window cost is so small.

Finally, all windows are double glazed, and east-wall absorption values tend to be higher than west ones, suggesting that east orientation may suffer from very cold early morning temperatures, and thus the improved wall construction, including thermal mass, might help to counteract that.

8.10 Conclusions of the chapter

Results from the Pareto based studies proved in general to be rather interesting, as understanding the trade-offs between conflicting objectives added another dimension towards our ability to interpret results.

The final algorithm implementation was quite successful in locating spread-out, well-defined Pareto fronts, what provided enough confidence on the results obtained. The Pareto front was usually quickly found by the GS [a similar behavior to that documented in the literature about NSGAs], and most of the computational effort was spent in small refinements to the front. This suggests computational time could be reduced if the algorithm was run for less generations, without significant loss of quality of solutions.

The materials studies were valuable in suggesting that using highly insulated walls and leaving thermal mass for floors and roofs slabs may be a cheaper and more efficient means of construction. The experiments using GGE as an objective function proved cellulose insulation to be an excellent alternative to conventional insulation, both in terms of environmental impact and to achieve good thermal resistance. Obviously, wall thicknesses become much higher when cellulose fill is used, but that can be used to architectural benefit too, for example, by having recessed windows in deeper facades, which would provide some ‘free’ shading.

Finally, the expansion of the method to include more than two objective functions could lead to other interesting results, and should be another of the future work developments that we would like to implement.

CHAPTER 9 Shape Generation

9.1 Introduction

The work described in chapter 7 had suggested the possibility of including shape manipulation in the realm of this dissertation, as a more encompassing approach towards understanding architectural adaptation to the exterior physical environment. The preliminary experiments with roof geometries described towards the end of that chapter gave rise to posterior speculations on the limits of the applicability of geometric variation to environmental performance studies. This chapter further develops that topic, not only by formalizing and implementing some experiments, but also by speculating about many other possibly interesting applications that may be pursued in the future.

We also combined in one of the experiments the concept of Pareto fronts with that of shape generation. The aim was to generate 'frontiers' of alternative shapes that would represent trade-offs between different evaluation criteria, and see how those conflicting objectives affected the three-dimensional molding of the building shape. Detailed analysis of physical phenomena happening in each space, like daylight-level fluctuations across the year, can also be made possible by the Generative System, as will be seen later in the chapter.

9.2 Initial experiments: Varying external wall tilts

A number of experiments using shape generation are described in this chapter. In the first experiment, the overall shape of the building was fixed, and the GS was allowed to play only with the tilt of the exterior wall surfaces. The initial, schematic building shape from which the GS departed can be seen in figure 9.1.

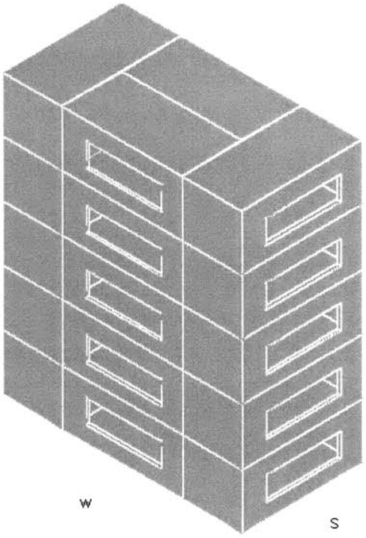


Figure 9.1 Initial building shape

The building has five identical floors, with the narrower sides facing South | North, and the longer sides East | West. Each floor has four rooms, identical in their dimensions. Windows are only allowed to exist in the longest wall in each space [only one window per space], even though the south and north rooms have also exposed exterior walls in their ends, so that results would be easier to interpret.

The parameters used in these experiments were: wall tilts, which could vary from 60º to 140º [a vertical wall having a 90º tilt, a smaller value representing a tilt inwards, and a larger value a tilt outwards], and window dimensions. The location chosen for the experiments was Oporto, representing a mild climate.

Results are shown in figure 9.2. The generated building shapes are formally complex and interesting, especially considering the very simple initial shape from where they originated. In general, the walls tended to tilt away from the sun, especially for the east and west facades. For these orientations, window size tended to increase with the tilt of the wall, as the less exposed windows would allow less heat gain into the building but admit natural light. The north and south walls tend to be less tilted. North windows are quite large, as already happened in the example of the building by Siza. Towards the south, windows in walls that remained vertical are smaller than those carved in the tilted walls. Towards the top floors, where more tilted surfaces emerge, larger openings appear in the facades, since they are less exposed to the sun.

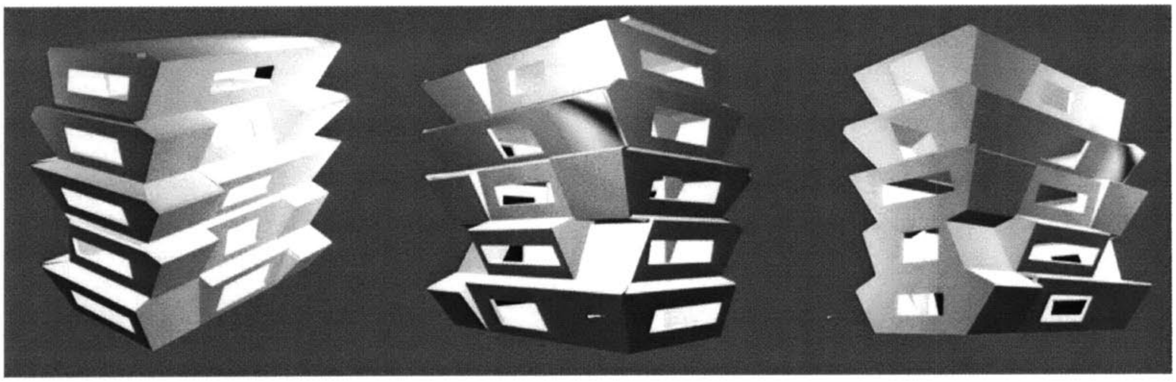


Figure 9.2 Results for Oporto. From left to right, three views of the same building: northwest, northeast and southeast.

These results suggested that from a simple initial layout, and through the application of very simple rules, or parametric deformations, it was possible to generate shapes with a high degree of complexity, that would be hard to predict from the outset and that were trying to use the simple mechanisms at their disposal to enhance the building's adaptation to the environment.

This seemed quite close to the original concerns that had led to the development of this project, and thus a path worth further exploration, both in terms of actual implementation and performance of experiments, and in terms of speculation about possible directions for future work and further experimentation.

The next section will describe the work developed in the context of these parametric shape manipulation experiments.

9.3 Parametric shape manipulation

In this section we researched the question of finding initial layouts that could allow for the emergence of a rich variety of formal solutions when manipulated by the Generative System. Determining adjacencies was perceived as a major problem, so a very simple experimental layout was chosen where all adjacencies between spaces were predetermined. Later in the chapter we speculate about other examples and interesting areas for pushing forward the limits of this research area.

9.3.1 Description of experimental setup

The basic layout from where the new solutions could emerge was very simple: in plan, it was a square divided in four similar squares. In the 1st floor, this corresponded to four rooms [R1, R2, R3 and R4, as can be seen in figure 9.3], which could vary in their length and width, but had a fixed, similar height. In the 2nd floor, there were four other rooms [R5, R6, R7, and R8], but instead of only varying in length and width, they could vary in height too. Moreover, the tilt of the roof could vary from a flat roof to a maximum of 45°. The azimuth of the tilt could either be 0° or 90°, as can be seen in figure 9.3. Whenever there was a tilted roof, a roof monitor was also generated.

This very simple problem setup, which has only 44 independent variables, generates however about 350 dependent variables that must all be correctly coded for the program to run without mistakes. Some of the possible mistakes would make the program crash, but others would still allow the program to run while providing adulterated results, which makes them more problematic than the ones that simply prevent the program from running and thus force the user to identify and correct them.

By looking at the specificities of transferring building geometrical information into a simulation program that calculates heat transfer, it is possible to understand the reason why so many dependent variables are generated from such a simple layout, as will be discussed later.

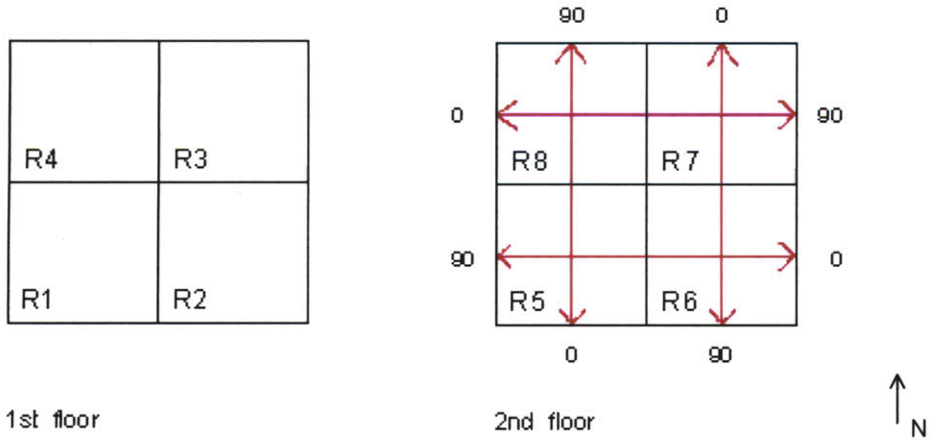


Figure 9.3 Basic layout for 1st and 2nd floor variations. The red arrows show the roof tilt direction. For example, it Room 6 roof azimuth is 0°, the roof will be tilted with its higher end towards the right, and a roof monitor would appear on the right side of the room.

As it was mentioned before, BDL has three coordinate systems, namely the building, room and wall systems. To locate a new space in the overall building layout, it is necessary to correctly determine its coordinates in the building coordinate system. The building coordinate system origin (0,0,0) was determined to be the intersection point of the 2 axes defining the 4 squares. From that reference point, it is possible to correctly locate each of the spaces, even though their dimensions are variables generated by the genetic algorithm. By using the insertion point of each room as its lower left corner, and considering the room's azimuth [North = 0, East = 90, South = 180, West = 270], it is possible to encode the insertion point of each space as shown in figure 9.4 and in table 9.1.

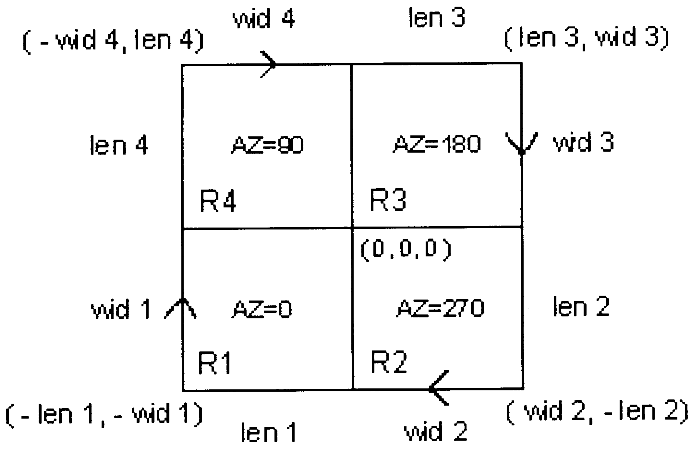


Figure 9.4 1st floor with room azimuths and coordinate systems. The arrows indicate the direction of the room's azimuth.

Room	**Azimuth**	**Insertion point**
1	0	(- len 1, - wid 1)
2	270	(wid 2, - len 2)
3	180	(len 3, wid 3)
4	90	(- wid 4, len 4)

Table 9.1 1st floor room azimuths and insertion point coordinates

As for the Z coordinate, which controls the height at which each room is inserted, it was assumed that for the 1st floor that value would be always 0. Later in the chapter, some considerations will be made about this assumption.

A similar procedure was used to locate the 2nd floor spaces, with the difference that the Z coordinate was not 0 as in the 1st floor, but it was set as the height of the 1st floor rooms [2.8 m]. Since the height of the 2nd floor spaces was allowed to vary, the Z coordinate for the roofs was variable too, and equal to the height of the space.

Because the roof azimuths could vary, the roof insertion point [X and Y coordinates] would differ if the roof azimuth was set by the GA to 0° or 90°. This was solved with a number of simple if-then rules, of the type:

```
if
roof5-azimuth = 0°
then
roof5-Xcoordinate = …
roof5-Ycoordinate = …
else
roof5-Xcoordinate = …
roof5-Ycoordinate = …
```

Until now, a description for the correct handling of a variable building geometry has been presented. However, this represents only one of the simplest parts of the problem setup. When a building geometry is fed into a thermal simulation program like DOE2, other layers of information apart from the purely geometrical ones become crucial. Topological information is one example. Adjacencies between spaces are important pieces of information for a program that calculates heat transfer across surfaces. In the specific example used in this section, that problem was simplified by the overall layout chosen, where space adjacencies are fixed. Room 1 is always adjacent to Room 2 towards the east, to Room 4 towards the north, and has external walls facing south and west. The floor is always adjacent to the ground, and the ceiling to Room 5. However, adjacencies to the external environment are not entirely described using this layout, once the spaces start to dynamically parameterized by the GA.

Figure 9.5 illustrates the simplest example of altering just one room's dimension. While in the first case, there is only an internal wall between Room 1 and Room 2, in the second case a previously nonexistent exterior wall has appeared in Room 1. In the third case, a new exterior wall also appears, but this time belonging to Room 2. Within BDL, the interior wall must be declared for both Room 1 and Room 2, but the new exterior wall should be declared specifically for the room it belongs to.

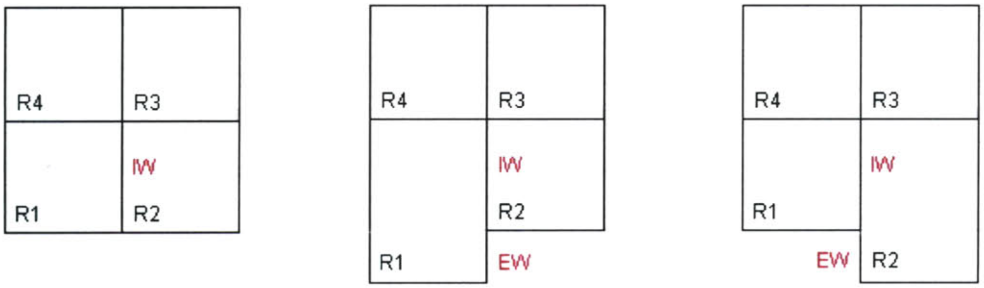


Figure 9.5 – Example of external walls appearing due to parametric variation of room dimensions

While in geometrical terms the interior and exterior walls might be similar [it is the same vertical plane], in construction terms those entities might already be different, since usually exterior walls are built with different materials than interior ones. In terms of heat transfer, they are even further different. An interior wall is just a boundary between two indoor environments that in most buildings are usually not too different, and thus heat transfer across the wall is small. In contrast, an exterior wall is a boundary between indoor and outdoor environments that can be very different, and thus the wall would have significant heat transfer across it. Furthermore, for the exterior wall the azimuth it faces is important, due to its relation to sun position [a south-facing wall behaves differently from a north-facing one]. Another important characteristic of exterior walls are that they may have windows, which are determinant factors in the environmental performance of the building.

In the 2nd floor, the generation of new external walls becomes more complex, since the rooms are also allowed to vary in height. For each adjacency situation, four different possibilities have to be predicted, which are depicted in figure 9.6. In the first case, no new external wall is created for Room 1. In the second case, a vertical external wall appears [E1]. In the third case, a horizontal one [E2] appears. In the forth case, an L shaped external wall emerges, which has to be decomposed in two pieces [E1 and E2], as shown in the figure, due to BDL format specificities.

These four possibilities, that in geometric terms are rather simple, pose nevertheless a data representation problem for the generative system. Because the GS does not have a CAD interface, from which it might be able to read information on building geometry and generate the respective BDL file, the system needs to work with a fixed-format BDL file. This file must include from the beginning all the possible geometric occurrences within the

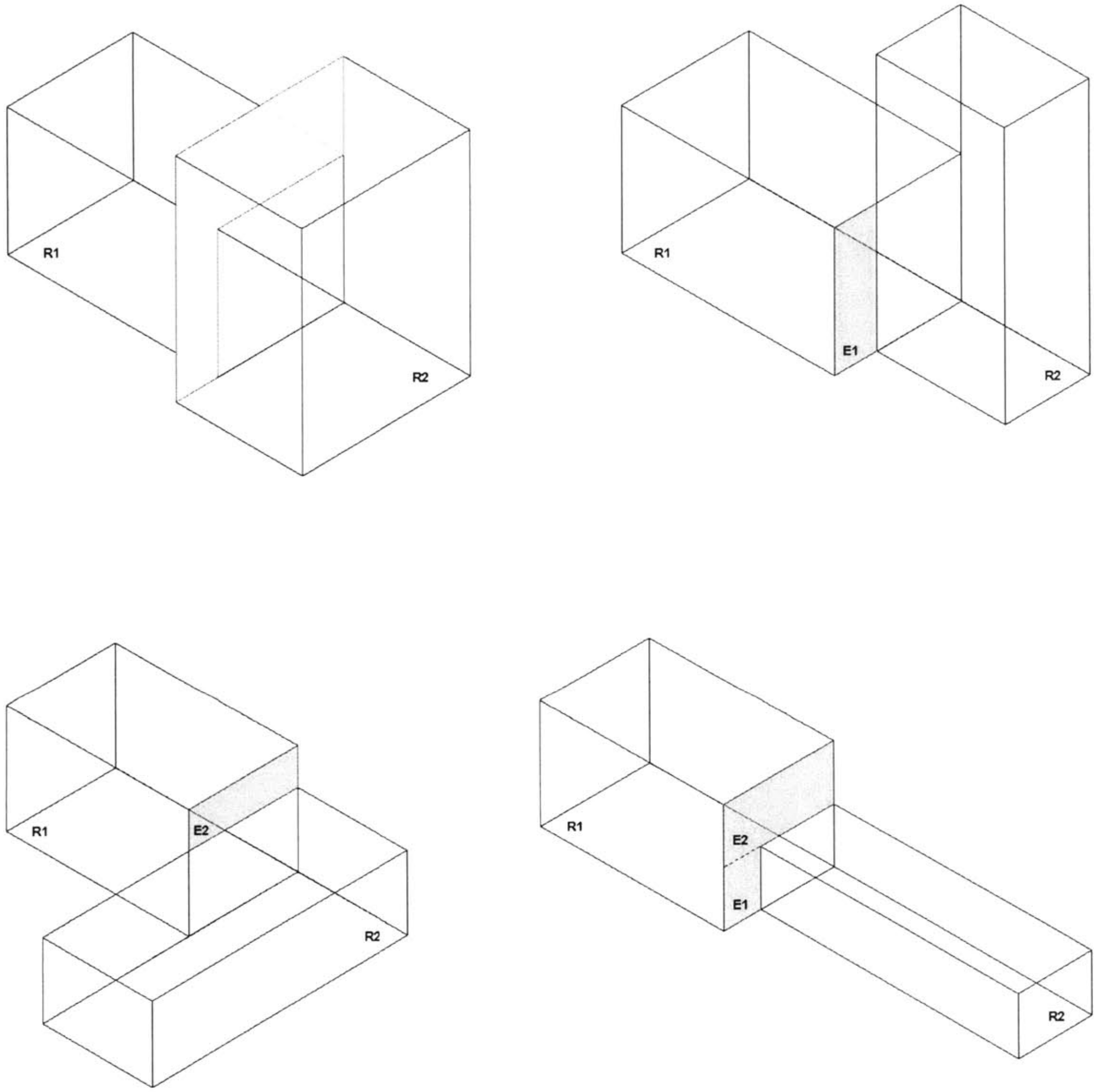


Figure 9.6 – Emergence of new external walls in Room1, according to Room2 dimensions

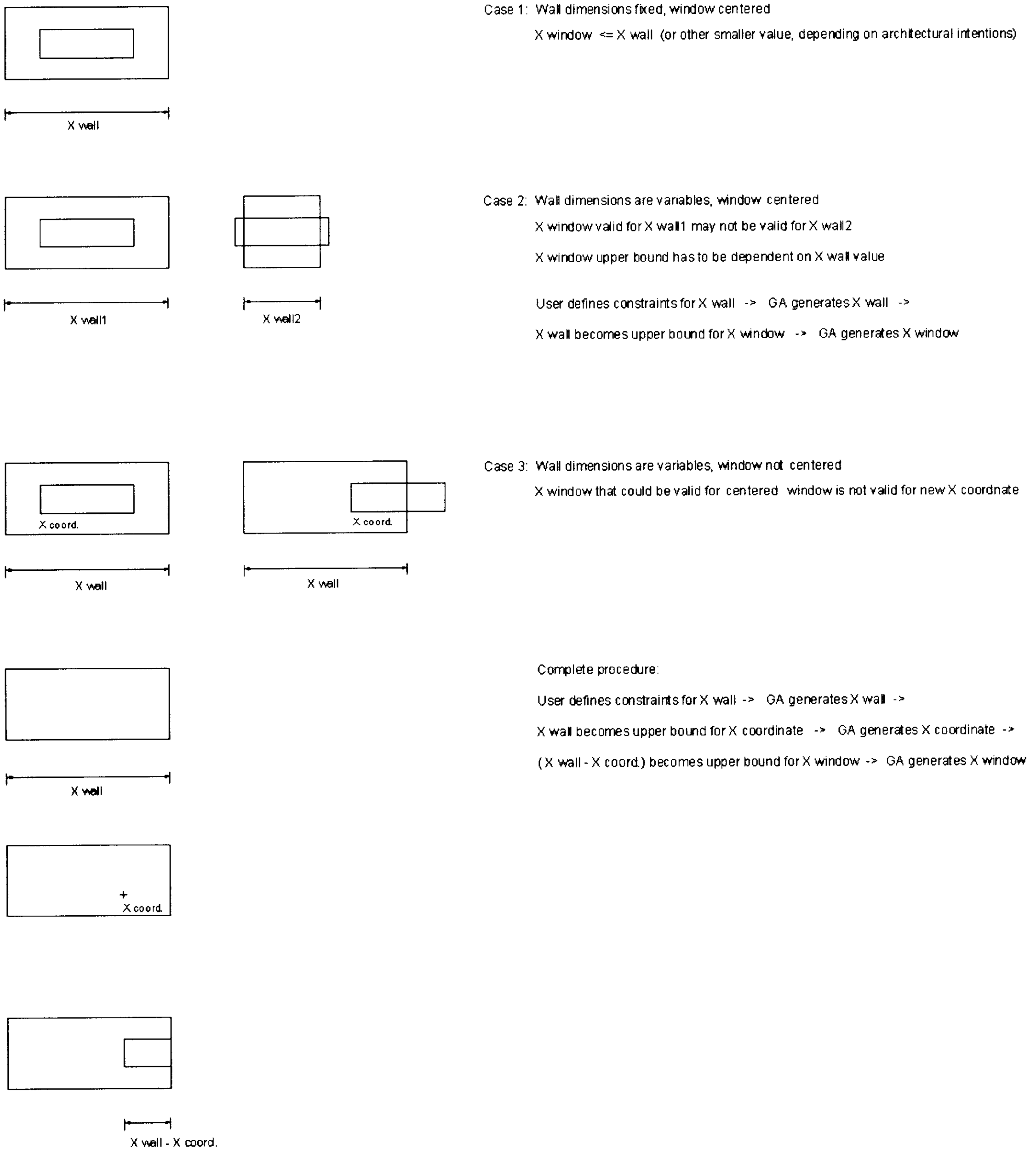
problem setup. In this example, the most complex case would be the L shaped wall, with occurrence of both E1 and E2 components. Given this, all rooms must predict the possible existence of E1 and E2 entities. If a given entity does not occur when the genetic algorithm generates a new individual, that entity parameters and coordinates must be driven to zero. If that entity does exist in the current geometric formulation, its parameter values [like width and height, for example, and other possible information about construction materials, wall solar absorptivity, internal light reflection values, etc.] must be calculated from the independent variables values, and its insertion point coordinates determined for correct positioning in relation to the space, using the room coordinate system [not the building coordinate system as before]. For simplicity, the code prevents the appearance of windows in these external walls, whose existence is uncertain, as this would add another layer of complexity to the problem. However, in a more complex problem formulation it would also be possible to predict the appearance of openings in the new facades created. Finally, it should be added that once an entity is declared in BDL, its dimensions [length, width, height] are not allowed to be zero. When that entity does not exist, its value cannot be driven to zero, but to a very small value close to it [like 0.001], which makes its existence negligible. If a zero value is given, the program will crash during execution.

