G. de Roo, C. Yamu, and C. Zuidema (Editors) **Handbook on Planning and Complexity**, Edward Elgar, Cheltenham, Glos UK, 302-317, 2020.

15. Complexity in design: optimal location through spatial averaging

Michael Batty

1. INTRODUCTION: THE PROBLEM OF DESIGN

As the digital age continues to invade and pervade all aspects of our lives and the places that we inhabit, the idea that our world is becoming ever more 'computable' is fast generating new notions about how we can act intelligently in organizing our cities. Traditionally our abilities to build cities have been based on the way we mobilize our native intelligence to fashion good design and architecture, but, increasingly, we are using artificial intelligence (AI) built from procedures and protocols embodied in algorithms to simulate and optimize our designs for future cities. In fact, for the last 60 or 70 years, ever since the digital computer was invented, we have been using these machines to enable us to make conditional predictions about the future as well generate new designs for that future in the quest to build a better quality of urban life for all.

In the last two decades, however, computation has become embedded in the very fabric of the city that we are intent on explaining and predicting. In short, as computers have got ever more powerful and ever smaller to the point where all of us own very powerful handheld devices that we can use to manipulate as well as communicate with one another, the very computers we have been using to understand and design are being embedded in the reality that we are intent on changing through design. Although we have used computers in this way since their very inception, they have remained separate from the objects that we are concerned with until now, when they are becoming part of those objects. This is changing our perspectives on cities in quite dramatic terms. The development of computation within the city itself is now being referred to as the 'smart city'. Our traditional concerns dealing with the processes of planning and design using computers are now able to utilize the products of 'smart' computation in the city in the form of new data – very often 'big data' – as well as generating new patterns of behaviours, new media and a host of other digital features that are changing and enriching urban living.

Into this mix of computers embedded in cities and our use of them to manage, control and design such mixes come some very new perspectives on how we might think of an augmented computational intelligence for building a new science of cities (Batty, 2013). Complexity theory underpins much of this thinking insofar as we articulate the city as a complex system and we are beginning to develop algorithms for steering and managing the city so that we are able to optimize various forms and functions that imply greater efficiency and equity in spatial location. In this chapter, we will sketch the way in which we might improve the planning and design process using new forms of intelligence that involve identifying the many factors that determine how and where urban development takes place. Currently most design processes are based on using factors that we consider are important to determining a good or even best solution to the problem in hand so that

we can generate a solution that gives appropriate weight to the importance and relevance of these factors. Design in this sense then is the process of reconciling and synthesizing how a range of disparate factors might be integrated to produce a solution which is optimal. The way the solution is optimal needs to be measured against the factors that are identified as requirements important in forming the solution. This is a very general description of a design problem and, to give it some substance, let us illustrate it using a simple example.

Imagine that we have a simple pattern – a solid circle – a circular city – that represents the location in a rectangular space for the development of some land for urban use. In the wider space, we consider that it is only this central circular area that is optimal for development. In the past, visionaries intent on planning the ideal city simply proposed such geometric designs with little recourse to a tightly reasoned logic from which such a solution might emerge. Today, we would expect such a logic to be made explicit for only then would we agree to such a solution which we would expect to be subject to discussion, debate and controversy. However, let us assume in this case, that there are four factors or requirements that suggest development might be in different positions within this space. We will also assume that the designer is mandated to come up with a solution to the problem that takes these into account. Now, in our example, the four factors in question suggest that development might be in any one of the four quadrants of the circle. In short, the first factor suggests the best place for development is in the north-west quadrant, the second factor in the south-west quadrant, the third in the south-east quadrant and the fourth in the north-east quadrant. We illustrate a final solution and the four factors that we assume will affect it in Figure 15.1.

The way we have arrayed these four factors in the figure is such that a very obvious solution which gives us the circular development immediately stands out. All we need to do is add the factors together. By the term 'add', we mean that we formulate the rules of addition in this case as follows. Take the first factor and array it against the second. If the area of the first factor is not developed – we could number this as 0 – and if the same area of the second factor is 0, then the addition is 0. If, however, the area in the first factor is developed where we number the development 1, but the area in the second factor is not

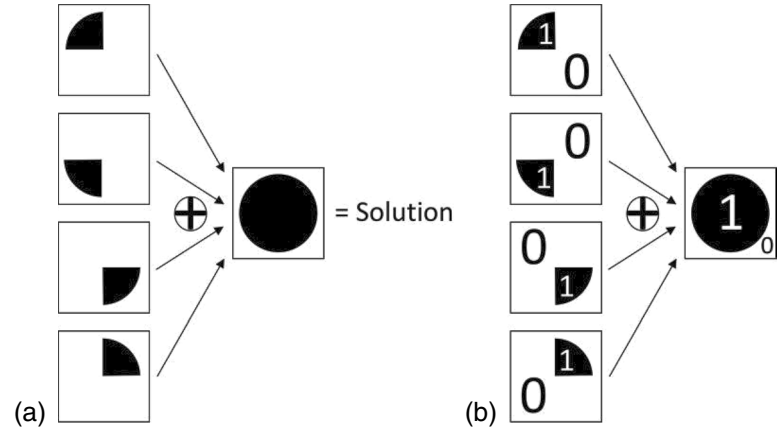


Figure 15.1 A simple design problem

developed and numbered 0, then the addition gives us 1, that is 1 + 0 = 1. If this were the other way around, the first not developed but the second developed, the result is still 0, that is 0 + 1 = 1. If we then take this partial solution, factor 1 added to factor 2 gives us half a circle. Continuing by adding factor 3 to 1 and 2 so far, we get three quarters of the circle developed, and then when we add the last quarter, we get the complete circle. So the solution to our design problem is simply adding the factors together.

