Kauffman, S., (1995). *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*. Oxford and New York: Oxford University Press. See especially 252-64.

Chapter 11

In Search of Excellence

Sidney Winter, an economist now at the Wharton School at the University of Pennsylvania after a four-year stint as chief economist at the General Accounting Office, spoke at a recent Santa Fe Institute meeting on "Organizational Evolution" and immediately captured our attention. After all, most of us were academic scientists. Sid spoke from experience near the center of the United States government about global changes in our economic life. "There are four horsemen of the workplace," he said.

> Technology, Global Competition, Restructuring, Defense Conversion. These are dominating the post–Cold War period. We need jobs, good jobs, but we don't know how to be certain that the economy will generate those jobs. Health reform, welfare reform, and trade policy are around the corner. We neither know how to achieve these nor understand their impact. The de facto tenure of jobs in U.S. firms is decreasing. Companies are out-sourcing. Rather than performing all parts of the total job inside the firm, many subtasks are being purchased from other firms, often in other countries. This is leading to disintegration of the vertical organization of firms. Mergers and acquisitions are pulling old companies into new forms, then spinning out components into new structures. Trade liberalization is upon us. We are downsizing. It is all captured by a common theme: repackaging. We are shifting packages of economic activity into new smaller units. The folk model of organization as top–down and centralized is out of date. Organizations are becoming flatter, more decentralized.

I listened with surprise. Organizations around the globe were becoming less hierarchical, flatter, more decentralized, and were doing so in the hopes of increased flexibility and overall competitive advantage.

Was there much coherent theory about how to decentralize, I wondered. For I was just in the process of finding surprising new phenomena, a new edge-of-chaos story, that hinted at the possibility of a deeper understanding of how and why flatter, more decentralized organizations—business, political, and otherwise—might actually be more flexible and carry an overall competitive advantage.

A few weeks later, just as I was absorbing this message, the Santa Fe Institute held an "outpost" meeting at the University of Michigan. The aim was to connect the work on the "sciences of complexity" going on in Santa Fe with that of colleagues at the University of Michigan. Computer scientist John Holland, who has had a major impact with his development of "the genetic algorithm," which uses landscape ideas, mutation, recombination, and selection to solve hard mathematical problems, is the glue between the institute and his home institution in Ann Arbor. The dean of the Department of Engineering, Peter Banks, was charismatic. "Total Quality Management is taking over, integrating new modular teams in our firms," he said. "But we have no real theoretical base to understand how to do this well. Perhaps the kind of work going on in Santa Fe can help." I nodded my head vigorously, hopeful, but not necessarily convinced.

Why would I, the other scientists at Santa Fe, or our colleagues around the globe studying complexity be interested in potential connections to the practical problems of business, management, government, and organizations? What are biologists and physicists doing poking into this new arena? The themes of self-organization and selection, of the blind watchmaker and the invisible hand all collaborating in the historical unfolding of life from its molecular inception to cells to organisms to ecosystems and finally to the emergent social structures we humans have evolved—all these might be the locus of law embedded in history. No molecule in the bacterium *E. coli* "knows" the world *E. coli* lives in, yet *E. coli* makes its way. No single person at IBM, now downsizing and becoming a flatter organization, knows the world of IBM, yet collectively, IBM acts. Organisms, artifacts, and organizations are all evolved structures. Even when human agents plan and construct with intention, there is more of the blind watchmaker at work than we usually recognize. What are the laws governing the emergence and coevolution of such structures?

Organisms, artifacts, and organizations all evolve and coevolve on rugged, deforming, fitness landscapes. Organisms, artifacts, and organizations, when complex, all face conflicting constraints. So it can be no surprise if attempts to evolve toward good compromise solutions and designs must seek peaks on rugged landscapes. Nor, since the space of possibilities is typically vast, can it be a surprise that even human agents must search more or less blindly. Chess, after all, is a finite game, yet no grand master can sit at the board after two moves and concede defeat because the ultimate checkmate by the opponent 130 moves later can now be seen as inevitable. And chess is simple compared with most of real life. We may have our intentions, but we remain blind watchmakers. We are all, cells and CEOs, rather blindly climbing deforming fitness landscapes. If so, then the problems confronted by an organization—cellular, organismic, business, governmental, or otherwise—living in niches created by other organizations, is preeminently how to evolve on its deforming landscape, to track the moving peaks.

Tracking peaks on deforming landscapes is central to survival. Landscapes, in short, are part of the search for excellence—the best compromises we can attain.

The Logic of Patches

I intend, in this chapter, to describe some recent work I am carrying out with Bill Macready and Emily Dickinson. The results hint at something deep and simple about why flatter, decentralized organizations may function well: contrary to intuition, breaking an organization into "patches" where each patch attempts to optimize for its own selfish benefit, even if that is harmful to the whole, can lead, as if by an invisible hand, to the welfare of the whole organization. The trick, as we shall see, lies in how the patches are chosen. We will find an ordered regime where poor compromises for the entire organization are found, a chaotic regime where no solution is ever agreed on, and a phase transition between order and chaos where excellent solutions are found rapidly. We will be exploring the logic of patches. Given the results I will describe, I find myself wondering whether these new insights will help us understand how complex organizations evolve, and perhaps even why democracy is such a good political mechanism to reach compromises between the conflicting aspirations of its citizens.

The work is all based on our now familiar friend, the *NK* model of rugged fitness landscapes. Since this is so, caveats are again in order. The *NK* model is but one family of rugged, conflict-laden, fitness landscapes. Any efforts here require careful extension. For example, we will need to be far surer than I now am that the results I will discuss extend to other conflict-laden problems, ranging from the design of complex artifacts such as aircraft, to manufacturing facilities, organizational structures, and political systems.