A similar procedure to that of 2nd floor external walls had to be used for roofs of 1st floor spaces, and exterior floors slabs of 2nd floor rooms. In all of those, the most complex case is the existence of an L shaped element, and thus all rooms must incorporate the possibility of this occurrence. If any elements do not exist in a given solution, the program must drive their values to a very small number close to 0. Forgetting to include any of these potential surfaces can introduce significant errors into the search procedure, as they represent important heat transfer areas in the building. Finally, the interior wall areas between each adjacent space must also be calculated.

As for window size and positioning, a number of new issues emerged within this new problem formulation of variable building shape. When building shape is fixed, it is possible to easily determine the upper bounds for the window size constraints, as those are limited by the size of the wall in which the window is located. However, if the wall size is not known in advance, since it is a variable for the GA, it is not possible to determine that upper bound in advance. This represents a major drawback in terms of the standard genetic algorithm functioning. In common GA implantations, the constraints for each variable are determined in advance, prior to running the program. To overcome the fact that constraint bounds are not known in advance, the constraints would have to change dynamically during the course of the program. This way, the value for a given constraint would be dependent from the value of a variable generated by the GA. This would imply a several-steps procedure. First, the upper and lower bounds for wall size are given. Then, the GA generates a valid variable value within those constraints, for both wall width and height. Those values are then used as the constraints for window sizes, again for both width and height. This way, it would be possible to assure that a given window would always fit in the corresponding wall.

Another possible complication emerges if the windows are not centered in the wall. If the windows are centered, determining the coordinates for the window insertion point [which is always its lowest left corner] can be done after knowing both the wall and window dimensions, using simple rules. However, if the insertion point coordinates are themselves variables [meaning that the window can be at any position in the wall], another problem appears. A valid window dimension for a centered window may not be valid for an off-centered one, since the window may not fit in the wall any longer. This would imply another revision of the GA code, introducing another steps into the algorithm. In this case, first the constraints for wall dimensions would be input by the user. The GA would generate valid wall instances within those bounds. The wall dimension values would then be used as constraints for the generation of insertion point coordinates (x, y) in the wall coordinate system, to make sure those insertion points would be inside the wall. Then, the window dimensions constraints would be calculated as the difference between the wall dimension and the insertion point coordinates, to ensure that valid window sizes were generated. Only then could the window dimensions be generated by the GA. This process is illustrated in figure 9.7.

The dynamic constraints algorithm described above has not yet been implemented, and constitutes one of the future developments mentioned in the 'conclusions and future work' section. Given that, it was necessary to find a simplified solution to allow for the initial running and testing of the variable-shape experiments described in this chapter. The simplest solution was to make the window width equal to wall width [thus becoming a dependent variable instead of an independent one], and that was the path followed here. In terms of height, the 1st floor windows posed no problems, as wall height was fixed and thus constraints could be determined in advance. For the 2nd floor, the maximum window height was set equal to the minimum wall height, to ensure windows would always fit into the respective wall, no matter what their height may be.



Case 1: Wall dimensions fixed, window centered

X window <= X wall (or other smaller value, depending on architectural intentions)

Case 2: Wall dimensions are variables, window centered

X window valid for X wall1 may not be valid for X wall2

X window upper bound has to be dependent on X wall value

User defines constraints for X wall -> GA generates X wall ->

X wall becomes upper bound for X window -> GA generates X window

Case 3: Wall dimensions are variables, window not centered

X window that could be valid for centered window is not valid for new X coordnate

Complete procedure:

User defines constraints for X wall -> GA generates X wall ->

X wall becomes upper bound for X coordinate -> GA generates X coordinate ->

(X wall - X coord.) becomes upper bound for X window -> GA generates X window

Figure 9.7 Rules for window generation and insertion, for a standard GA environment [case 1] and a dynamic constraints environment [cases 2 and 3]. Y coordinates obey similar rules.

These simplified rules have the drawback of allowing little variation in façade design. Windows always stretch from wall to wall, and can only vary in height. This led to a certain standardization of the window solutions generated, which is nevertheless counteracted by the great variety of shapes that are generated by the GS. To introduce more diversity into the experiments, and also as an useful environmental analysis strategy, the window height can be driven to a very small value close to 0, meaning that if the GA finds that reducing a window area introduces benefits in terms of overall building performance, it is allowed to make it disappear.

The location of daylighting reference points has to be calculated by the program for each particular space geometry. The rule for placing the sensors were: one sensor in the center of the space, and the other 2 meters away from the innermost walls, that is, the walls that have no windows. This tries to ensure that natural light is used throughout the space, and that it penetrates into the deeper areas of the rooms. The Z coordinate of the sensors is 0.75 m, approximately desktop height. Figure 9.8 shows sensors locations for a randomly generated shape. Sensors for 1st floor spaces are in red, and for 2nd floor are in yellow.

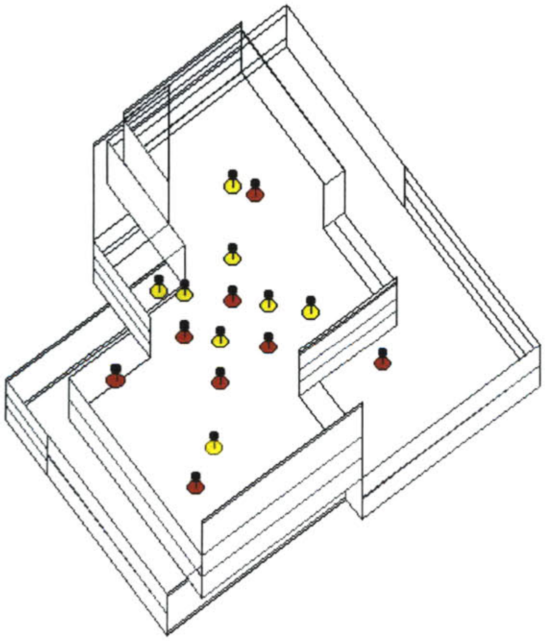


Figure 9.8 Automatic daylight sensors placement in a randomly generated building configuration. Sensors in red are for the 1st floor, in yellow for the 2nd floor.

Other parameters, like space area and volume, have to be calculated after the independent variable values are generated. In calculating the volume, it should not be forgotten that when a tilted roof exists, the corresponding volume under it must be added to the basic parallelepiped volume.

A remark should be added about self-shading calculations. DOE-2 will not calculate building self-shading unless appropriate surfaces are explicitly declared as shading surfaces. Since from the outset it is unknown what surfaces will be shading others, all exterior planes [wall, exterior floors, roofs] should be declared as shading surfaces so that they are considered as such. This can be particularly important in cases where a 2nd floor space projects over a 1st floor one, and acts as an overhang for it, influencing both solar gains and daylighting levels, or for rooms with different heights, tilted roofs, and recessed external walls in relation to adjacent ones.

9.3.2 Experiments

Some initial random configurations generated by the GS within the above described framework are shown in figure 9.9.

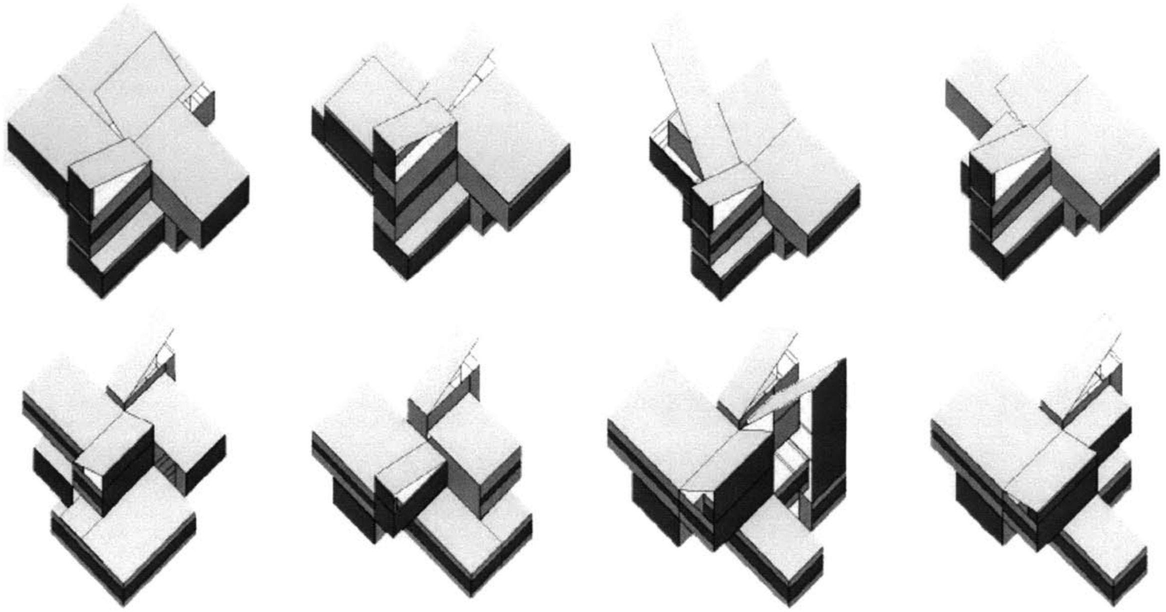


Figure 9.9 Some initial random configurations generated by the GS using the previously described layout

A major concern in these experiments that related shape manipulation and space layout to environmental performance and energy use was that the main strategy the Generative System would use to reduce building energy consumption would be to reduce building area. No matter how efficient and adapted to the outdoor environment it is, a much larger building will always consume more energy than a very small building. So, the predicted outcome from experiments using annual energy consumption as the objective function would be a population of minimum possible dimensions buildings, with some variation in façade design that would generate some differences between them.

It was thus evident from the outset that building area would have to be included in the fitness function somehow. This was implemented in two different ways. One implied the use of penalty functions, and the other used Energy Use Intensity [EUI] as the objective function, which translates energy use per unit area.

9.3.2.1 Penalty functions

The use of penalty functions was first experimented. The main idea was that each of the building's floors should have a certain area, hypothetically related to functional and programmatic requirements. The GS could assign different areas to each of the four spaces in that floor, in the way that it found to be better in terms of environmental performance, but the total area of the floor would have to equal a predetermined number of square meters. The penalties were then calculated according to the amount by which that area requirement had been violated by that solution. The absolute value of the difference between the required area and the actual one was used, so penalties would be applied both for too small and too large spaces. A similar method was applied to calculate the penalty for violation of 2nd floor area requirements. These penalties would then be added to the original fitness value of the solution [annual energy consumption], and would degrade that individual's fitness according to the extent of its violation of the area requirements.

This method intended to ensure that if the GS tried to reduce floor area to a minimum so that the energy consumption would be low, a high penalty function would degrade that solution's performance and make the GS move away from it. The penalty values could vary by a large extent, since they are based on floor area calculations, and floor area is allowed to vary significantly in this problem set. Each room dimension can vary between 3m and 15m, what means that each room area can go from 9m^2 to 225m^2. This represents already a large variation, but if we aggregate those areas into total floor area, each floor plan can vary from 36 m^2 to 900 m^2, a very significant range. For this reason, penalty values for solutions that seriously violated area requirements could be quite high. Penalties were based on a required area for each floor of approximately 470 m^2, which is about half of the permissible range. So, the penalty for the 1st floor area violation was calculated as:

penalty1= (abs (470-floorarea1) / 470) * 70

The factor of 70 at the end was adopted after experiments using other factors, as it was found that too small penalty factors would still make area reduction the best strategy for energy savings.

The penalty function method was developed because there is no formal way in genetic algorithms to constraint the outcome from a combination of variables. For example, it is possible to place upper and lower bounds in a room's length and width, but not in a room's area, since that is the result of a multiplication of two independent variables.

Figures 9.10 and 9.11 show results for experiments using penalty functions, for Oporto's climate. A typical GA implementation was used here, with total population size equal to 30 individuals, from which only 9 are represented here. Figure 9.10 shows six good results, and figure 9.11 shows three poor-performance ones. The GS was run for 200 generations.

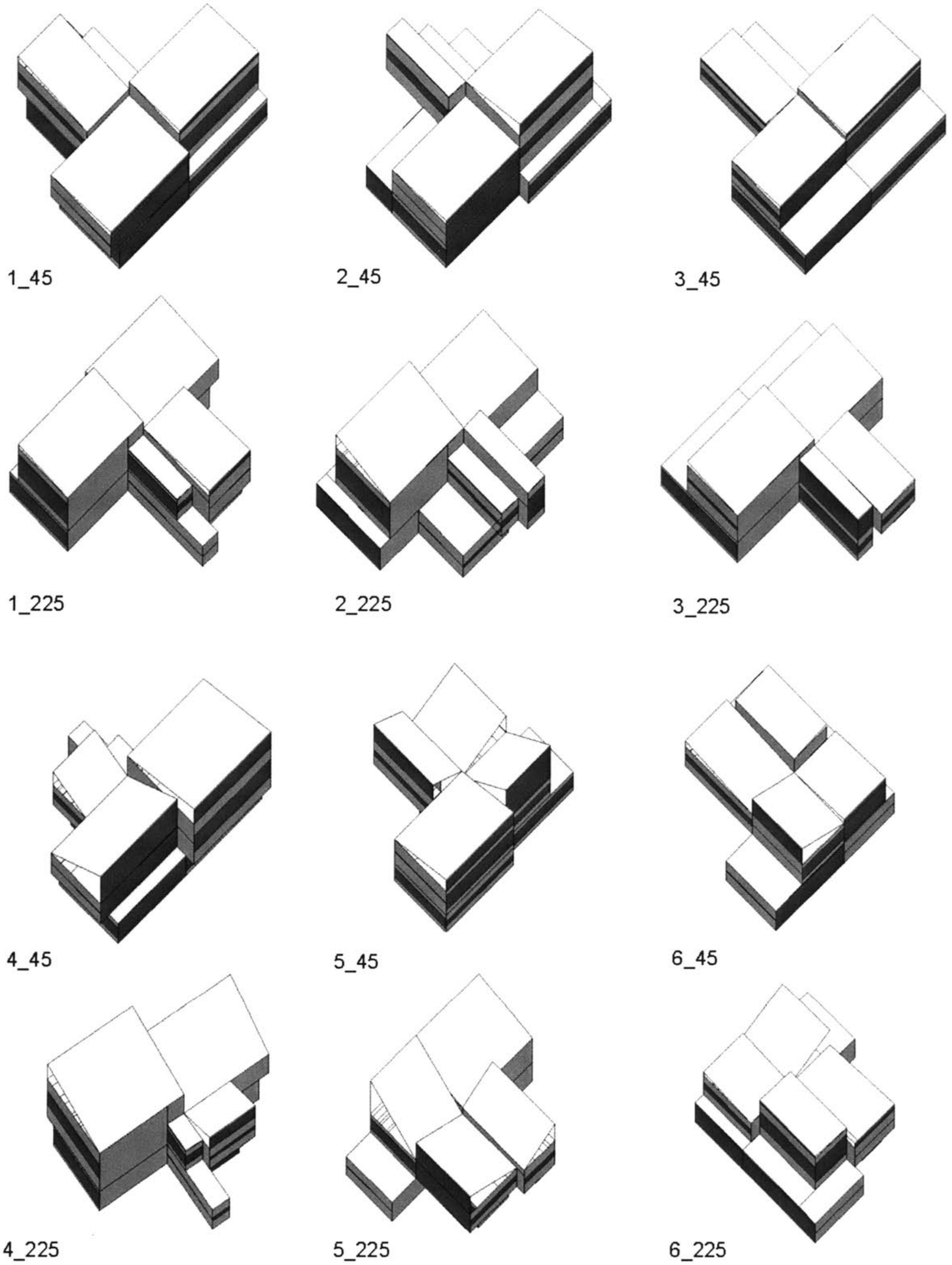


Figure 9.10 Best solutions for Oporto using penalties method. The numbers show solution fitness in ascending order [1 being the best solution]. For each building there is a SW view [45º] and a NE view [225º]. The darker areas in the walls represent windows. Drawings are not to scale.

Solution #	Fitness without penalty [MWh]	Penalty for area violation	Final fitness value
1	96	1	97
2	82	26	108
3	98	11	109
4	92	19	111
5	105	9	114
6	85	30	115

Table 9.2 Initial fitness values in MWh of the best solutions [solution numbers correspond to those in image 9.10], penalty size for area constraints violation, and final fitness values of the solutions

From figure 9.10 and table 9.2, some conclusions can be drawn. Observing the values in the table, it is possible to see that the best solutions in the population use different strategies to achieve small fitness values. While some stay closer to the required areas, increasing energy consumption but decreasing penalty values, others reduce their energy consumption levels by cutting to some extent on floor area, but still achieving a good final fitness value.

Solution 1 is the best. Even though it complies with area requirements, it manages to do so without greatly increasing energy consumption values, what reveals a good degree of adaptiveness to the environment. It chose to create elongated spaces facing south, with generous opening sizes to south, but reducing them toward the west. In general, east / west facades are smaller that south/north ones. Toward the northern side on the building, it avoided the northeast orientation, which is always unfavorable due to reduced lighting levels for most of the day, placing in that corner very small spaces that could be used as service areas. To the northwest, it used larger volumes [even if smaller than those towards the south], and used the 2nd floor volume to shade the 1st floor west-facing opening, to control solar gains, such as it had done with the southwest room. In the overall, this seems a balanced and reasonable solution, and demonstrates the GS is able to create appropriate geometries for a given problem.

Solution 2 is not very dissimilar to 1 towards the south side, but towards the north it substantially reduced the areas of the rooms. This generated a substantial energy reduction, but also a quite large area penalty, which lead to a solution more than 10% worse that 1.

Solution 3 is again not too dissimilar to 1, but has higher energy consumption levels, even though its area has been reduced [as shown by the penalty function value], probably because of the exposed south-facing 1st floor roofs, since the 2nd floor is recessed into some kind of terrace. It also has quite larger west-facing openings.

Solution 4 is interesting in that it plays with the tilts of the roofs to generate south and north lighting sources, and uses projected volumes again to shade the 1st floor west facing windows. However, it reduces too much the north side volumes, and is thus penalized in terms of areas.

Solutions 5 and 6 start to be more hybrid. Solution 5 stays close to the area requirements, but has high energy consumption levels, probably due to the large glazing areas towards east and west, namely due to roof tilts facing those orientations. Solution 6 has a very high area penalty, and probably too big east-facing windows.

Solution 1 seems to be a quite adapted solution to Oporto's mild climate. One can argue that some of the other solutions may be more appealing in aesthetics terms. This brings back the issue of the interactivity between the architect's intentions and the GS. After being presented with this initial solutions, it is then up to the architect to decide what paths of exploration he is willing to take to achieve solutions closer to his intentions. He can decide to change some constraints, run a MicroGA to explore the neighborhood of good solutions like 1, or manually perform same changes and simply do a DOE2 simulation of the modified design to assess the impact of those changes.

Looking at some poor-performance solutions may also be a useful exercise, to assess which design features may have a more negative impact on a solutions' performance. In figure 9.11 and table 9.3 we present some of the worst solutions found in the final population.

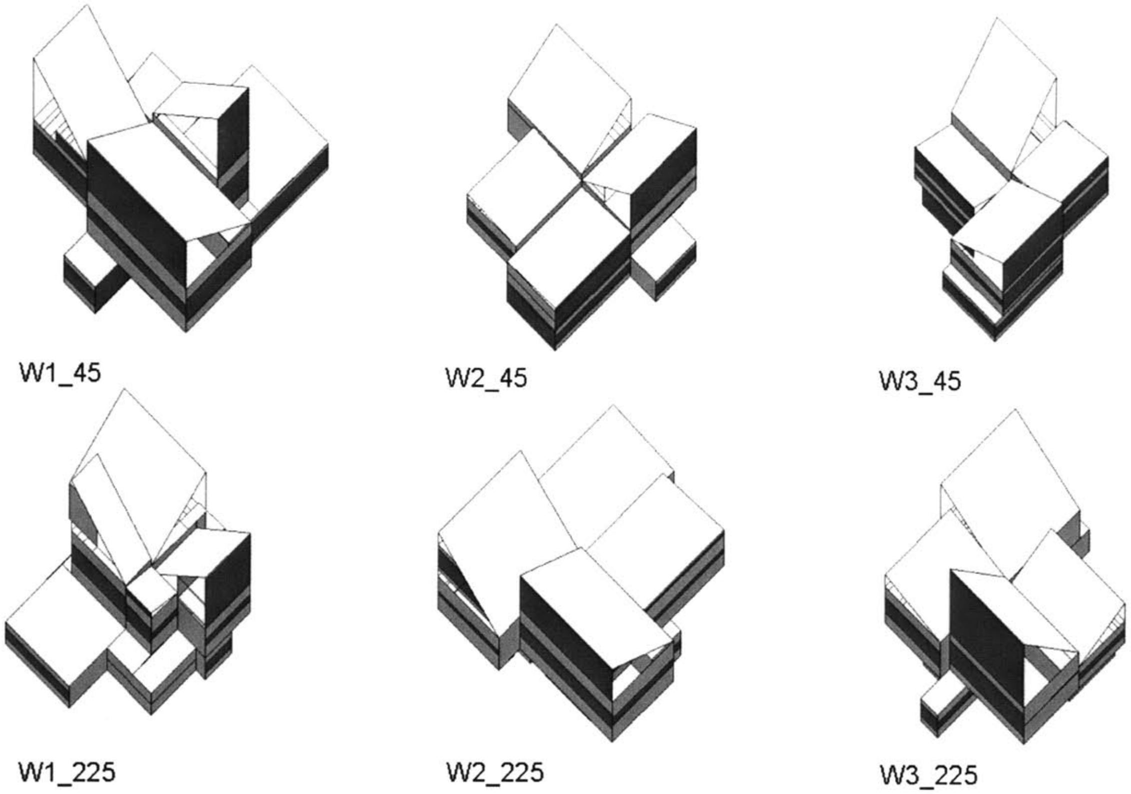


Figure 9.11 Worst solutions present in the last generation for Oporto, using penalties method. The numbers show the solution fitness in descending order [1 being the worst solution]. For each building there is a SW view [45º] and a NE view [225º]. The darker areas in the walls represent windows. Drawings are not to scale.

