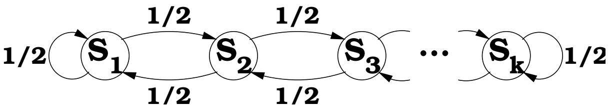
The Propp-Wilson Algorithm in practice: the need for sandwiching

When using the Propp-Wilson algorithm we in general need to start a chain in every possible state of the state space. This poses no problem when S is a set of moderate size, but when S is a very large set the algorithm becomes cumbersome.

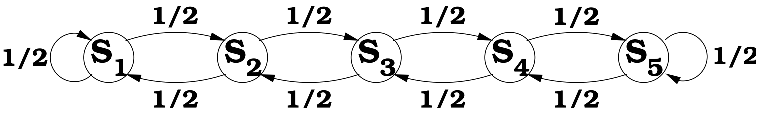
One way around this problem is to start chains in every state of a small subset of S and still be able to determine when "all" chains have coalesced.

An example: the ladder walk

Consider a Markov chain on state space $S = \{s_1, ..., s_k\}$, having the following transition graph.



This chain tries to move up or down at each time. If it is in one of its endpoints and tries to make a "wrong" move it stays put. Let $k = 5$ and let us simulate the chain according to the Propp-Wilson algorithm and see what happens.



See fig 11.1 for a typical simulation of the ladder walk with $k = 5$. In this case we use the following update function.

$$\phi(s_j, u) = \begin{cases} s_j & \text{, if } u < 0.5 \text{ and } j = 1 \\ s_{j-1} & \text{, if } u < 0.5 \text{ and } j \in \{2,3,4,5\} \\ s_{j+1} & \text{, if } u \geq 0.5 \text{ and } j \in \{1,2,3,4\} \\ s_j & \text{, if } u \geq 0.5 \text{ and } j = 5 \end{cases}$$

The coupling is done by using *the same* random numbers in all five chains. Due to the coupling they all try to move up or down at each time. Note that all we have to do is to simulate the chains starting in s_1 and s_5, when they meet every other chain is squeezed in between them.