NK fitness landscapes are examples of what mathematicians call hard combinatorial optimization problems. In the framework of *NK* landscapes, the optimization problem is to find either the global optimum, the highest peak, or at least excellent peaks. In *NK* landscapes, genotypes are combinatorial objects, composed of N genes with either 1 or 0 allele states. Thus as N increases, one finds what is called a combinatorial explosion of possible genotypes, for the number of genotypes is 2^N. One of these genotypes is the global peak we seek. So as N goes up, finding the peak can become very much harder. Recall that for $K = N - 1$, the maximum density of interconnectedness, landscapes are fully random and the number of local peaks is $2^N/(N + 1)$. In Chapter 8, we discussed finding maximally compressed algorithms to perform a computation, and noted that such algorithms "live" on random landscapes. Therefore, finding a maximally compressed program for an algorithm amounted to finding one or, at most, a very few of the highest peaks on such a random landscape. Recall that on random landscapes, local hill-climbing soon becomes trapped on local peaks far from the global optimum. Therefore, finding the global peak or one of a few excellent peaks is a completely intractable problem. One would have to search the entire space to be sure of success. Such problems are known as *NP*-hard. This means roughly that the search time to solve the problem increases in proportion to the size of the problem space, which itself increases exponentially because of the combinatorial explosion.

Evolution is a search procedure on rugged fixed or deforming landscapes. No search procedure can guarantee locating the global peak in an *NP*-hard problem in less time than that required to search the entire space of possibilities. And that, as we have repeatedly seen, can be hyperastronomical. Real cells, organisms, ecosystems, and, I suspect, real complex artifacts and real organizations never find the global optima of their fixed or deforming landscapes. The real task is to search out the excellent peaks and track them as the landscape deforms. Our "patches" logic appears to be one way complex systems and organizations can accomplish this.

Before discussing patch logic, I am going to tell you about a well-known procedure to find good fitness peaks. It is called simulated annealing, and was invented by Scott Kirkpatrick at IBM and his colleagues some years ago. The test example of a hard combinatorial optimization problem they chose is the famous Traveling Salesman problem. If one could solve this, one would break the back of many hard optimization problems. Here it is: you, a salesperson, live in Lincoln, Nebraska. You must visit 27 towns in Nebraska, one after the other, and then return home. The catch is, you are supposed to travel your circuit by the shortest route possible.

That's all there is to it. Just get in your flivver, pack 27 lunches in your carry-along ice chest, and get on with the tour. Simple as pie. Or so it sounds.

If the number of cities, N, which is 27 here, increases to 100 or 1,000, the complexity of the pie becomes one of those hyperastronomical kinds of messes. You see, if you start in Lincoln, you have to choose the first city you will go to, and you have 26 choices. After picking the first city on your tour, you must pick the second, so 25 choices remain. And so on. The number of possible tours starting from Lincoln is $(27 \times 26 \times 25 \times \ldots 3 \times 2 \times 1)/2$. (Divide by 2 because you could choose either of two routes around any circuit, and dividing by 2 keeps us from counting the same tour twice.)

One would think there was some easy way to find the very best tour; however, this appears to be a dashed hope. It appears that in order to find the shortest tour, you have to consider them all. As the number of cities, N, grows large, you get one of the combinatorial explosions of possibilities that make genotype spaces and other combinatorial spaces so huge. Even with the fastest computer, it can be impossible to guarantee finding the shortest tour in, say, the life of the human species or the universe.

The best thing to do—indeed, the only practical thing to do—is to choose a route that is excellent, but not necessarily the very best. As in all of life, the salesperson in search of excellence will have to settle for less than perfection.

How can one find at least an excellent tour? Kirkpatrick and his colleagues offered a powerful approach in their concept of simulated annealing. First we need a fitness, or cost, landscape. Then we will search across that landscape to find good short tours. The landscape in question is simple. Consider our 27 cities and all possible tours through them. As we have seen, there is a very large number of such tours. Now we need the idea of which tours are "near" each other, just as, for genotypes, we needed the idea of near mutant neighbors. One way of doing this is to think of a "swap" that exchanges the positions of two cities in a tour. Thus we might have gone A B C D E F A. If we exchange C and F, we would go A B F D E C A (Figure 11.1).

Once we have defined this notion of a "neighboring tour," we can arrange all the possible tours in a high-dimensional space, rather like a genotype space, in which each tour is next to all its neighbors. It's hard to show the proper high-dimensional tour space. Recall that in the *NK* model, each genotype, such as (1111), is a vertex in a Boolean hypercube and is next to N other genotypes: (0111), (1011), (1101), and (1110). Adaptive walks occur from genotype to neighboring genotype

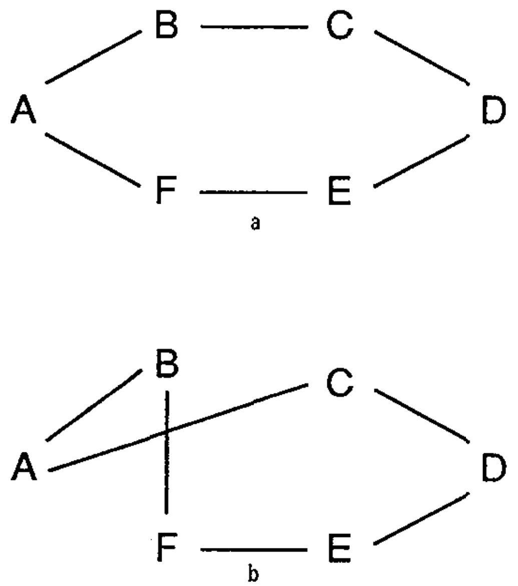

Figure 11.1 *The Traveling Salesman problem. One tries to find the shortest route through several cities.* (a) *Traveling Salesman tour through six cities, A–F.* (b) *Exchanging two cities from* (a) *alters the tour to a "neighboring" tour.*

until a local peak is reached. In tour space, each tour is a "vertex" and is connected by a line to each of its neighboring tours. Since we are seeking the shortest tour from Lincoln through the 27 cities and back to Lincoln, it makes sense to think of the length of the tour as its "cost." Since each tour has a cost, we get a cost landscape over our tour space. Since we are trying to minimize cost rather than maximize fitness, on a cost landscape we seek the deepest valleys rather than the highest peaks. The idea, however, is obviously the same.

Like any other rugged landscape, the tour space may be correlated in a variety of ways; that is, neighboring tours will tend to have the same length, and hence cost. If so, it would be smart to use the correlations to help find excellent tours, even if we cannot find the best. Recall from Chapter 8 that we found a rather general feature of many rugged landscapes: the deepest valleys "drain" the largest region of the space of possibilities. If we think of the valleys as real valleys in a mountainous terrain, water can flow downhill to the deepest valleys from the greatest number of initial positions on the landscape. This feature is essential to simulated annealing, as we are about to see.