Solution #	Fitness without penalty [MWh]	Penalty for area violation	Final fitness value
W1	109	44	153
W2	113	29	142
W3	119	22	141

Table 9.3 Initial fitness values in MWh of the best solutions [solution numbers correspond to those in image 9.11], penalty size for area constraints violation, and final fitness value of the solutions

In general, these solutions not only have high penalty functions, but they also have high energy consumption levels. This is due mainly to large west- and east-facing glazing areas from roof monitors generated by steep roofs, and deep overhangs shading south and north openings, thus blocking daylight and useful solar gains. It should be added that the external floors were modeled with no insulation [they are just concrete slabs with some interior finish] so in general the appearance of large projected volumes is not encouraged. This could be changed in subsequent experiments by improving floor insulation levels.

The use of penalty functions introduced another level of complexity in the interpretation of results. The two factors [energy use and floor area violation] are somehow difficult to isolate for result analysis. For this reason, it was decided to do another set of experiments where floor area would also be included in the final evaluation, but using a different method. We used Energy Use Intensity [EUI], that is, energy consumption for unit of floor area. This way, it was easier to evaluate the relative environmental performance of different solutions, independently of the overall floor area.

9.3.2.2 Energy Use Intensity [EUI] for Oporto experiments

Figure 9.12 shows results for the Oporto climate using EUI as the final objective function. The best individual is solution 1, which is somehow different from that found using penalty functions. Since in this case the GS is not asked to assign a given total area to the building, it sizes and distributes spaces in a different way. Very slim and shallow, all glazed elements are used towards the south, in a configuration that resembles some kind of sunspace. Larger spaces, of more bulky proportions, are used towards the north of the building. Those spaces have lower surface-to-volume ratio, and thus have less heat transfer surfaces to lose energy through. However, they may be more difficult to bring daylight into, since they are deeper. For that reason, the GS generates quite large openings in these rooms, mainly facing north [including roof monitors too in that direction], but also towards east and west, as those large spaces require large glazing areas for appropriate daylighting. In general, the proportions of the building tend to orient the longest facades to south and north, and the shorter ones to east and west, as expected. The best performing shapes are in general quite compact, with reduced roof areas in the first floors and few overhangs or projecting elements.

This overall layout is kept in solutions 2 and 3, with some variations, but starts to suffer more significant changes as the fitness of the solutions decreases, with results becoming more difficult to interpret for these intermediate solutions.

Table 9.4 shows the fitness function values for each of the solutions shown in figure 9.10, in terms of EUI [kWh per square foot]. Among the six best solutions, there is already more than 10% variation in energy use intensity.

Solution #	**Final fitness value [EUI]**
1	9.02
2	9.57
3	9.69
4	9.79
5	9.88
6	10.04

Table 9.4 Fitness values in EUI of the best solutions [solution numbers correspond to those in figure 9.12]. EUI is in kWh per square foot.

Finally, we look at the worst performing solutions to try to find patterns of elements leading to poor behavior. Figure 9.13 shows three of the worst individuals, with the worst one on the left [W1]. The main characteristics of these solutions are: high surface-to-volume ratios, caused by a large number of slim, elongated shapes; a large number of exposed external surfaces, like external walls, 1st floor roofs and exposed floors from projected 2nd floor elements [with the aggravation that exposed floors have no insulation]; and large glazing areas facing unfavorable orientations like east and west.

Knowing that these configurations lead to lower performance solutions can either make the designer move away from them, or if the type of architectural language the architect is looking for somehow matches these configurations, they should be counteracted by properly insulated external surfaces, carefully chosen wall and roof solar absorptivity, more sophisticated glazing systems [double clear glass is used in these examples], and appropriate use of shading systems. Nonetheless, these solutions will usually imply higher construction costs, so if low-cost construction is an issue, the architect may try to infer from the GS results the type of building shapes and layouts that would lead to better performance by adaptation to the environment.

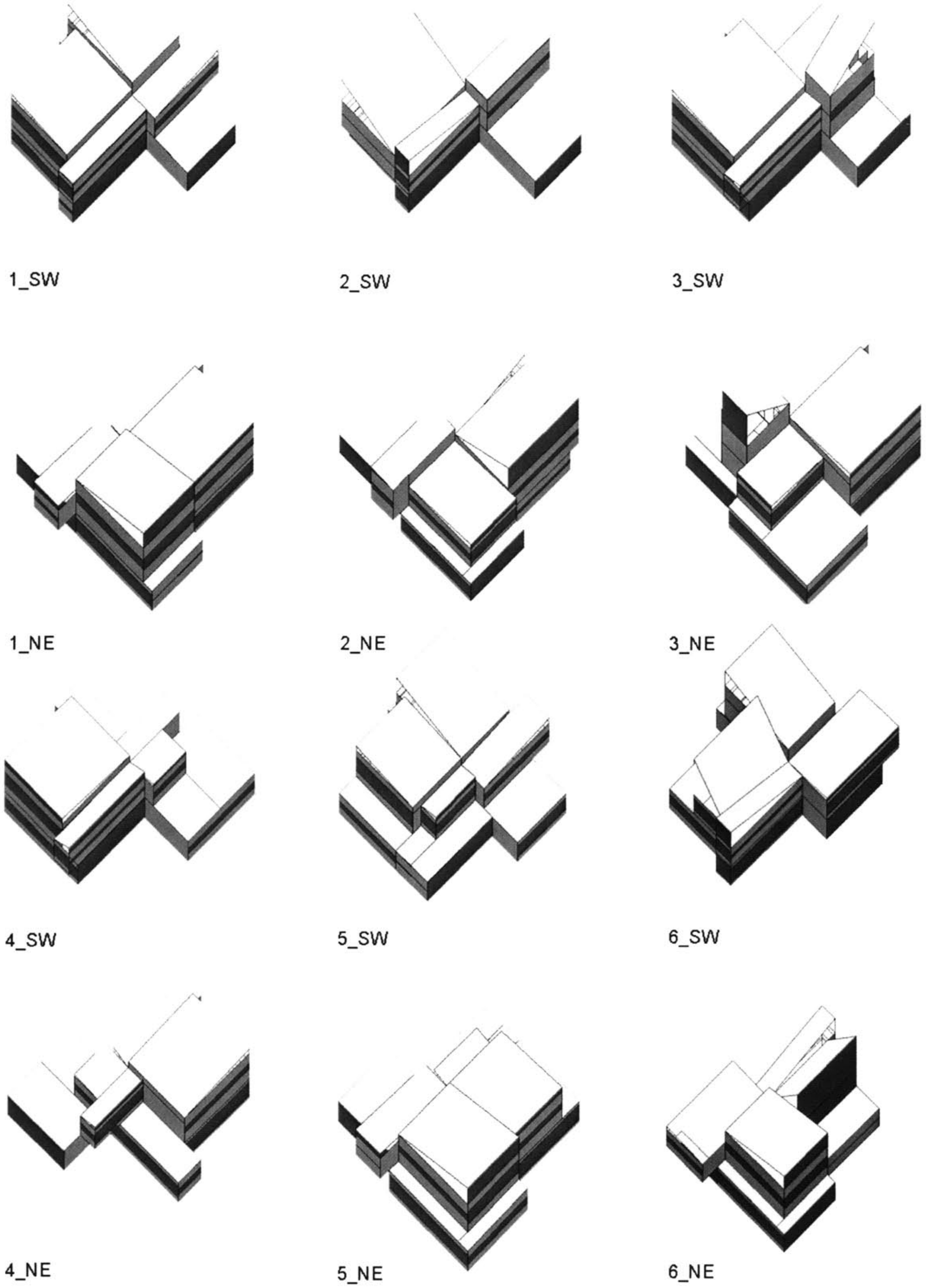


Figure 9.12 Best solutions for Oporto using EUI as fitness function. The numbers show solution fitness in ascending order [1 being the best solution]. For each building there is a SW and a NE view. The darker areas in the walls represent windows. Drawings are not to scale.

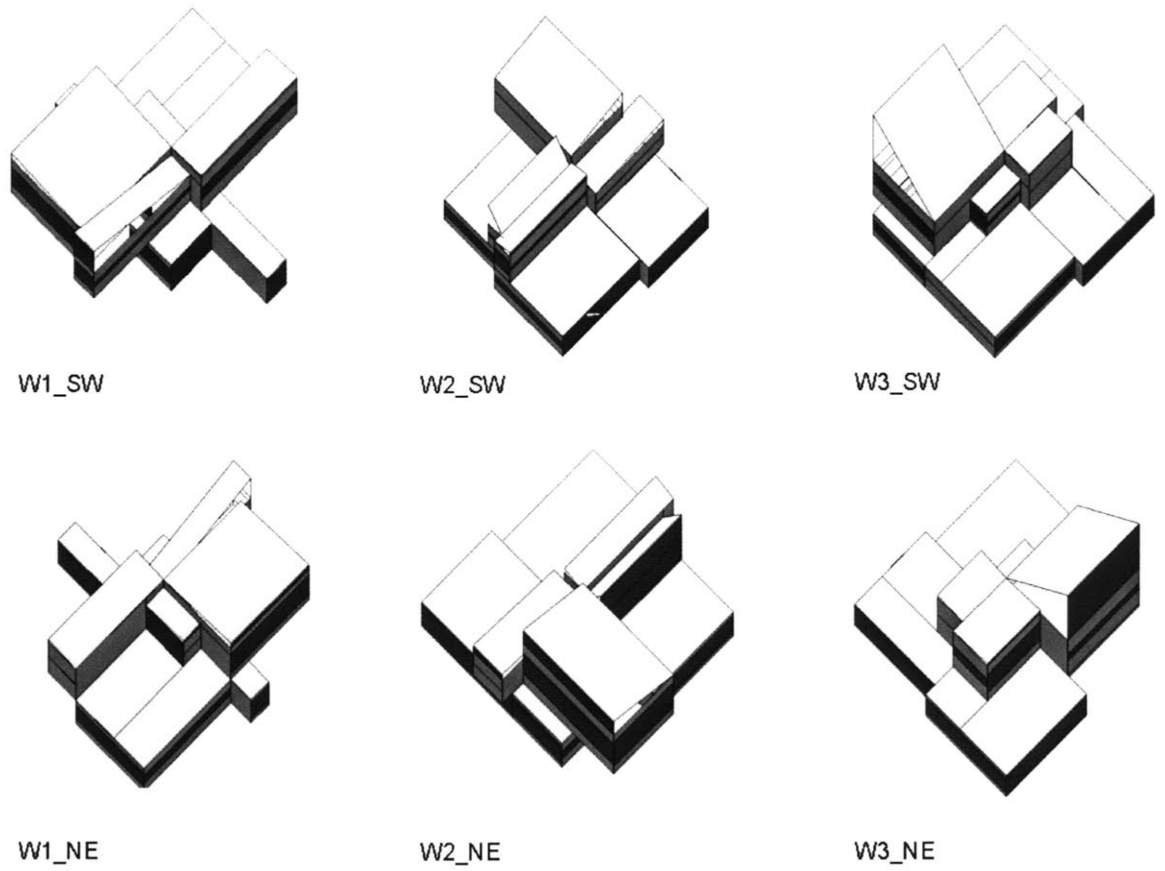


Figure 9.13 Worst solutions present in the last generation for Oporto, using Energy Use Intensity [EUI]. The numbers show the solution fitness in descending order [1 being the worst solution]. For each building there is a SW and a NE view. The darker areas in the walls represent windows. Drawings are not to scale.

Solution #	**Fitness value [EUI]**
W1	13.18
W2	11.85
W3	11.79

Table 9.5 Worst solutions for Oporto using Energy Use Intensity [EUI]. Solution numbers correspond to those in image 9.13.

Finally, to further illustrate the effect of exposed external surfaces on the overall performance of the building, an image is included that shows some results obtained when an error was by mistake introduced into the GS. In some of the preliminary experiments for this chapter on shape generation, it was forgotten to account for the appearance of external exposed floors when rooms from the 2nd floor were larger than the 1st floor ones below then. The results from running the GS with that error are illustrated in figure 9.14. What happened in this case was that the 2nd floor rooms all projected above the 1st floor ones, which remained very small. This was because the GS found it could generate extra area without extra heat transfer surfaces attached to it, so it explored that opportunity. The 1st floor rooms, which had uninsulated ground floors that lost heat to the ground, remained very small, and the 2nd floor rooms grew equally in all directions, creating the squares that can be seen in the images on the left on figure 9.14, which correspond to the best solution. Those squares project deeply in relation to the 1st floor rooms, but have no external floor attached to them, because they would only be generated if they were declared in BDL. No feature is automatically created by BDL from a certain geometry, and an important source of errors is that the files can still run without some existing features being declared. The example presented here serves thus as an illustration of the extent to which results can be jeopardized by the introduction of such errors.

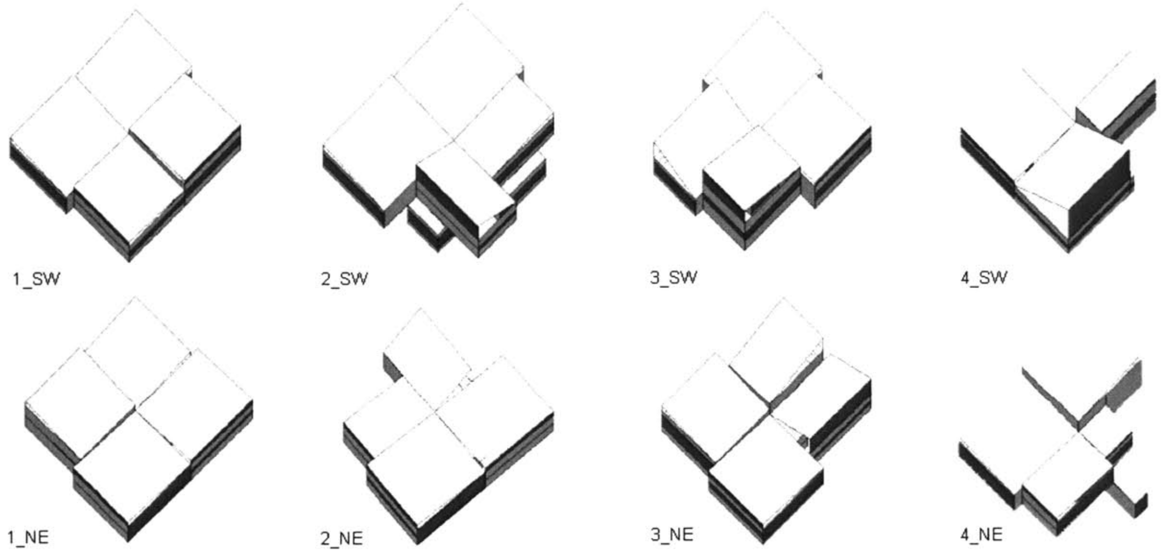


Figure 9.14 Solutions for Oporto using Energy Use Intensity [EUI], but forgetting to account for the external floors that would appear when 2nd floor rooms projected over 1st floor ones. The 'best' solution is on the left. For each building there is a SW and a NE view. The darker areas in the walls represent windows. Drawings are not to scale.

9.3.2.3 Energy Use Intensity [EUI] for Chicago experiments

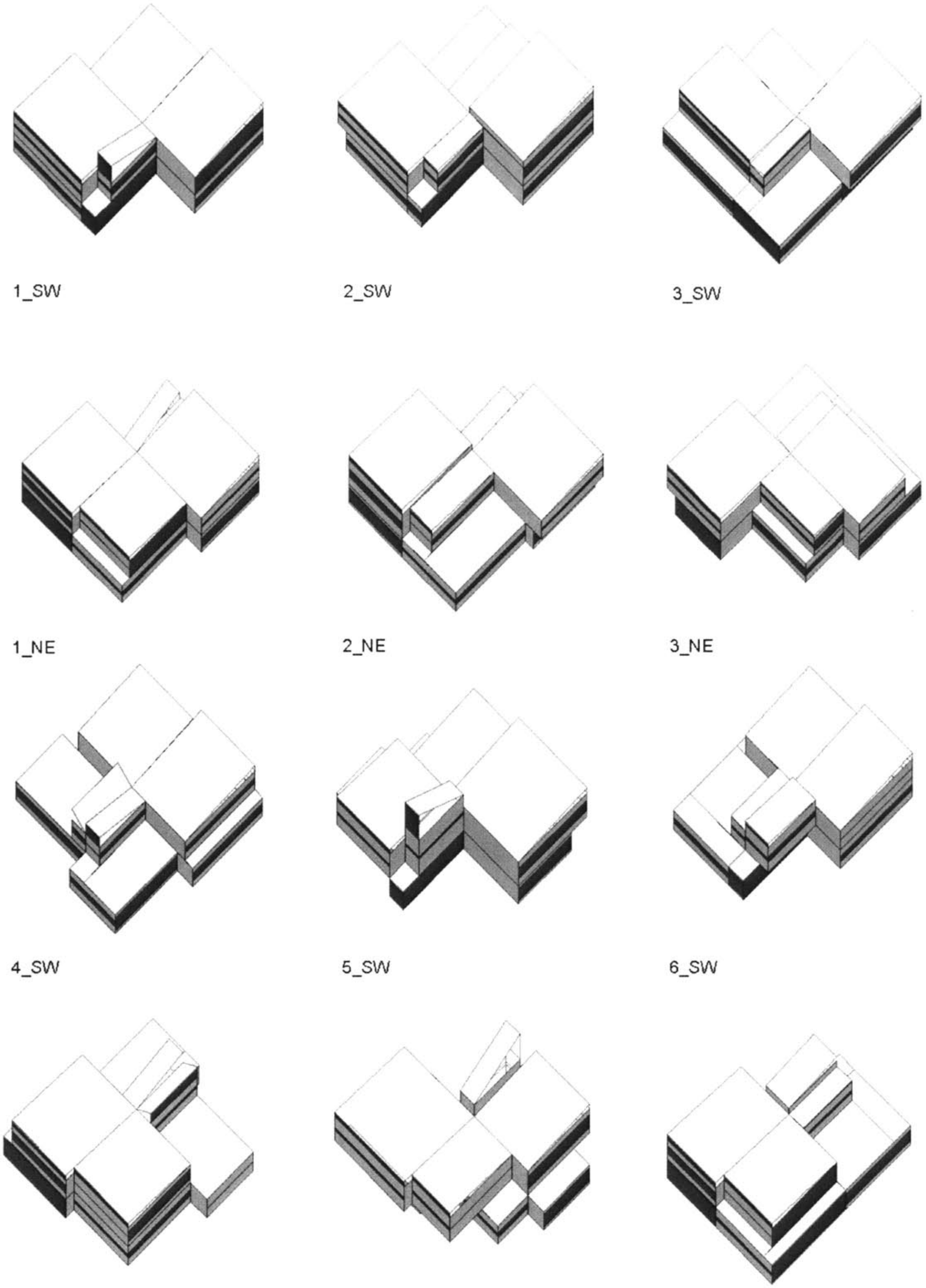
EUI experiments were also done for Chicago, mainly due to its similarity to Beijing's climate. Results are shown in figure 9.15 and in table 9.6.

Solution #	Final fitness value [EUI]
1	26.27
2	27.60
3	28.79
4	28.88
5	30.18
6	30.27

Table 9.6 Fitness values in EUI of the best solutions [solution numbers correspond to those in figure 9.15]. EUI is in kWh per square foot.

The first observation from table 9.6 is that energy use intensity is much higher in Chicago than in Oporto, as expected from the extreme winter conditions. The other observation is that the performance among these six best solutions degrades faster than for the Oporto case. While for Oporto there was a decrease of 10% among the 6 best solutions, here the decrease is more than 15%, showing that the design conditions may have a larger impact in the overall performance of the building.

The best solution for Chicago [1] is a very compact shape, with minimum exposed roofs and floors. It has three bulky, deep spaces in each floor, plus a sunspace-type of room towards the south, very slim and all glazed. Some manual experiments were performed, assessing the consequences of reducing the glazing area of these 'sunspaces' [both towards south and west], and the building's performance actually decreased. In the other 'deep' spaces, glazing areas are also quite high, for daylighting purposes, and because window surface area is not too significant in relation to the whole volume of the space. Again, as the building shape starts being more fragmented and more exposed areas appear, including new exterior walls, the building performance degrades, suggesting that for a cold climate like Chicago, compact shapes perform better. Small wall-to-volume ratios work better. Sunspaces can be beneficial too. It is interesting to note that in the large volumes, due to the depth of the rooms, small window sizes usually correspond to a decrease in performance, due to a lack of daylighting.



1_SW 2_SW 3_SW

1_NE 2_NE 3_NE

4_SW 5_SW 6_SW

4_NE 5_NE 6_NE

Figure 9.15 Best solutions for Chicago using EUI as fitness function. The numbers show solution fitness in ascending order [1 being the best solution]. For each building there is a SW and a NE view. The darker areas in the walls represent windows. Drawings are not to scale.