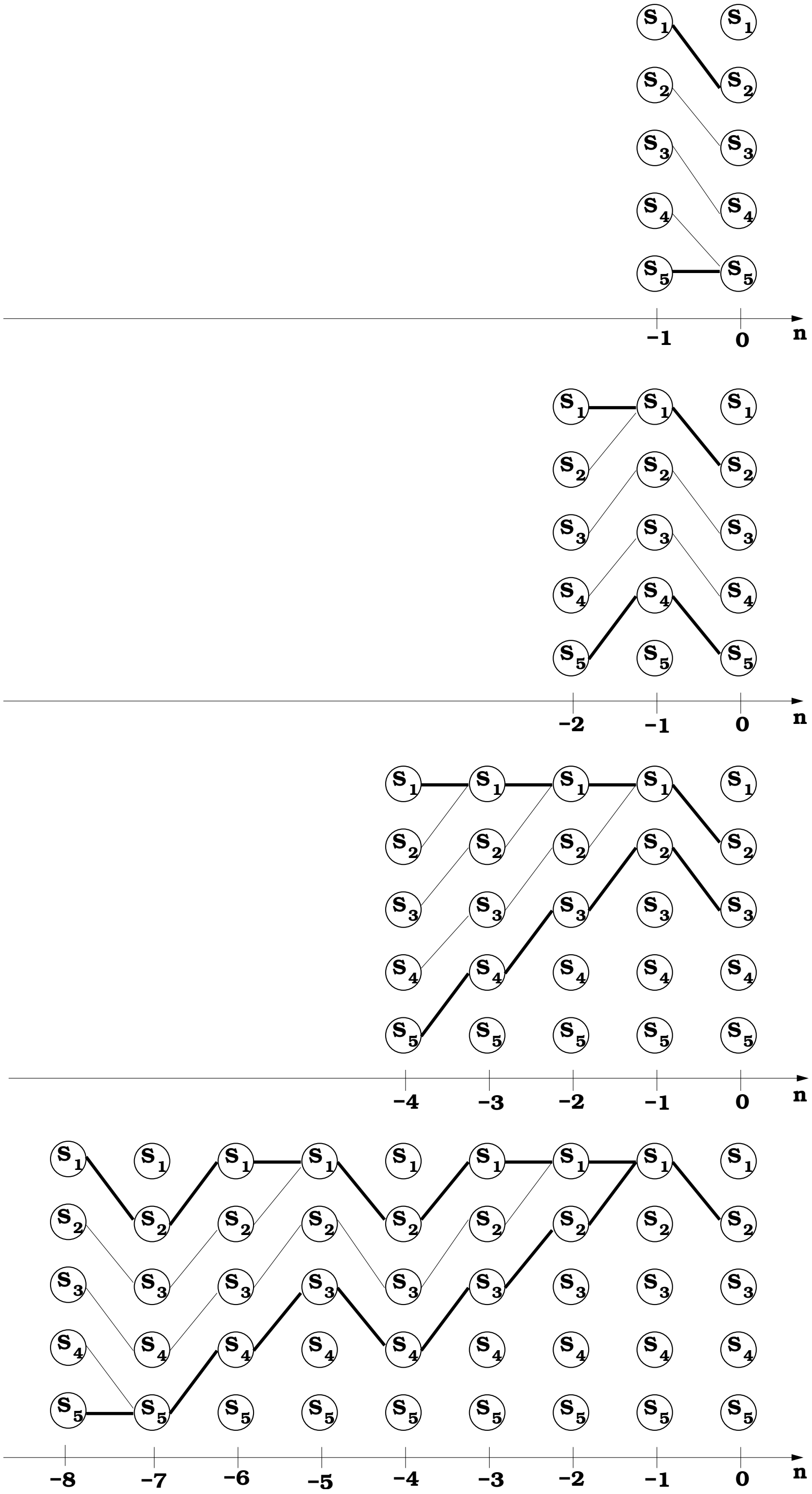


Figure 11.1: A typical Propp-Wilson simulation of the ladder walk. The chains starting in the maximum and minimum state a drawn by thick lines. Here the maximum and minimum are defined by the ordering, $s_1 > s_2 > s_3 > s_4 > s_5$, of the state space.

Sandwiching

The technique used in the example is called sandwiching. Before using it the chain have to fulfill some requirements.

- There has to exist an ordering, $\preceq$, of the state space. A partial order will be enough, as long as there exists a largest and a smallest element according to that order.
- The update function $\phi : S \times [0,1] \to S$ must preserve that order, i.e.

  $$s_i, s_j \in S : s_i \preceq s_j \quad \Rightarrow \quad \phi(s_i, U) \preceq \phi(s_j, U)$$

  must hold.

Another example: the Ising model

We will see that we can apply this sandwiching technique when simulating the Ising model using MCMC. First we introduce the model, then we present the Gibbs sampler, an ordering of the state space and an order preserving update function. After that we put everything together to a working Propp-Wilson simulation algorithm.

The Ising model: Let $G = (V, E)$ be a graph. The Ising model is a way of randomly assigning $-1/+1$ to each vertex in G according to a certain probability distribution. Two quantities determine the model, the inverse temperature β and the energy, H. The energy of an element $\xi \in \{-1, +1\}^V$ is defined as

$$H(\xi) = -\sum_{\langle x,y\rangle \in E} \xi(x)\xi(y)$$

Many edges with different configurations in each end give us high energy, and a configuration with many vertices having the same configuration gives a low energy. To each configuration ξ we assign probability

$$\pi_{G,\beta}(\xi) = \frac{1}{Z_{G,\beta}}\, e^{-\beta H(\xi)}$$

where $Z_{G,\beta}$ is just a normalizing constant.

$$Z_{G,\beta} = \sum_{\xi \in \{-1,+1\}^V} e^{-\beta H(\xi)}$$

The Gibbs sampler: To be able to run the MCMC algorithm at all we need an update function. For MCMC's we do the following. Given a configuration ξ we update it to ξ' by the following procedure.

1. Pick a vertex $v \in V$ uniformly at random.
2. Let $\xi(v) = a$ with probability $\pi_{G,\beta}(\xi(v) = a|\xi^v)$ for any $a \in \{-1, +1\}$. For the Ising model this conditional probability becomes ...

$$\begin{aligned}\pi_{G,\beta}(\xi'(v) = +1|\xi^v) &= \frac{\pi_{G,\beta}(\xi(v) = +1, \xi^v)}{\pi_{G,\beta}(\xi^v)} = \frac{\pi_{G,\beta}(\xi(v) = +1, \xi^v)}{\pi_{G,\beta}(\xi(v) = -1, \xi^v) + \pi_{G,\beta}(\xi(v) = +1, \xi^v)} \\ &= \left(1 + \frac{\pi_{G,\beta}(\xi(v) = -1, \xi^v)}{\pi_{G,\beta}(\xi(v) = +1, \xi^v)}\right)^{-1} = \cdots = \left(1 + e^{-2\beta \sum_{\{z:\langle z,v\rangle \in E\}} \xi(z)}\right)^{-1}\end{aligned}$$

3. For $w \in V \setminus \{v\}$ let $\xi'(w) = \xi(w)$.

With this probability we can later construct the right update function.