Picture a water droplet or a ball. Once it reaches one local minimum,

it is stuck there forever unless disturbed by some outside process. That is, whether hiking uphill toward fitness peaks or downhill toward cost minima, if one can take only steps that improve the situation, one will soon become trapped. But the minimum or maximum by which one is trapped may be very poor compared with the excellent minima or maxima. Thus the next question is how to escape.

Real physical systems have an utterly natural way to escape from bad local minima. Sometimes they randomly move in the wrong direction, taking a step uphill when it would seem natural to go down. This random motion is caused by thermal vibration and can be measured by temperature.

Think of a system of interacting molecules colliding with one another. The rate of collisions depends on the velocities of the molecules. Temperature is a measure of the average motion of the molecules, the average kinetic energy. High temperature means that the molecules are in violent random motions. Zero temperature means the molecules are not moving at all.

At a high temperature, a physical system jostles around in its space of possible configurations, molecules colliding with one another and exchanging kinetic energy. This jostling means that the system does not just flow downhill into local energy minima, but can, with a probability that increases with the temperature, jump uphill in energy over "energy barriers" into the drainage basins of neighboring energy minima. If the temperature were lower, the system would be less likely to jump uphill in energy over any given barrier, and hence would be more likely to remain in any given energy "well."

Annealing is just gradual cooling. Real physical annealing corresponds to taking a system and gradually lowering its temperature. A smithy hammering red-hot iron, repeatedly plunging the forming object into cold water and then reheating it and hammering it again, is practicing real annealing. As the smithy anneals and hammers, the microscopic arrangements of the atoms of iron are rearranged, giving up poor, relatively unstable, local minima and settling into lower-energy minima corresponding to harder, stronger metal. As the repeated heating and hammering occurs, the microscopic arrangements in the worked iron can at first wander all over the space of configurations, jumping over energy barriers between all local energy minima. As the temperature is lowered, it becomes harder and harder to jump over these barriers. Now comes the crux assumption: if the deepest energy minima drain the biggest basins, then as temperature is lowered, the microscopic arrangements will tend to become trapped in the biggest drainage basins, precisely because they are the biggest, and drift down to the deepest, stablest energy minima. Working iron by real annealing will achieve hard, strong metal because annealing drives the microscopic atomic arrangements to deep energy minima.

Simulated annealing uses the same principle. In the case of the Traveling Salesman problem, one moves from a tour to a neighboring tour if the second tour is shorter. But, with some probability, one also accepts a move in the wrong direction—to a neighboring tour that is longer and "costs" more. The "temperature" of the system specifies the probability of accepting a move that increases cost by any given amount. In the simulation, the algorithm wanders all over the space of possible tours. Lowering temperature amounts to decreasing the probability of accepting moves that go the wrong way. Gradually, the algorithm settles into the drainage basins of deep, excellent minima.

Simulated annealing is a very interesting procedure to find solutions to conflict-laden problems. In fact, currently it is one of the best procedures known. But there are some important limitations. First, finding good solutions requires "cooling" very slowly. It takes a long time to find very good minima. There is a second obvious problem with anything like simulated annealing if one also has in mind how human agents or organizations might find good solutions to problems in real life. Consider a fighter pilot streaking toward battle. The situation is fast-paced, intense, life-threatening. The pilot must decide on tactics that optimize chances for success in the conflict-laden situation. Our fighter pilot is unlikely to be persuaded to choose his tactics in the heat of battle by making what he knows are very many mistakes with ever-decreasing frequency until he settles into a good strategy. Nor, it would appear, do human organizations try to optimize in anything like this fashion. Simulated annealing may be a superb procedure to find excellent solutions to hard problems, but in real life, we never use it. We simply do not spend our time making mistakes on purpose and lowering the frequency of mistakes. We all try our best, but fail a lot of the time.

Have we evolved some other procedure that works well? I suspect we have, and call it by a variety of names, from federalism to profit centers to restructuring to checks and balances to political action committees. Here I'll call it patches.

The Patch Procedure

The basic idea of the patch procedure is simple: take a hard, conflict-laden task in which many parts interact, and divide it into a quilt of nonoverlapping patches. Try to optimize within each patch. As this occurs, the couplings between parts in two patches across patch boundaries will mean that finding a "good" solution in one patch will change the problem to be solved by the parts in the adjacent patches. Since changes in each patch will alter the problems confronted by the neighboring patches, and the adaptive moves by those patches in turn will alter the problem faced by yet other patches, the system is just like our model coevolving ecosystems. Each patch is the analogue of what we called a species in Chapter 10. Each patch climbs toward fitness peaks on its own landscape, but in doing so deforms the fitness landscapes of its partners. As we saw, this process may spin out of control in Red Queen chaotic behavior and never converge on any good overall solution. Here, in this chaotic regime, our system is a crazy quilt of ceaseless changes. Alternatively, in the analogue of the evolutionary stable strategy (ESS) ordered regime, our system might freeze up, getting stuck on poor local peaks. Ecosystems, we saw, attained the highest average fitness if poised between Red Queen chaos and ESS order. We are about to see that if the entire conflict-laden task is broken into the properly chosen patches, the coevolving system lies at a phase transition between order and chaos and rapidly finds very good solutions. Patches, in short, may be a fundamental process we have evolved in our social systems, and perhaps elsewhere, to solve very hard problems.

By now you know the *NK* model. All it consists of is a system of N parts, each of which makes a "fitness contribution" that depends on its own state and the states of K other parts. Let's put the *NK* model onto a square lattice (Figure 11.2). Here each part is located at a vertex connecting it to its four immediate neighbors: north, south, east, and west. As before, we let each part have two states: 1 and 0. Each part makes a fitness contribution depending on its own state and that of its north, south, east, and west neighbors. This fitness contribution is assigned at random between 0.0 and 1.0. As before, we can define the fitness of the entire lattice as the average of the fitness contribution of each of the parts. Say all are in the 1 state. Just add up the contributions of each part and divide by the number of parts. Do this for all possible configurations and we get a fitness landscape.