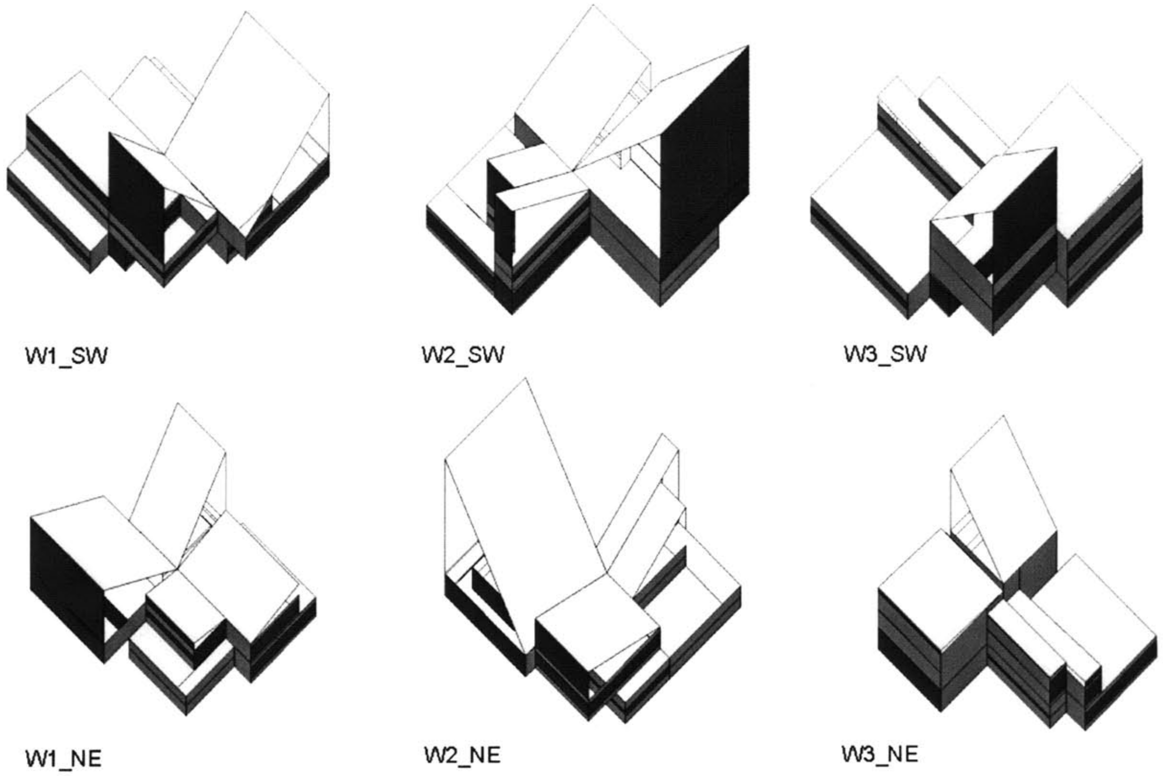


Figure 9.16 Worst solutions present in the last population for Chicago, using Energy Use Intensity [EUI]. The numbers show the solutions in descending order [1 being the worst solution]. For each building there is a SW and a NE view. The darker areas in the walls represent windows. Drawings are not to scale.

In relation to the worst performing solutions [figure 9.16] it seems that the most prominent characteristic is the presence of largely tilted roofs, which are associated with large glazing areas by the rules controlling solutions generation. The worst solution [W1] has both east and west facing roof monitors. The solutions that have south facing ones are also bad, but not as much as the first one. The overall geometry of the buildings is also quite fragmented, the spaces small, the roofs exposed, and there are several overhangs, which represent large sources of heat losses.

Recalling once again what has been mentioned before, this GS is not regarded as an 'optimization' tool, but instead it may be used as a generative mechanism whose goals are not only to reduce energy consumption in buildings, but also to suggest alternative building configurations. However, at this stage of results analysis, it is pertinent to analyze results from the perspective of energy efficiency, because that was the objective function used to guide the search. Other more abstract or formalistic interpretations may potentially find some of the poor-performance results more appealing in some other aspects, but that is not the path we pursued in this chapter.

9.4 Combining Pareto fronts with shape generation

Chicago solutions presented some challenges in terms of interpretation of the results from the Generative System, since the requirements for heating and lighting are quite conflicting. For that reason, the idea emerged of combining the concept of Pareto fronts with that of shape generation. The problem the GS had to solve was to find the best trade-offs between solutions that provided adequate daylighting and minimized the need for heating. One must keep in mind that the objective of finding a good Pareto front is not only to achieve solutions that perform well either in terms of heating or lighting, but to find the solutions that, while having a good performance in terms of heating, also have the best possible performance in terms of lighting given the priority given to heating, and vice-versa. Middling solutions that perform reasonably well in terms of the two conflicting criteria are usually located towards the center of the front.

A Pareto run was thus performed having as the two objective functions both minimizing energy spent in lighting [corresponding to maximizing use of daylighting], and minimizing energy for heating. The progression of the search is shown in figure 9.17. Because the problem was quite complex, due to the large number of variables in the problem [44 independent variables] and the use of two objective functions, the GS was run for 400 generations. The population size was 30 individuals.

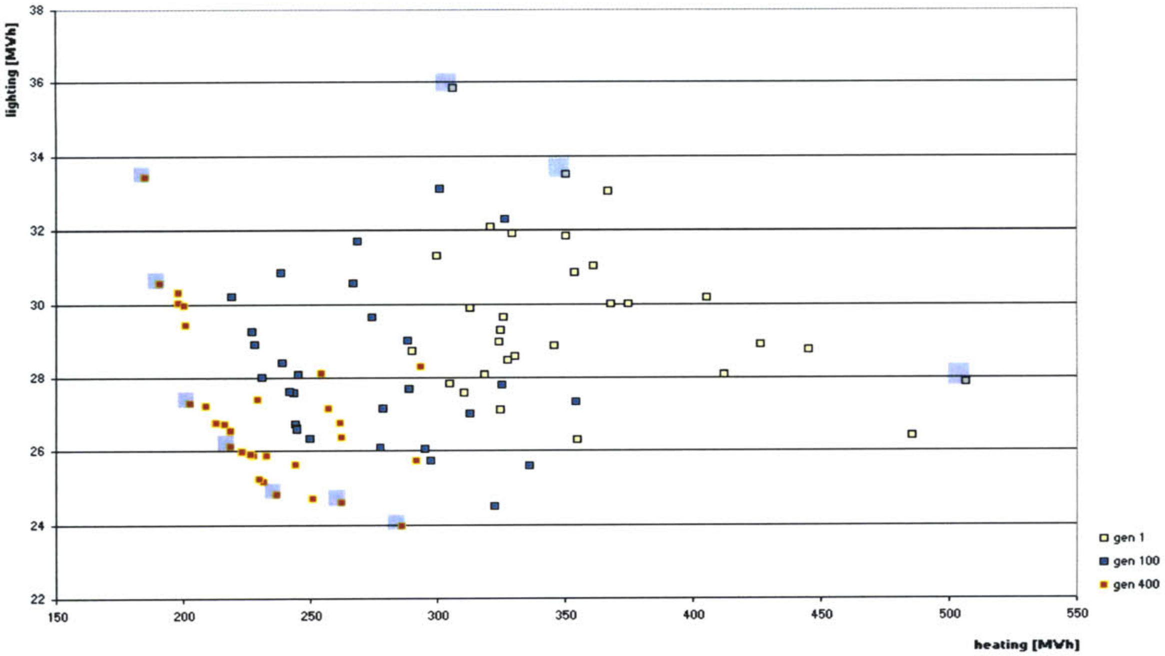


Figure 9.17 Progression of Pareto front search, from generation 1 to 400. The shaded squares indicate the points which have been visualized in the following images, both for good [red squares] and bad [white squares] solutions.

It can be seen that from the initial random population [white squares in the image], the points moved towards the regions of lower objective function values and by generation 100 [in blue] the points were starting to define an initial boundary. By generation 400 [in red], that boundary was much more clearly defined, and been further pushed towards the lower regions of the solution space. It might be the case that running the GS for more generations would further increase the definition of this boundary, since some of the points of the final generation are not yet at the Pareto front, but for the demonstration purposes of this exercise that definition level is close enough. We chose to visualize only some of the most significant points of the front, which are highlighted by the larger blue squares in figure 9.17. The three blue squares towards the top left of the graph highlight the poor-performance solutions also chosen to be visualized, for comparison purposes. Figure 9.18 illustrates the good-performance points, together with the building shapes that they represent.

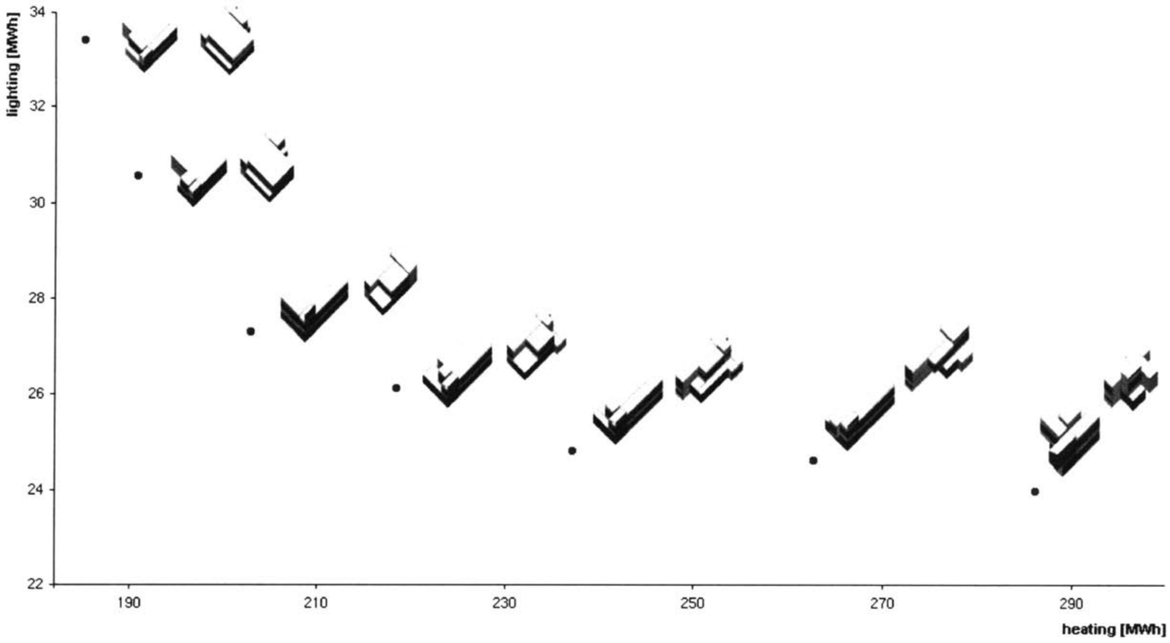
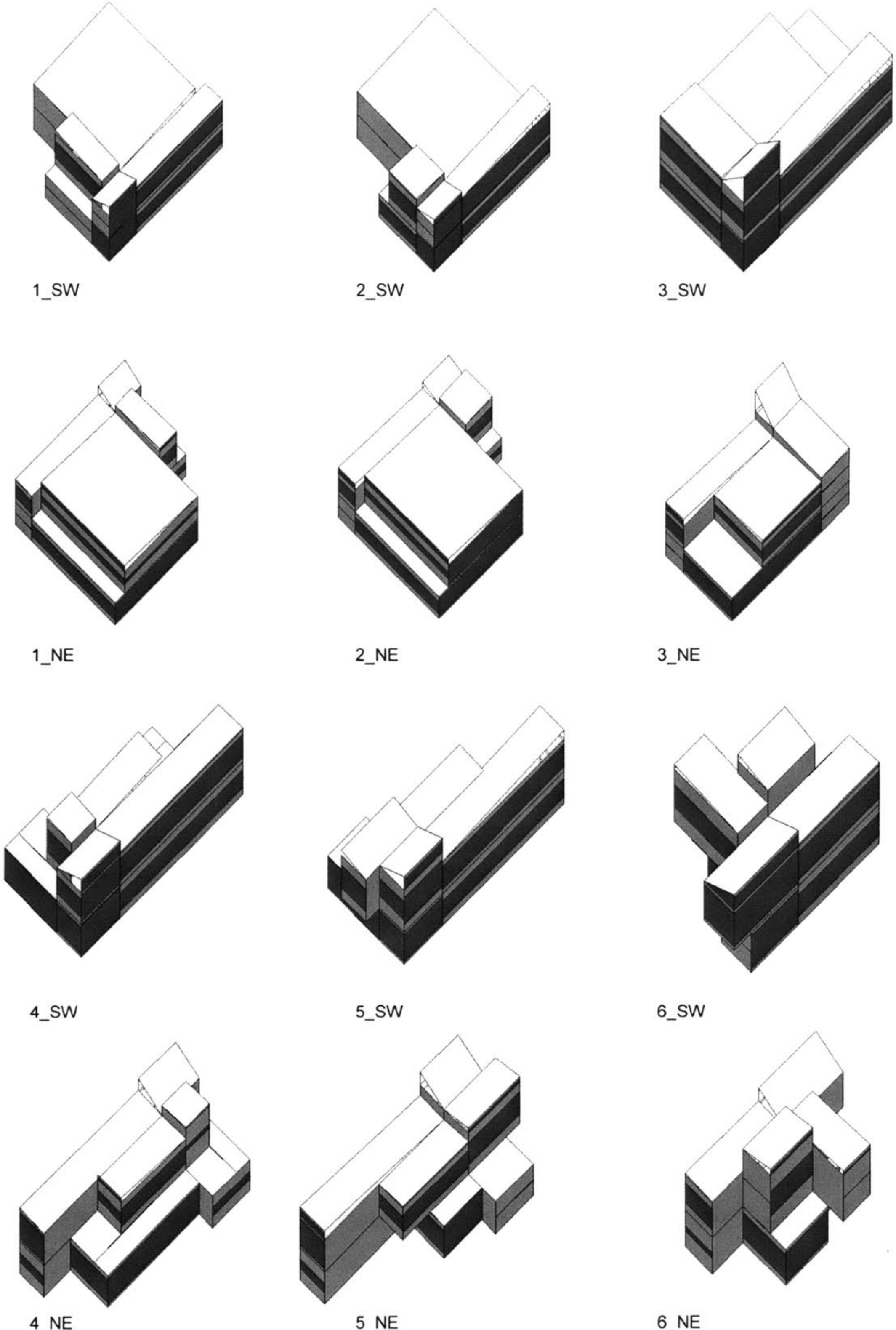


Figure 9.18 Pareto front points visualized in the following images. Larger versions of the images can be seen in figure 9.19

The building shapes can be better visualized at a larger scale in figure 9.19. The best solution in terms of heating basically constitutes a single, compact, large space facing northeast, with thin, sunspace-like, all glazed south and west elements surrounding it. This happens both in the 1st and 2nd floors. The best solution in terms of lighting is formed by small spaces, where daylight can easily penetrate. The south-facing large glazing areas still exist in this solution, in long and thin rooms facing south. The intermediate solutions show basically a morphosis transformation from one solution to the other. Solutions #4 and #5 are also interesting ones, showing very long and thin south-facing elements, and then a number of smaller, north-facing spaces.

Finally, figure 9.20 illustrates some of the poor examples also identified in figure 9.17. Solution W1 performs quite poorly in terms of lighting mainly because many of the spaces have small or nonexistent windows. One might think this could lead to a good performance in terms of heating, but the fragmented distribution of spaces creates many external surfaces that represent high wall-to-area ratios, and many exposed roofs and floors too. Solution W3 is reasonably good in terms of lighting, but quite poor for heating, mainly due to large roof tilts and glazing surfaces, and to the use of overhangs and small volumes.


1_SW 2_SW 3_SW

1_NE 2_NE 3_NE

4_SW 5_SW 6_SW

4_NE 5_NE 6_NE

Figure 9.19 Pareto front points. Solution 1 represents the best building shape in terms of heating. Solution 6 is the best building shape in terms of lighting. The other images represent intermediate solutions [see figure 9.18 for comparison]

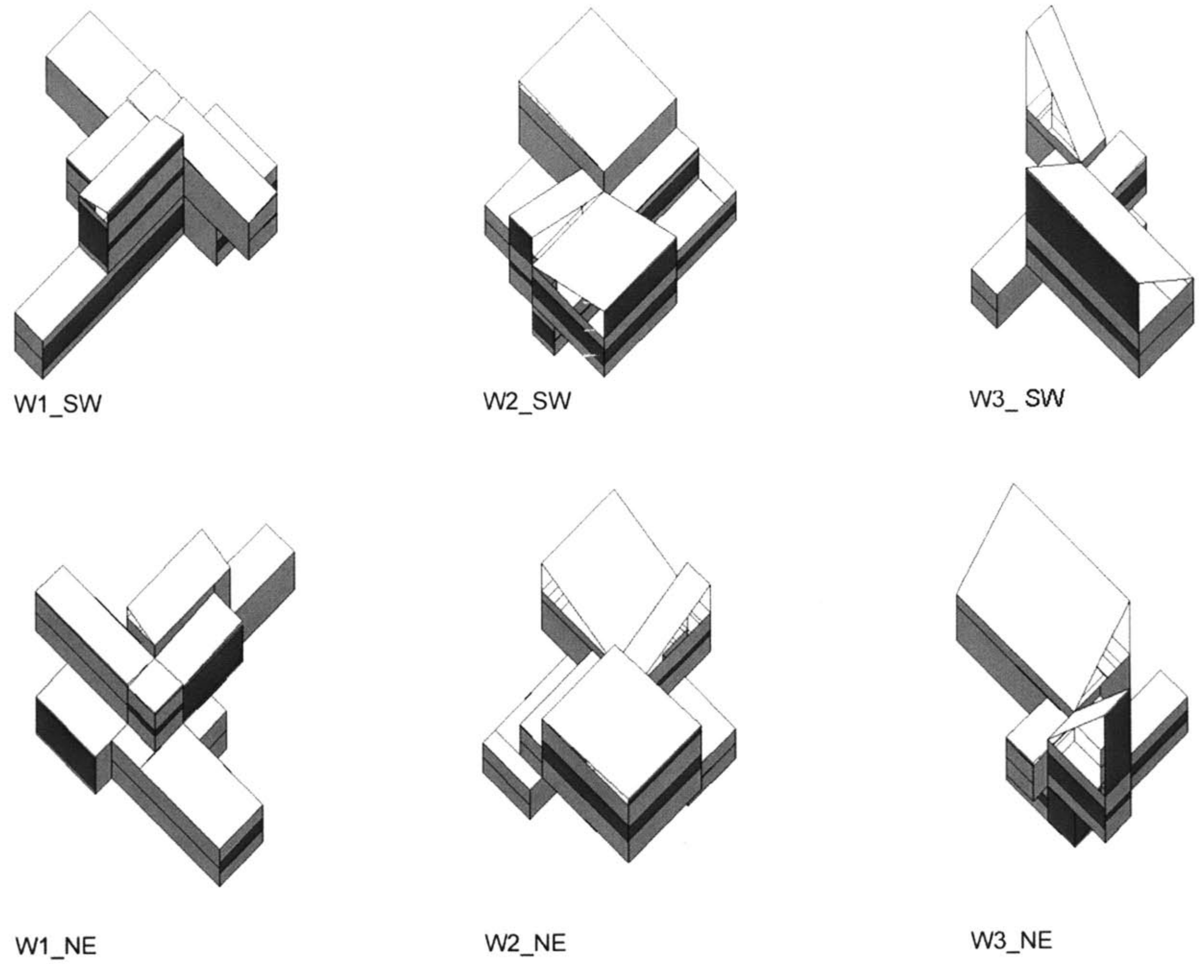


Figure 9.20 Some poor performance solutions from the initial generation. See text for analysis.

The last step of this section involves some 'zooming' into solution 1 from figure 9.19. This building configuration has the best heating performance found. However, it shows some intriguing features, like very large north- and east-facing windows in some of the spaces, what might not be expected in such a cold climate. Our hypothesis was that the configuration of having a very compact space, with a small wall-to-surface area ratio, could make the use of very large openings not too detrimental in terms of heating, while allowing for adequate daylighting of the space. Using DOE2-generated lighting reports, the percentage of artificial lighting savings due to daylighting use was plotted for both room 3 [the large, bulky space in the ground floor] and for room 7 [the similar space located above room 3, in the 2nd floor]. Results are shown in figures 9.21 and 9.22.

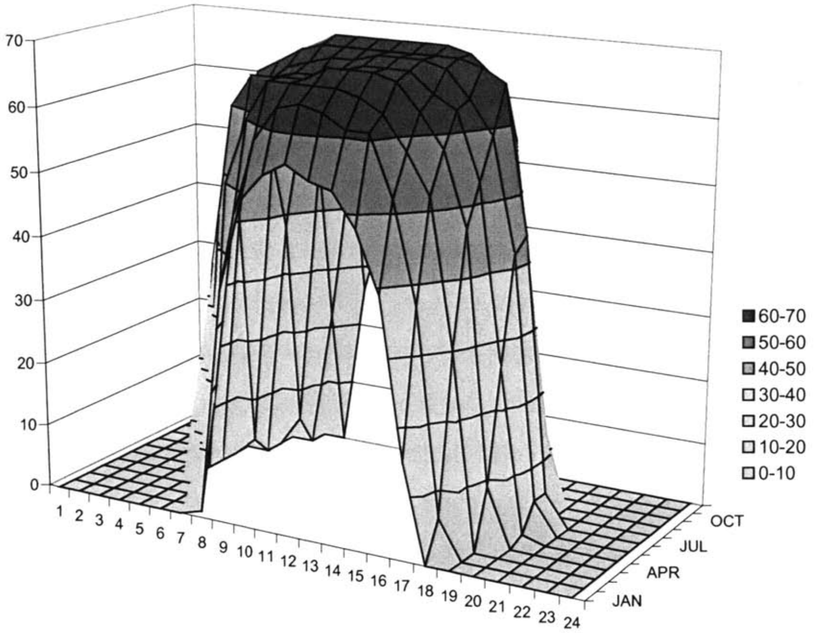


Figure 9.21 Plotting percentage of artificial lighting savings due to daylighting use for Room 3 of solution 1

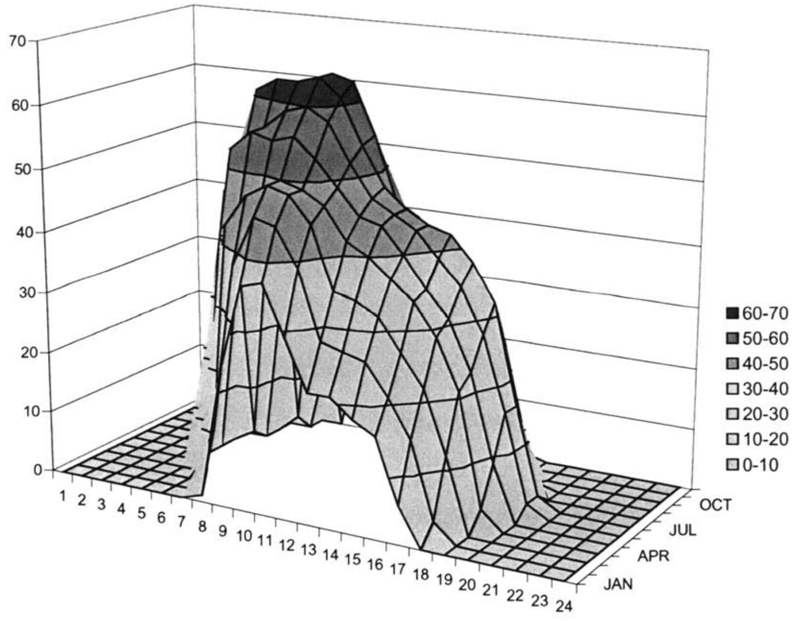


Figure 9.22 Plotting percentage of artificial lighting savings due to daylighting use for Room 7 of solution 1

From figure 9.21 it is possible to conclude that by using large north facing windows, room 3 achieves a 70% artificial lighting reduction for most of the year, except for months like January and February [70% reduction is the maximum allowed, since it is assumed that there will always be some lights on, like task lighting at individual workplaces]. So, even though the space is quite deep, the combination of east- and north-facing windows is very successful in providing good daylighting to the space.

