

Chapter 2

Optimisation

The Approach

The methodology I am going to describe is called *Optimisation*: this design approach has been already proven to be successful in several disciplines and also in engineering, but it is not popular in the building industry yet, and I must admit I find it quite surprising.

In the extreme summary, I would say that optimising means to look for the best solution: when we are facing a problem, we generally aim at solving it in the best possible way. We don't want to find an answer or a good answer; we want to identify the answer that addresses our questions more effectively than any other possible solution.

I'll first try to explain this thing in a general way, and then I'll give an example.

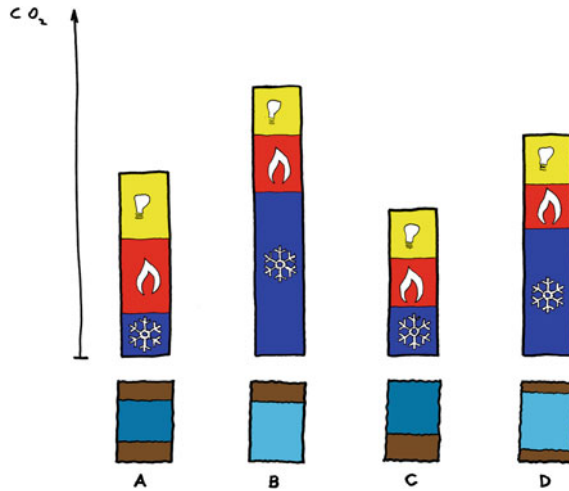
The first step is to be very clear in mind what you're after: you have to determine the criterion (or criteria) you are going to use to compare different solutions. In optimisation terms, this means defining the *objective function* and you have to refer to this in order to understand whether option A is better than option B. Once this has been clearly defined, then it will be possible to compare all potential solutions to your problem and determine which one is the best. Things become a bit more complicated if you want to compare options using more than one criterion, because in this case you can't find one optimum solution, but you can determine the set of options representing the best compromises between the different criteria. This approach is called *multi-objective optimisation*.

Now I'll give you an example of what I mean. Let's say that we are designing the façade of a building and we want to determine the right combination of type of glass and percentage of vision area, in order to achieve the best level of energy efficiency of the building. This is the very generic question, from which we have to define the objective function. In order to do so, it is important that we understand how the parameters we're playing with (i.e. percentage of vision area and glass type) affect the energy performance of the building. The three major sources of energy consumption, which can be directly related to the performance of the building envelope are: heating, cooling and artificial lighting. Glazed areas lead to

higher heat losses than the ones through an opaque, well-insulated wall; at the same time, solar radiation can enter through the glass, introducing solar gains within the internal environment, which lead to cooling energy demand. But it is also very important to guarantee a reasonable amount of daylighting, in order to improve visual comfort of the occupants and reduce the energy demand for artificial lighting. From this simple example, you can see that it's not that easy to define an objective function summarising all these aspects in an effective way. There are two possibilities: we can either make some assumptions and combine together the three sources of energy consumption in one single element, or we can keep them separate and consider three different criteria. In the first case, we will run a *single-objective optimisation*, the second case corresponds to a *multi-objective optimisation*.

Single-Objective

As I mentioned earlier, if we are to run a single-objective optimisation then we have to consider different aspects (in this example the energy consumption for heating, cooling and artificial lighting) and the objective function will have to combine the different criteria. In order to do so, we need to find a way to measure all of them and to sum up the partial results. Once this is done, it is possible to determine the overall performance of each façade option and describe it with one parameter. In this way the different façade options can be compared and we can run a single-objective optimisation. In the case of our example it is quite easy to combine the different criteria as they all represent energy consumptions. Since they are different types of energy, a good way of summing up the three contributions is to 'translate' them in terms of carbon emissions. This can be done by assuming the efficiency of the different systems (cooling, heating and lighting) and by considering some factors relating energy demand with carbon emissions—these factors differ for different countries, depending on the way energy is sourced there. The image below shows an example where four different façade options—A, B, C and D—are compared in terms of total annual carbon emissions, due to cooling, heating and artificial lighting.



Single-objective optimisation, carbon emission scheme

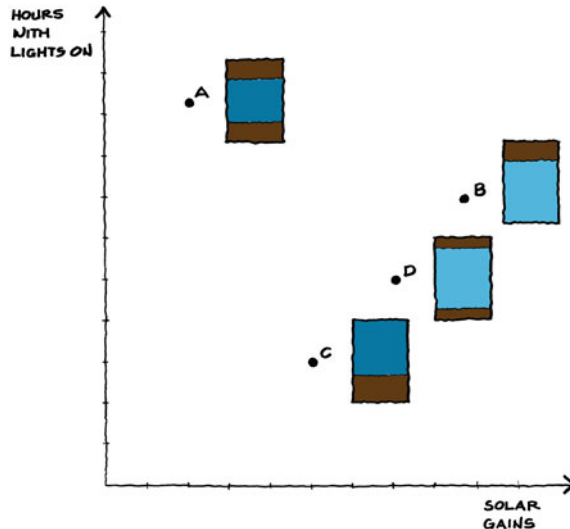
The best design will be the one corresponding to the lowest value of the objective function. It is very straightforward to understand that the best solution—among the ones considered—is C, whilst the worst one is B.

But everything becomes much more complicated if we are dealing with criteria which are significantly different. For example, if we wanted to run a single-objective optimisation where the criteria are energy consumption and internal comfort, we would have to think harder in order to find a common denominator. A possibility is to compare options in terms of cost: for energy, as we did when we assumed the corresponding carbon emission, it is relatively straightforward to assess reasonable values of corresponding costs. But what about comfort? Well, if we are designing an office building, we can say that the level of comfort is related to the employees' productivity, therefore better comfort levels will result in lower costs for the work of employees. And this cost can be combined with the one for the provision of energy. Obviously, the more 'indirect' the relationship between the different criteria, the more intricate the mechanism to combine them and the more heavily the objective function relies on assumptions.

Yes, I think I can see how difficult it can get to summarise completely different aspects in a single parameter. For example, I guess that mixing energy and space performance can become pretty academic! I'm not sure I'm convinced this can always work.

Multi-Objective

I probably agree with you: when you can't or don't want to merge different criteria in one single-objective, but you can choose to keep different aspects separate, in order to have a better understanding of the overall situation. In this case you are running a *multi-objective optimisation*. Let's come back to our example, but let's simplify it a little bit to keep things clearer: let's compare different façade options and see how they perform in terms of control of solar gains and allowance for daylighting. This is actually one of the most challenging questions for the performance design of façades: 'how can I achieve the best compromise between limitation of solar gains (and hence cooling energy) and provision of daylighting'? Answering this question can be a never-ending process, if we don't follow a proper procedure. For each of the considered façade options, let's calculate the annual amount of solar gains and the amount of hours during the year when artificial lights have to be on, because daylighting does not provide adequate levels of illuminance. In this way we can assign a quantity to each of our criteria for all our options. The image below shows how four façade options are distributed within what we can call *performance space*: options A, B, C and D are compared in terms of their performance considering the criteria we are interested in.