Macready, Dickinson, and I have been looking at fairly large lattices, 120×120, so our model hard problems have 14,400 parts. That should be enough to be hard. Since *NK* rugged landscapes are very similar to the landscapes of many hard conflict-laden optimization problems, including the Traveling Salesman, finding means to achieve good optima here is likely to be generally useful. Notice, as usual, the vast space of possibilities. Since each part can be in two states, 1 and 0, the total number of combinations of states of the parts, or configurations of the

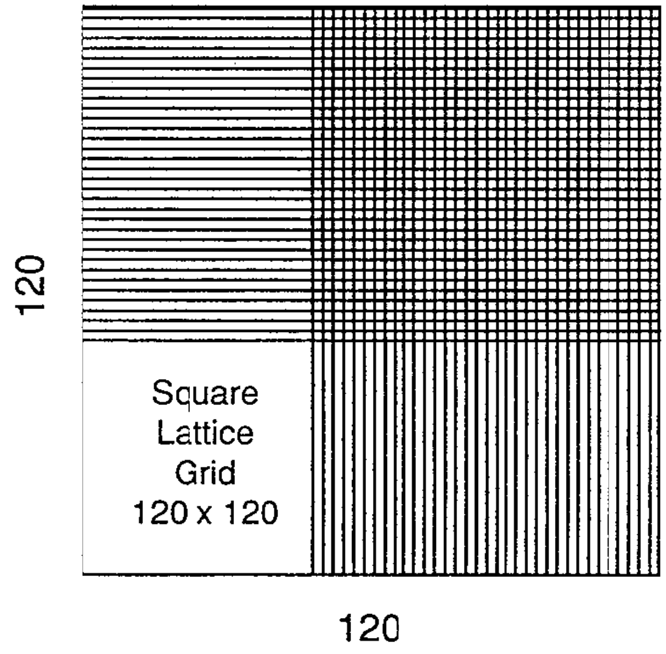

Figure 11.2 *An* NK *model in the form of a 120 × 120 square lattice. Each site, which can be in one of two states, 1 or 0, is connected to its four neighbors: north, south, east, and west. (The lattice is bent into the surface of torus; that is, the top edge is "glued" to the bottom edge, and the left edge is glued to the right edge so that each site has four neighbors.)*

lattice, is $2^{14,400}$. Forget it, there is not enough time since the Big Bang to find the global optimum. We seek excellence, not perfection.

Because Bill Macready is a physicist and physicists like to minimize an "energy" rather than maximize a "fitness," and because we are also used to minimizing on cost surfaces, let us think of the *NK* model as yielding an "energy" landscape and minimize energy. Each configuration of the 14,400 parts on the 120 × 120 lattice is a vertex on the 14,400-dimensional Boolean hypercube. Each vertex is an immediate neighbor of 14,400 other vertices, each corresponding to flipping one of the 14,400 parts to the other state, 1 or 0. Each vertex has an energy, so the *NK* model generates an energy landscape over this huge Boolean hypercube of configurations. We seek the deep, excellent minima. The *NK* landscape remains unchanged here, we just go "downhill" rather than "uphill" on it. The statistical-landscape features are the same in either direction.

Now I will introduce patches. Suppose we use the same *NK* landscape, leave the parts coupled in the same ways, but divide the system into nonoverlapping quilt patches of different sizes. The rule is always going to be the same: try flipping a part to the opposite value, 1 to 0 or 0 to 1. If this move lowers the energy of the patch in which the part is located, accept the move; otherwise, do not accept the move.

Figure 11.3 shows a smaller version of our 120 × 120 lattice, here reduced to a 10 × 10 square. In Figure 11.3*a*, we consider the entire lattice as a single "whole" patch. I'll call this the "Stalinist" limit. Here a part can flip from 1 to 0 or 0 to 1 only if the move is "good" for the en-

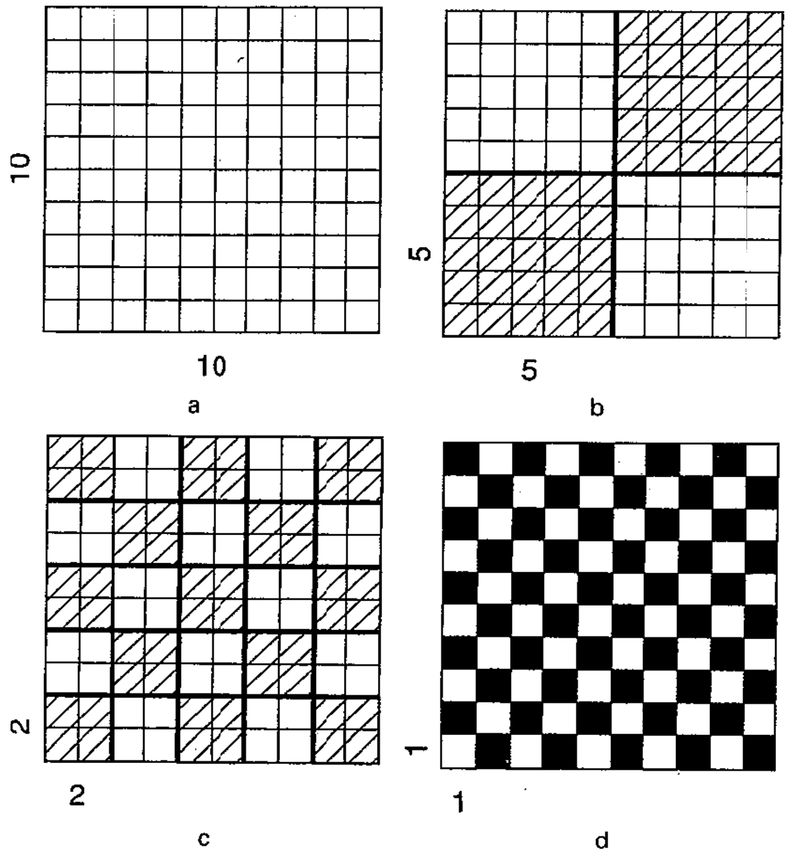

Figure 11.3 *The patches procedure.* (a) *A 10 × 10* NK *lattice; the entire system is one patch.* (b) *The same lattice divided into four 5 × 5 patches,* (c) *25 2 × 2 patches, and* (d) *100 1 × 1 patches.*

tire lattice, lowering its energy. We all must act for the benefit of the entire "state."