To get this far, we have to make some quite severe assumptions. First, we do not worry about the fact that an area is undeveloped according to one factor and that it is developed according to another, for when we add them together, we assume that the undeveloped area is not in conflict with the development. If it is, then there is no solution by this kind of addition. If we call development 1 and non-development −1 then what we get when we add all four factors is an empty space. Development cancels out non-development. In short, we have defined a synthesis of factors as a process of addition: where there is something to add, then this is equivalent to development taking place. Figure 15.1 makes this quite clear. And what is important to get this very simple geometric solution is the fact that the geometry of each individual piece fits the total picture like a jig saw. Returning to Figure 15.1, then in (a) we see the original problem and what happens if we add the four pictures and in (b) we can see what happens when we label development 1 and non-development 0. Noting that when we add it up in this way, the only thing that matters is whether the area of development is called 1, for non-development (= 0) is neutral to the addition process. If it is not, everything about this process of synthesis would be different. It is clear that the process of addition is one of logic akin to Boolean algebra and instead of using the addition sign we use the operator Å.

Let us continue with this problem. There are many ways of synthesizing these four patterns of development to make one. We are assuming of course that the process of addition is such that when we add individual patterns – factors – each of which implies a best location, that the process gives us an overall best pattern. In this case four factors which are different but imply individual best locations are combined to one. The process of addition which in Figure 1 is a set of four additions implies a simple combination, but there could be many other ways of combining them in that we might add factor 1 to factor 2 and then the composite factor 1–2 to factor 3 to get 1–2–3 and then finally to factor 4, thus giving us the output. In fact in our example this would give us the same result so this order would not make any difference. But we could weight each of the factors, that is, give them different degrees of importance in the addition. We might also assume that if the shapes do not fit in this close jigsaw, then we could get overlapping patterns of different intensity.

Let us illustrate this by specifying different shapes for each of our four factors, which are still defined in the same square space or map. Adding two factors where the shapes are irregular and overlapping, using a somewhat different additive logic, then one would be adding 1 + 1 where the result would be 2, and the range of variation over the shapes would be from 0 to 2. One might then assume that 2 was better than 1 and this variation introduces different degrees of optimality into the solution. Gradations of optimality are produced by adding the four factors where one might have any combination of development 1s and non-development 0s. With four factors specified as 0 or 1, we then have combinations like 1–0–0–0, 1–1–0–0, 1–1–1–0, 1–1–1–1, 0–1–0–0 and so on. In total there are $2^4 = 16$ possibilities and four gradations 1,2,3, and 4.

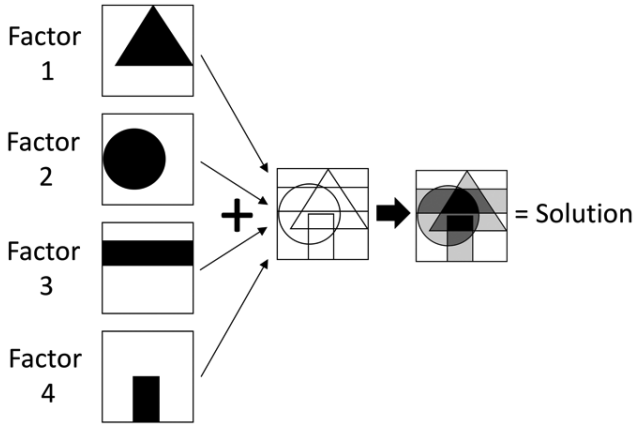


Figure 15.2 Elaborating the design: overlapping requirements

We have modified our simple design problem to take this into account and the simplicity of the first solution is no longer the result of this synthesis. We show the result in Figure 15.2 where there are some very obvious extensions to this logic. We have already noted that, if development and non-development are of equal weight, the process is one where good and bad rather than good or neutral is involved. We might also introduce gradations in the degree of optimality for each factor and instead of each factor being represented in terms of 1 developed and 0 non-developed there might be a range between 1 and 0 or over some predetermined and comparable interval. The factors need to be commensurate, of course. There are many such elaborations and we will show some in the following section.

2. DESIGN AS SPATIAL AVERAGING

To some extent, the process we have outlined is one of making compromises in the sense of taking an average of the different factors or requirements that the design must take account of. If we weight the elements of each factor differentially, then we imply different degrees of importance to each requirement in two ways: first, spatially we can give each factor a degree to which development might take place, ordering this over a scale ~~from~~ example from 0 to 1; and second, in terms of the relative weighting of each factor, then each factor will have a different weight in the solution. Without worrying too much about the first feature based on differential development potential, let us illustrate the second one with different weights. Our example in Figures 15.1 and 15.2 can thus be elaborated as follows. Assume that the arrows in these figures indicate how each factor is to be combined into a common solution and now assume that we can give weights to each of these additions. Let us now define each factor as F_i^k, where k defines the factor in question and i defines the area of the map which in our simple example is 0 or 1. We can of course specify this over a range if we consider that each area has a degree of development potential. We now give a weight to each factor and we define this as w_i^k. The simplest weighted solution is then the average of these factors and we thus define the solution F_i as

$$F_i = \sum_k w_i^k F_i^k \qquad \text{where} \qquad \sum_i w_i^k = 1 \tag{15.1}$$

We show this process in Figure 15.3(a). In fact, all we are doing is adding a weight to each arrow, which shows how we combine the four factors. We will normalize the weights to sum to 1 over all the factors but, as long as the weights are commensurate meaning they refer to the same scale, normalization is not essential.