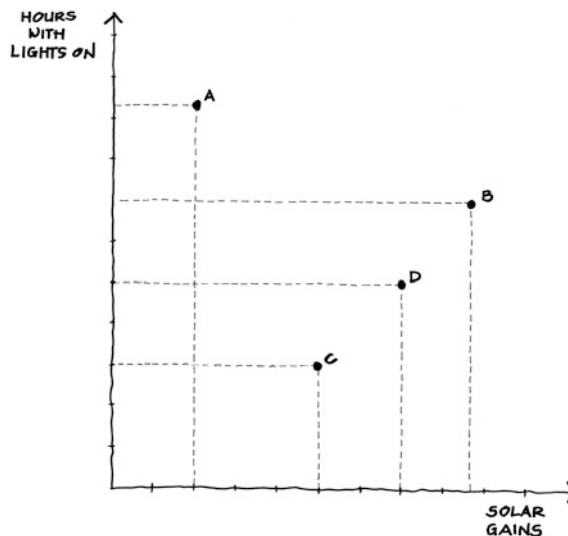
Multi-objective optimisation, performance space scheme

Option A is the one leading to the lowest amount of solar gains, but it is also the one where artificial lights will have to be on for the largest amount of time. Let's now compare options A and B: if we go for façade option B instead of A, we will have a higher amount of solar gains, but also we'll have more daylighting. Can we

say that option A is better than option B or vice versa? The answer is no because they reach different types of compromises between the two considered criteria. And the same happens if we compare options A and C or options A and D. Let's now compare options C and D: in this case option C leads to lower solar gains and better daylighting, so it is better than option D from the point of view of both criteria at the same time. In this case we can say that option C is better than option D: we should say that *option C dominates option D*. Let's define this concept properly: a solution X is said to be non-dominated by solution Y if both the following conditions are satisfied:

- (a) Solution X is no worse than Y in all criteria;
- (b) Solution X is better than Y in at least one criterion.

As it appears clear, we can't simply compare options separately, because we would get lost immediately if we did that: it's very important that we use an instrument that compares all the options at the same time. Let's use the concept of *domination*: we can say that all *non-dominated options are optimum solutions*. And it's easy to understand whether a solution is dominated or not: let's have a look at the image below.

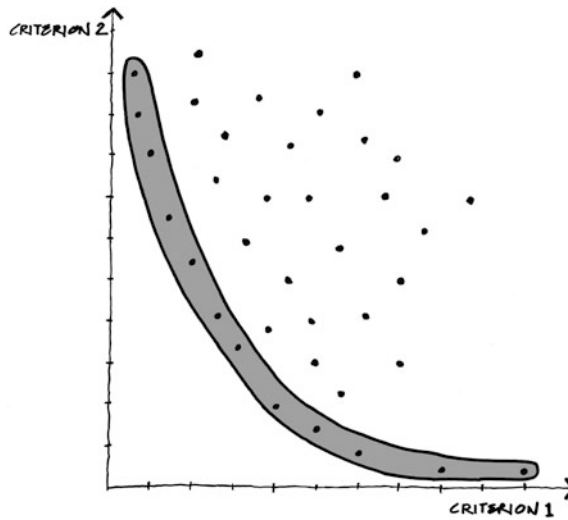


Multi-objective optimisation, domination scheme

If we draw a line parallel to each axis from the positions corresponding to the performance of each option, we'll immediately see if this option is dominated by any other options: if there are points representing other solutions between the two lines, our option is dominated. In our example, options A and C are non-dominated, option D is dominated by option C and option B is dominated by C and D.

In this case, then, we can say that only options A and C represent optimum compromises between the considered criteria: all other solutions can be dismissed.

Obviously, in a real case we would deal with a much higher amount of options, and possibly with more performance criteria. In such real cases, we aim at finding a lot of options representing optimum compromises between the design criteria: in technical terms, the scope of a multi-objective optimisation analysis is to find the *Pareto front*, which comprises all the non-dominated solutions. If we have two criteria, the Pareto front is represented by a curve, if we have three criteria, the Pareto front is a surface, and so on. What if we have only one criterion? The Pareto front will be a point, corresponding to the only optimum solution of a single-objective optimisation analysis.



Multi-objective optimisation, Pareto front scheme

Single- and multi-objective optimisation are strictly connected to each other, they are basically different interpretations of the same process. In the case of a single-objective optimisation process, we have to make some assumptions in order to connect the different criteria and determine one parameter that we use to compare different solutions. In other terms, we can say that we introduce some *weights* to combine the different criteria: in our previous example, the performance criteria were cooling energy, heating energy and daylighting availability, and we used some weights (the assumed energy efficiencies of the different systems) to calculate a total energy consumption. These weights are defined at the beginning of the optimisation process, *a priori*. In the case of multi-objective optimisation we

don't do this, but we keep the different criteria separated during our research for the Pareto front. Once we have identified all the non-dominated solutions, i.e. the optimum compromises between the different criteria, we can introduce weights to sum up the objectives. In this case we use the weights after the optimisation process, *a posteriori*, and this can give us several opportunities. If we have done things properly, the solution we get from a single-objective optimisation is within the Pareto front, and we will find it if we apply *a posteriori* the same weights that we applied *a priori* for the definition of the single-objective performance scale.

This description should make it clear why optimisation can represent a good method to achieve a balanced-design: it basically makes it possible to quantify the qualities of different design solutions and hence to compare them, in order to identify the good answers to our questions. The final result is that we can focus our attention on a limited number of options, which we know are very effective, and can exclude the ones that don't provide good value but could generate confusion for the design team. I would also like to highlight the fact that by using optimisation we are also forced to think hard about ways of describing the qualities of our design, in ways that can be compared clearly and indisputably.

Optimisation and Objective Function

Excuse me, but I have to stop you to make some comments about what you said, before it gets too broad.

As far as I know, optimisation is a mathematical procedure used in management to find the best configuration in a set of given data. So, if I had to apply it to the construction industry, I would have associated it with the management of work, resources, materials, etc. But, according to your definition, the concept of optimisation seems already entirely integral to the process of design (rather than to management).

When we design something, what are we doing if not looking for the best solution? So, when we talk about design, in a way we're already talking about optimisation.

Yet, I would point out some distinctions.

I don't like it when people talk about solving problems. We, as designers, don't solve problems, we rather foresee them in order to pursue a will. I would say that our task is to make this will possible. The will demands the most effective answer to its request. We measure its effectiveness according to the fulfilment of our expectations. So, to get an effective answer, we must know very well what we are asking.

Having design in mind, the thing I find most interesting about optimisation is the importance given to this issue: how to define the quality of the product. It's interesting that, instead of presenting optimisation as a user-friendly tool, you said right from the start that the only way to use it is to know exactly how to ask what we want.

I very much share this vision applied to design: I think that the beginning of the process has its development in itself and theoretically everything has already been decided from the very first step. This is fundamental for me. The objective function looks like the summary of a design, or, in other words, a magic formula (even if the process that it triggers has nothing magical).

I like to think that if we build a system considering all the variables and every single solution we could write an objective function for the whole design, where attentively considered requests lead to optimised results.

Sure, the procedure you present is a brilliant and efficient tool, but there's a problem: in order to use it, first we have to conceive a performance and to define its purpose. If we don't have the standards that help us decide what to expect from the design, we need a vision, a point of view, a will that requires specific performances to the project as a whole.

What I fear when I think of the word 'optimisation'—a process that leads to the optimum, the search for what's absolutely excellent—is that it might trap the design into a deterministic and irreversible process.