Since in the Stalinist limit any move must lower the energy of the entire lattice, then as successive parts are tried and some are flipped, the entire system will carry out an adaptive walk to some local energy minimum and then stay there forever. Once at such a local minimum, no part can be flipped and find a lower energy for the total lattice, so no flips will be accepted. All parts "freeze" into an unchanging state, 1 or 0. In short, in the Stalinist regime, the system locks into some solution and is frozen there forever. The Stalinist regime, where the game is one for all, one for the state, ends up in frozen rigidity.

Now look at Figure 11.3*b*. Here the same square lattice, with the same couplings among the parts, is broken into four patches, each a 5×5 sublattice of the entire 10×10 lattice. Each part belongs to only a single patch. But the parts near the boundaries of a patch are coupled to parts in the adjacent patches. So adaptive moves, parts flipping between 1 and 0 states, in one patch will affect the neighboring patches. I emphasize that the couplings among the parts are the same as in the Stalinist regime. But now, by our rule that a part can flip if it is good for the patch in which it is located, a part might help its own patch, but hurt an adjacent patch.

In the Stalinist limit, the entire lattice can flow only downhill toward energy minima. Thus the system is said to flow on a potential surface. The system is like the ball on a real surface in a valley. The ball will roll to the bottom of the valley and remain there. Once the lattice is broken into patches, however, the total system no longer flows on a potential surface. A flip of a part in one patch may lower the energy of that patch, but *raise* the energies of adjacent patches because of the couplings across boundaries. Because adjacent patches can go up in energy, the total energy of the lattice itself can go up, not down, when a single patch makes a move to lower its own energy. And since the entire lattice can go up in energy, the total system is not evolving on a potential surface. Breaking the system into patches is a little like introducing a temperature in simulated annealing. Once the system is broken into quilt patches so an adaptive move by one patch can "harm" the whole system, that move causes the whole system to "go the wrong way."

We reach a simple, but essential conclusion: once the total problem is broken into patches, the patches are coevolving with one another. An adaptive move by one patch changes the "fitness" and deforms the fitness landscape, or, alternatively, the "energy" and the "energy landscape," of adjacent patches.

It is the very fact that patches coevolve with one another that begins to hint at powerful advantages of patches compared with the Stalinist

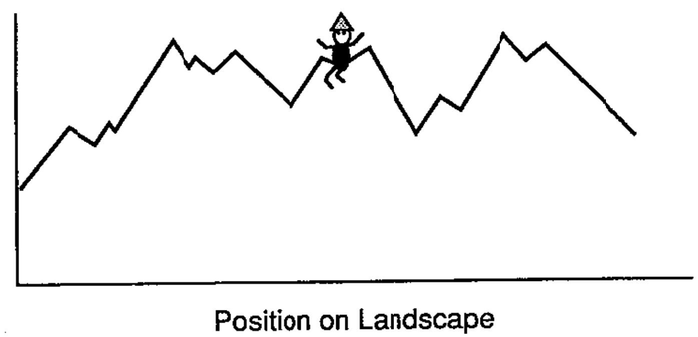

Figure 11.4 *An energy landscape showing a system trapped on a poor, high-energy local minimum.*

limit of a single large patch. What if, in the Stalinist limit, the entire lattice settles into a "bad" local minimum, one with high energy rather than an excellent low-energy minimum, as in Figure 11.4? The single-patch Stalinist system is stuck forever in the bad minimum. Now let's think a bit. If we break the lattice up into four 5×5 patches just after the Stalinist system hits this bad minimum, what is the chance that this bad minimum is not only a local minimum for the lattice as a whole, but also a local minimum for each of the four 5×5 patches individually? You see, in order for the system broken into four patches to "stay" at the same bad minimum, it would have to be the case that the same minimum of the entire lattice happens also to be a minimum for all four of the 5×5 patches individually. If not, one or more of the patches will be able to flip a part, and hence begin to move. Once one patch begins to move, the entire lattice is no longer frozen in the bad local minimum.

Well, the intuitive answer is pretty obvious. If the entire lattice, in the Stalinist limit, flows to a bad local minimum, the chance is small that the same configuration of the parts is also a local minimum for all four of the 5×5 patches, so the system will not remain frozen. It will "slip" away and be able to explore further across the total space of possibilities.

The Edge of Chaos

We have now encountered a phase transition between order and chaos in model genomic networks and in coevolutionary processes. In Chapter 10, we saw that the highest average fitness in coevolving systems appeared to arise just at the phase transition between Red Queen chaos and ESS order. Breaking large systems into patches allows the patches literally to coevolve with one another. Each climbs toward its fitness peaks, or energy minima, but its moves deform the fitness landscape or energy landscape of neighboring patches. Is there an analogue of the chaotic Red Queen regime and the ordered ESS regime in patched systems? Does a phase transition between these regimes occur? And are the best solutions found at or near the phase transition? We are about to see that the answer to all these questions is yes.

The Stalinist limit is in the ordered regime. The total system settles to a local minimum. Thereafter, no part can be flipped from 1 to 0 or 0 to 1. All the parts, therefore, are frozen. But what happens at the other limit? In the extreme situation shown in Figure 11.3*d*, each part constitutes its own patch. In this limit, on our 10 × 10 lattice, we have created a kind of "game" with 100 players. At each moment, each player considers the states, 1 or 0, of its north, south, east, and west neighbors and takes the action, 1 or 0, that minimizes its own energy. It is easy to guess that in this limit, call it the "Leftist Italian" limit, the total system never settles down. Parts keep flipping from 1 to 0 and 0 to 1. The system is in a powerfully disordered, or chaotic, regime.

Since the parts never converge onto a solution where they stop flipping, the total system burbles along with quite high energy. In the *NK* model, the expected energy of a randomly chosen configuration of the *N* parts without trying to minimize anything is 0.5. (The 0.5 value follows because we assigned fitness or energy as random decimals between 0.0 and 1.0, and the average value is halfway between these limits, or 0.5.) In the chaotic Leftist Italian limit, the average energy achieved by the lattice is only a slight bit less, about 0.47. In short, if the patches are too numerous and too small, the total system is in a disordered, chaotic regime. Parts keep flipping between their states, and the average energy of the lattice is high.

