**Markov Chains as Random Walks in State Space**

- **Model:** Set of states + 1-step transition probabilities
- 1-step transition probabilities determine the topology of the state space → **feasible walks**
- Initial conditions + 1-step transitions determine what is achievable during a chain walk
Long-Term Behavior

- **Long-term predictions:** What is the state probability distribution in the long run? → With which probability after a certain long time the state of the process will be $E_i$?

- If an asymptotic probability distribution does exist for the n-step probabilities then the chain has a **limiting distribution** that says what is the probability of finding the chain in a particular state in the long run.

- If the limiting distribution is *independent of the initial distribution* → **Invariant distribution**

- The limiting distribution may exist but be different for different (set of) initial distributions.

- If a chain has a limiting distribution it also has a **stationary distribution** for the states (but the reverse is not always true).

- **Most interesting / useful chains:** Have limiting invariant distribution → We can say something *general* about the chain!
What does a stationary MC output?
- A probability distribution!

At a “large” time \( n \), when convergence has been reached, the state \( E_i \) entered by the chain at step \( n+1 \) should be seen as a state sampled with probability \( p_i \) from the probability distribution \( p_{\text{stationary}} \).

This is very useful!
- If just setting 1-step probabilities I can get the desired asymptotic distribution, then I have a way to sample from a complex / unknown distribution by making the chain reaching stationarity.
procedure SimulatedAnnealing()
    \( S = \{ \text{set of all feasible solutions} \} \);
    \( \mathcal{N} = \text{neighborhood structure defined over } S \);
    \( s \leftarrow \text{Generate a starting feasible solution}; \quad \text{ // e.g., with a construction heuristic} \)
    \( s^{\text{best}} \leftarrow s \);
    \( T \leftarrow \text{Determine a starting value for temperature}; \)
    \( \text{while (NOT YET frozen)} \quad \text{ // termination criterion} \)
        \( \text{while (NOT YET AT equilibrium FOR THIS TEMPERATURE)} \)
            \( s' \leftarrow \text{Choose a random solution from neighborhood } \mathcal{N}(s); \quad \text{ // e.g., select a random 2-opt move} \)
            \( \Delta E \leftarrow f(s') - f(s); \)
            \( \text{if } (\Delta E \leq 0) \quad \text{ // downhill, locally improving move} \)
                \( s \leftarrow s'; \)
                \( \text{if } (f(s) < f(s^{\text{best}})) \)
                    \( s^{\text{best}} \leftarrow s; \)
            \( \text{else} \quad \text{// uphill move} \)
                \( r \leftarrow \text{Choose a random number uniformly from } [0,1]; \)
                \( \text{if } (r < e^{-\Delta E/T}) \quad \text{ // accept the uphill, not improving, move} \)
                    \( s \leftarrow s'; \)
        \( \text{end if} \)
    \( \text{end while} \)
    \( T \leftarrow \text{Lower the temperature according to the selected cooling schedule}; \)
\( \text{end while} \)
\( \text{return } s^{\text{best}}; \)
MC and Simulated Annealing

- **States** (let’s assume they are countable): $x \in X^n \subseteq \mathbb{R}^n$
- We don’t know where the max is, but we have the function $f(x)$
- **Neighborhood $N$**: which states can communicate with each other
- The chain / search moves from one state to the other based on $N$ and **transition probabilities** $\Rightarrow$ What are they? $\Rightarrow \exp(-\Delta E/T)$
- **What is a desired target stationary distribution?** Something that assigns probability 1 to the state corresponding to the global max

$$\max_x f(x) \quad x \in X^n \subseteq \mathbb{R}^n$$
MC and Simulated Annealing

\[
\max_x f(x) \quad x \in X^n \subseteq \mathbb{R}^n
\]

- For each choice of T, the transition probabilities “enforce” to a certain extent the next state being better (higher value of \( f \)) than current one
- The stationary distribution will assign higher probabilities to the states with higher function values
- Each new while loop, will start from a progressively better state
- Unfortunately, the decrease in T must follow an exponential scheme to guarantee that at T=0 the stationary distribution is the desired one
procedure SimulatedAnnealing()
    \( S = \{\text{set of all feasible solutions}\}; \)
    \( \mathcal{N} = \text{neighborhood structure defined over } S; \)
    \( s \leftarrow \text{Generate a starting feasible solution}; \)
    \( s^{\text{best}} \leftarrow s; \)
    \( T \leftarrow \text{Determine a starting value for temperature}; \)
    while (NOT YET frozen) // termination criterion
        while (NOT YET AT equilibrium FOR THIS TEMPERATURE)
            \( s' \leftarrow \text{Choose a random solution from neighborhood } \mathcal{N}(s); \)
            \( \Delta E \leftarrow f(s') - f(s); \)
            if \( (\Delta E \leq 0) \) // downhill, locally improving move
                \( s \leftarrow s'; \)
                if \( (f(s) < f(s^{\text{best}})) \)
                    \( s^{\text{best}} \leftarrow s; \)
            else // uphill move
                \( r \leftarrow \text{Choose a random number uniformly from } [0,1]; \)
                if \( (r < e^{-\Delta E/T}) \) // accept the uphill, not improving, move
                    \( s \leftarrow s'; \)
            end if
        end while
        \( T \leftarrow \text{Lower the temperature according to the selected cooling schedule}; \)
    end while
    return \( s^{\text{best}}; \)
General on Monte Carlo methods

- **Monte Carlo** is the art of approximating an expectation by the sample mean of a function of simulated random variables.

- Hit-or-miss (acceptance-rejection) Monte Carlo integration of a function $f(x)$

  \[
  y = \int_{a}^{b} f(x) \, dx
  \]

  - Draw $N$ random points from a uniform distribution in $[a, b] \times [0, y_0]$
  - Count the number $n$ falling below $f(x)$ to estimate the area below the function
  - $\hat{y} = \frac{n}{N} \cdot [y_0(b - a)]$

- Crude / Simple Monte Carlo: generate $N$ random $x$ points from a uniform distribution in $[a, b]$ and compute:

  \[
  \hat{y} = [b - a] \cdot \frac{1}{N} \sum_{i} f(x_i) \quad \hat{y} \to y \quad \text{as} \quad N \to \infty
  \]

- Convergence of Monte Carlo methods to the true expectation is based, in general, on the law of large numbers