Now this is still quite a simple method of averaging, equivalent to assuming that every factor is combined in one step with every other to produce the solution. However, imagine that each factor has a specific set of weights that relate to all other factors and these weights will differ for each factor. Then for any factor k, the set of weights for that factor relative to every other factor j are given as w_i^{kj}. For a particular factor k, we now combine the weights as

$$F_i^k(t+1) = \sum_j w_i^{kj} F_i^j(t), \sum_j w_i^{kj} = 1 \tag{15.2}$$

and it is clear that, all other things being equal, then $F_i^k(t+1) \neq F_i^j(t+1)$, $j \neq k$. That is, as the weights are individualized for each factor, this process of averaging each one against all others will lead at this first stage to a set of sub-solutions, one for each factor, which are not equal to one another. Note that we have appended a temporal index t to each factor implying that this is a sub-solution to the problem and, using this augmented weighting, a unique solution is usually not possible in one shot, in one temporal iteration. In Figure 15.3(b), we show this process where the factors are combined to produce a sub-solution for each factor, rather than one solution for all factors. We do not show the actual sub-solutions but only indicate the outline of how the original factors overlap as we have not actually worked out the final solution. The reader could easily have a go at doing this with such a simple set of factors.

In equation (15.2) and Figure 15.3(b), it is clear that the first set of sub-solutions $F_i^k(t+1)$ can then be fed back as ~~through~~ they were the original factors and passed through the network of weights once again. We can keep on doing this until the factors begin to converge towards a single solution. You may ask why we can guarantee that there is a single solution. The answer is easy enough to see – it is because every factor influences every other factor but the weights determine what this influence will be. Ultimately all the differences between the factors are ironed out, and the sub-solutions become the one solution. Now if all the weights were identically distributed, that is, if $w_i^{kj} = w_i^{zj}$, the problem would reduce to the one we examined in Figure 15.2. If the weights were all the same, that is as in the case of our four factor problem $w_i^{kj} = w_i^{zj} = w_i^k = w_i^j = 1/4$, again we would be back at the simplest design problem where each factor has the same weight in the simplest of averaging.

Thus Figure 15.3(b) implies that the key issue in design of this kind is the set of weights that are specified. We have elaborated this figure to show how we can feed-forward the sub-solutions implying that these ultimately converge in one solution. In fact, what in essence we are doing is feeding back the sub-solutions using the same weights. We can see this formally in equation (15.2). If we substitute $F_i^k(t+1)$ for $F_i^k(t)$ on the right-hand side of (15.2), we get $F_i^k(t+2) = \sum_j w_i^{kj} F_i^j(t+1)$, $j \neq k$, which we can simplify and generalize over m temporal iterations or cycles to

$$F_i^k(t+m) = \sum_j w_i^{kj} F_i^j(t+m-1) = \sum_j w_i^{kj}(m) F_i^j(t) \tag{15.3}$$

Note that all the action is now loaded onto the weights $w_i^{kj}(nm)$, which change at each time period. A little bit of algebra lets us figure out how we change the weights from one iteration to another, that is from $(t+m-1)$ *to* $(t+m)$ and the following formula specifies this operation.

$$w_i^{kj}(m) = \sum_z w_i^{kz} w_i^{zj}(m-1) \tag{15.4}$$

This process of averaging is linear and additive, meaning that the operations we have specified for combining factors are the simplest possible. Assuming all the factors are connected to all others, meaning that all the weights are positive and normalized to sum to 1 with respect to the relationships between one factor and every other, that is $\sum_j w_i^{kj} = 1$, then convergence will take place and we are then able to derive a unique solution $F_i^k = F_i^k(t+m)$ where we assume that m is very large and is the limit of the process. In Figure 15.3(c), we show how we can represent the feedback as an iterative loop in the sequence of generating sub-solutions and, in this sense, our pictorial illustration begins to suggest the kind of iterative procedures that are characteristic of machine learning. We will not elaborate this extension yet but first we will note other features of the iterative process. We will then generalize this scheme of things to ideas about opinion formation and consensus building in social networks, which will cast our design problem into a social context.

There is one last step we need to make in thinking of design as successive averaging and that involves the formal process of additive iteration. We can think of the weighting matrix w_i^{kj} as a probability matrix W. As we normalize each row of this matrix to sum to 1, this defines W as a transition probability matrix which is applied to each cell of the map i. We can drop the cell notation in fact because all this relates to is the fact that we characterize the map as being divided into ~~npixels~~ or cells and the weights are applicable to each one. In fact, the weights could be different for different cells but we do not invoke this extension here. We can now write the iterative process as follows in matrix notation as $F(t+m) = WF(t+m-1)$ where $F(t+m)$, $F(t+m-1)$ are $1 \times n$ vectors and W is an nxn matrix where n is the number of factors. This defines a Markov chain where, as the temporal averaging continues, the equation can be written as $F(t+m) = W^m F(t)$. In the limit of m, W^m converges to a steady state stochastic matrix. Details of this process are spelt out in *The New Science of Cities* (Batty, 2013). Readers who are unfamiliar with the algebra need only note that the process converges. A little thought reveals that as everything affects everything else and is continually averaged, the weighting system establishes the relative importance of the factors that define the design problem.

3. A NETWORK VIEW OF DESIGN

If you examine Figures 15.3(b) and (c), you will see that as every factor has a unique weighting with respect to every other factor, this establishes a network where everything is connected to everything else. You can see that there are n^2 links between the four factors – 16 in all – and we will graph these links as a simple network in the manner that