We come to the fundamental question: At what patch size does the overall lattice actually minimize its energy?

The answer depends on how rugged the landscape is. Our results suggest that if *K* is low so the landscape is highly correlated and quite smooth, the best results are found in the Stalinist limit. For simple problems with few conflicting constraints, there are few local minima in which to get trapped. But as the landscape becomes more rugged, reflecting the fact that the underlying number of conflicting constraints is becoming more severe, it appears best to break the total system into a number of patches such that the system is near the phase transition between order and chaos.

In the current context, we can look at increasing the number of conflicting constraints by retaining our square-lattice configuration, but

considering cases in which each part is affected not only by itself and its four immediate neighbors—north, south, east, and west—but by its eight nearest neighbors—the first four, plus the northwest, northeast, southeast, and southwest neighbors. Figure 11.5 shows that increasing the range across the square lattice over which parts influence one another allows us to think of the four-neighbor, eight-neighbor, 12-neighbor, and 24-neighbor cases. Hence in terms of the *NK* model, *K* is increasing from four to 24.

Figure 11.6 shows the results obtained, allowing the patched lattice to evolve forward in time, flipping randomly chosen parts if and only if the move lowers the energy of the patch containing the part. We carried

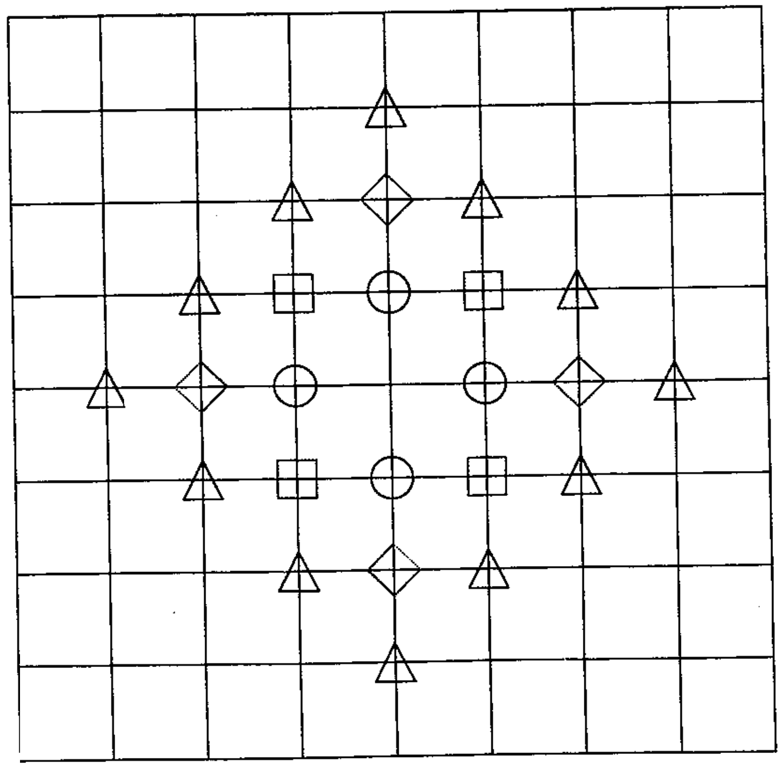

Figure 11.5 *Increasing the range. A site on an* NK *square lattice may be connected to* K = 4 *neighbors (circles); or may "reach" farther out into the square lattice to be coupled to* K = 8 *sites (circles plus squares), or may reach still farther out to be coupled to* K = 12 *sites (circles plus squares plus diamonds); or may reach out to be coupled to* K = 24 *sites (circles plus squares plus diamonds plus triangles).*