OK, OK, I see your point, but wait a second: so far we have spoken of optimisation in general terms, and the procedures explained earlier are applicable only if we have the possibility to evaluate all the façade options we want to consider. Obviously, in the real-world, this is not necessarily the case: the number of alternatives that we want to consider can potentially be unmanageable. Traditionally, the design team decides to analyse the performance of a very limited amount of options, which are selected because they proved successful in previous projects or because of some rule of thumbs. This approach is perfectly legitimate and for sure it can lead to very good design solutions with reasonable calculation effort. But there is a massive limitation in this design strategy: how can we find out innovative solutions if we limit ourselves to a small number of options we consider safe because of previous experiences? If we want to achieve unprecedented results, we need to adopt a new procedure! The one I want to talk to you about is *evolutionary optimisation*: this is a procedure which combines the aspiration of considering extremely wide spectra of potential solutions with the limitations of calculation effort. In very general terms, in evolutionary optimisation the research for optimum solution (or solutions if we are running a multi-objective analysis) is not a static process, but it *evolves* through different *generations* of calculations, which can test options spread along the whole range of possible solutions. The idea is that the research starts in a completely random way: some possible solutions are picked within the so-called *research space*. I want to stress the fact that these starting points are not chosen by the design team (who defines only the limits of the research space and the objective function), but are randomly selected: this is very important because only in this way it is possible to explore options that would normally be excluded a priori. These selected options are analysed and the corresponding values of the objective function(s) are calculated; in this way it is possible to start having an idea of some trends of how the objective function varies with the parameters describing the different façade options. From these

preliminary trends, the optimiser can select a second-generation of possible solutions, which are expected to achieve better performance in terms of the considered objective function, since we are interested only on good options. It may be that the preliminary trends were misleading and potentially the options considered in the second-generation could perform worse than the previous ones: never mind, because even so, this second-generation of calculations will provide more information to the optimiser. In this way, trends can be better understood and the following generation of options can be determined. This process goes on for a certain number of generations, and in general along generations the quality of tested options increases more and more. At the end of the procedure, the optimum option or the options corresponding to the Pareto front will be identified.

During the previous description, I used the very general term *optimiser* because I didn't want to enter too much detail, but actually this noun has everything in it. An optimiser could be a person who can interpret results and understand what I called trends, and in this way identify the following generation. In general, though, optimisation has to be automated, in order to be an effective design tool, therefore the optimiser is an *optimisation algorithm*. There are a lot of different types of optimisation algorithms, and this is not the place to go through all of them: I'll describe to you only one type, which is the most popular. I'll describe these algorithms in a simplified way because I don't think it is useful for our discussion to enter the very details of them. I hope in this way things will be clear.

Genetic Algorithms

Genetic algorithms take inspiration from the laws of Darwin's theory reported in his "*The Origin of Species*". In extreme synthesis, Darwin shows that in nature individuals compete for their own survival, and within populations there is a huge variation of individuals: different characteristics of individuals are inherited through generations. The key thing is that not all individuals have the same opportunities to transfer their characters to following generations: beings which are more suited to their local environment have more opportunities to survive and hence to reproduce. In this way they transmit their successful characters to following generations, and therefore populations evolve and become more adapt to the environment.

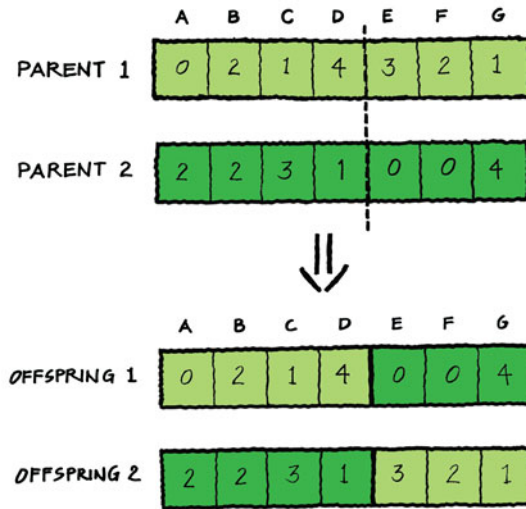
Genetic algorithms try to reproduce natural laws and apply them in the search for solutions of optimisation problems: like individuals, solutions evolve towards conditions of better fitness that make them more appropriate answers to problems. As I mentioned earlier, the research for optimum solutions goes from generation to generation of potential options. After one generation has been analysed, each option of this generation can be assigned a fitness: options can hence be compared and it is possible to determine a ranking. The following generation of potential solutions is identified by means of three different operations: *selection*, *recombination* and *mutation*.

Selection

As for the Darwinian's survival of the fittest, selection is the mechanism through which options with a better fitness are more likely to survive and to move to the following generation. It is important to note that, as for the natural process, we don't want to necessarily get rid of all 'bad' solutions within one generation step: we want only to give more opportunities to good solutions to 'survive'. And this is because there may be some good aspects also in options which, overall, perform badly. There are several ways we can follow to achieve this. One of them is to recreate the mechanism of the roulette wheel: we can imagine to have a roulette wheel whose different slices represent the individuals. The size of the different slices is proportionate to the fitness of the different solutions, i.e. the solution with the highest ranking has the largest slice. Depending on where the wheel stops, the corresponding individual 'survives' the selection: obviously it is much more likely that solutions having good fitness are selected. The wheel turns as many times as the number of options we are considering for each generation.

Recombination

Selecting fit individuals is clearly not enough because we want to explore new options as well. The idea is that it is likely that the combination of solutions with a good fitness can generate other fit options: the good qualities of different solutions can be complementary and their combination can lead to very successful characteristics. As for selection, also recombination can be carried out in different ways, but the most common is the mechanism called *crossover*. Let's say that we can describe the considered options by using six different characteristics (from A to G) and each characteristic can express itself in five different ways (from 0 to 4): in this way we can assign to each individual a 'chromosome' that describes it completely. Crossover cuts the chromosome of two *parents* and mixes them, generating two new *offspring* individuals. The image below describes how this mechanism works.

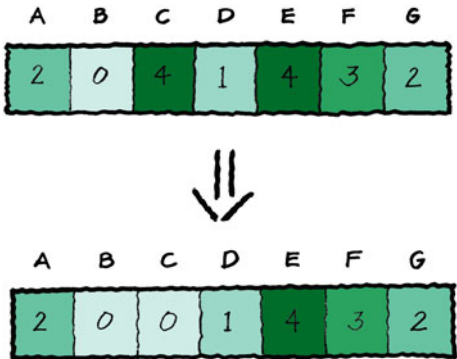


Genetic algorithms, recombination scheme

As for the selection process, also recombination is significantly driven by random, casual conditions: the location of the *cut* within chromosomes is not controlled. This allows to keep the research for the optimum more open.

Mutation

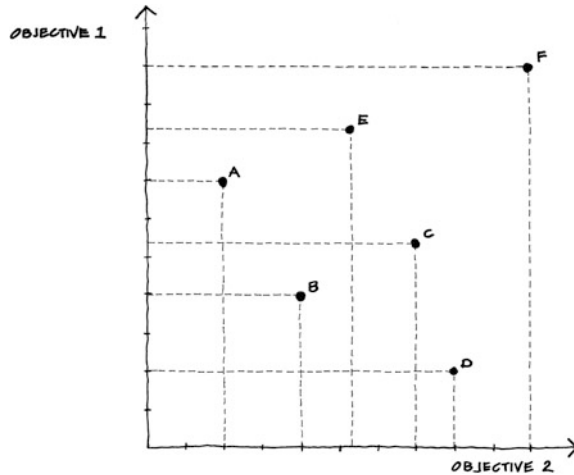
In theory, selection and recombination can make different generations evolve towards optimum solutions, because they guarantee that fit options have adequate opportunities to transmit to the following generations the character that make them successful. But along the natural selection process, another mechanism occurs and has an impact: mutation introduces some changes in the individuals' chromosomes, which are unexpected and random. Changes due to mutation are not related to the fitness of solutions and don't happen because they are expected to provide improvements. But mutation introduces an unpredictable element that allows to explore areas within the problem space that could be completely disregarded otherwise. The mechanism of mutation is very simple: a number of individuals within a generation is randomly selected and one element of their chromosome is randomly changed.



Genetic algorithms, mutation scheme

Depending on the type of problem we want to analyse, these three procedures have to be adjusted. In some cases it is preferable not to recombine all the ‘survived’ individuals, but to keep the fittest ones in-haltered, in order to keep their chromosomes for the following generations. And also mutation can be relatively frequent or extremely rare: in the first case the algorithm will explore a wider region of the problem space. If mutation occurs very rarely, the algorithms will find optimum solution(s) in a quicker way. Unfortunately, it is not possible to define precise rules that define how to adjust these parameters in the best way for each problem: these adjustments rely a lot on the user’s competence and experience.