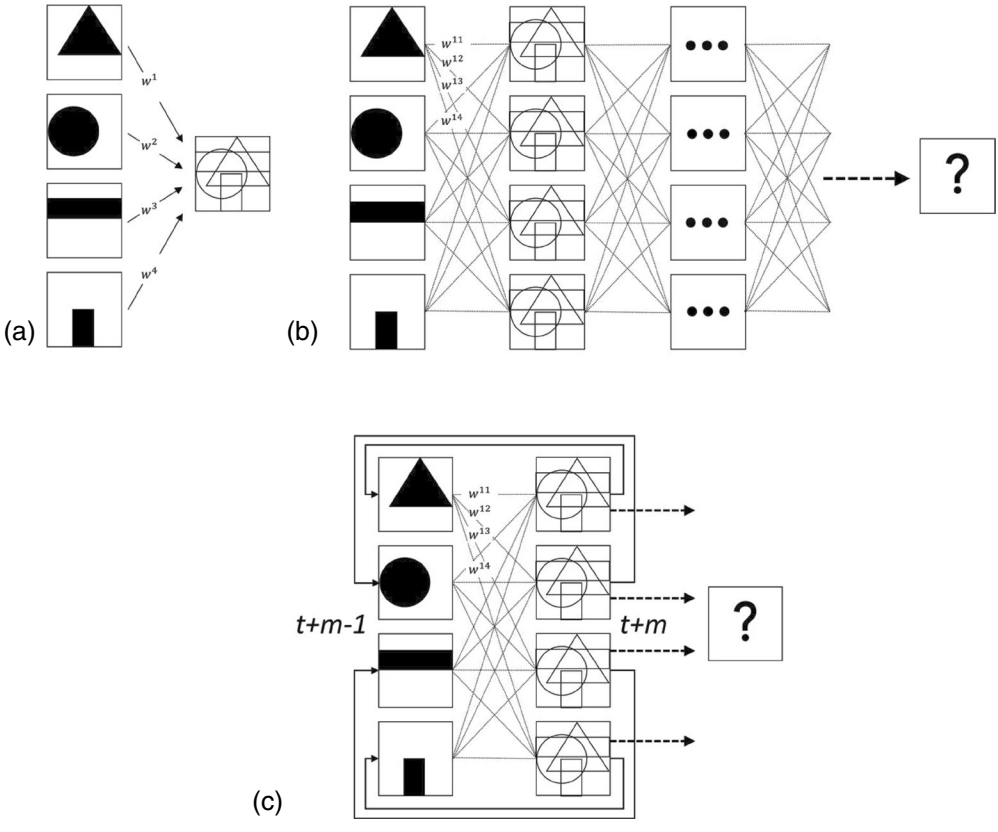


Figure 15.3 Iterative averaging: (a) simple weighting averaging; (b) successive averaging from distributed factor weights; (c) successive averaging illustrated as a type of back propagation

we show in Figure 15.4(a). We can interpret this network in the following way. If each factor is regarded as being a sub-solution to the problem, the designer wishes to use each sub-solution to produce one that is nearer the optimal and as yet unknown solution. So if the designer then compares the sub-solution to all those sub-solutions which are linked to it as shown in the network, the designer produces a new sub-solution by combining the sub-solution in question with all those to which it is linked. This process establishes a sub-solution according to the weights and, if the weights are all equal, this is still our simple arithmetic average. The rationale for combination is that each of the factors has a degree of importance and this should be used to weight the sub-solution. Once all sub-solutions have been produced, the process occurs again with the sub-solutions producing new sub-sub-solutions, and so on. The sub-solutions thus get ever closer and closer to the final result. One can see this network problem as one of communicating factors with respect to all others – pooling them if you like. If they are regarded as opinions about how important the designer considers the particular factors, this might be a process of averaging or pooling the opinions and thus the network might even be considered as one where

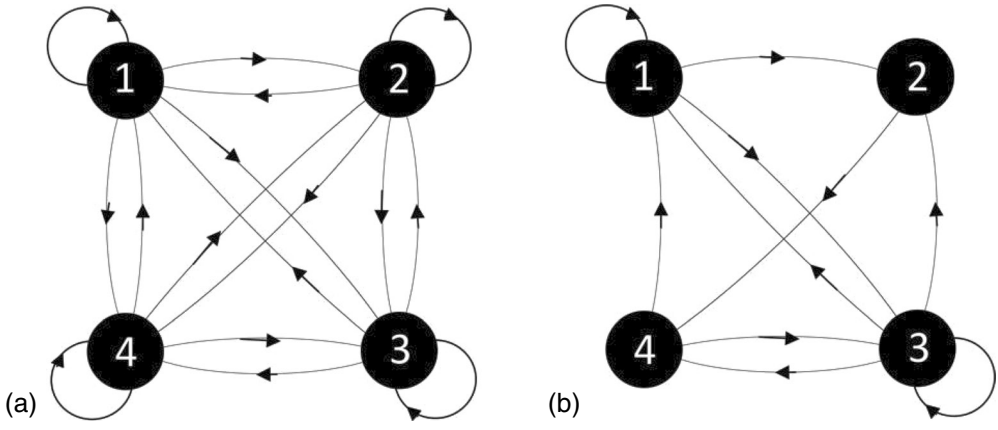


Figure 15.4 The simple design problem as (a) a completely connected communications network, and (b) a strongly connected network

the links are based on statistical correlations between the factors. In short, the weights might be how close one factor is to another and the pooling takes place according to how important these related factors are. There are many ways of thinking about this but, in essence, the process is one of successively reducing differences between what the factors as sub-solutions imply about the best location, so that ultimately a single best location emerges from n possibilities, the number of factors.

We might also think of the network as not one where the designer is in control but where the factors are individuals or experts with their own view about the best solution to the problem. Each factor k, then, is a map of the best location from the point of view of that person k. The problem is thus one of an idealized process of coming to some solution through the pooling of these different solutions. In this interpretation, we need to consider the network in more detail. We have shown a network in our simple example where every factor is linked to every other and in this case it is clear that every factor can communicate with every other and that ultimately all the factors will influence each other with respect to their relative weights. There will always be a solution to the problem. But if one factor is not linked to any of the others or if a factor is only linked to the others but is not linked in the reverse direction, the solutions are more complex. In the case where one or more factors do not link with one another, there may be two or more solutions. In the case where one factor is linked to all others and influences them, but is not influenced in the reverse direction, the ultimate solution comes to be the initial best location for that factor. In short, the connectivity of the network (as well as the weights for each link) makes an enormous difference to the final solution. What we show in Figure 15.4(b) is the original 4×4 network where, of the original 16 links, we have deleted six. In fact, the network is still strongly connected, meaning that one can trace links from any one of the four factors – the four nodes – to every other node and thus everything still influences everything else in the limit. If there was some barrier to influence due to the lack of network links, the problem would not be soluble or rather more than one solution would be the result.