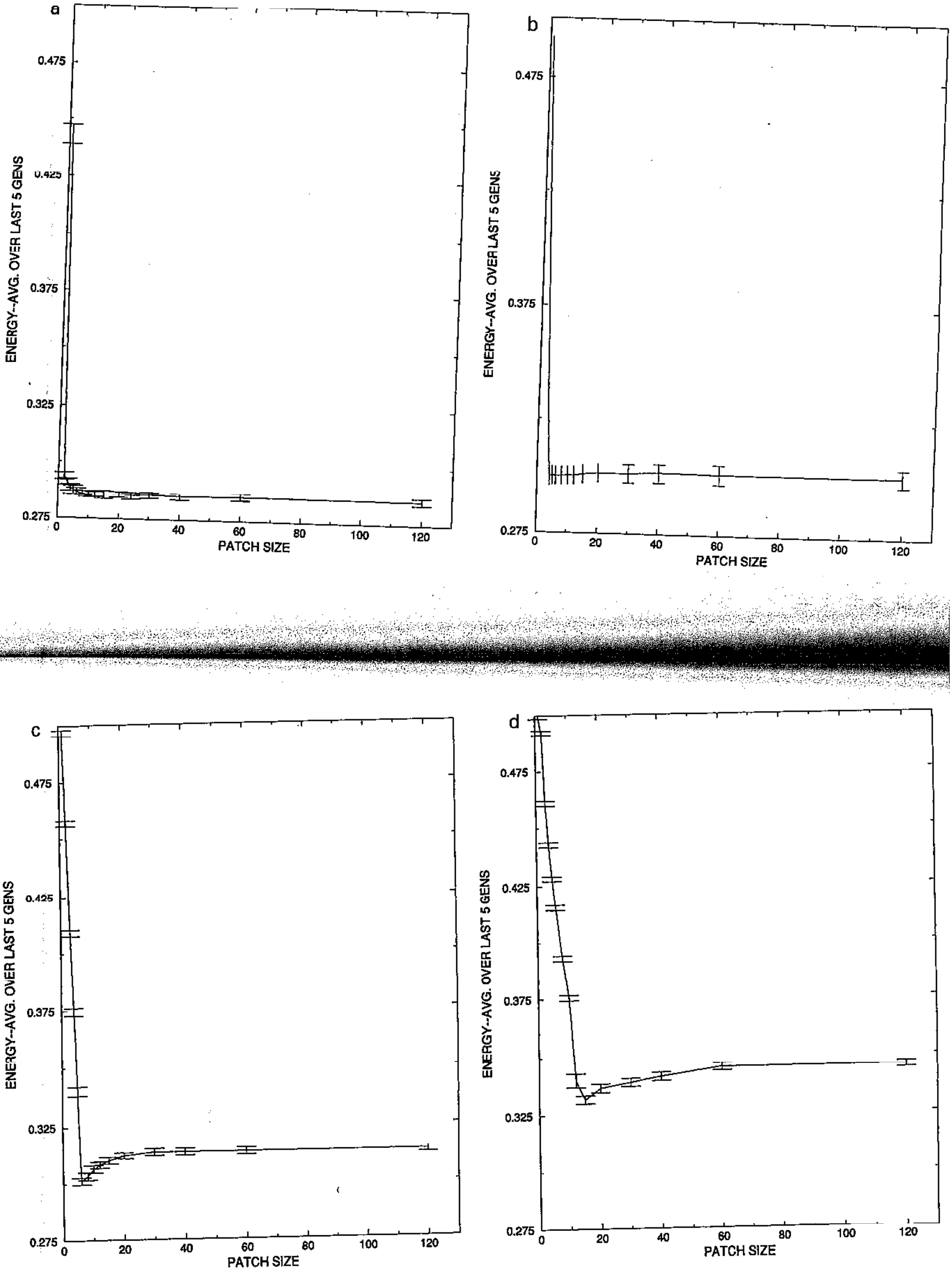

Figure 11.6 *Divide and conquer. As landscapes become more rugged, energy minimums can be more easily found by dividing the problem into patches. Patch sizes are plotted on the* x-*axis. The average low energy reached at the end of the simulation runs is plotted on the* y-*axis. (Bars above and below the curve at each patch size correspond to plus and minus 1 standard deviation.)* (a) *For a smooth,* K = 4, *landscape, the optimum patch size, minimizing energy, is a* 120 × 120 *single patch—the "Stalinist" limit.* (b) *For a more rugged,* K = 8, *landscape, the optimum patch size is* 4 × 4. (c) *For* K = 12, *the optimum patch size is* 6 × 6. (d) *For* K = 24, *the optimum patch size is* 15 × 15.

out these simulations as the total energy of the lattice decreased until the energy attained was either fixed or only fluctuating in a narrow range. Figures 11.6*a–d* plot energy on the *y*-axis against the size of each of the patches on the *x*-axis. Here all patches are themselves square, and our results are from the 120 × 120 lattice. All our simulations were carried out on exactly the same set of *NK* lattices. Only the sizes of the patches were changed. Thus the results tell us the effect of patch size and number on how well the lattice as a whole lowered its overall energy.

The results are clear When $K = 4$, it is best to have a single large patch. In worlds that are not too complex, when landscapes are smooth, Stalinism works. But as landscapes become more rugged, $K = 8$ to $K = 24$, it is obvious that the lowest energy occurs if the total lattice is broken into some number of patches.

Here, then, is the first main and new result. It is by no means obvious that the lowest total energy of the lattice will be achieved if the lattice is broken into quilt patches, each of which tries to lower its own energy regardless of the effects on surrounding patches. Yet this is true. It can be a very good idea, if a problem is complex and full of conflicting constraints, to break it into patches, and let each patch try to optimize, such that all patches coevolve with one another.

Here we have another invisible hand in operation. When the system is broken into well-chosen patches, each adapts for its own selfish benefit, yet the joint effect is to achieve very good, low-energy minima for the whole lattice of patches. No central administrator coordinates behavior. Properly chosen patches, each acting selfishly, achieve the coordination.

But what, if anything, characterizes the optimum patch-size distribution? The edge of chaos. Small patches lead to chaos; large patches freeze into poor compromises. When an intermediate optimum patch size exists, it is typically very close to a transition between the ordered and the chaotic regime.

In Figure 11.7, I show something of this "phase transition." These two figures concern the same *NK* landscape on the same lattice, begun from the same initial state. The only difference is that in Figure 11.7*a* the 120 × 120 lattice was broken into 5 × 5 patches, while in Figure 11.7*b* the lattice was broken into 6 × 6 patches. The figures show how often each site on the lattice flips. Sites that flip often are dark; ones that do not flip at all are white. In Figure 11.7*a*, most sites are dark—indeed, darkest along the boundaries between patches. Sites keep flipping in a chaotic, disordered regime. But just increase patch size to 6 × 6 and the results are starting. As shown in Figure 11.7*b*, almost all sites stop flip-