Room 7, on the contrary, uses mostly east-facing openings [both a window and a clerestory type of opening, due to the slightly tilted roof], with a very reduced north-facing window. This causes a substantial change in the profile for artificial lighting savings, as can be observed in figure 9.22. In this room, savings are more reduced, happen mostly during the morning hours, and only reach the 70% level in the peak summer months.

The analysis of these graphs helped thus to shed some light on the reasons behind the use of such large north facing windows in a climate like Chicago, what could not be obvious from the outset. Even though solution 1 is not the best one for lighting, for it to be in the Pareto frontier it should be able to perform reasonably well in terms of lighting too. However, in the same building, other spaces exist that have very poor daylighting use, such as room 4 [in the northwest corner, 1st floor], whose graph is plotted in figure 9.23

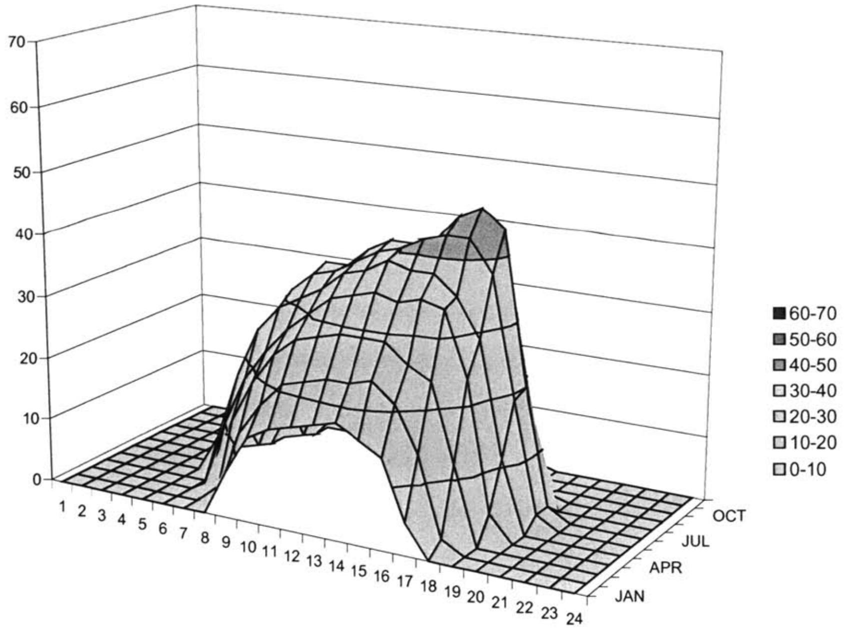


Figure 9.23 Plotting percentage of artificial lighting savings due to daylighting use for Room 4 of solution 1

Room 4 has only a north-facing window, of relatively reduced dimensions, and has no openings to the west. This causes that the daylighting use profile is quite poor, with percentage of artificial lighting savings being quite low for most of the year, and only reaching the 40-50% range in July and August. However, since this is a rather small space, its impact on overall daylighting use on the building is not too significant.

Many other types of reports can be requested to the Generative System, which allow the user to look at the physical phenomena happening in each particular space of the building in extreme detail [at the limit, hourly reports can be generated, showing the variables evolution for each hour of the year]. This includes not only lighting information, of which only one type was shown here, but also information about cooling and heating loads in each space, what elements [walls, roofs, underground floors, windows, etc.] are causing more energy loss / gain during which time of the year, etc.

The intention here is to demonstrate that the architect does not have to rely on a single measure, like annual building energy consumption, to assess the relative quality of a solution in environmental terms. He can ask for solutions in terms of Pareto fronts, considering multiple criteria simultaneously, and can also get very detailed information about what is happening in each of the spaces generated by the GS. In many situations, users may only be interested in high-level, general information. But in cases where detailed information is required, it can be easily made available by the GS. Also, in educational terms, for students to experiment with shape generation with the GS, this type of detailed report can help them to more deeply understand all the complex interaction of variables happening in a particular architectural design.

9.5 Further experiments

The schematic setup defined for the shape-generation experiments described in the previous section is only one of an almost limitless number of possibilities. In this section, some speculations are made about other potentially interesting experiments that could be carried out.

Studying alternative courtyard house layouts could be an easy to implement experiment, which could potentially lead to interesting results. The proportions of the courtyards could be made to vary, as would the proportions of the rooms surrounding it. Different widths, lengths and heights for the several elements could make very different courtyard-type configurations emerge, which could be quite different for distinct climates.

Other experiments could imply a fixed building perimeter, and having the GS change only the disposition of internal walls. This could become some sort of floor plan assignment problem, as exemplified in figure 9.24. Penalty functions could be used to achieve spaces with dimensions close to those intended by the architect. The interior space layout would have consequences for the façade solutions, and the use of dynamic constraints would probably be necessary in this case.

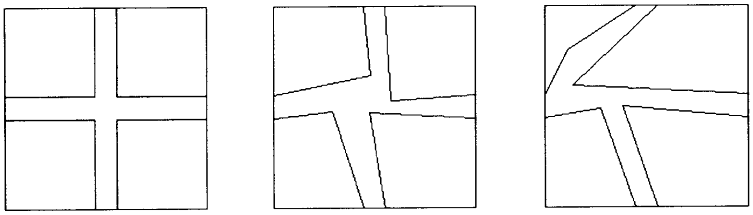


Figure 9.24 From a hypothetical initial schematic layout like the one on the left, new spatial solutions could emerge.

Another possibility would be to use wall width and height as the independent variables, instead of room dimensions. Using room dimensions assumes that the rooms have rectangular or, at the most, quadrangular shapes. If wall dimensions were used, the room plans could have any kind of trapezoidal shape [if only four walls are used], or even become triangular if the width of a wall would become zero. Figure 9.25 illustrates this concept. The varying wall widths would make the resulting wall azimuths to change in relation to those usually present in a rectangular room, and its calculation would have to be carried out, along with each walls insertion point, which would depend on the previous wall end point. Some mechanism would have to be implemented to ensure the walls would eventually define an enclosed space. The image on the left illustrates the case where a west facing wall width is tending to zero, what could be a possible occurrence in a hot climate, where west orientation is highly unfavorable. The resulting configuration would then be close to a triangle. Finally, the dependencies of some variable values on previous ones might make the use of dynamic constraints necessary in this type of problem too.

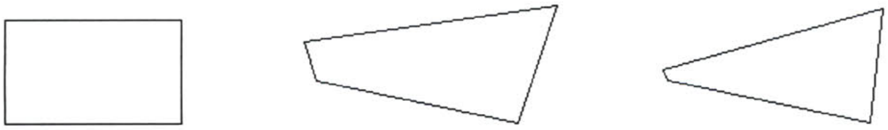


Figure 9.25 Illustrating the possible evolution of a rectangular plan if wall widths are used as variables. In the left image, one of the walls is tending to zero, turning the plan into a triangle.

Another potential extension would be towards a procedure closer to a rule-based system. Figure 9.26 illustrates one possible formalization of this method. The GS could be made to generate a first shape [shape 1 in the images], with any type of proportions. Then it could randomly select a point in one of the façades [in this case, the pink dot in the south façade] as the insertion point for shape 2. Shape 2 could also have any dimensions within the user-defined constraints. Then the lower left corner of shape 2 could become the insertion point for another randomly generated shape 3. And so on. Again, dynamic constraints might be needed for this type of problem.

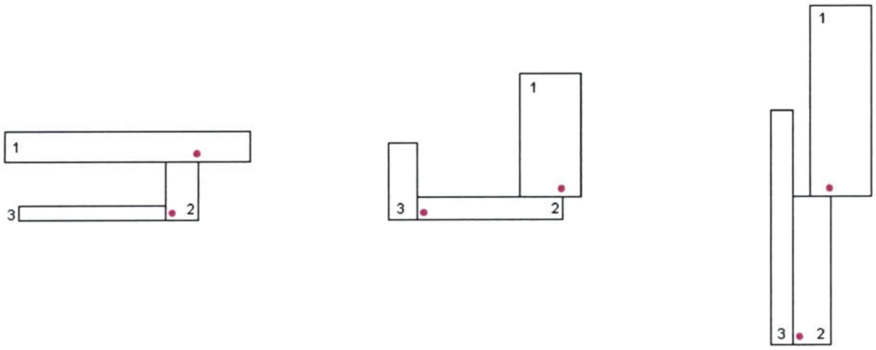


Figure 9.26 Possible implementation of a rule-based type of system.

Investigating the possible relations between the rule-based approach proposed here and Shape Grammar methods [mentioned in the literature review in chapter 2] could potentially be a very interesting path of research. The shape grammar paradigm has been very thoroughly researched, both in theoretical terms (Stiny, 1980), and through practical implementations. Recent work by Duarte (2000, 2001), implementing a shape grammar that encodes Siza's Malagueira housing project rules, is an example of that. Combining this well-established approach to shape generation with the evolutionary Generative System proposed here could open new paths of exploration leading to innovative methods and results.

Urban scale applications would also be possible to pursue, by studying the impact of the position of buildings in relation to each other. Since This GS does not include CFD simulations, those studies would have to relate more to shading patterns and light reflectance from one building's façade to other buildings' interiors, but could still represent an interesting study. In the near future, a combined CFD/energy simulation software may make this type of study even more appealing. Nevertheless, a more ambitious proposal regarding urban scale applications will be presented in the 'further work' section.

Another possible extension would be to consider buildings that are standing in sloped sites, instead of flat ones. Constraining all the Z coordinates of the ground floor insertion points to zero, as we did in the first exercise, implies that the building would be standing on a flat terrain. If the building was standing on a sloped surface, different ground floor rooms would have different Z coordinates for their insertion points, which would have to be determined by the terrain characteristics. In the case the terrain was sloped but the ground floor Z coordinates were all set as similar, the possibility that part of the building was below ground level would have to be considered, what could be important both to determine minimum window sill height constraints, and to allow that part of the wall could be an underground wall. An underground wall could have different construction materials than an above-ground exterior wall, and would also behave differently in terms of heat transfer with the external environment.

As a final note, it is important to mention that in all these potential problem settings, other variables can be included, like wall tilts, construction materials, glazing types, shading elements, and even adaptive materials like electrochromics, which would push the adaptation paradigm to the building operation level too. One must keep in mind that the more variables are included into a problem, the larger the solution space becomes and the more difficult it is for the GS to find good performance solutions. On the other hand, it may introduce a new wealth of ideas and combination of elements that can be very interesting to observe and act as potential triggers for future project developments. It is always possible, at a later stage, to isolate smaller sections of the building to perform a more detailed performance improvement for that particular area, while using the overall, large combination of variables as a source of potentially new ideas during the initial stages of the design process.

These are just some speculations about possible new applications for the Generative System presented here. It is considered that this system could be an excellent teaching tool to use in architecture studios at schools of architecture, combining architectural design problems with low-energy, sustainable approaches. Each student or group of students could design their own problem framework and schematics [like some of those proposed above], run the GS, try to interpret the results in terms of which variables are having the most impact of the buildings performance, make modifications into the design, assess their impact, etc. Apart from overall annual energy consumption, DOE2 can generate a large variety of detailed reports that would allow the students to 'zoom' into specific parts of the design, and try to understand the processes going on in that space, both in terms of lighting, cooling and heating loads. This way they could gain more insight into the interaction of those different aspects, and also how they change across the day and the year. DOE2 also allows that the GS may be run for only part of the year, so it would be possible to see what solutions would be more adapted if only the summer season was considered, for example, or only the winter season. All these possibilities also represent rich opportunities outside the teaching rooms, both in academic research projects and actual architectural practice.

CHAPTER 10 Conclusions and further work

10.1 Review of the conclusions from each chapter

Many conclusions from the work developed for this dissertation were already included at the end of each chapter. Those conclusions will be shortly reviewed here, and some higher level, more general conclusions will be added in the next section.

In chapter 4 the Generative System was validated against an example for which the optimal solution was known. Results proved highly satisfactory and provided enough confidence for the process to be extended to other solution spaces for which the best solutions could not be known in advance.

In these initial experiments, building geometry and space layout were fixed, and only alternative façade solutions were assessed. It was observed that for different climates and different building orientations within that climate, solutions obtained varied substantially, but the introduction of other variables like different glazing types or shading systems also had a large impact on the final solutions. This proves that the dynamics involved in whole-building performance are complex, hard to predict, and the use of a Generative System like the one proposed here can provide useful insight into them.

Solutions also showed that different configurations may correspond to similar standards of environmental performance. This may constitute valuable information to the architect, who is provided with a number of alternative solutions over which he can further overlap other design criteria not included in the search process. Also, by choosing the type of variables that are manipulated by the GS, the architect is also to a large extent exerting control and manipulating the final outcome.

In chapter 5, three heuristic techniques [Genetic Algorithms, Simulated Annealing and Tabu Search] were described and compared, partially using results from the literature, and partially by computational experiments performed for that effect. The computational experiments proved Simulated Annealing to perform almost as well as GAs in terms of finding good performance solutions in reasonable computational time, but other factors influenced our choice for GAs, mainly that SA searches for a single solution, while GAs use populations of individuals. From an architectural design point of view, the latter has the advantage of providing alternative solutions to the architect. That could also be achieved by doing more than one run of an SA-based system, but if alternative solutions are an issue, GAs seemed more appropriate from the outset. Also, GAs represented from the beginning a move towards the adaptation paradigm, as mentioned in the introduction.

The possibility of using Tabu Search for architecture-related problems was kept as a possible future extension, due to some interesting features associated with TS, like extensive use of memory, use of path-relinking strategies to create new solutions from building blocks associated with high-performance individuals [some kind of 'learning' process], and the possibility of performing local search [intensification] in the neighborhood of good solutions to get alternative options. Hybrid implementations, taking advantage of the best characteristics of different search methods, may also be a path for further exploration.

In chapter 7, the possibility of incorporating architectural-design intentions into the GS, and encoding 'language' constraints, was researched by applying the system to an existing building by Álvaro Siza. The Generative System proved to be flexible enough to incorporate constraints that allow the user to manipulate certain architectural design intentions. The close coincidence between GS and Siza's solutions in some situations was of particular interest. On the other hand, the departures from the existing design proposed by the algorithm suggests that this GS may be a useful tool in exploring multiple paths during the design process, which can lead to lower energy designs while not necessarily degrading the initial aesthetic intentions of the architect. Also, the range of solutions the GS offered for different geographical locations showed that the system is able to adapt the architectural design to the climate where it is located, even within the same language constraints.

Another interesting dimension the GS showed in this more complex architectural test environment was is its capability to account for interactions between different elements of the building, and to make the design for each specific element dependent on its integrated role on the architectural whole. These interactions happen not only due to explicit relations caused by constraints, compositional axis, rules for repetition of elements, etc., but also from implicit relations created by design occurrences such as the same space having more than one light source.

The possibility of extrapolating from the façade-level results to other dimensions like building geometry or spatial organization suggested new directions for further work where those aspects might also be manipulated by the generative system. The first attempts of shape manipulation using the GS, while confined to just varying roof tilts, proved that it was possible to use the system to alter building geometry in order to make it more adapted to its environment. This was the seed for much of the work later developed in chapter 9.

Chapter 8 introduced a major change in relation to the previous chapters, in that multiple evaluation criteria were considered simultaneously for assessing the quality of solutions, instead of the single objective function used until then. These multicriteria studies used Pareto front-based methods, with a NSGA being chosen as a result of an extensive literature review. The final algorithm implementation was quite successful in locating well-defined Pareto fronts, what provided a good degree of confidence in the results obtained.

Results proved in general to be quite satisfactory. Understanding the trade-offs between conflicting objectives enhanced our ability to interpret results. The materials studies [for wall construction materials, using initial costs and building energy consumption as objective functions] were quite valuable in suggesting that using highly insulated, lightweight walls and leaving thermal mass for floors and roofs slabs may be a cheaper and more efficient means of construction than building heavy walls, even for hot climates. The experiments also suggested that walls facing different orientations be constructed with different materials, and that considering wall solar absorptivity was also important, what may have an architectural impact not only on the type of cladding materials adopted, but also on the use of different colors in different facades.

Using greenhouse gas emissions [GGE] from building materials as an objective function added yet another dimension to the GS, relating it at a deeper level with the concerns mentioned in the introduction on reducing GGE emissions, which have been an issue of concern for the European Union [including Portugal], and for most of the industrialized world, as shown by the Kyoto protocol targets [despite that fact that there are currently many uncertainties surrounding the future of such protocol]. Following the previous results that lightweight, highly insulating materials seemed to be a good option for wall construction, those extra experiments showed that less conventional insulation materials, like cellulose fill, could be a good alternative to conventional insulation [often based in polymeric materials, thus with higher levels of embodied energy and GGE emissions], both in terms of environmental impact and by achieving good degrees of wall insulation.

Experiments in chapter 8 were done using only two objective functions, but the method could be expanded to include a larger number of criteria, knowing however that result interpretation and decision making will become harder as the number of objective functions increases. One interesting experiment would be to combine building energy consumption, materials first costs, and GGE to create a three-dimensional Pareto front that could show best trade-offs between these three criteria.

Finally, chapter 9 introduced the concept of shape generation, where the GS is no longer dealing with a previously specified building geometry and manipulating mainly the building envelope characteristics, but is instead manipulating the building shape itself. Experiments for different climates showed that the GS was able to generate distinct architectural shapes adapted to a particular climate. Results become somewhat harder to interpret when so many variables are under play, and further 'zooming' into the performance of each solution may be necessary to understand the GS output.

Another possibility that was explored to deal with this extra level of complexity was to combine Pareto methods with shape generation. That way, it was possible to see what shapes were better adapted to different and conflicting performance criteria [like lighting and heating, for example], and from there be able to draw further conclusions. The problem size becomes quite large at this level, so it was found that the GS had to run for a much larger number of generations to generate results that provided enough confidence to the user, by clearly defining an output Pareto front.

Several further extensions to the shape generation experiments were suggested at the end of chapter 9. The main idea behind those extensions is that the architect can exert a high degree of control over the solutions by properly formulating a problem layout that represents his design intentions, by manipulating the constraints placed on each variable, and by using methods such as penalty functions to try to ensure that solutions stay within some target goals for the design.

However, within that universe of constraints, the GS will still be able to generate a remarkable variety of solutions, a fact which the architect can take advantage of, both as an aid in understanding the interplay of forces in shaping a certain form to respond to a certain environment [climate], and also as a platform of creative suggestion of many different possibilities and variations within a specific framework [augmented design], from which the architect can chose those that are more aesthetically appealing to him.

10.2 Overall conclusions

In general, we consider that the initial statements from the introduction, which place this Generative System within a triangle whose vertices are architectural design, environmental performance, and computational methods, were given substantial support from the experiments and conclusions of the different chapters.

We would like to expand our analysis in terms of the 'environmental performance' vertex, since it relates to a complex issue in the realm of architecture, that of 'evaluation.' From the thesis results, we could see that the GS could improve a building's environmental performance, mostly by reducing its energy consumption levels [the work on GGE embedded in construction materials was less developed during the course of this dissertation]. The relevance of these results is twofold. First, reducing energy consumption adds to a building's sustainability, an issue of current concern to the architecture discipline, and also at a more global level to society in general, due to both economic and environmental reasons. Secondly, high consumption levels work as indicators of problems in the architecture of the building. In cases where mechanical systems are not installed to offset those deficiencies [and thus no energy consumption will result from the design deficiencies], users will eventually suffer from discomfort inside the building, bringing the human factor into the problem. Thus, reducing energy consumption in buildings seems to have a relevance in itself.

However, from the literature review we could see that other, similar studies were being done using other evaluation measures. Monk's work aims at improving the acoustical quality of spaces. Shea's work looks at buildings' structural performance. Testa's and O'Reilly's work had more subjective, aesthetics-related evaluation forms. A pattern is thus present in the research in this area, where computational methods are incorporated into the architectural design process to help designers to improve the final architectural artifact. It represents thus a new approach to the architectural design process, where many evaluation measures are valid. The question if there should be an effort to integrate all these measures into a single system, that would look for the 'optimal' performance in terms of all these [and possibly other] evaluation criteria, remains an open one, but in our opinion, that might not be a desirable goal, as the concept of an 'optimal' solution may not be applicable to a complex domain like architecture, where so may factors are always interplaying. It may be better to get some insight into each of these evaluation criteria separately, and then leave to the architect the role of the final decision-maker.

In terms of environmental performance, results suggest this GS may be a useful tool to architects during the design process, with potential for applications at many different levels. The architect can apply the Generative System to a building geometry and layout he has mostly predefined, and use it to study more particular problems such as building envelope design. He can use the GS as a diagnosis tool for his proposed layout, by identifying potentially problematic areas and suggesting ways to approach them.

On the other hand, the GS can act itself as a shape-generation tool, where the architect provides the basic framework for the problem mostly in terms of abstract relationships, and the GS generates a population of solutions that comply with the rules established by the architect [the 'initial rules' stated in the introduction], while simultaneously displaying improved performance in environmental terms, and providing a large number of possible choices for future exploration. Once an overall shape has been found that in general terms complies with the architect's intentions, some variables can become fixed, such as space dimensions, and the GS can run on a more restricted solution space, looking only at façade design or construction materials, for example. This way, different degrees of freedom could be included in the search, and local searches could be implemented to investigate progressively more specific problems. This variability on problem dimension/domain could be a useful method to explore. Another option is to include from the outset all the variables in the search, and although this could provide very interesting and challenging solutions, results may become quite difficult to interpret. Also, because the solution space would become extremely large, the GS would have to be run for a very large number of generations, to ensure that high performance solutions are being found. However, if one excludes the idea of 'optimization,' [trying to find the best possible solution], and instead is looking at improvement, the GS could become a very powerful conceptual tool in providing many different, unexpected options, which might not be 'optimal' but would nevertheless have a good environmental performance and be architecturally interesting at the same time.