There are several disciplinary fields that have thrown up communications problems in the manner we have illustrated here through concepts such as design as averaging,

networks and social power, opinion pooling, and collective action, and it is worth noting these. First, the idea of formulating the locational design problem – which is essentially based on the notion that we want to find the best location for some land use or facility within a wider space such as a region or a town or even a smaller scale space such as a residential neighbourhood – goes back more than half a century to the development of physical and landscape planning. The idea of building requirements as sub-solutions to the location problem was first implemented in methods of mapping such solutions and overlaying them to produce some synthetic average. These methods were quite widely used from the 1950s onwards but were first developed 50 years before that in the development of park design by Olmsted ~~among~~ others (Steinitz, Parker and Jordan, 1976). In the 1960s, Alexander (1964) developed more formal varieties of these methods based on hierarchical averaging while McHarg (1969) used overlay methods in landscape location problems where the requirements were mainly physical involving flooding, topography, and so on. The author developed the network design methods sketched above and linked these ideas to social networks. This built on theories of social power (French, 1956) in which factors were identified with actors involved in reaching a consensus – the design solution – using the communications methods noted above (Batty, 1971). Lately, Steinitz (2012) has automated traditional methods of overlay analysis within GIS, which he terms 'geodesign'.

These methods have been exploited in various contexts by Harary (1959), De Groot (1974), and Kelly (1981) and have been further extended by social network theorists such as Friedkin (1998). More recently, opinion pooling based on similar ideas has been developed in computer science through multi-agent modelling relating to opinion changes associated with notions about flocking and herding (Blondel et al., 2005). There are various extensions to such network analysis using bipartite graphs linking these notions to ideas about collective action (Coleman, 1994), which in turn relate such problems to indices of social power providing the basis for a social interpretation of weighting schemes in design problems. The author has also extended these ideas to site design where the weighting of factors pertains to ownership, finance, and use (Batty, 2016).

Before we explore how we can reformulate the design problem as one which relates to how actual development takes place, we will illustrate the method of successive averaging that is based on the procedures contained in equations (15.3) and (15.4). This example is also presented in Batty (2013) but we will repeat it here, as it crystallizes the essence of the method of successive averaging. The problem is one of searching for land for urban expansion in a small town in Northern England which has a population of some 50,000 in the town itself and some 20,000 in the map area around the town in which we are searching for appropriate development land. We identify 12 factors that we consider critical to the search for the best land for urban development and these we will map as areas suitable and not suitable for development. We have in fact exercised a hard boundary around areas judged to be suitable and not suitable although we could have specified these more rigorously as different degrees of development potential. These factors range from physical constraints pertaining to topographic costs to social and economic potentials based on accessibility to retailing and residential development. The list is: accessibility to existing urban services, costs of spatial congestion, accessibility to recreational amenities, areas of acceptable micro-climate, areas of water catchment and poor drainage, institutional constraints imposed by government, accessibility to external urban markets, subsidence and extensive industrial pollution, areas of suitable topography, rural amenity areas,

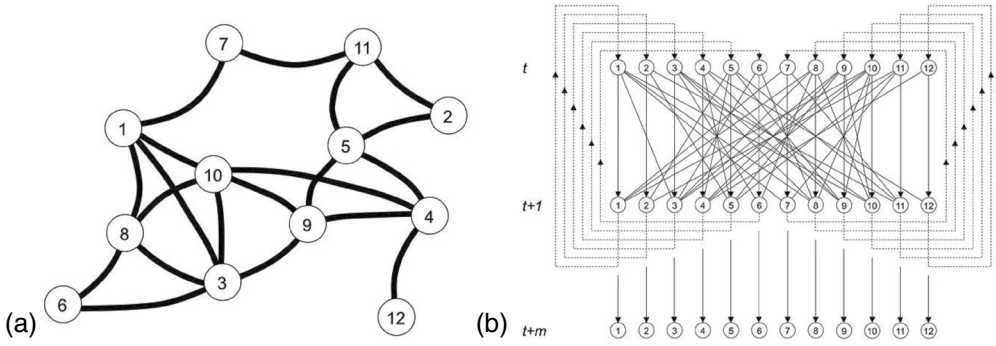


Figure 15.5 Structure of the urban development design problem

historic urban areas, and conservation of high quality agricultural quality. These factors are mapped below in Figure 15.6 where we show them as the first inputs or sub-solutions to the successive averaging, but before we explain this, we need to specify the linkages between the different factors.

In Figure 15.5(a), we have specified the direct links between each factor. These links are in fact directional, although we represent them as symmetric and we do not show the self-links, as every factor is linked to itself. The rationale for linking any two factors is based on the notion that, if you develop land for urban use based solely on the first factor, this makes it highly likely that you would develop similar land based on the second factor. In short, if the two factors reinforce themselves, we specify a link which gives their relationship some importance to the design. We are then in a position to use this network to successively average all the factors in the manner that we illustrate in Figure 15.5(b). To give an example, if we take factor 6, we produce a new factor 6 on the first iteration as a simple weighted average of factors 3 and 8 to which 6 is linked, not forgetting the self-link for factor 6 itself. You can read this directly off the graph in Figure 15.5(a).