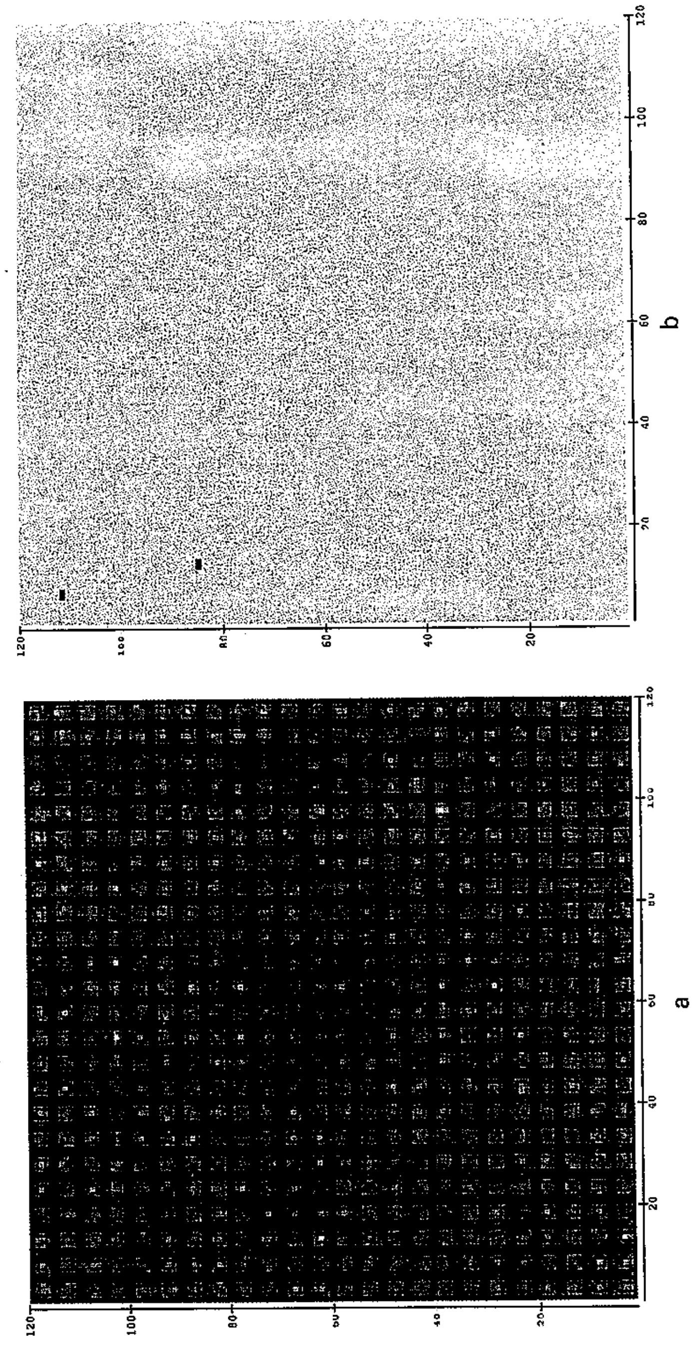

Figure 11.7 *Optimum patch sizes occur near the edge of chaos. What happens when we change the patch size of an* NK *lattice in which* K = *12? (Sites that flip often are dark; sites that do not flip are white.)* (a) *When broken into 5 × 5 patches, the lattice is in the unfrozen, chaotic regime. Most sites are dark and darkest along the boundaries between patches.* (b) *But watch what happens when we break the lattice into 6 × 6 patches. Almost all sites stop flipping and freeze. The transition from* (a) *to* (b) *is the phase transition from chaos to order.*

ping. A few quarrelsome sites across a few boundaries keep flipping, but almost all sites have settled down and stopped changing.

The system, if broken into 5 × 5 patches, never converges onto a solution and the energy of the whole lattice is high. The same system, if broken into 6 × 6 patches, converges onto a solution in which almost all patches and almost all their parts are no longer changing. And the energy of the entire lattice is very low. A kind of phase transition has occurred as the same lattice is broken into 5 × 5 and then into 6 × 6 patches, passing from a chaotic to an ordered regime.

In terms of a coevolutionary system, it appears that if the lattice is broken into 6 × 6 patches, each patch reaches a local minimum that is consistent with the minima of the patches adjacent to it. This global behavior is now like a Nash equilibrium among the patches, or an evolutionary stable strategy. The optimum found by each patch is consistent with the optima found by its neighbors. No patch has an incentive to change further. The patches therefore stop coevolving across their landscapes. The system converges on a total solution.

Across lots of simulations, it appears to be the case that the lowest energy is found for a given lattice in the ordered, ESS regime, somewhere near the phase transition. In some cases, the lowest energy is found at the smallest patch size that remains in the ordered regime, thus just before the phase transition to chaos. In other cases, the lowest energy is found when patch size is still a bit larger so the system is a bit deeper into the ordered regime and farther from the phase transition to chaos. Therefore, as a general summary, it appears that the invisible hand finds the best solution if the coevolving system of patches is in the ordered regime rather near the transition to chaos.

Patch Possibilities

I find it fascinating that hard problems with many linked variables and loads of conflicting constraints can be well solved by breaking the entire problem into nonoverlapping domains. Further, it is fascinating that as the conflicting constraints become worse, patches become ever more helpful.

While these results are new and require extension, I suspect that patching will, in fact, prove to be a powerful means to solve hard problems. In fact, I suspect that analogues of patches, systems having various kinds of local autonomy, may be a fundamental mechanism underlying adaptive evolution in ecosystems, economic systems, and cultural systems. If so, the logic of patches may suggest new tools in design problems. Moreover, it may suggest new tools in the management of complex organizations and in the evolution of complex institutions worldwide.

Homo sapiens sapiens, wise man, has come a long way since bifacial stone axes. We are constructing global communication networks, and whipping off into space in fancy tin cans powered by Newton's third law. The *Challenger* disaster, brownouts, the Hubble trouble, the hazards of failure in vast linked computer networks—our design marvels press against complexity boundaries we do not understand. I wonder how general it has become as we approach the year 2000 that the design of complex artifacts is plagued with nearly unsolvable conflicting constraints. One hears tales, for example, of attempts to optimize the design of complex manufactured artifacts such as supersonic transports. One team optimizes airfoil characteristics, another team optimizes seating, another works on hydraulics, but the multiple solutions do not converge to a single compromise that adequately solves all the design requirements. Proposals keep evolving chaotically. Eventually, one team makes a choice—say, how the hydraulic system or the airfoil structure will be constructed—and the rest of the design becomes frozen into place because of this choice.

Does this general problem of nonconvergence reflect "patching" the design problem into too many tiny patches such that the overall design process is in a nonconverging chaotic regime, just as would be our 120 × 120 lattice broken into 5 × 5 rather than 6 × 6 patches? If one did not know that increasing patch size would lead from chaos to ordered convergence on excellent solutions, one would not know to try "chunking" bigger. It seems worth trying on a variety of real-world problems.

Understanding optimal patching may be useful in other areas in the management of complex organizations. For example, manufacturing has long used fixed facilities of interlinked production processes leading to a single end product. Assembly-line production of manufactured products such as automobiles is an example. Such fixed facilities are used for long production runs. It is now becoming important to shift to flexible manufacturing. Here the idea is to be able to specify a diversity of end products, reconfigure the production facilities rapidly and cheaply, and thus be able to carry out short production runs to yield small quantities of specialized products for a variety of niche markets. But one must test the output for quality and reliability. How should this be done? At the level of each individual production step? At the level of output of the entire system? Or at some intermediate level of chunking? I find myself thinking that there may be an optimal way to break the total production process in each case into local patches, each with a