One might argue that even used in that way, this GS would always be a limited, restricted design tool, because some of the most important issues in architecture are not quantifiable and thus not prone to be approached using computational processes. We propose two levels of argumentation to deal with that possible critique. One is that the higher-scale issues that relate the architectural artifact with the social, economic and political issues that have a crucial role in shaping architecture, are to be analyzed by the architect, not by the computer. The result from this analysis, however, can be conveyed in a specific representation of the design problem that the architect feeds to the Generative System. This would represent a different way of thinking and producing architecture, in relation to what is the standard practice for most architects. It would require the ability to distill a complex problem, where many higher scale variables exert their forces, into an overall framework on which the Generative System could work. The interactive loop between architect and GS that we have been advocating throughout this dissertation could then begin at this higher, conceptual level. The GS would be acting as an augmented design tool.

The 'representation of a design problem' we have referred to somewhat resembles the concept of a diagram that Alexander emphasizes in his preface to 'Notes on the Synthesis of Form' (1971). Alexander says: "The idea of a diagram, or pattern, is very simple. It is an abstract pattern of physical relationships which resolves a small system of interacting and conflicting forces, and is independent on all other forces, and of all other possible diagrams". It also draws, to some extent, on Mitchell's ideas (1977) of dimensionless representations of shapes, more in terms of their 'building blocks' and relations than in terms of their actual dimensions and final configurations.

The other argument is more radical, and relates to the modes of representation in architecture that were mentioned in the introduction. At the limit, any architectural artifact can be fully represented using a symbolic language, be it numeric [computer graphics also work on numeric representations] or written [like materials specifications, etc]. Since computers use symbolic language systems, there is theoretically no reason why a piece of architecture cannot be fully manipulated by a computer. At the very limit, we could entirely describe a piece of architecture through a long, unique chromosome, just like nature encodes 3D shape in DNA [what represents a fascinating field of research for biologist studying morphogenesis].

Mitchell (1977) distinguishes between three levels of representation that can be used in generative systems in architectural design: analogue, which imply operations that change the state of the system (based on mechanics, like Gaudi's wire-frame suspended structures); iconic, which represents design in a literal way; and finally, symbolic generative systems that represent potential solutions by means of symbols such as words, numbers, and mathematical operations, and where the concepts of design variables and data structures emerge. The necessity of working with representations of reality emerged from the realization that a representation was quicker, easier and cheaper to manipulate than reality itself. The combinatorial possibilities of the symbolic systems used for those representations have long been explored in several fields (literature, music, science, engineering, architecture). An excellent review on the topic is provided by Mitchell (1977). Adding 'evaluation' to those combinatorial mechanisms lead to the application of quantifiable, numerical methods to many of these areas, and in some way to the emergence of the field of optimization, mainly drawing from operations research theories. The development of heuristic procedures to explore the huge combinatorial spaces that quickly grow with the number of variables involved was based on the need to search those spaces in a non-exhaustive, guided way.

Our aim is to explore mechanisms of adaptation by moving away from the 'optimization' paradigm, and instead focus on the idea that many different solutions can be equally well adapted to a given environment, by using different strategies both in terms of active and reactive adaptation. Also, because we believe architecture design in not about finding 'optimal' solutions in terms of a set of criteria, but is also to a great extent an aesthetic endeavor, there is the need to let the architect interfere with his preferences in the adaptation process [the frozen accidents]. Drawing from Whitehead's ideas (1933), 'beauty is left as the only aim which is by its very nature self-justifying'. He continues: 'a truth-relation is not necessarily beautiful. It may not even be neutral. It may be evil.' So we saw from our experiments that sometimes the most 'optimized solutions' (maybe those that expressed a truth-relation to some design objectives) were not necessarily the most interesting in aesthetics terms.

A question may then be asked, why use generative systems that are evolutive, randomized and adaptive, instead of relying on the architect's intuition, knowledge and creativity. The answers may be manifold, but two possible arguments are: First, that the computer works as a way of augmenting the architect's capabilities and creativity, such as the machine did before it, entailing a first 'loss of innocence' (Alexander, 1964), from what many architects of the time retracted. The introduction of computation in the design process corresponds to a second 'loss of innocence,' according to Alexander too, this time an intellectual rather than mechanical one, that many architects may not be willing to accept either. Second, that the idea that an architectural design must be totally intentional, in the sense that all decisions have been made by the architect in response to a program and a site, may be passive of reflection. Programs change [a building is designed to be occupied in some way and it is later adapted for some other use]; sites change, and a building's response to a site may become just a trace after time has passed by. So one may question if the design shape does indeed need to be fully responsive and intentional in relation to its time/place contingencies. Álvaro Siza, who has always taken the 'site' as a main departing point for a design development, has recently put forward the idea of 'a shape looking for a site.'

Despite the fact one cannot incorporate in the symbolic representation provided to the generative system all the network of relations that the architectural artifact establishes with the above-mentioned higher-level forces that surround it [social, economic, political], or other subjective processes [metaphorical, imagetic, conceptual] that lead to a certain design, those can to some large extent be embedded in the framework the architect establishes for the GS. This is obviously a matter of great complexity and sensitivity to the discipline of architecture, that relates to the limits of the application of computational generative systems to architectural design, and would in itself represent an excellent topic for a dissertation, so we shall no further pursue this argument.

10.3 Further work

Many further extensions have already been suggested in the course of this dissertation, related to the contents of the different chapters. Among those, some of the most interesting could be those suggested in the 'shape generation' chapter. In this section we propose other possible directions for future work, extending it to new directions like parametric solid modeling, variable-length chromosomes and complex adaptive systems.

One of the most immediate needs in terms of further work is to develop interfaces for the Generative System, so that it can become more easily usable by other people, be they architects, researchers or students. Those interfaces should be developed at two levels. One is a simple user interface, that allows the input of variable values, like constraints bounds, maximum number of generations, strategies to apply, etc.

Another, much more important and also much harder to develop, is a graphical CAD interface. The relevance of this development is that it would allow the architect to run the Generative System from a 3D model of the building, and have the results from the GS drawn back automatically, in a modified 3D model of the same building. This would be a procedure that any architect that usually constructs 3D CAD models of his projects could easily use, to study specific questions such as façade design or choice of construction materials [and possibly to make extrapolations to other aspects like space layout or even building geometry].

The CAD interface would promote the emergence of the previously advocated interactive loop between architect and GS, because it would provide the results in a graphic format, very easy to read and much more intuitive for the user than a large number of values in an output file. The architect should also be able to draw the constraint areas in the model, graphically visualizing the solution space that he is defining, instead of just inputting numbers as upper and lower bounds for each variable.

The impact of the CAD interface would be much more evident if we consider the case where the architect is using the Generative System for shape generation. In this case, the CAD system does not only represent enhanced user interface, but it would actually add new possibilities and capabilities to the GS. In chapter 9, on shape generation, it was mentioned that because of BDL specificities, it is necessary to predict all possible occurrences of new entities when doing a shape generation experiment, and then drive to zero the values of the variables that did not occur in a given solution generated by the GS. This process is not only tedious, hard to implement, and a potential source of serious errors in the program, but it somewhat limits from the beginning the possible solutions created by the GS.

On the contrary, if a properly designed CAD interface existed, the GS could generate any type of solutions, and the corresponding BDL file would be created by reading from that CAD model all the entities that exist in it, in BDL format. There would be no longer the need to predict all possible occurrences, because the BDL file would not have a fixed format; instead its format would dynamically change according to the CAD model generated by the GS.

Two main issues are crucial in developing such a system. One is to use the appropriate computer graphics paradigm that would allow a BDL format to be created out of a 3D CAD model[4]. Because DOE2 models heat transfer across surfaces, adjacency information must be included in the BDL file. That is, DOE2 needs to know that a certain surface is adjacent to other space in the building, or that other surface is adjacent to the outside environment. This seems relatively simple information, but it is actually information that is not possible to be retrieved from a 3D CAD model made using a common modeling package such as Autocad. This happens because there is no such entity as a 'space' in Autocad. There are surfaces, or volumes, but those are just geometric entities, that do not 'know' that they define a space, that to one side there is another space, that to the other side there is the exterior environment, etc. To have both geometric and topological information, one must resort to other computer graphics paradigms, such as Parametric Solid Modeling [PSM] (Hoffman et al., 2001). In PSM, topological information exists along with geometric information. Furthermore, entities are solids that are parametric, what makes their manipulation extremely easy for the GS, giving rise to a wealth of possible shape combinations that can be unexpected from the outset. Even if one departs from a simplified PSM program, where the user can only chose from a limited palette of forms, the possibilities are almost endless. The possible link to commercially available, powerful PSM programs [such as CATIA©] could make this graphically-based Generative System an extraordinarily powerful tool, with almost limitless possibilities.

The other implementation issue that we envision as a difficulty if such a PSM interface is developed for the GS is that since all possible occurrences in the generated designs are not predicted, and new, unexpected forms and entities can emerge, the length of the chromosomes used to encode those designs has to be variable too. Until now, we have always worked with fixed-length chromosomes, with a specific number of bits assigned to encode each variable. If new variables are allowed to emerge, or existing ones are allowed to disappear, the length of the chromosome must also be adaptive and dynamically change as the GS is running. Although this does not seem too hard to implement, another issue emerges: how does one perform crossover between two chromosomes that have different lengths. Moreover, how does one perform this crossover without destroying meaningful building blocks? In fixed-length chromosomes, we know that an allele at a particular position always relates to a certain variable. In variable length ones, that information is no longer available, because it changes all the time.

To deal with this problem, we intend to use a method proposed by Holland (1992), where a punctuation notation is added to the chromosomes. Using this additional notation, it is possible to know that after some punctuation element, some particular bit of information starts, and another one ends. Using a similar method, we could probably do crossover between variable-length chromosomes without incurring the risk of crossing over alleles coding different variables, like alleles coding a room's dimension crossing with those for a construction material, for example. That would be completely unmeaningfull, destroy building blocks and jeopardize the search process.

Finally, due to the almost endless wealth of possible shapes that could be generated by such a system, the application of Rapid Prototyping technologies could become very helpful in such a system for the designer to quickly construct small 3D models of the shapes generated by the GS, to better visualize and evaluate them.

The other process we envision to cope with the high diversity of solutions that would emerge from such a Generative System would be to make use of Design Galleries (Marks et al., 1997), where the user himself would participate in guiding the search progression. Using design galleries, the architect could be presented, at the end of each 10 generations or so, with all the shapes generated by the GS in the computer screen, in some kind of tiled format, together with their performance values, and would manually choose the ones that he preferred, by assigning them some fitness value that could then be combined with the performance-based fitness value [maybe using Pareto approaches] to continue to guide the search. This would include the user's preference in the search, and could act as an extra mechanism to give the architect further control over the final solutions. In any case, this mechanism would not exclude the careful design of the problem layout and of the variables under study and their constraints, as a way to encode the architect's design intentions into the Generative System. The use of Design Galleries would represent a nice method to incorporate what we previously called 'frozen accidents,' that is, the intervention of the architect and his personal preferences into the search, possibly changing its course from more predictable paths. Using design galleries the architect

[4] Some work is currently being developed at LNBL to create a link between CAD programs and DOE2.

could also 'freeze' some part of a solution, while letting other elements free for further exploration.

Another possible dimension to incorporate in this GS would be studies of modularity to pre-fabricated, pre-assembled units. The GS could, for example, study modular façade systems where the objectives would be not only to achieve fenestration schemes that would allow an appropriate indoor environment, but also to simultaneously minimize the number of different modules [opaque and glazed] present in a given solution, to decrease costs and construction complexity.

Finally, we would like to add a final remark about handling complex geometries. There is currently a significant interest in the architecture discipline in the used of curved, free-form and complex geometries. The application of sophisticated CAD modeling software and the links to CAD/CAM technologies have made possible the construction of many free-forms that were previously extremely hard to build (as the example of Utzon's Sydney Opera House tells us). Until now, the main technical concerns regarding these new possible geometries have been buildability issues. However, we would like to argue that architects should also be showing a concomitant concern about what kind of internal environment is being created inside those structures, both in terms of daylighting, solar control, heat losses, etc. It has been mentioned before that a program like DOE2 has a difficulty in handling complex curved geometries. However, the use of tessellation techniques applied to complex curved surfaces could transform free-form curved surfaces into panels, which would be important to deal with the buildability aspects of those geometries. Simultaneously, those tessellated geometries could then be included in a DOE2 model, which could read them as a series of planar modules. The choice of different materials for those modules could then include criteria like daylighting admission and other environmental control aspects. Again, we would like to push forward the integration of architecture, environmental control and structural engineering, while always grounded on sound architecture theory.

.

10.3.1 Extending the research to Complex Adaptive Systems

In earlier work, Holland [1975] describes adaptation as a process whereby a structure is progressively modified to give better performance in its environment. Three majors components are required for an adaptation process to occur:

1) the environment of the system undergoing adaptation
2) the adaptive plan, whereby the system's structure is modified to effect improvements
3) a measure of performance, i.e., the fitness of the structures in the environment

The generative system initially has incomplete information about which structures are most fit. To reduce this uncertainty, the system must test the performance of different structures in the environment, and start 'learning' which features make a structure successful [building blocks]. The 'adaptiveness' of the plan enters when different environments cause different sequences of structures to be generated and tested, and different final outcomes are created with similar levels of performance.

This view of adaptation reflects what was called 'reactive' adaptation in the introduction, where the individuals are mainly reacting to the exterior environment. Another adaptation paradigm was proposed by Popper (1992), which we called 'active' adaptation in the introduction. Popper's 'active' view of adaptation nicely matches John Holland's latest work on complex adaptive systems (Holland, 1995), which departs from his previous work on classifier systems, and from the 'Echo' model [see section 1.1] (Holland, 1992). The remaining of this section will look at possible means through which a model like 'Echo' could be adapted to the study of spatial relations between buildings, and of the interstitial spaces and urban configurations generated by those buildings.

The classifier system concept enhances an individual's connection to the external environment by introducing a number of detectors in each individual, which check for the occurrence of certain events or characteristics in the environment. If those exist, an input message is sent to the individual. That message is compared to the individuals' message list, and if it matches any message in its list, a certain action is taken. There can also be messages that act in groups, that is, only if two or more messages are matched at the same time is a certain action taken. The outputs from that process are directed to the system's effectors, which cause some action to take place either in the environment or over other individuals. The effectiveness of that action is then reflected on the fitness of that individual.

Classifier systems have other components, such as the 'bucket brigade,' that are not going to be reviewed here. The basic mechanisms previously described are enough to set the framework for understanding the next model, the Echo system.

In the Echo system, a number of individuals exist in an environment where space layout, or geometry, plays an important role. Each individual is tagged to perform certain actions once another individual gets close enough to it. The tags allow for three kinds of actions: combat, trading, and mating. Combat implies a kind of tournament between the two individuals, and can be initiated unilaterally, that is, if the combat tag of one individual matches another one, that individual can start the tournament by itself. Trading represents some kind of cooperation between individuals, and can only be triggered bilaterally, that is, it can only happen if the two individual's trading tags agree. Mating is also bilateral, and is very much like crossover in genetic algorithms.

Echo behaves like a complex adaptive system in that it is a multiagent system that takes only a few primitive activities to generate an amazing array of structures and behaviors [Holland, 1992]. Individuals can organize themselves in clusters of cooperation, create 'war' focus, isolate themselves, etc. This is a basic characteristic of complex adaptive systems, that from simple initial rules complex behaviors can emerge, which themselves generate new rules or patterns of behavior that could not be predicted from the initial ones, due to all the non-linearities of the model.

We propose that a model based in Echo could be applied to the generation of urban patterns, or spatial relations between buildings, which could have a significant interest for experiments in the discipline. Each individual would be an autonomous building, which would interact with the other buildings around it by, for example, depriving them from valuable daylight, or by usefully shading them from direct solar gains in hot climates, or perniciously shading them in cold climates. Such a model would be possible to implement in DOE-2, simulating each building as a three dimensional entity as it happened in the other experiments described in previous chapters. A more sophisticated model, which would also account for air flow patterns between the several buildings, would have to include Computational Fluid Dynamics and would probably have to wait for the development of more efficient CFD algorithms and more powerful computers than those we have at our disposal today.

There are two important differences between the previous experiments and the model that is proposed here. First, there would be a number of different buildings, not a single one as until here; and second, what would be encoded in the chromosome of each building would be not its shape, geometry, dimensions and construction materials, but instead what kind of actions that building could perform in relation to its neighbors. The chromosomes would thus be composed of tags only, one for each façade of the building, which would specify what kind of action that building would trigger towards its neighbor facing that façade. The shape of the buildings, and all its other characteristics, would be fixed [at least during the initial experiments], and could not be varied by the generative system anymore, so that results from actions taken would be easier to interpret. This model would thus exclude reactive adaptation, and would allow for active adaptation only. That means the buildings could improve its environmental performance only by acting over the buildings around them, and not by changing their own characteristics.

Each building would be able to detect the position of its neighbors in space, and could have tags for several possible actions, like repel, attract, couple, make rotate or mate. 'Repel' would mean the building would make the other building move further away from it in some direction, for example to prevent the other building from shading it. 'Attract' would have the opposite effect, making the other building move closer. Repel and attract could both be triggered unilaterally. 'Couple' would turn the other building adjacent to the original one, that is, they would be joined from one façade. This would mean that both buildings would lose their windows on the adjoined façade, losing daylight but at the same time getting rid of a significant heat transfer area with the outer environment, both in terms of heat gain and loss. This could start to generate some urban configurations like rows of buildings, possible streets, etc. 'Couple' would have to be bilaterally agreed. 'Make rotate' would mean the other building would have to rotate by a certain number of degrees in relation to the original one. Finally, 'mate' would mean the two buildings would perform crossover, just like in regular GAs, but what would be crossed over are action tags, not geometrical or materials characteristics. These are just some initial examples for possible actions the buildings generate between they and their neighbors. Other possible interactions could be proposed, causing different types of behavior to appear. Developing and testing this type of actions could be an important part of the research.

Our prediction is that this simple set of actions could make very interesting spatial relations and interstitial spaces between buildings start to emerge, with some buildings starting to form clusters like streets or other urban configurations where several buildings are joined together, others keeping a larger distance between each other if that would allow for more beneficial interactions, some buildings being twisted and rotated in plan in relation to others, other buildings isolating themselves from any neighbors, etc. These 'urban configurations' would also very likely be different for different types of climates. Factors like minimum regulatory distances between buildings that are not adjoined could also be included in the model.

In terms of implementation, because each building would have to correspond to a different DOE2 model, the program would have to run in parallel in several computers, or in a parallel processor computer. Even though the overall system would be quite computationally intensive, using parallel computation could keep simulation times close to that experienced at the moment, or even faster if more powerful computers are used. The same experiments could probably be done using a series of DOE2 runs in a single computer, but with the corresponding time penalties.

In more complex experiments, reactive and active adaptation could coexist in the model. The building could not only act on its neighbors to create itself a more favorable microenvironment [active adaptation], but it could undergo changes in its own configuration [reactive adaptation], as it has happened until now in the experiments described in this dissertation. Only actual implementation and testing of such a system could tell us which type of experiments would be more useful in providing information about urban patterns that are more adapted to a given external environment.

Nevertheless, drawing from experiences in other areas where complex adaptive systems are simulated using computers, simplified, stripped-down models can often be more useful than detailed and complex ones (Gell-Mann, 1995). It is probable that, at least during a first stage, keeping the building layout simple, unchanged, and with the possibility of only a few actions over adjacent buildings, can tell more about the dynamics of complex urban patterns than using a model where the buildings themselves can also vary in their geometry, space layout, façade design and materials, etc.

According to Gell-Mann (1995), if the model is crowded with too many variables, two main problems can occur: First, the model would become too difficult to handle mathematically, and second, once the model is run it would be difficult to understand the results. But it may be a useful extension that, after the model is run with only a few variables and possible actions, and the emergent patterns and behaviors start to be understood, more detail and further actions [including 'reaction'] can be added to further explore emergent spatial relations.

However, it can be predicted from the outset that allowing for building geometry to change during the course of the process could have a significant effect on the effectiveness of certain tags. An 'attract' tag that might work well if the adjacent building was 3 floors high could turn to be a very unsuccessful action if the same building suddenly became much taller, for example. So, the effect of tags would be difficult to isolate from the effect of geometric changes, and it is thus recommendable that at least some initial experiments would be done with fixed geometries and no reactive adaptation. Of course it would be possible to include all types of variables used until now, including building materials [which could affect factors like wall solar absorptivity and light reflectance, apart from insulation values for the walls], window dimensions, etc. However, the solution space could become extremely large, and results could become very difficult to interpret.

Even considering only the stripped-down, simplified version, DOE2 seems to be a particularly suitable simulation program for this application. To study the interaction between different buildings in an urban pattern, both lighting and thermal impacts have to be considered [air flow problems have been mentioned above]. While having reasonable computing times, DOE2 can account for shading between buildings, daylight reflected from a building's façade into another building's interior, and a number of other interactions that can make the results meaningful. The impact of those interactions can then be expressed in terms of the individual performance of each building. Using a simulation program that could not do these kind of calculations would not ensure that any emergent pattern at urban scale would indeed prove beneficial for the performance of each individual building, which is what we would like to achieve.

Implementing such a system would certainly pose new computational challenges, but could give rise to a very interesting research tool, to explore novel spatial relations between buildings that would be more environmentally responsive and potentially previously unthought of. It could also be a powerful teaching tool, by making students trying to understand the dynamics of the selforganizing urban relations that could emerge from such an approach. In more complex problems, other evaluation measures could be included in the model by using Pareto-front methods, to study the possible trade-offs between environmental factors and other criteria pertinent to urban configuration studies.

Other interesting possible outcomes from experiments with such a system would be to shed new light on the interactions between architectural shape and its spatial relations with other buildings. What was described in here is a very general framework that would allow each researcher or student to generate their own research questions and investigate them, by specifying particular building shapes, materials, etc., and also the type of relations | actions the buildings could perform.