In Figure 15.5(b), we use the network to show how we can produce each iteration of the weighted factors. Note that our network is now the other way around from that shown in Figures 15.3(b) and (c) with the progression from stage to stage being vertical not horizontal. This is merely a matter of presentation. Now if we specify the land for development in each factor as 1 and for non-development as 0, we could use Figure 15.5(b) to compute the maps through several iterations of the process. In fact, this is how we produce the sub-solutions which we show in Figure 15.6 for a series of stages. At each stage we strengthen the sub-solution by slicing it to reinforce the 1–0 distinction between development and non-development. It is clear that the process converges and, within 10 iterations, the factors have more or less converged to one stable solution. In fact, in a real design problem, this formal structure would most likely be used as a process for considering how different factors need to be resolved towards a solution. In fact, where there is considerable uncertainty about the factors and a good deal of intuition involved in their averaging, the factors might be strengthened at each stage. The method is not so different from Alexander's (1964) method for resolving conflicting factor maps, which is best demonstrated by Alexander and Manheim (1962) for a highway location problem, similar to that discussed by McHarg (1969). In fact, Alexander argues that this kind of process

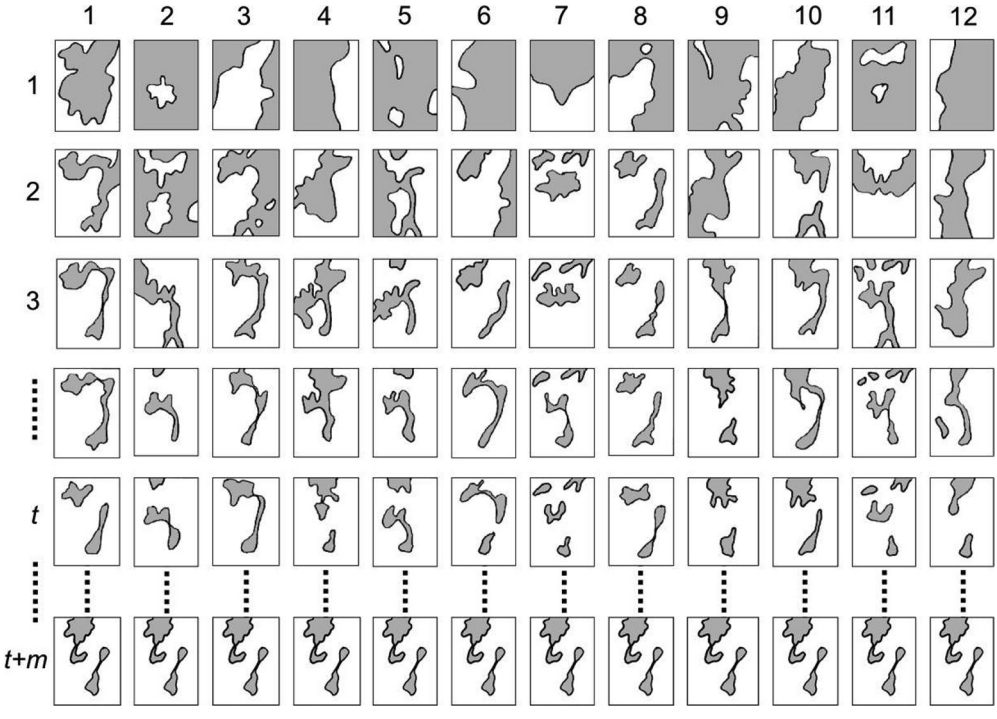


Figure 15.6 Averaging of the 12 initial factors through successive synthesis

gives the designer an opportunity to think hard about what is being compromised at each stage of the averaging and opens up the possibility of adding additional information, thus softening quite markedly the formal numerical process that is implicit in the method.

4. THE INTELLIGENCE IN URBAN DESIGN

In the development of computational methods in urban planning, we make a distinction between how the city system evolves and develops, and the way we as planners and designers tackle the problems and realize the potential associated with making the city better, more liveable, more equitable and more efficient. In short, we have models of the city which are our focus of interest in contrast to models of the way we go about solving problems associated with the city which is the focus of our design. In some respects, the most consistent approach would be one where the way the city evolves is closely mapped onto the methods we use for resolving urban problems associated with the city's development. But all too frequently there is a strong mismatch between the way the city evolves and the way in which the designs and plans we have been dealing with here are constructed.

To an extent, the approach we are advocating here, which is one of systematic design, is intelligent in that it is consistent and considered and based on a rational decision-making process. We identify a series of factors which are clearly relevant to the solution and we devise a solution which optimizes the requirements that are implicit in the factors we

define. However, it is unlikely that the actual process of development would lead to the kind of design solution that we have devised. In short, what we need is to do is to figure out what kind of process actually leads to different kinds of urban development and then possibly adjust the real process to bring it closer to the design. This is quite a different activity from design: it is one of science, the explanation of what we observe rather than the development of something new. In design, we start with requirements as factors, weight them in various ways and devise a solution. Operating in the opposite direction, we wish to define a series of factors and weights that give rise to a certain pattern of development. In fact, it is unlikely that there is any computational process that would let us identify a series of relevant factors other than the setting up of hypotheses about how development occurs from our own particular knowledge of the development process. But what we might be able to do is figure out the set of weights that might lead to a certain sort of development using techniques which operate in a similar fashion to those we have used to identify an optimal design. In short, in terms of our original specification of the design problem, we now know the form of the urban development as we observe this as F_i, we also hypothesize that factors F_i^k, determine what we observe, and the problem is then one of finding the weights W_i^{kj}, which will generate development as closely as possible to that we observe.

In other words, in design, we identify requirements that any development needs to meet. These conflict and we need to figure out how important they are to one another and then successively average them to produce a composite and hopefully best solution. In contrast, in examining actual development, we first identify a series of factors that might appear to influence the actual development, and then we devise a procedure that weights these factors in such a way that it leads to the development we observe. This is a similar process to design but in design we do not know the outcome but we do have consistent ways of showing how the factors can be weighted to produce the best solution. In examining actual urban development, we have some sense of the factors but do know the actual outcome. We want to determine the weights that pertain to each factor so that the factors, when combined, explain the outcome. One obvious way of approaching this is to see how close the factors are to the outcome and adjust the weights to enable this to take place. In this sense, we are introducing some intelligence into the process and, in the current terminology, the techniques for doing this are referred to as artificial.