Genetic algorithms can be used for both single- and multi-objective optimisation problems. The three processes of selection, recombination and mutation occur in the same way. The main difference is in the way the fitness of individuals is determined, in order to initialise the process of selection. For the case of single-objective optimisation, the fitness of each individual corresponds to the level of the objective function. For multi-objective optimisation, things are slightly more complicated. In this case the fitness of the different individuals is their *domination index*: for each individual it represents the number of other options (within the analysed generation) dominating it. Let’s have a look at the chart below: it shows the positions of six solutions within the solution space, defined by two objective functions. If we consider the case that both objectives are to be minimised, we can see that solutions A, B and D are not dominated and solutions C, E and F are dominated. But this is not a sufficient distinction, because we also need to differentiate in terms



Multi-objective optimisation, domination scheme

of how many options dominate each individual, in order to establish a ranking of the generation. The chart in the image shows the following ranking: options A, B and D have fitness 0, option C has fitness 1, option E has fitness 2 and option F has fitness 5, therefore will be the most likely to ‘disappear’ after the selection process. Now, without entering in the very details, it is important for the optimisation algorithm not just to identify as many non-dominated solutions as possible, but also to find solutions along the whole Pareto front, in order to provide adequate information for the decision-making process. There are several ways of adjusting the genetic algorithms for that, but this becomes too specialist a discussion.

Randomness and Hypothetical Logic

I think we should reflect on the connection of optimisation and evolutionary theory. Please, spare me a moment to think about what you’ve just said.

Somehow, it seems to me that there’s a semantic contradiction between the two concepts that should cooperate in your method: optimisation and evolution. If optimisation searches for the optimum, i.e. what in a given set fulfils in the best way an external requirement, and evolution generates the ‘evolved’, i.e. what over time has most effectively reacted to a given environment, then what is evolved is

not necessarily the optimum and vice versa. Actually I would say they are quite the opposite: optimum is a requirement and the ‘evolved’ a result.

But probably this attempt to bring together the contrasts, this ‘I want everything’ attitude, goes well with the ideal of the architect, who wants the design at all costs.

We’ve already spoken about optimisation, but now I’d like to spend some words on the evolutionary process, that plays a key role in what you said.

On the basis of orthodox Darwinism, nature has no purpose, no direction and no inevitable outcomes. It evolves according to a certain randomness.

Now, every design is a statement, a declaration of will. It is a law. You will agree with me that law and randomness don’t get along very well. So, I wonder, why should we leave to randomness the search for what we want? (Even if it is some kind of benevolent and friendly chance). And why should we trust this randomness right in the field of architectural design? When architecture is the realm of care, accuracy, informed decisions etc., basically the opposite of randomness?

Hence, another objection: the result we get from optimisation, through the randomness that rules the procedure, is not exactly the optimum but, more precisely, something that comes very close to the optimum: the hypothetical optimum. The logic that rules the system is a hypothetical logic. In this sense optimisation seems to embrace the principle, typical of the scientific knowledge, that there’s no final or undeniable Truth.

This is the most modern, disenchanted and smartest aspect of our scientific-technological culture. No result is absolute! But, as human beings, how can we be satisfied with something that’s only highly likely?

Even if we are happy with our result, we know that probably there’s a better one. And this awareness, i.e. that what we got is not the best we could get, makes us unhappy.

At least, when we design and make decisions in the traditional way, we can experience some happiness in the finite and imperfect awareness that we’ve found, once and for all, what is for us the optimal solution. It might be a flawed happiness but it’s certainly rewarding.

But I don’t want to divert from the topic of our discussion. Getting back to what you’ve said, I must say I find very meaningful the two issues of randomness and hypothetical logic.

Enhancement Methods and the Idea of Perfection

In nature, things evolve reacting to external stimuli, changing their biology according to the environment and as far as we know the way they do it is not planned. This means that the earlier stage of an organism (or a generic being), is not necessarily worse than the latest or most advanced one. Each stage has a

certain autonomy in terms of progress, because it doesn't have necessarily a relation of cause and effect with the following one.

Every being is perfectible and tends to perfection. The result, though, is not predetermined or predictable. So, since this concept of perfection is relative (not absolute) it's impossible to assert the existence of a plan in nature. And this is precisely the main difference between natural evolution and architectural design, because the latter takes its origin from a desire and an idea.

To apply to design a method inspired by natural selection, then, means to separate the idea from its completion. Perfection only belongs to the phase of conception, while enhancement is pertinent to the following stages.

So, we can use randomness as a procedural element, functional in finding the best solution, but the thoughtful questions and planned choices that trigger the process at the very beginning don't leave anything to chance.

Anyway, the fact that we obtain the best solution through a process led by randomness is eventually an advantage for the designer: he/she doesn't have to take the final decision by choosing personally a specific solution, but still has the responsibility of asking the right question at the very beginning.

The system makes it easier to choose (doubt) and strengthen the decision (will). It lightens the means and deepens the end.

The process of optimisation seems to be extremely smooth and efficient, even seductive. Hence a few doubts: how do we question the process? Aren't we likely to twist the result, at the cost of asking the right questions? Is there a risk to bend the end to the means? In other words, could optimisation take over architects, instead of being a handy tool at their service?

Optimisation of Solutions Versus Optimisation of Decisions

Isn't it interesting that solutions evolve in the same way as species do? This way, they seem to be more like living creatures than working machines. Therefore I would speak of decisions rather than solutions (since decisions are better suited to living beings), and I would call the process *evolutionary optimisation of decisions*.

Things get even more interesting when you say that solutions are like individuals. At this point, the process appears to me like an experiment, a model of reality, even though it is obviously an abstraction. Pushing it further, we could say that thoughts behave like individuals.

The process replicates the activity of the mind. This mind, though, doesn't think like a machine, but follows closely the interaction of thoughts inside a human brain. In theory we could adopt this method every time we need to take a decision, choosing from a wide range of variables. It would be nice to have an ongoing process of optimisation in the background of our mind for all the decisions of our everyday life. It would be like having a constant technical support to our brain.

Getting back to architecture, would it be possible at some point to feed the process not only with the data from a generic situation, but also with historical

information related to the specific problems that we are facing from time to time? This could be a way to improve the decision-making process, enriching it with previous architectural experiences. It might provide us some extra tools to estimate the value of a solution.

In comparison to the past we have much better analytical skills in terms of logistics—we are able to manage and organise massive amounts of data—but when it comes to make a qualitative synthesis, we have to resort to subjective judgement. In this sense things get problematic when we have to assign a value to different solutions and determine their fitness.

In addition to comfort, feasibility, energy performance, amount of CO₂ emissions, what are the other parameters that can help us evaluate fitness?

We can define a hierarchy of values only among elements that are comparable according to some rules or standards. But how can we decide for (qualitative) features that haven't been prescribed or that are difficult to regulate?

A decision-making process is efficient as long as there is a definition of values. So if we want to extend optimisation to our disciplines, how can we determine the value of a design? I believe that we can't answer this question just in terms of efficiency and beauty.

And besides, once we have some judgement criteria and we've started the optimisation process, to me there's a big difference between the natural evolutionary process and its artificial counterpart. It's about the very concept of fitness. Because what is fittest in nature is not necessarily the strongest, the smartest or the most obedient to the laws of nature. The concept of survival cannot be reduced to a matter of fair (benevolent) chance. In other words, I find it difficult to convert the so-called struggle for life into an artificial model.

