At the end of the simulation, the algorithm provides only a relative density of states. To determine the normalized density of states $g_n(E)$, we can either use the fact that the total number of states for the Ising model is $\sum_E g(E) = 2^N$ or that the number of ground states (for which $E = -2N$) is 2. The latter normalization guarantees the accuracy of the density of states at low energies, which is important in the calculation of thermodynamic quantities at low temperature. If we apply the former condition, we cannot guarantee the accuracy of $g(E)$ for energies at or near the ground state, because the rescaling factor is dominated by the maximum density of states. We can use one of these two normalization conditions to obtain the absolute density of states and use the other normalization condition to check the accuracy of our result.

Problem 15.30 Sampling the density of states

(a) Implement the Wang–Landau algorithm for the two-dimensional Ising model for $L = 4$, 8, and 16. For simplicity, choose $p = 0.8$ as your criterion for flatness. How many Monte Carlo steps per spin are needed for each iteration? Determine the density of states and describe its qualitative dependence on E.

(b) Compute $P(E) = g(E)e^{-\beta E}/Z$ for different temperatures for the $L = 16$ system. If $T = 0.1$, what range of energies will contribute to the specific heat? What is the range of relevant energies for $T = 1.0$, $T = T_c$, and $T = 4.0$?

(c) Use the density of states that you computed in part (a) to compute the mean energy, the specific heat, the free energy, and the entropy as a function of temperature. Compare your results to your results for $\langle E\rangle$ and C that you found using the Metropolis algorithm in Problem 15.16.

(d) Use the Wang–Landau algorithm to determine the density of states for the one-dimensional Ising model. In this case you can compare your computed values of $g(E)$ to the exact answer:

$$g(E) = 2\frac{N!}{i!(N-i)!}, \tag{15.77}$$

where $E = 2i - N$, $i = 0, 2, \ldots, N$, and N is even. How does the accuracy of the computed values of $g(E)$ depend on the choice of p for the flatness criterion? (Exact results are available for $g(E)$ for the two-dimensional Ising model as well, but no explicit combinatorial formula exists. See the article by Beale.)

*(e) The results that you have obtained so far have probably convinced you that the Wang–Landau algorithm is ideal for simulating a variety of systems with many degrees of freedom. What about critical slowing down? Does the Wang–Landau algorithm overcome this limitation of other single spin flip algorithms? To gain some insight, we ask, given the exact $g(E)$, how efficiently does the Wang–Landau sample the different values of E? Use either the exact density of states in two dimensions computed by Beale or the approximate one that you computed in part (a) and set $f = 1$. Because the system is doing a random walk in energy space, it is reasonable to compute the diffusion constant of the random walker in energy space:

$$D_E(t) = \langle [E(t) - E(0)]^2\rangle/t, \tag{15.78}$$

where t is the time difference, and the choice of the time origin is arbitrary. The idea is to find the dependence of D on the energy E of the system at a particular time origin. How long does it take the system to return to this energy? Run for a sufficiently long time so that D_E is independent of t. Plot D_E as a function of E. Where is D a maximum? If time permits, determine D_E at the energy E_c corresponding to the critical temperature. How does D_{E_c} depend on L? ■

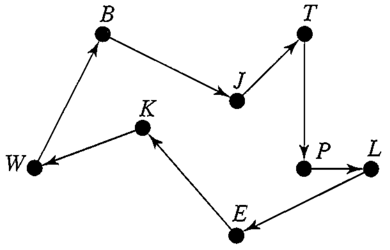

Figure 15.7 What is the optimum route for this random arrangement of $N = 8$ cities? The route begins and ends at city W. A possible route is shown.

15.13 ■ MORE APPLICATIONS

You are probably convinced that Monte Carlo methods are powerful, flexible, and applicable to a wide variety of systems. Extensions to the Monte Carlo methods that we have not discussed include multiparticle moves, biased moves where particles tend to move in the direction of the force on them, bit manipulation for Ising-like models, and the use of multiple processors to update different parts of a large system simultaneously. We also have not described the simulation of systems with long-range potentials such as Coulombic systems and dipole-dipole interactions. For these potentials, it is necessary to include the interactions of the particles in the center cell with the infinite set of periodic images.

We conclude this chapter with a discussion of Monte Carlo methods in a context that might seem to have little in common with the types of problems we have discussed. This context is called *multivariate* or *combinatorial optimization,* a fancy way of saying, "How do you find the global minimum of a function that depends on many parameters?" Problems of this type arise in many areas of scheduling and design as well as in physics, biology, and chemistry. We explain the nature of this type of problem for the *traveling salesman problem,* although we would prefer to call it the traveling peddler or traveling salesperson problem.

Suppose that a salesman wishes to visit N cities and follow a route such that no city is visited more than once and the end of the trip coincides with the beginning. Given these constraints, the problem is to find the optimum route such that the total distance traveled is a minimum. An example of $N = 8$ cities and a possible route is shown in Figure 15.7. All known exact methods for determining the optimal route require a computing time that increases as e^N, and hence, in practice, an exact solution can be found only for a small number of cities. (The traveling salesman problem belongs to a large class of problems known as NP-complete. The NP refers to nondeterministic polynomial. Such problems cannot be done in a time proportional to a finite polynomial in N on standard computers, though polynomial time algorithms are known for hypothetical nondeterministic (quantum)