Finally, this notion of adaptive behavior in architecture could be taken to the realm of a single building, where the different individuals, competing or collaborating, could be the individual spaces inside the building. Somewhat different rules and actions to those proposed for an urban scale would have to be designed. For example, a space could collect information about the lighting conditions and internal loads of an adjacent space. If the adjacent space was better than itself, it could 'donate' some of its space to its neighbor, by moving an internal wall, for example. If the adjacent space was to some extent similar to itself, it could try to 'copy' some of the adjacent space characteristics. Individual spaces could also have information on the best orientations for views out, and could try to actively search for forms that would maximize their exposure to good views, while considering other factors simultaneously. Many other possibilities exist, as these are only preliminary suggestions for domains of research and do not intend to be detailed methodological proposals.

10.4 Final remark

All the future work hinted at in this chapter is a demonstration that, even though many conclusions could be drawn from the work presented in this dissertation, a wealth of further explorations and new paths of research remains unvisited. However, this may be one of the best final conclusions for one to draw from a dissertation. Challenging and promising inquiries wait for our time and commitment to bring them to light.

Bibliography

Aarts, E., Korst, J., van Laarhooven, P., 1997, Simulated Annealing, in Local Search in Combinatorial Optimization, Aarts, E., Lenstra, J. K. (eds), John Wiley & Sons, pp. 91-120

Aherne, F., Rockett, P., Thacker, N., 1997, Automatic parameter selection for object recognition using a parallel multiobjective genetic algorithm, Lecture Notes in Computer Science, V. 1296, p. 559

Alexander, C., 1964, Notes on the Synthesis of Form, Harvard University Press, Cambridge, Massachusetts

Almeida, M., Centeno, M., 1998, A composite heuristic for the single machine early/tardy job scheduling problem, Computers & Operations Research, V.25 N.7-8, pp. 625-635

Baker, N. and Steemers, K. 2000. Energy and Environment in Architecture: A Technical Design Guide, E & FN Spon, London

Broadbent, G., 1973, Design in Architecture: Architecture and the Human Sciences, John Wiley & Sons

Caldas, L., 1995, Thermal Performance of Electrochromic glazings in a Naturally Ventilated Office for North and South European Locations, unpublished Master thesis, University of London, UK.

Caldas, L., Norford, L, 1999. A genetic algorithm tool for design optimization, Proceedings of ACADIA '99, Salt Lake City, October 28-31, USA, pp. 260-271

Caldas, L., Norford, L., 2000, Energy Design Optimization Using a Genetic Algorithm, *Automation in Construction,* Special Issue 2000

Caldas, L., Rocha, J., 2001, A Generative Design System Applied to Siza's School of Architecture at Oporto, *Proceedings of CAADRIA'01*, Sydney, April 19-21, pp.

Caldas, L., Norford, L. 2001, Architectural Design Constraints in a Generative Design System: Interpreting Energy Levels, *Proceedings of Building Simulation 2001*, Rio de Janeiro, August 13-15

Cagan, J. and Mitchell, W. J., 1993, Optimally directed shape generation by shape annealing, *Environment and Planning B*, 20, 5-12

Camp, C., Pezeshk, S, Cao., G., 1998, Optimized design of two-dimensional structures using a genetic algorithm, *Journal of Structural Engineering*, V. 124 n. 5, pp. 551-559

CCE, 2001, Proposta de Directiva ao Parlamento Europeu e do Conselho relativa ao Rendimento Energético dos Edifícios, Comissão das Comunidades Europeias, Bruxelas, Maio 11, 2001

Chiang, W., Chiang, C., 1998, Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation, European Journal of Operational Research, V.106, pp. 457-488

Chipperfield, A., Fleming, P., 1996, Multiobjective gas turbine engine controller design usinhg genetic algorithms, IEEE Transactions on Industrial Electronics, V. 43, N. 5, Oct. 1996, pp. 583-587

Corana, A., Marchesi, M., Martini, C., Ridella, S., 1987, Minimizing Multimodal Functions of Continuous Variables with the "Simulated Annealing" Algorithm, ACM Transactions on Mathematical Software. vol. 13, n 3, pp. 262-280

Damsky, J. and Gero, J., 1997, An evolutionary approach to generating constraint-based space layout topologies, in Junge, R. (ed.), *CAAD Futures 1997*, pp. 855-874, Kluwer, Dordrecht

Davis, L., (Ed.), 1987, Genetic Algorithms and Simulated Annealing, Pitman Publishing, Morgan Kaufmann Publishers, Inc.

De Jong, K., 1975, An analysis of the behavior of a class of genetic adaptive systems, Doctoral Dissertation, University of Michigan, Dissertation Abstracts International, 36 (10), 5140B

Deb, K. and Goldberg D., 1989, An Investigation of Niche and Species Formation in Genetic Function Optimization. ICGA 1989.

Diário de Notícias, 2001, UE conseguiu baixar electricidade desde 1998
http://www.dn.pt/neg/home/fpna.htm

Dickinson, S. and Bradshaw, A. 1995. Genetic Algorithm Optimization and Scheduling for Building Heating Systems, Genetic Algorithms in Engineering Systems: Innovations and Applications, 12-14 September 1995, pp. 106-111, University of Sheffield: Conference Publication No. 414, Institution of Electrical Engineers

Duarte, J., 1997, Order and diversity within a modular system for housing, Proceedings of the 5th conference on Structural Engineering Applications of Artificial Intelligence, Lahti, Finland

Duarte, J., 2000, A digital framework for augmenting the architect's creativity, Greenwich 2000 – Digital Creativity Symposium, London, January 12-15

Duarte, J., 2001, Customizing Mass Housing: The Grammar of Siza's houses at Malagueira. In, Environment and Planning B.[forthcoming].

Eiben, A., van Kemenade, C., Kok, J., 1995, Orgy in the computer. Multi-parent reproduction in genetic algorithms, Lecture Notes in Artificial Intelligence Advances in Artificial Life Proceedings of the 3rd European Conference on Artificial Life Jun 4-6 1995 n929

Fonseca, C. and Fleming, P., 1993, Genetic Algorithms for Multiobjective Optimization: formulation, discussion and generalization, Proceedings of the 5th ICGA, pp. 416-423 Evolutionary Computation 3(1):1-16. MIT, 1995.

Fonseca, C. and Fleming, P., 1995, An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation 3(1), pp.1-16. MIT Prsee

Fonseca, C. and Fleming, P., 1995, Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction. Genetic Algorithms in Engineering Systems: Innovations and Applications. IEEE Conference Publication No. 414. 12-14 September 1995.

Frazer, J., 1995, An Evolutionary Architecture, Architectural Association Press, London

Galante, M., 1996, Genetic algorithm as an approach to optimize real-world trusses, *International Journal for Numerical Methods in Engineering*, Vol. 39, pp. 361-382

Gallego, R., Monticelli, A., Romero, R., 1998, Comparative studies on non-convex optimization methods for transmission network expansion planning, IEEE Transactions on Power Systems, V.3, N.3, pp. 822-828

GASA, 2000, Emissão e Controlo de Gases com Efeito de Estufa em Portugal, DCEA-FCT, Universidade Nova de Lisboa, Portugal

Gell-Mann, M., 1995, Plectics, in The Third Culture: Beyond the Scientific Revolution, John Brockman [ed.], Simon&Schuster, New York, pp. 316-326

Gero, J. and Radford, A, 1978. A dynamic programming approach to the optimum lighting problem, Engineering Optimization, 3 (2), pp. 71-82

Gero, J., Kazakov, V. 1998, Evolving Design Genes in Space Layout Problems, *Artificial Intelligence in Engineering*, Vol. 12 (3), pp. 163-176

Glover, F., Tabu Search – Part 1, 1989, ORSA Journal on Computing, V. 1, n 3, pp. 190-206

Glover, F., Tabu Search – Part 2, 1990, ORSA Journal on Computing, V. 2, n 1, pp. 4-32

Godart, C., Kruger, M., 1995, Genetic algorithm with parallel steady-state reproduction, Lecture Notes in Computer Science Artificial Evolution Proceedings of the 1995 European Conference Sep 4-6 1995 v1063

Goffe, W., Ferrier, G., Rogers, J., 1994, Global optimization of statistical functions with simulated annealing, Journal of Econometrics, vol. 60, no. 1/2, Jan./Feb., pp. 65-99

Goldberg, D., and Richardson, J., 1987, "Genetic Algorithms with Sharing for Multimodal Function Optimization."

Goldberg, D. 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company

Grefenstette, J. 1986. Optimization of control parameters for genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics, SMC-16 (1), pp. 122-128

Hertz, A., Taillard, E., de Werra, D., 1997, Tabu Search, in Local Search in Combinatorial Optimization, Aarts, E., Lenstra, J. K. (eds), John Wiley & Sons, pp. 121-136

Hoffman, C., Joan-Arinyo, R., Parametric Modeling, Report...

Holland, J., 1975, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press

Holland, J., 1992, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, The MIT Press

Holland, J., 1995, Hidden Order: How Adaptation Builds Complexity, Addison-Wesley Publishing Company

Horn, J., Nafpliotis, N., and Goldberg, D., 1994, Niched Pareto Genetic Algorithm for Multiobjective Optimization. Proceedings of the 1st IEEE Conference on Evolutionary Computation, Part 1, Jun 27-29, Orlando, FL, pp. 82-87

Huang, W., Lam, H. 1997. Using genetic algorithms to optimize controller parameters for HVAC systems, Energy and Buildings, 26, 277-282

Incropera, F., 1996, Fundamentals of Heat and Mass Transfer, Wiley, New York.

Ishida, R., Sugiyama, Y. (1995), Proposal of constructive algorithm and discrete shape design of the strongest column, *AIAA Journal*, Vol. 33 (3), pp. 401-406

Jo, J., Gero, J. (1998), Space layout planning using an evolutionary approach, *Artificial Intelligence in Engineering*, Vol. 12 (3), pp. 149-162

Jozefowska, J., Mika, M., Rozycki, R., Waligora, G., Weglarz, J., 1998, Local search metaheuristics for discrete-continuous scheduling problems, European Journal of Operational Research, V.107, pp. 354-370

Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983, Optimization by Simulated Annealing, Science, 220, pp. 671-680

Koumousis, V., Panos, G., 1994, Genetic algorithms in discrete optimization of steel roof trusses, Journal of Computing in Civil Engineering, v. 8, n 3, pp. 309-325

Krishnakumar, K., 1989. Micro-genetic algorithms for stationary and non-stationary function optimization, in Rodriguez, G. (ed.), Intelligent Control and Adaptive Systems, 7-8 November (pp. 289-296), Philadelphia, Pennsylvania: SPIE – The International Society for Optical Engineering

Law, Sinyan, 1997, Daylighting design using a non-linear optimization technique, Master in Architecture Thesis, Massachusetts Institute of Technology, Cambridge.

LNBL (Lawrence Berkeley National Laboratory), 2001, http://gundog.lbl.gov

Mantawy, A., Abdel-Magid, Y., Selim, S., 1997, A new simulated annealing-based tabu search algorithm for unit commitment, 1997, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Oct. 12-15 1997, Orlando Fl., Part 3 (of 5), pp. 2432-2437

Marcelin, J., Trompette, P., Dornberger, R, 1995, Optimization of composite beam structures using a genetic algorithm, *Structural Optimization*, Vol.9 (No.3-4), pp. 236-244

Marks, J., Andalman, B., Beardsley, P., Freeman, W., Gibson, S., Hodgins, J., Kang., T., 1997, Design Galleries: A general approach to setting parameters for computer graphics and animation, Computer Graphics Proceedings, Annual Conference Series, pp. 389-398

Martin, L., March, L., 1966, Land use and built forms, Cambridge Research

Martins, R., 2001, Factura Energética Portuguesa Aumenta 108 por Cento, Jornal Público http://jornal.publico.pt/2001/05/04/Ambiente/G01.html

Mitchell, W., 1974, Techniques of automated design in architecture: a survey and evaluation, University of California, School of Architecture report, Los Angeles, California.

Mitchell, W., Steadman, J., Ligget, R., 1976, Synthesis and optimization of small rectangular floor plans, in Environment and Planning B, Vol. 3, pp. 37-70

Mitchell, W., 1977, Computer-Aided Architectural Design, Petrocelli / Charter, New York.

Moeck, M., Lee, E., Rubin, M., Sullivan, R., Selkowitz, S., 1998, Visual quality assessment of electrochromic and conventional glazings, Solar Energy Materials & Solar Cells. V. 54 n 1/4 Aug, pp. 157-164

Monks, M., Oh, B., Dorsey, J., 1998, Audioptimization: Goal based acoustic design, MIT Technical Report MIT-LCS-TM-588

Muhlenbein, H., 1997, Genetic Algorithms, in Local Search in Combinatorial Optimization, Aarts, E., Lenstra, J. K. (eds), 1997, John Wiley & Sons, pp. 137-171

MURE, 1998, Mesures d'Utilisation Rationelle de l'Energie, European Commission

Norris, D., 1997, Daylight and productivity: Is there a causal link? In Glass Processing Dayus Conference, Tampere, Finland, Sept. 13-15

O'Reilly, U-M, Ramachandran, G., 1998, A preliminary investigation of evolution as a form design strategy, In Adami, C., Belew, R., Kitano, H and Taylor, C., (Eds), Proceeedings of the 6th conference on Artificial Life, MIT press

O'Reilly, U-M., Ross, I., Testa, P., 2000. Emergent Design: Artificial Life for Architecture Design, *Artificial Life* 7

Osman, I., Kelly, J. (Eds.), Meta-Heuristics: Theory and Applications, Kluwer Academic Publishers, 1996

Ospelt, C., 2001, Private Communication

Otten, R., van Ginneken, L., 1989, The Annealing Algorithm, Kluwer Academic Publishers

Pennebaker, W., Mitchell, J., 1993, JPEG: Still image data compression standard, Van Nostrand Reinhold, New York

Petrie, C., Webster, T., Cutkosky, M., 1995, Using Pareto optimality to coordinate distributed agents, Artificial Intelligence for Engineering Design, Analysis & Manufacturing: Aiedam. V. 9 n. 4 Sep. pp. 269-281.

Pham, D. & Onder, H., 1992*,* A knowledge-based system for optimizing workplace layouts using a genetic algorithm, *Ergonomics*, Vol. 35, No. 12, pp. 1479-1487

Pham, D., Jin, G., 1995, Genetic algorithm using gradient-like reproduction operator, Electronics Letters v31 n18 pp.1558-1559

Popper, K., 1992, In Search for a Better World: Lectures and Essays from Thirty Years, Routledge, London

Radford, A., 1978, Some Room / Environment Optimization Models Using Dynamic Programming, Computer Report CR30, Department of Architectural Science, University of Sydney

Radford, A., Gero, J., 1978, On the design of windows, *Environment and Planning B*, 6 (1), 41-45

Radford, A., Gero, J., 1985, Multicriteria optimization in architectural design, in Gero, J. (ed.), Design Optimization, 1985, Academic press, Inc.

Reedy, G., Cagan, J., 1995a, Optimally directed truss topology generation using shape annealing, ASME Journal of Mechanical Design, V. 117, N. 1, pp. 206-209

Reedy, G., Cagan, J., 1995b, An improved shape annealing algorithm for truss topology generation, ASME Journal of Mechanical Design, V. 117, N. 2A, pp. 315-321

Reilly, S., Arasteh, D., Selkowitz, S., 1991, Thermal and optical analysis of switchable window glazings. Solar Energy Materials. V. 22 n 1 Mar, pp. 1-14

Selkowitz, S., Kim, J., Navvab, M. & Winkelman, F., 1982, The DOE-2 and Superlite Daylighting Programs, In Passive 82, The National Passive Solar Conference, August 29-September 3, Knoxville TN

Shea, K., Cagan, J., Fenves, S., 1997, A shape annealing approach to optimal truss design with dynamic grouping of members, ASME Journal of Mechanical Design, V.199,N.3, pp.338-394

Shea, K., Cagan, J., 1997, Innovative dome design: Applying geodesic patterns with shape annealing. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 11, pp.379-394

Shea, K., Cagan J., 1998, Generating Structural Essays from Languages of Discrete Structures. In *Artificial Intelligence in Design '98*, Gero, J. S. and Sudweeks, F. (Eds). Kluwer Academic Publishers. London, pp. 365-404

Simon, H. ,1969, *The Sciences of the Artificial.* Cambridge, Mass.: MIT Press

SRG, Simulation Research Group, 1993. DOE-2 Supplement – Version 2.1E, Lawrence Berkeley National Laboratory, LBL-34946

Sohn, M., Small, M., 1999, Parameter estimation of unknown air exchange rates and effective mixing volumes from tracer gas measurements for complex multi-zone indoor air models, Building and Environment, v. 34, n.3, pp.293-303

Srinivas, N. and Kalyanmoy D., 1995, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation 2(3): 221-248. MIT Press

Stiny, G., 1980, Introduction to Shape and Shape Grammars, Environment and Planning B 7, pp. 343-352.

Sullivan, R., Arasteh, D., Papamichael, K., Kim, J., Johnson, R., Selkowitz, S., McCluney, R. 1988, An Indices Approach for Evaluating the Performance of Fenestration Systems in Nonresidential Buildings, *ASHRAE Transactions*, Vol. 94 (Part 2)

Sullivan, R., Lee, E. and Selkowitz, S, 1992. A Method for Optimizing Solar Control and Daylighting Performance in Commercial Office Buildings, ASHRAE/DOE/BTECC Conference on the Thermal Performance of the Exterior Envelopes of Buildings V, December 7-10, Clearwater Beach, FL

Sullivan, R., Rubin, M., Selkowitz, S., 1997, Energy performance analysis of prototype electrochromic windows, ASHRAE Transactions. V. 103 pt 2, pp. 149-156

Syswerda, G., 1989, Uniform crossover in genetic algorithms, Proceedings of the Third International Conference on Genetic Algorithms, June 4-7, George Mason University, pp.2-9

Szykman, S., Cagan, J., 1993, Automated design of optimally directed three dimensional component layouts, Advances in Design Automation, ASME, Albuquerque, NM, September 19-22, Vol. 1, pp. 527-537

Tamaki, H., Mori, M., Araki, M., Generation of a set of Pareto-optimal solutions by genetic algorithms, Transactions of the Society of Instrumental and Control Engineers, V. 31, N.8, pp.1185-1192

Tamaki, H., Kita, H., Kobayashi, S., 1996, Multi-Objective Optimization by Genetic Algorithms: A Review, IEEE 1996

Testa, P., 1994, Álvaro Siza 1958-1994, *El Croquis* .Milan: Fiorucci International.

Testa, P. ,1999, Evolutionary Space in the Architecture of Álvaro Siza, A+U. Architecture and Urbanism. 355. 00:04. pp. 8-13

Testa, P., O'Reilly, U., Weiser, D., Ross, I., 2001, Emergent Design Studio: Interactive model of design and computation. in Environment and Planning B: Planning and Design [forthcoming]

Van Laarhoven, P., Aarts, E., 1987, Simulated Annealing: Theory and Applications, D. Reidel Publishing Company

Wang, B.& Chen, J., 1996, Applications of genetic algorithm for the support location optimization of beams, *Computers and Structures*, Vol. 58, No. 4, pp. 797-800

Weile, D., Michielssen, E., Goldberg, D., 1996, Genetic algorithm design of Pareto optimal broadband microwave absorbers, IEEE Transactions on Electromagnetic Compatibility, V. 38, N. 3, pp. 518-525

Wetter, M., 2000, Design Optimization with GenOpt, Building Energy Simulation User News, Vol. 21, September/October 2000

Whitehead, A., 1933, Adventures of Ideas, Macmillan Company

Winkelman, F., 1983. Daylighting Calculation in DOE-2, Lawrence Berkeley National Laboratory Report, LBL-11353, May 1983

Wright, J., 1996, HVAC optimization studies: Sizing by genetic algorithm, Building Services Engineering Research and Technology, Vol.17, No.1, pp.7-14

Yoshiaki, I., Toshimitsu, T., Atsuya, F., 1996, Elevator installation design by genetic algorithms for large buildings, Transactions of the Japan Society of Mechanical Engineers, Part C, V.62, n601, pp. 3413-3419

Illustration credits

Figures 7.6, 7.2, 7.3, 7.20, 7.21, 7.22 (right) and 7.23 (right) are taken from El Croquis n. 68-69 (1994)

Figure 7. 7 is taken from blueprints provided by Álvaro Siza

Figure 8.2 is adapted from Petrie et al. (1995)

Figures 8.3 and 8.4 are adapted from Schaffer (1985)

Figures 8.8, 8.9, 8.10, 8.11 are adapted from Tamaki et al. (1996)

Figures 8.12, 8.13, 8.14 and 8.15 are taken from Deb et al. (1989).

Figure 8.16 is adapted from Srinivas et al. (1995)

I would like to acknowledge the help of João Rocha in creating some of the digital images presented in chapter 7 [as well as his contribution to the research presented in that chapter], and the collaboration of Jamie Buller, Joe Distefano, Michael Frauens, Anupam Sahai and Robb Wirthlin in chapter 8.