I'm not sure I agree with you when you say that you would speak about 'decisions' rather than 'solutions'. Solutions are static, they do not deliver any action: they are simply analysed, they kind of *ask for the permission to exist* but it is not up to them to give any contribution during the optimisation process. If, by chance, they are good enough and they survive the selection, they will transmit their characters to the following generation, in a way that they cannot control. Having said that, the algorithm does not take any decisions either: it works in a rigid way, relying heavily on randomness. The only decisions are taken by the design team during the definition of the problem, when the criteria to describe the fitness of solutions are established. And then, after the optimisation process has delivered the list of optimum solutions, everything is again in complete control of the design team. I think this is something very important to stress properly, because many people are reluctant in using the computational power available as they do not want to leave the decision process in control of *soulless machines*.

I don't think we should be worried about the fact that randomness is an important ingredient of optimisation. I know it can sound extremely uncomfortable to have some design decisions taken by means of a random process—but is this really the case? I'm going to explain why I don't think this is what actually happens. First of all, let's have a look at what is driven by arbitrary operations. The

first thing we cannot control is the selection of the first-generation of solutions: we define how many options there are in each generation, but then the algorithm picks the starting points of the evolutionary process completely by chance. We already discussed that it is very important that this happens, because only in this way we are sure that the whole space of potential options are explored, without limitations dictated by our previous experiences/prejudices. Once the different options of one generation have been analysed and their fitness has been assessed, the three operations followed by the algorithm to determine the following generation are strongly influenced by randomness. For selection, the dimension of each slice of the roulette wheel is proportionate to the level of fitness of each option, but then the way the wheel turns and hence the position where it stops is completely arbitrary. For recombination, randomness acts at two levels: first of all when couples of parents are formed—there is indeed no control about which option is mixing its genes with which—and also the location where chromosomes are cut and then combined is random. Eventually, mutation is the most arbitrary of all the operations, as we cannot control which option is going to mutate, which gene will change and how it will be modified.

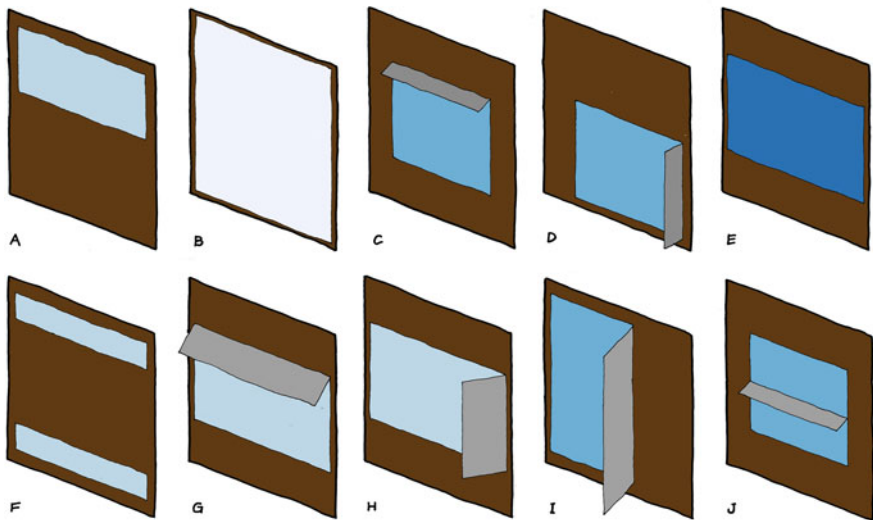
But does this mean that the final result is completely random as well? Of course not, otherwise the whole thing wouldn't make any sense. It is good practice to run the same evolutionary optimisation a number of times: if we get the same result, it is very likely that the optimisation process has been successful and that the result is the actual optimum or the Pareto front. And this is because randomness should impact only on the way the optimum is identified by the algorithm, without impacting on the final result, which is determined uniquely by the definition of the problem—i.e. the range of potential options we want to consider and the fitness function we have chosen. Since the optimisation procedure aims to find out the optimum solution without really testing the whole set of options, it is possible that the algorithm hasn't found the best solution, and we cannot be completely sure that the evolution didn't get stuck in what we called 'a local optimum'. Hence the importance of having different runs of the algorithm, because in this way we limit the likelihood that the same local optimum 'confuses' the algorithm all the times. So, in conclusion, randomness is a tool to reduce the risk of not finding the optimum, but it doesn't represent a real part of the decision-making process. Therefore we, designers, should not be worried about having our project being influenced by arbitrary, irrational decisions taken by some strange behaviour of one of the microchips within the engine of our soulless computer! Every important decision which impacts the result of our design is under our full control—which also means that we can't blame our computer if our design is wrong... Therefore, it is really up to us to define the problem, or the question we are asking the algorithm to answer, as the real answer is simply embedded in the question itself: the quality of the optimum is a direct consequence of the quality of our definition of the range of options and of the fitness function.

Let's Try to Optimise a Façade

After this brief, general description of genetic algorithms, I think it's worth understanding how they work for the façade design. As an example, let's say that we have to optimise a façade where the following parameters can be varied:

- The number of windows (within a module);
- The percentage of glazing;
- The presence of shading devices (overhangs or vertical fins);
- The geometry of the shading devices (in terms of depth and angle);
- The position of windows;
- The type of glass.

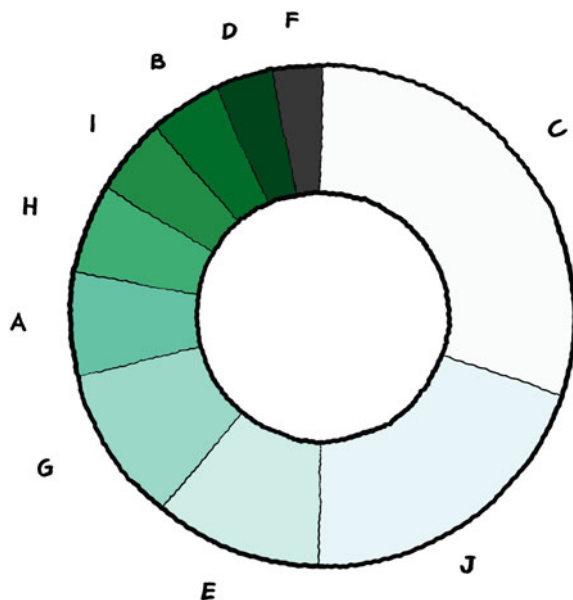
As you can imagine the number of potential solutions is enormous and it is impossible to analyse all of them, therefore genetic algorithms can represent an adequate tool to tackle this problem effectively. Just for the sake of discussion, let's say that for each generation there are ten individuals (the actual number should be larger than this, but I'm trying to keep things simple for this example). Let's imagine that, for the x-generation we are considering, the ten solutions correspond to the ones shown in the images below.



Optimisation of a façade, x-generation scheme

Let's say that the fitness of these options has been assessed and it results that option C is the best one, with the worst option being F. By knowing the fitness of each option it is possible to build the roulette wheel: option C will have the largest

slice and option F the smallest. The size of the different slices is proportional to the level of fitness of each option: the ratio between the size of a specific slice and the size of the whole wheel represents the probability that the option corresponding to that slice is selected at each turn of the wheel. In the example, the fitness of option C is very good and it makes the slice corresponding to option C occupy 30 % of the whole wheel: this means that, for each of the ten turns of the wheel, option C will have 30 % of probabilities of being extracted.