Ordering of the state space: Let us introduce an ordering $\preceq$ of $\{-1,+1\}^V$ by the following. Note that this is not a total ordering, just a partial ordering. There are configurations $\xi, \eta \in \{-1,+1\}^V$ such that neither $\xi \preceq \eta$ nor $\eta \preceq \xi$ holds. For any $\xi_1, \xi_2 \in \{-1,+1\}^V$ we say that $\xi_1 \preceq \xi_2$ if for all $v \in V$ we have

$$\xi_1(v) \leq \xi_2(v)$$

We cannot compare every configuration with every other configuration, but there are a minimum and maximum configuration. The maximum configuration is the with $+1$ at every vertex, and the minimum one has -1 at every vertex, let us denote them ξ^+ and ξ^- respectively.

$$\xi^{\max} = \{\xi^+(v) = +1 : v \in V\}$$

$$\xi^{\min} = \{\xi^-(v) = -1 : v \in V\}$$

Order preserving update function: We have several choices for an update function, but we have to choose the right one, otherwise it will not be order preserving. Let us try the following version of $\phi : \{-1,+1\}^V \times [0,1] \to \{-1,+1\}^V$ here defined for each vertex.

$$\phi(\xi(w),u) \begin{cases} +1 & \text{, if } w = v \text{ and } u \leq \left(1 + e^{-2\beta \sum_{\{z:\langle z,v\rangle \in E\}} \xi(z)}\right)^{-1} \\ -1 & \text{, if } w = v \text{ and } u > \left(1 + e^{-2\beta \sum_{\{z:\langle z,v\rangle \in E\}} \xi(z)}\right)^{-1} \\ \xi(w) & \text{, otherwise} \end{cases}$$

Let ξ_1, ξ_2 be two configurations such that $\xi_1 \leq \xi_2$ and let u be a $U[0,1]$ random number. We use the update function to generate configurations ξ_1', ξ_2'. Suppose $\xi_1'(v) = +1$, to show that ϕ is order preserving we must show that $\xi_2'(v) = +1$.

Since $\xi_1'(v) = +1$ we know that

$$u \leq \left(1 + e^{-2\beta \sum_{\{z:\langle z,v\rangle \in E\}} \xi_1(z)}\right)^{-1}$$

From $\xi_1 \leq \xi_2$ we get that

$$\left(\sum_{\{z:\langle z,v\rangle \in E\}} \xi_1(z)\right) \leq \left(\sum_{\{z:\langle z,v\rangle \in E\}} \xi_2(z)\right)$$

which in turn implies

$$u \leq \left(1 + e^{-2\beta \sum_{\{z:\langle z,v\rangle \in E\}} \xi_1(z)}\right)^{-1} \leq \left(1 + e^{-2\beta \sum_{\{z:\langle z,v\rangle \in E\}} \xi_2(z)}\right)^{-1}$$

and we see that $\xi_2'(v) = +1$ as required.

Putting it all together: We simulate two chain $(X_0, X_1, ...)$ and $(X'_0, X'_1, ...)$ by letting

$$X_0 = \xi^{\min}$$

and

$$X'_0 = \xi^{\max}.$$

We fix a sequence of starting times $(N_1, N_2, N_3, ...) = (1, 2, 4, ...)$. At each integer time we update the chains by using the above defined update function in the following manner.

$$X_{-n+1} = \phi(X_{-n}, U_{-n+1})$$

and

$$X'_{-n+1} = \phi(X'_{-n}, U_{-n+1})$$

Where the two chains are coupled by using the same random number. We follow the Propp-Wilson simulation schedule and stop when we for some sequence of times $-N_m, -N_m + 1, ..., 1, 0$ have $X_0 = X'_0 = \eta$ for some $\eta \in \{-1, +1\}^V$ and we output η as distributed according to $\pi_{G,\beta}$. configuration.