Appendix 1 Illustrations in black and white

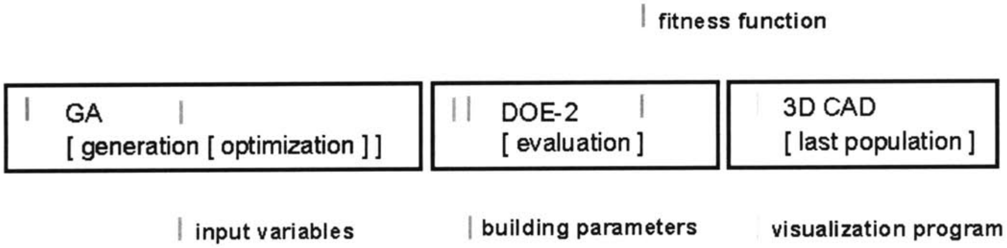


Figure 3.1 Flow of information across the Generative System components

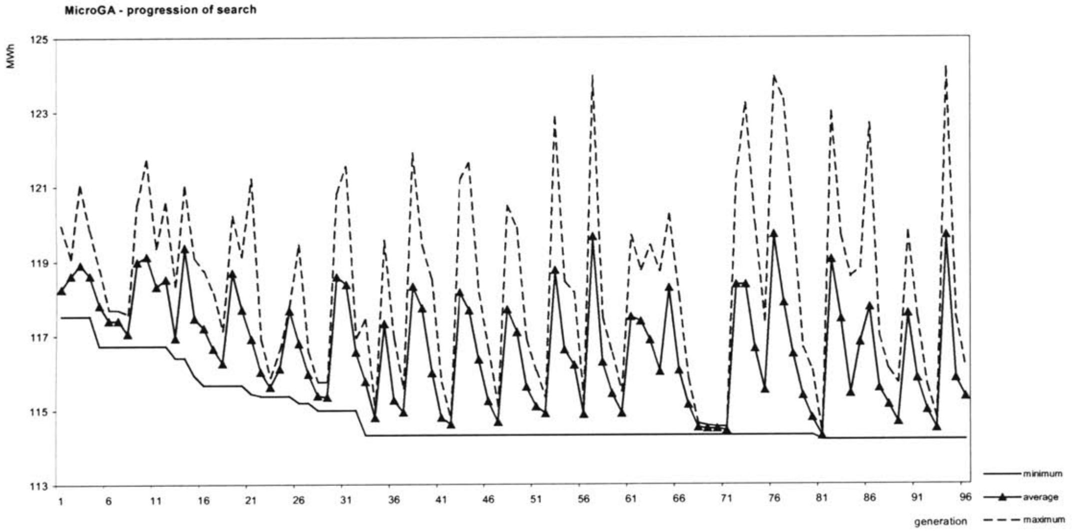
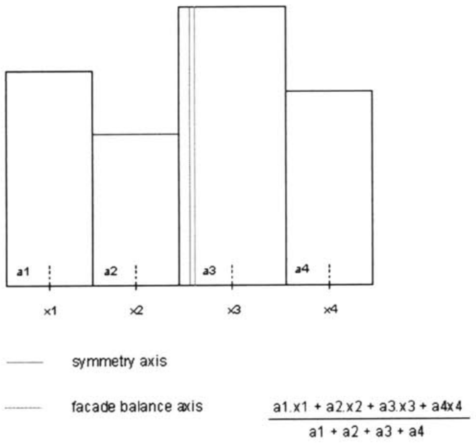


Figure 5.3 Search progression for one experiment using a MicroGA



$$\frac{a1.x1 + a2.x2 + a3.x3 + a4x4}{a1 + a2 + a3 + a4}$$

Figure 6.2 Façade profile, showing location of symmetry and façade balance axes. From left to right, we can see modules 1,2,3 and 4

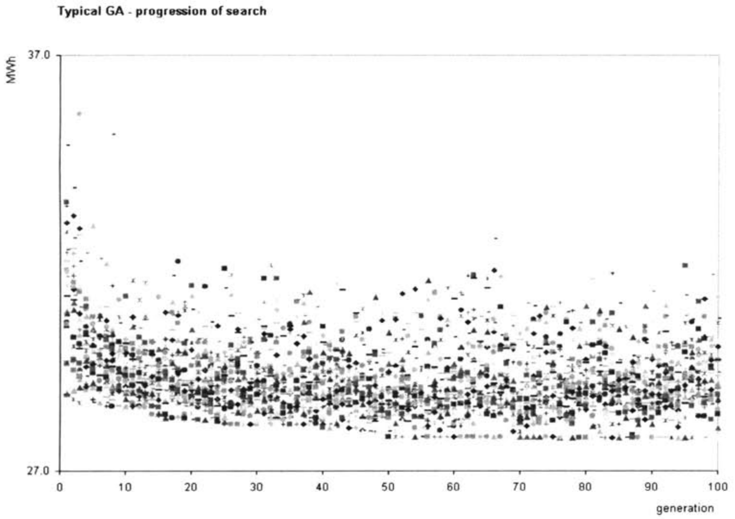


Figure 6.5 All points tested by the GS during the search [3000]

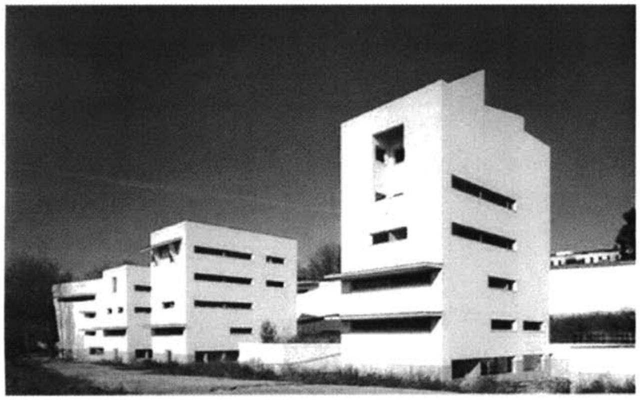


Figure 7.2 Southeast view of the four studio towers. Tower H is on the foreground

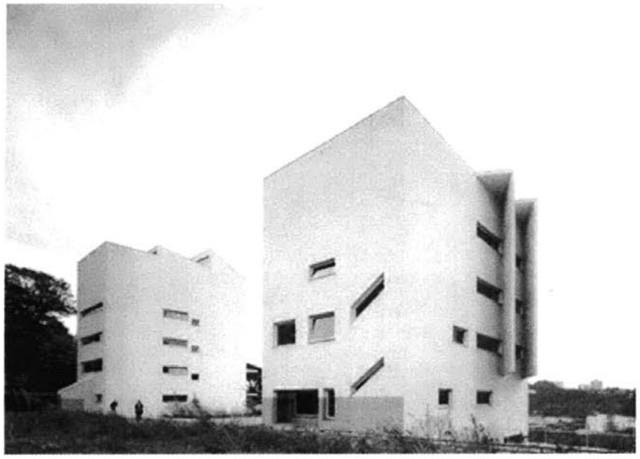


Figure 7.3 Northwest view of towers H and G. Tower H is in the background. Tower G is in the foreground, showing the large vertical fins that protect the west-facing studio windows from direct sun.

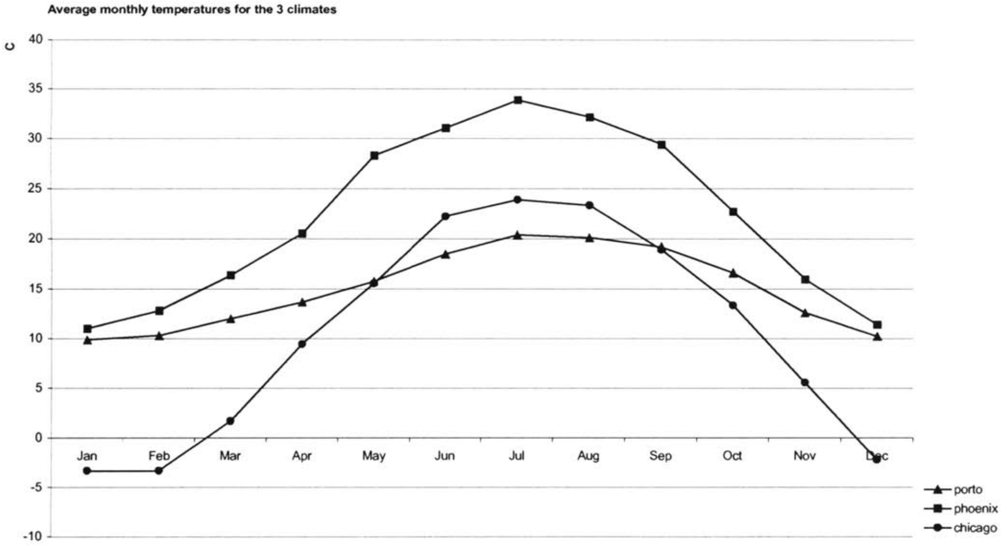


Figure 7.15 Average monthly temperatures for the three climates

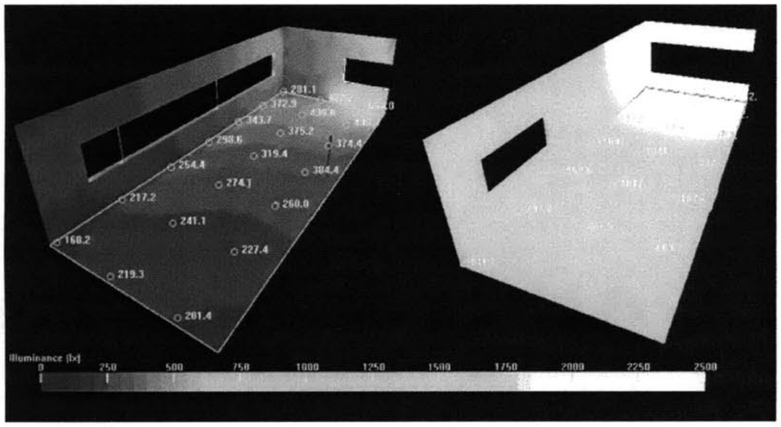


Figure 7.26 Comparison of existing [left] and GS [right] daylighting levels in 4th floor studio at 3pm, June 21.

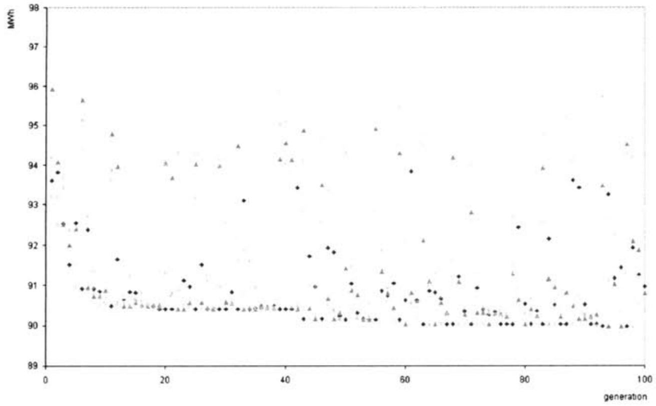


Figure 7.41 Each point in the graph represents a solution tested by the algorithm

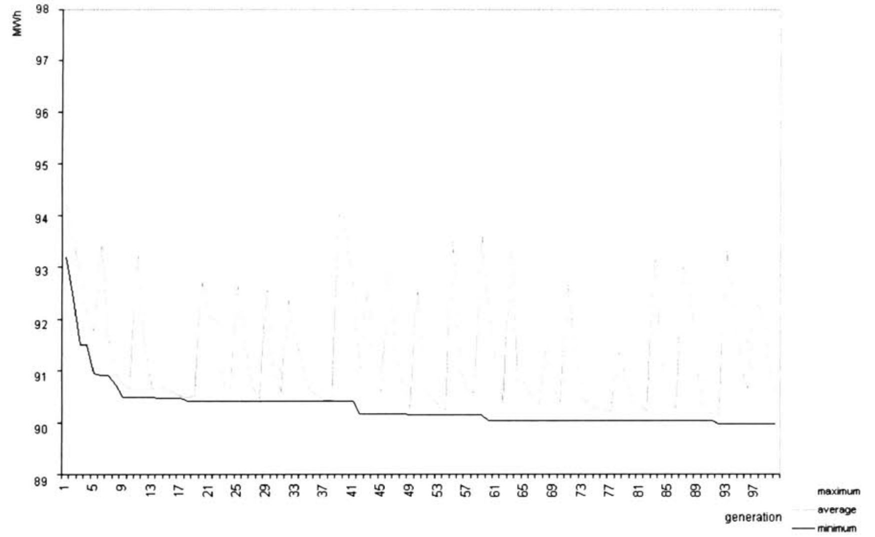


Figure 7.43 Progression of GA search for 100 generations

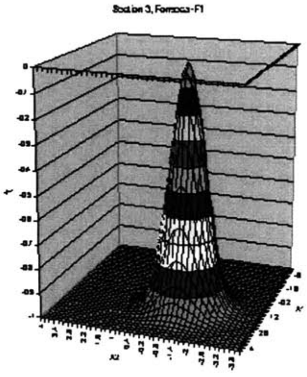


Figure 8.5 Function F1

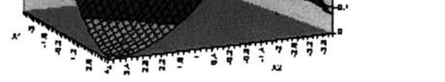


Figure 8.6 Cost Function of VEGA with Functions F1 and F2

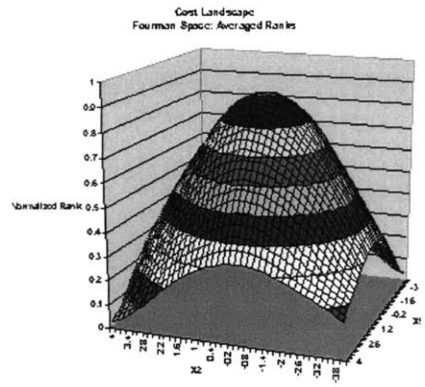


Figure 8.7 Cost Function using Fourman's implementation

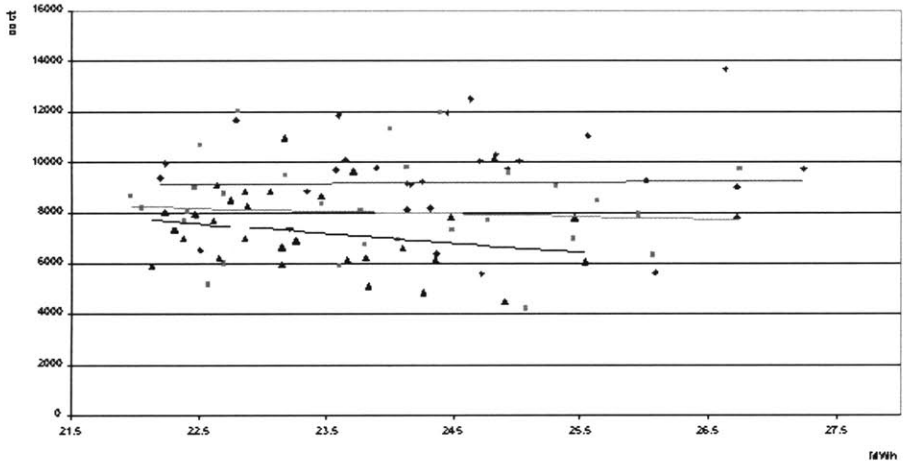


Figure 8.17 Experiments with the initial NSGA implementation showed solutions were not converging to a Pareto front, even though they were improving the overall population performance. Further code refinements were necessary.

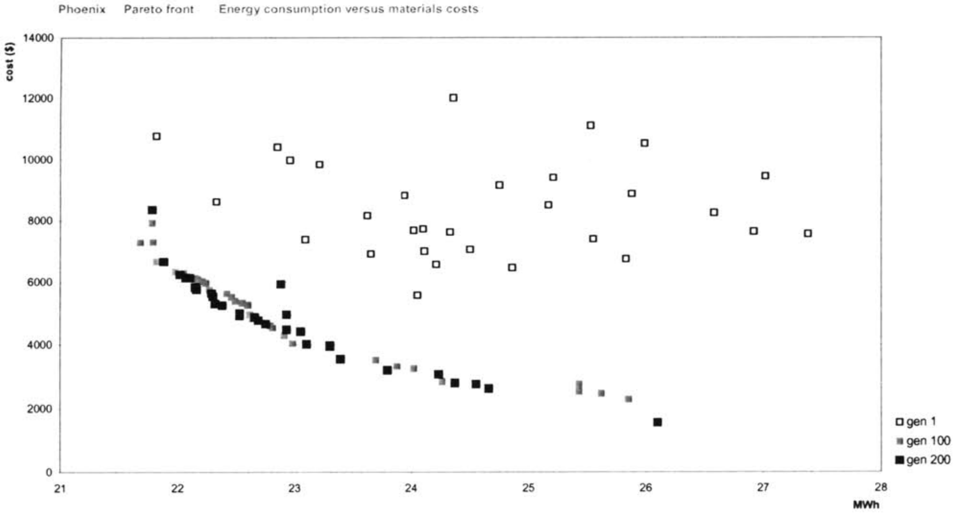


Figure 8.22 Pareto front for Phoenix climate

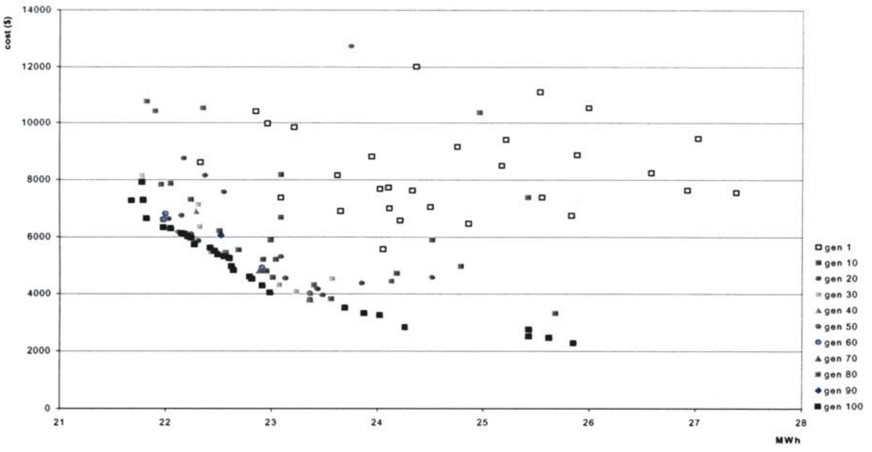


Figure 8.23 Search progression for the Pareto front for Phoenix climate

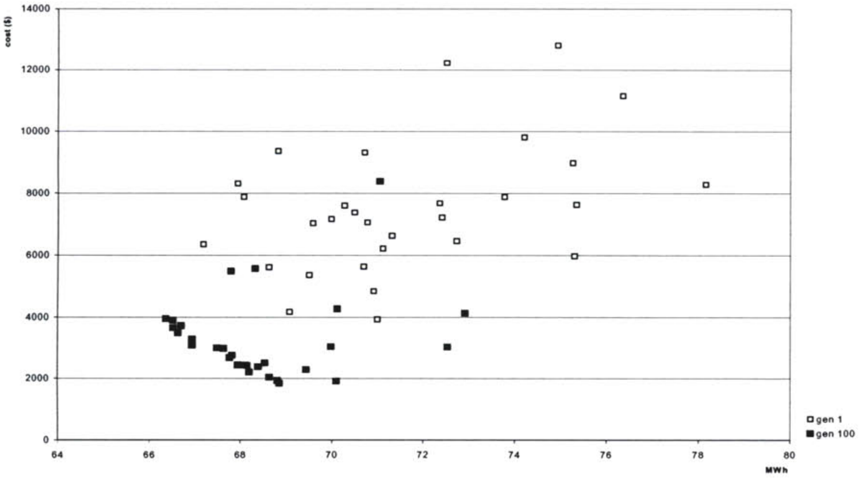


Figure 8.26 Pareto front for Chicago climate

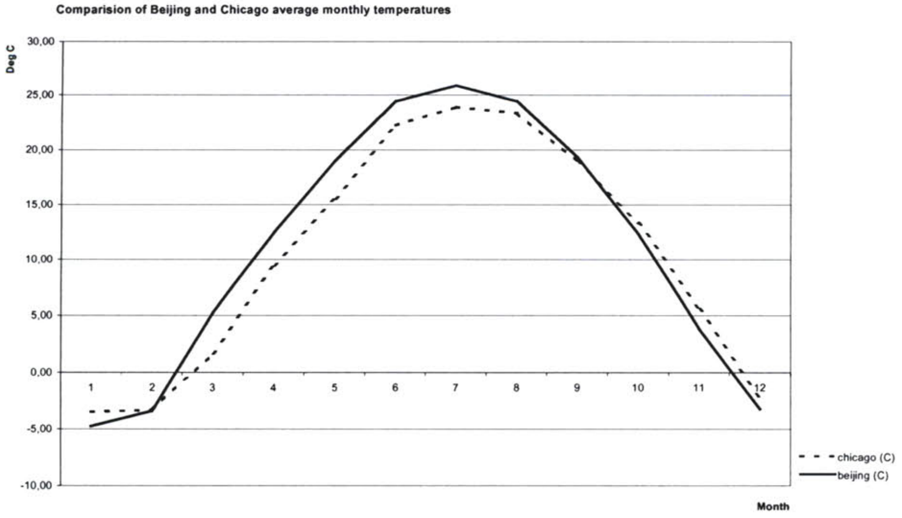


Figure 8.21 Comparison of Chicago and Beijing's monthly average temperatures

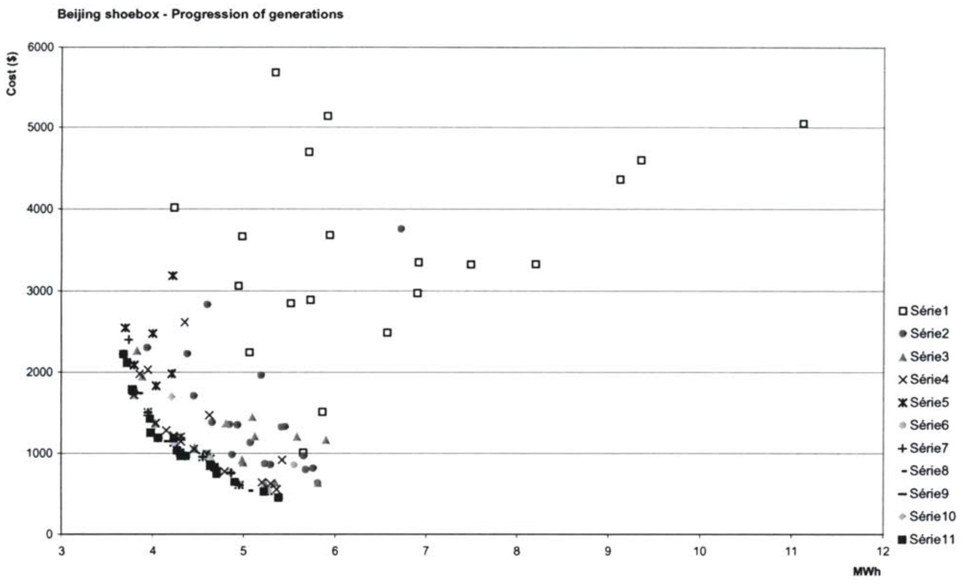


Figure 8.23 Progression of generations, from gen 1 to gen 100

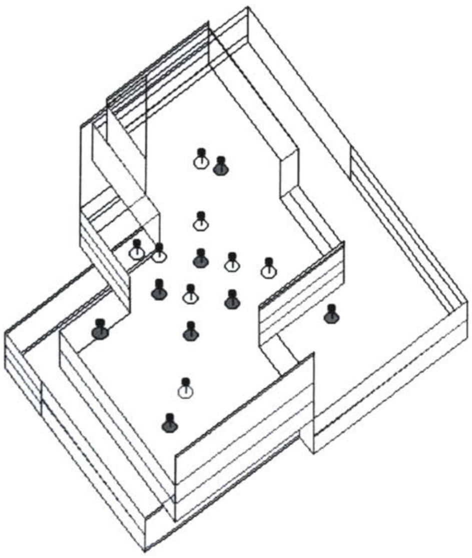


Figure 9.8 Automatic daylight sensors placement in a randomly generated building configuration. Sensors in gray are for the 1st floor, in white for the 2nd floor.