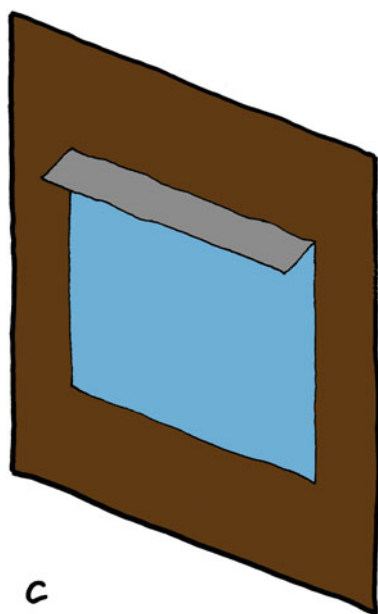


Roulette wheel, fitness scheme

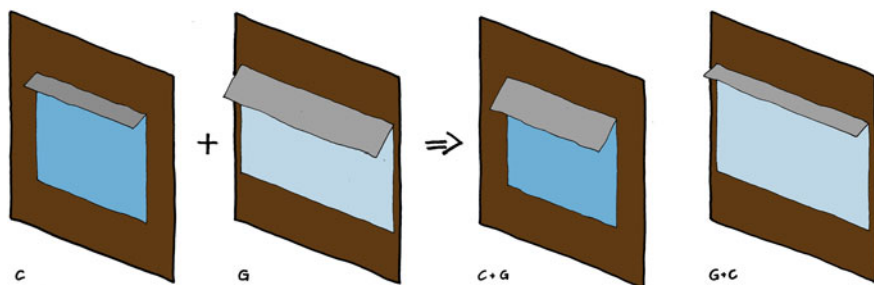
I have made the wheel twist 10 times and the following options have come out:

C; C; G; A; J; C; C; D; J; H.

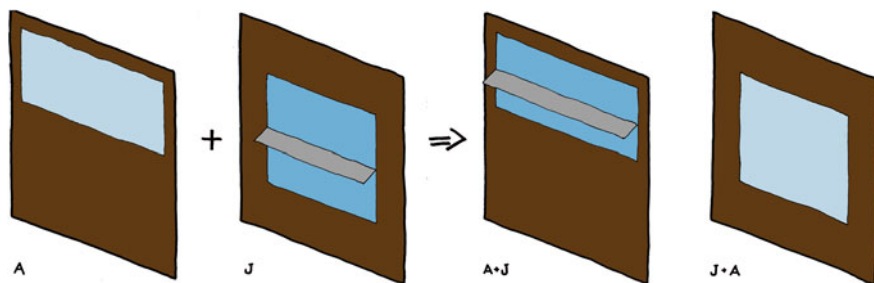
If we say that we want to keep one option in-haltered and to mutate one solution, we will have:



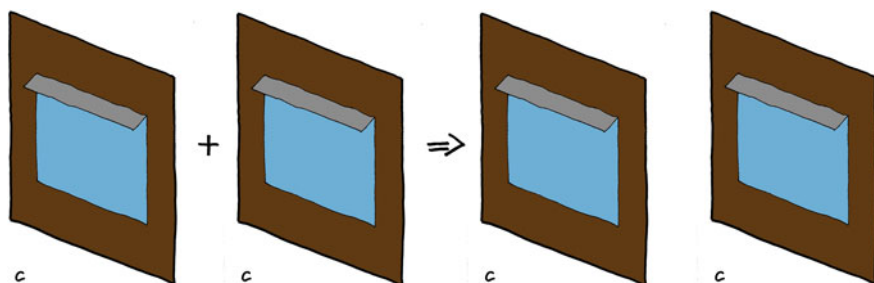
In-haltered C



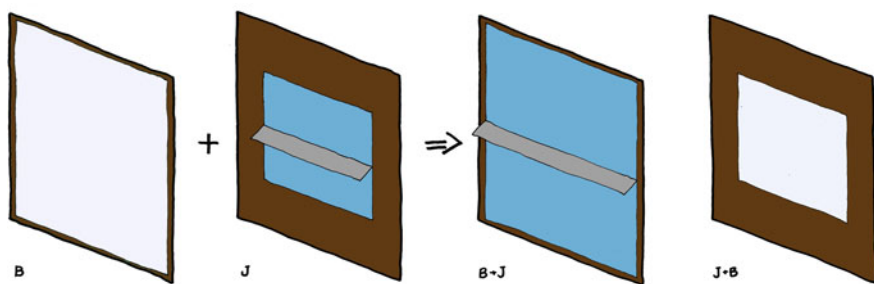
Cross-over between C and G



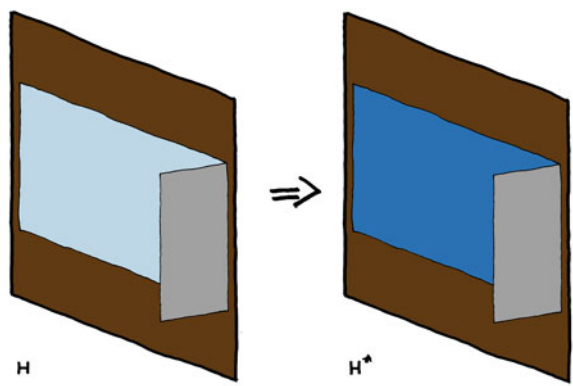
Cross-over between A and J



Cross-over between C and C

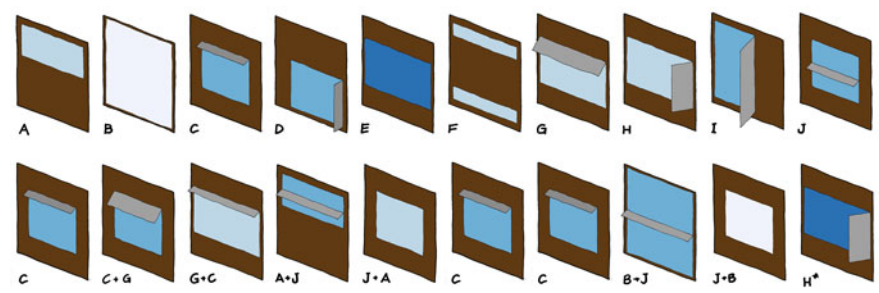


Cross-over between B and J



Mutation of H

As it can be seen, moving from one generation to the following one has led to almost a complete disappearance of vertical shading devices, and there are six options out of ten which present the same type of glass. This is just an example, but it shows how genetic algorithms pursue the research of optimum solutions.



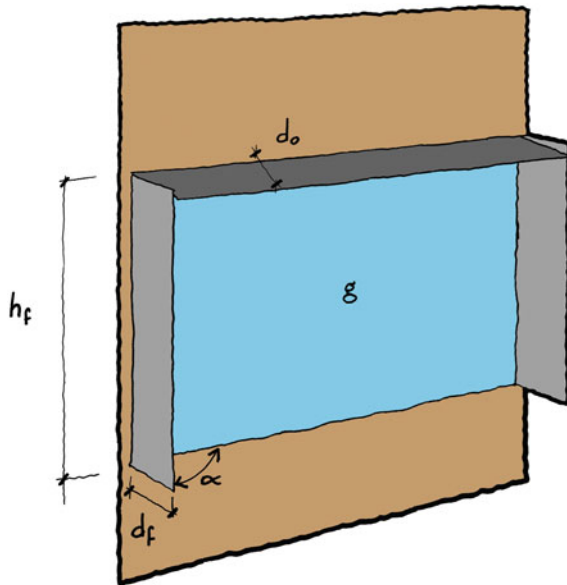
Research of the optimum solutions

So far we've seen how the genetic algorithms work in principle to evolve from randomly selected options towards the optimum one(s). This procedure will be different for the other types of algorithms that can be applied for evolutionary optimisation. What is the same for all the different versions of algorithms is what we expect from them:

- Algorithms have to explore a very wide range of possible solutions;
- They have to identify the best solution(s);
- They have to do so within a reasonable amount of time.