Let us review what we have just said. In design, we start with key factors or requirements, we weight them in various ways and we then produce a solution – a design – that enables these factors to be combined according to the weights we specify. In real development, we also start with key factors, but we do not know how to weight these so we start with an arbitrary set and see how easily the factors can be combined to produce the known outcome. At the beginning of the process, there will be a substantial mismatch, but the weights are gradually adjusted so that eventually the outcome can be explained by combining the factors using the weights that we evolve during the process. In short, in design we continually alter the solution until it becomes consistent with the input factors while, in real development, we keep the outcome fixed and gradually alter the weights until they let the inputs combine to produce the observed outcome. These are quite similar processes but before we continue we need to demonstrate a simple example. In fact, we will take use our original problem to demonstrate how we can explain a known outcome in this way.

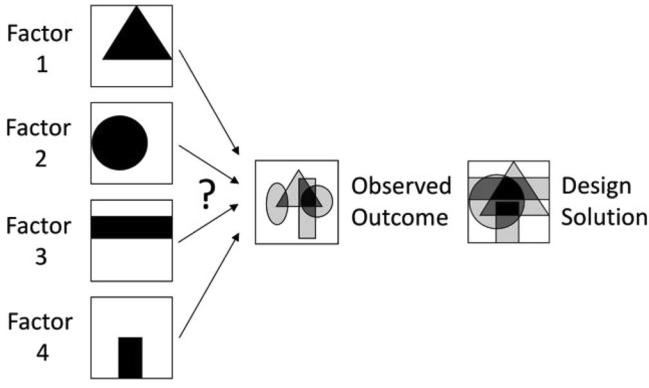


Figure 15.7 An observed outcome compared to a designed outcome

Let us go back to our simple example as we conceived it in Figure 15.2. We have four factors and this time we have an outcome that we observe, not a design that we generate. The outcome that we observe is the result of a very large number of processes and decisions that define how the city functions and evolves. We assume that our four factors are the key ones but we note that these may not be enough, nor may they be the correct ones and thus the process we invoke is of course limited by these assumptions. By changing these, we can improve the explanation. The outcome that we picture in Figure 15.7 is by no means the same as the design, but we have constructed it so that there are some elements in it that reflect the geometry of some of the factors. The idea is that we choose the weights to combine the factors so that they come as close as possible to the observed outcome. We know of course from our earlier discussion that the optimal design also shown in Figure 15.7 can be produced directly from an equally weighted combination of the factors and, although we cannot show that this is unique, it is likely to be the best combination to produce the design. It is unlikely, however, that any simple combination of weights in Figure 15.7 will lead to the observed outcome and thus we require a much more elaborate procedure for determining an appropriate set of weights as well as a formal procedure for doing this.

What we need to do is introduce another layer of sub-solutions in the same way we built a network for the design problem. In fact, with only four factors, it is likely that we might need to build more than one layer – in fact we can introduce as many such hidden layers as we like that enable different weights to be evolved so that the inputs come as close as possible to the output. We are not intending to demonstrate the result of determining actual weights for this example, as the process would take too long to illustrate, but we will sketch the way this procedure works. First the inputs are combined using an arbitrary weighting and sub-solutions at the $t + 1$ layer are determined. These are then compared against the observed output and on the basis of the mismatch the weights just used are recomputed and the sub-solutions are then combined again using these weights. In the simple net shown in Figure 15.8, this procedure with a simple feed-forward, then feedback, is not likely to be enough to determine sub-solutions that ultimately match the observed outcome. Thus let us introduce a second hidden layer – it is called hidden because it does not have any interpretative meaning when it is introduced, for all it is doing is acting as

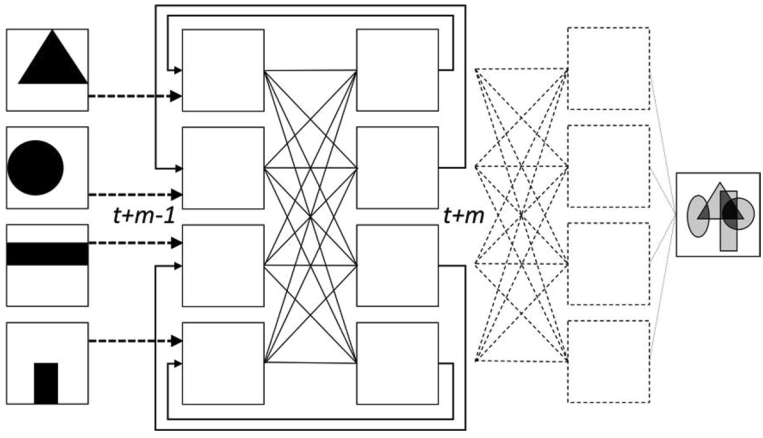


Figure 15.8 An artificial neural net

a way of elaborating the weights. We thus show the augmented network with the second hidden layer as well in Figure 15.8 by the broken-dashed lines. Clearly the complexity of the net can be increased indefinitely by adding hidden layers in this manner.

We can give a rough sketch of the formal mechanisms that are used to solve the neural net that we imply in Figure 15.8. Our formal model can now be stated as follows in analogy to equation (15.3).

$$F_i^k(t+m) = \sum_j w_i^{kj}(t+m-1)F_i^j(t+m-1) \tag{15.5}$$

Now we do not know the weights $w_i^{kj}(t+m-1)$ but we do know the initial factors F_i^j and the observed outcome F_i. For any set of weights, starting with the factors F_i^j, we compute the outcome $F_i^k(t+m)$ and compare this to the observed outcome F_i to see how close this is. We then compute the error $\varepsilon_i^k(t+m) = f(F_i - F_i^k(t+m))$ and use this to change the weights for the next iteration when we feedback the computed outcome so far for the factors as implied by equation (15.5). The weights are chosen so that they move the sub-solutions closer to the observed outcome and in general $w_i^{kj}(t+m) \sim 1/\varepsilon_i^k(t+m)$. In other words, the bigger the error the smaller the weight in the next round, ensuring of course that the weights still sum to 1. Unless we have a specific example, this is simply a generic indication of how we might begin to change the weights. There are many schemes for doing this and in general this stage is referred to as back propagation.