Once the optimisation has run, we have a series of options: if we ran a single-objective optimisation, we'd have the best solution and the ones whose fitness is close enough to the optimum. After a multi-objective, we'll deal with the solutions within the Pareto front. Potentially, the amount of data can be really difficult to handle and generate confusion. It is important to find a good way of fully understand the results and hence take full advantage of the optimisation process. The main questions that we want to be able to answer properly are: (1) What are the specific configurations of façade (for example, percentages of glazing, types of glass, etc.) that should be avoided? (2) Reversing the question, are there some elements that have to be very prescriptive? (3) Are there parameters that don't affect the overall performance and which can therefore be designed considering only aesthetic drivers? (4) Are there ways of improving significantly the performance with little impact on the aesthetics? If we can't find a way to use the results of our analysis to answer these questions, no matter how efficient the optimisation process was: we have failed!

Let's try to answer these questions through an example, represented by the image below.



Optimisation of a façade module

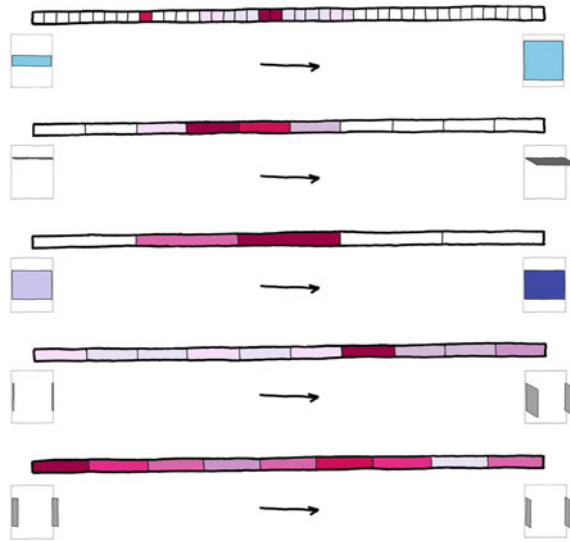
In this case we want to optimise the design of a façade module where we can vary the following parameters:

- (a) The height of the glazing (h_g), which defines the percentage of vision area. We want to test a wide range (from 20 to 80 %) and we split this range in 43 steps, in order to keep the research comprehensive.
- (b) The depth of the overhang (d_o), which can vary between 100 mm and 1 m at steps of 100 mm.
- (c) The type of glazing (g), which can be very transparent or dark: 5 different types of glass are considered.
- (d) The depth of the vertical fins (d_f), which can vary between 100 mm and 1 m at steps of 100 mm.
- (e) The inclination of the vertical fins (α), which varies between 10 and 90° at steps of 10°.

In this case our research space is made out of almost 200,000 possible options. Let's say that we want to run a single-objective optimisation, where the objective function is the annual amount of carbon emissions due to cooling, heating and artificial lighting. We are considering an office building in London and the façade we are looking at is facing west.

The outcome of the optimisation exercise is a series of combinations leading to very low carbon emissions, i.e. the optimum solution and the ones leading to an increase of less than 2 % in emissions. This means that the design team is provided with a huge number of options from which decisions are to be taken. The problem is that the most immediate way of looking at the results is a long table, which has all the answers, but it's very confusing and doesn't help that much. As I mentioned earlier, we need a clever way of presenting the results, in order to have some immediate answers that can drive the design process effectively.

We need to get some rules that we can follow in order to identify a façade design that meets the aspirations of the design team and which is, at the same time, performing properly from an energy point of view. The idea is to visualise the intensity of the different levels of each parameter: this will tell us how frequently a specific type of glass or a certain depth of the overhang is present within the series of good options identified by the optimisation. This can be done in the following way: let's visualise the different variables as bars, spanning between the two extreme levels we have considered. We can use different colours to indicate how frequently the levels of the different variables are present within the series of good options we want to focus on: the more intense the colour, the more frequently that specific variable's level occurs within the optimality region. The image below shows what I mean, applied to this specific example.



Frequency of the good options

The image above is telling us that:

- (a) There is a small range of glazed percentages within the optimality region, and neither very little nor a lot of vision area are beneficial;
- (b) The overhang is particularly beneficial when it is in the region of 400 mm/ 600 mm depth;
- (c) Only two glass types lead to good overall performance: the intermediate one and a slightly more transparent version;
- (d) Vertical fins are beneficial only when they are quite deep (at least 700 mm);
- (e) The inclination of vertical fins has a limited impact, since all the considered angles are present.

From the analysis of these results we can immediately understand what measures should be taken to end up with an energy efficient façade.

What this representation does not tell us is how the different variables need to be combined: what if we want to know the implications of selecting a specific level of one variable? For example, let's say that we want to understand how we can play with the geometry of the shading devices if the height of the vision area (i.e. h_g) is 2 m. Well, in this case the answer is not that immediate, because there are too many possible questions that can rise, and a 'static' representation of the results cannot satisfy all of them. We need to adopt an interactive way of selecting options. This can be done with a number of tools available in the market, for example interactive pdf can be adopted. We are not particularly interested in how to do this, but we want to focus on what sort of representation we need.

When we don't choose any level of the considered variables, the interactive tool will show us the same image shown before, and will declare the amount of 'available options', i.e. the number of combinations selected during the optimisation run. The corresponding screen shot will look like the image below:

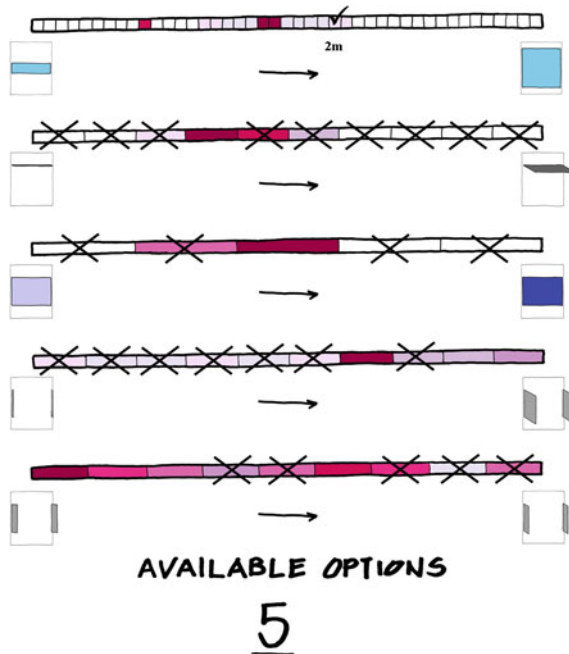


Available options

If we want to know what happens when the height of the glazing is 2 m, we click on the corresponding button and we'll have the view of the following image:

This shows immediately the implications of selecting a specific level: only a few buttons can still be clicked, and the number of available options has dramatically dropped. This has happened because we selected a level with a very low intensity.

The main advantage of this approach is that it does not refer to a specific type of analysis: we could have used it also if we had run a multi-objective optimisation with as many variables as needed.



Available options with 2m high glazing

In summary, we have seen how we can take advantage of software tools and algorithms to identify good solutions, within a huge amount of potential options and at a reasonable computational/time cost. We have also seen how the wide set of answers can be organised in a way that provides viable help to the members of the design team. In the examples above we have always referred to fitness functions based on performance/energy indicators. But this does not necessarily have to be the case. As I mentioned at the very beginning of this section, the first step to be done during an optimisation exercise is the definition of the fitness function, i.e. the scale we use to compare the different design options. This is not an easy task, but I think it is very good that the design team has to make an effort to think about how to assess whether one façade type (or, in general, a building option) is better than another: this exercise will generate a virtuous cycle for the whole design process. It will force everybody to state what they're after for the project, and to do so in a very explicit, honest way.

But this process, so dramatically honest, also requires a completely open mind-set for the designers. Because, once the objectives have been identified and properly defined, the answer we get may bring us to a solution which is very far from where we were expecting to end up. Let's imagine that the design team has worked out a solution to the design problem both in a 'traditional way' and by means of an optimisation exercise. It is possible that the two solutions coincide,

but it is infinitely more likely that they are very different. I guess there are three possible ways to deal with such a situation:

- The design team is very keen on the solution identified by means of a traditional design approach and doesn't accept to be driven by a computer (even if the computer has simply followed the rules set by the design team itself!). But, at the same time, there is the need to demonstrate, scientifically, that the identified solution is very rational and satisfies all the client's requirements in a very effective way. Therefore, the objective function is 'adjusted' in a way that an optimisation run identifies the same, preferred solution.
- The 'traditional' solution is considered more satisfactory because of some 'instinctive' criteria that the optimisation algorithm cannot perceive. Even so, the optimised solution is still taken into account and considered as a benchmark, against which the 'traditional' design needs to be compared to. It will then be clarified whether or not the improvements delivered by instinct can justify the overall lower value of the building.
- The optimised solution is deeply analysed and appraised with no prejudices due to the efforts spent for the production of the 'traditional' solution. With a complete open mind-set, the design team understands why the optimised design is better and buys it.

In principle, scenario c is the one that I think should be followed. Actually, in theory it would be better if the design team didn't develop a parallel solution in a traditional way, because in a way this will inevitably create a prejudice, which then will make it harder to analyse the optimised design. I think this is a very delicate issue to deal with: I'm fully aware that it is natural to follow the traditional route, especially during the phase when optimisation is not a common practice (yet, I'd say). And it could be beneficial, as long as it doesn't represent an intellectual barrier for the design team—and I'm very aware that this is something extremely difficult to avoid. But, let's go back to our three scenarios. As I said, situation c is the ideal one, because the final result is not just the realisation of the optimised design, but it is also a very educative process: the design team has critically understood the implications of the original decisions, which defined the objective function. And it is possible to compare the results of a fully comprehensive approach, with the ones of a traditional procedure, which we have grown up thinking it's the best one. I think this is something incredibly powerful, isn't it?

I find scenario b very interesting as well for two reasons. First of all it indicates a limitation of the optimised process, which cannot deal with some 'instinctive' criteria, and therefore can't find the actual optimum design. But, is this really the case? Or is it a matter of a mistaken definition of the objective function? This is something that we should tackle properly later on. The second reason why I find this scenario interesting is that also in this case there is a good educative process: the design team has analysed the situation in depth and there is an awareness of some mistakes or limitations in the way the objective function was defined. And there is a honest recognition of the limits of the proposed solution: the members of

the design team are aware that the proposed solution doesn't represent the best answer to the brief (represented by the criteria defining the objective function), but they know how far they are from the optimum.

What I find really disgraceful is what I described in scenario a, which, I'm afraid, is an extremely common situation. In this case the members of the design team don't accept the results of optimisation, but at the same time doesn't have the honesty to admit that and they tweak the rules until their design is scientifically proven to be the best one.

Backgrounding

In my opinion the worst and most detrimental scenario is b, because it confuses things, preferring vague 'instinctive criteria' to a logic design.

If we don't want to adopt optimisation in the architectural process, because we don't trust it or we are afraid that it might harness the design, diverting it from its ultimate aim, then it's more honest to dismiss it straight away, relying only on our traditional set of skills and ideas. It's OK to stick to what you know. But I also believe that the task of designing, either alone or in a team, is an ongoing negotiation. And design should try to bring to unity all the contributions provided by the different variables.

What fascinates me of the optimisation method is the neat distinction between conception and the decision-making stage. There's some sort of conceptual purism in this.

Being the description of the objectives, the design becomes a statement, a formula. We can follow different paths to apply this formula. Numerical analysis will help us choose the one that is probably the best.

With this I don't mean that the form will automatically create itself, in an act of self-production. No, it just remains suspended in a potential stage and will then find a solution during the decision-making process.

Nowadays there are many examples of science and technology contributions in architecture, as in all aspects of life. But I think it's evident that there's a gap between the possibilities offered by computer aided design and our present construction skills: graphic software tools give a scientific patina to our fantasies, but usually the aesthetics of the designs conceived with their help is automatically betrayed in the construction stage.

Parametric designs are beautiful and I'm waiting for the technology that will turn them into reality. On paper (or, better, on the screen) they are fascinating previews, allegories of the future, but once built they just highlight all the limitations of our materials science, and therein lies the rub.

The way scientific means are used nowadays is often just a formal statement.

By prefiguring post-Earth scenarios, computationalism seems to generate a new Nature, a nature that is supposed to be kind to men, but that ultimately might do perfectly well without them. Then, why don't we all pack and move elsewhere,

letting this self-produced architecture freely express itself on the planet? Because, what really looks out of place in the renderings of these computational designs is people: these tiny insects, these hundreds of ants going back and forth in buildings that look like huge sugary cakes.

In a moment of discouragement my professor once told me that architects put themselves into troubles from the moment they discarded classical orders. In fact, the syntax of architectural styles used to solve the problem of the form allowing architects to focus on the spatial elaboration of buildings.

The world, added my professor, needed a new Leon Battista Alberti! I've never fully understood what he meant, I just took it as a provocation detached from reality. Why should we wait for a new Leon Battista Alberti? There's no space for theories nowadays because sooner or later every theory is doomed to be proven wrong and thus to become useless.

Optimised design is not a theory but an approach, an operational framework or, better said, a *background*, that replaces the architectural as well as the technological syntax.

But on the foreground there must be an idea of space that comes from a specific desire. The stronger the initial idea of space, the more we can delegate its implementation.

In this sense I think optimisation might be an alternative to the compositional syntax, to what architectural styles used to be, a background in which syntax is left (temporarily) pending and the design can be accomplished regardless of the impositions of style and formal choices.

Today we find ourselves in front of the same tool, all of us at the same time sitting in front of the screen, as we produce and consume. When a tool is universally spread and shared by everybody, we can take its specific features for granted and finally start to compare the results, triggering that personal challenge that I mentioned before.

And I'm not saying this because I want at all costs to find a revolutionary value in the spreading of computational design.

Everyone says that an epochal change is happening, but sometimes I think that it's just a great sleight of hand, that delays the suspense and excites our anticipation. Just as when we expect the release of a new computer, that, before being a technological product is the promise of a new technological life.

Every day we are surrounded by countless signs that testify the ongoing change. But I suspect this change is slower than we think. Or maybe there's no change at all. Maybe we just want to convince ourselves that we are right in the middle of a cultural revolution.

I have this feeling because for years there's been much talk of technology, without really talking of technology. The technological society has not been fully accomplished yet. It will probably take over when we will acknowledge technology not as a mere means but as foundational feature of our world.

I believe we might take a substantial step in this direction by giving (or returning) architecture a specifically technical content. And optimisation could

help us, because it requires breaking the design into quantitative factors and features that are easily and univocally describable.

On the basis of decisions that are not stylistic but spatial, optimisation provides a technical and impersonal answer, the possibility of a form.

This is not a functionalist approach, but rather the celebration of architecture as the art of space, freed from any formal imperative.

The objective function sums up the design in a non-visual representation. This immaterial formulation, though, is very powerful because it already contains in embryo its outcome what will or might be realised. A power that's obviously proportional to the quality and accuracy of the formulation: the stronger the ability to desire, the more successful the final result.

For this reason, I would say that the objective function defines the successful building.

Evolutionary Optimisation of Façade Design

A New Approach for the Design of Building Envelopes

Zemella, G.; Faraguna, A.

2014, VIII, 121 p. 60 illus., 50 illus. in color., Hardcover

ISBN: 978-1-4471-5651-2