This indeed is close to what AI calls a neural net, so called because its original inspiration came from work 70 or so years ago on attempting to simulate the network of neurons and synapses that make up the dendrites of the brain (McCulloch and Pitts, 1943). There are many elaborations of such nets so that the inputs might ultimately generate the outputs. For example, we could add a pulse at every iteration which was of a standard identical form that would ensure that the net reproduced certain constrained elements that were always present in the observed outcome or met predetermined thresholds. But of more import is the way in which we are able to combine inputs at any stage. In this chapter, we have introduced extremely simple additive weighting. In fact, in neural nets used for

pattern recognition there can be very different functions which act to produce different responses – usually non-linear rather than linear – for different stimuli. There can be leads and lags in successive iterations, and there can be many different methods of enabling the mismatch between inputs and the observed outcome to be handled using different forms of back propagation – feedback – which can be likened to methods of searching a solution space, which is akin to a problem of optimization.

5. THE KEY CHALLENGES

One major issue is the choice of the inputs. Clearly if not enough inputs are chosen, there is no way a specific outcome could be trained. For example, in Figures 15.7 or 15.8, if we only had one input different from the output, it is obvious there is no way a particular output could occur. And then we might define far too many inputs and thus many solutions might be possible and, in this case, the net is over-specified. So far in urban planning, there are very few applications of these kinds of method. Design rather than science has predominated, but there is now the prospect that we might take many different cities with different patterns of urban development and attempt to define different inputs that give rise to these patterns. In this way we might be able to explore the process by which certain patterns emerge, and, in this context, generalize across different urban forms and functions. In some respects, the application of AI techniques such as these is largely dependent on choosing good examples. In fact, urban development patterns might be far too complex for neural nets and machine learning to make many advances in terms of their explanation, largely because such tools best work on highly routinized patterns such as regular purchasing behaviour, and physically deterministic problems such as those encountered in basic science, biology, and health.

It is not hard to see how routinized behaviour can be explained by neural nets. Essentially if we are regular in our behaviour, this regularity will be reflected in various data associated with that behaviour and these can then act as inputs which are associated with observed outcomes. To an extent, all that neural networks are doing is associating independent variables with dependent ones and defining a very large number of parameters that determine the outcomes exactly. Sometimes it is appropriate to think of such problems as ones where one is producing a perfect fit between inputs and outputs and, in some respects, we might even think of the model of the output as being the data itself or some manipulation thereof. There are countless problems in urban planning and design that might be treated in this way, but despite the very rapid advances currently in machine learning, which rely on big data and massive computational processing, the key problems in our domain are intrinsically more complex than those that currently take pride of place in AI.

The challenge for us is to catalogue the kinds of problems that define urban planning and design so that we clarify the distinctions between the scientific problems of understanding how actual cities function, develop and evolve and the way society defines the design task, which inevitably involves intervention in the process of development. In this chapter, we have mainly focused on the design task while also suggesting that we can use similar pattern recognition techniques to explore the actual process of development. A bigger task awaits us and that is to evolve processes of actual development that incorpo-

rate certain design requirements. Our goal, then, is to merge science and design and, in this quest, the idea of artificial neural networks appears to offer considerable promise.

REFERENCES

Alexander, C. (1964) *Notes on the Synthesis of Form*, Harvard University Press, Cambridge MA.
Alexander, C. and Manheim, M.L. (1962) The Use of Diagrams in Highway Route Location: An Experiment, *Research Report RR-R62–3*, Department of Civil Engineering, MIT, Cambridge MA.
Batty, M. (1971) An Approach to Rational Design: Part 1: The Structure of Design Problems, & Part 2: Design Problems as Markov Chains, *Architectural Design*, *41*, 498–501 & 436–439.
Batty, M. (2013) *The New Science of Cities*, The MIT Press, Cambridge, MA.
Batty, M. (2016) Evolving a Plan: Design and Planning with Complexity, in J. Portugali and E. Stolk (Editors) *Complexity, Cognition, Urban Planning and Design*, Springer Proceedings in Complexity, Springer International Publishing, Switzerland, 21–42.
Blondel, V.D., Hendricks, J.M., Bolshevik, A., and Tsitsiklis, J.N. (2005) Convergence in Multiagent Coordination, Consensus, and Flocking, in *Proceedings of the Joint 44th IEEE Conference on Decision and Control*, European Control Conference, Seville, Spain, December, 12–15.
Coleman, J.S. (1994) *Foundations of Social Theory*. Harvard University Press, Cambridge MA.
De Groot, M.H. (1974) Reaching a Consensus, *Journal of the American Statistical Association*, *69*, 118–121.
French, J.R.P. (1956) A Formal Theory of Social Power, *Psychological Review*, *63*, 181–194.
Friedkin, N.E. (1998) *A Structural Theory of Social Influence*, Cambridge University Press, New York.
Harary, F. (1959) A Criterion for Unanimity in French's Theory of Social Power, in D. Cartwright (Editor) *Studies in Social Power*, Institute for Social Research, Ann Arbor, MI, 168–182.
Kelly, F.P. (1981) How a Group Reaches an Agreement: A Stochastic Model, *Mathematical Social Sciences*, 2, 1–8.
McCulloch. W.S. and Pitts, W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, *5*, 115–133.
McHarg, I. (1969) *Design with Nature*, American Museum of Natural History, Natural History Press, Garden City, NY.
Steinitz, C. (2012) *A Framework for Geodesign: Changing Geography by Design*, ESRI Press, Redlands, CA.
Steinitz, C., Parker, P., and Jordan, L. (1976) Hand-drawn Overlays: Their History and Prospective Uses, *Landscape Architecture*, *66* (5), 444–455.