Appendix B

Algorithms

Here we adopt a standard notation (pseudocode) for defining algorithms [33, 154]. Assignment of a value α to a variable β is denoted by $\beta \leftarrow \alpha$. The branching probability $P_{\alpha,\beta}$ is represented as $P[\alpha, \beta]$ if it is stored in matrix form, and as $P[\beta][\alpha]$ if it is stored in the form of adjacency lists.

Algorithm B.1 Calculate the pathway sum $\mathcal{S}^{C_N}_{\alpha,\beta}$

Require: Chain nodes are numbered $0, 1, 2, \ldots, N-1$
Require: $1 < N$
Require: $\alpha, \beta \in \{0, 1, 2, \ldots, N-1\}$
Require: $P[i, j]$ is the probability of branching from node j to node i

if $\alpha < \beta$ **then**
    $h \leftarrow 1$; $t \leftarrow N-2$; $s \leftarrow 1$
else
    $h \leftarrow N-2$; $t \leftarrow 1$; $s \leftarrow -1$
end if
$L \leftarrow 1$
for all $i \in \{h, h+s, h+2s, \ldots, \alpha\}$ **do**
    $L \leftarrow 1/(1 - P[i-s, i]P[i, i-s]L)$
end for
$\Pi \leftarrow 1$
for all $i \in \{\alpha+s, \alpha+2s, \alpha+3s, \ldots, \beta\}$ **do**
    $\Pi \leftarrow P[i-s, i]L\Pi$
    $L \leftarrow 1/(1 - P[i-s, i]P[i, i-s]L)$
end for
$R \leftarrow 1$
for all $i \in \{t, t-s, t-2s, \ldots, \beta\}$ **do**
    $R \leftarrow 1/(1 - P[i+s, i]P[i, i+s]R)$
end for
return $LR\Pi/(L - LR + R)$

Algorithm B.2 Calculate the pathway sum $\mathcal{S}_{a,b}^{G_N}$

Require: $1 < N$
Require: $a, b \in \{0, 1, 2, \ldots, N-1\}$
Require: W is a boolean array of size N with every element initially set to True
Require: N_W is the number of True elements in array W (initialised to N)
Require: $P[i,j]$ is the probability of branching from node j to node i
Require: $AdjIn[i]$ and $AdjOut[i]$ are the lists of indices of all nodes connected to node i via incoming and outgoing edges, respectively

Recursive function $F(\alpha, \beta, W, N_W)$

$W[\beta] \leftarrow$ False
$N_W \leftarrow N_W - 1$
if $\alpha = \beta$ and $N_W = 0$ **then**
  $\Sigma \leftarrow 1$
else
  $\Sigma \leftarrow 0.0$
  for all $i \in AdjOut[\beta]$ **do**
    for all $j \in AdjIn[\beta]$ **do**
      if $W[i]$ and $W[j]$ **then**
        $\Sigma \leftarrow \Sigma + P[\beta, j] F(j, i, W, N_W) P[i, \beta]$
      end if
    end for
  end for
  $\Sigma \leftarrow 1/(1 - \Sigma)$
  if $\alpha \neq \beta$ **then**
    $\Lambda \leftarrow 0.0$
    for all $i \in AdjOut[\beta]$ **do**
      if W[i] **then**
        $\Lambda \leftarrow \Lambda + F(\alpha, i, W, N_W) P(i, \beta)$
      end if
    end for
    $\Sigma \leftarrow \Sigma\Lambda$
  end if
end if
$W[\beta] \leftarrow$ True
$N_W \leftarrow N_W + 1$
return Σ

Algorithm B.3 Calculate the pathway sum $\mathcal{S}^{\mathcal{G}_N}_{\alpha,\beta}$ from every source to every sink, and the mean escape time for every source in a dense graph $\mathcal{G}_N$

Require: Nodes are numbered $0, 1, 2, \ldots, N-1$
Require: Sink nodes are indexed first, source nodes - last
Require: i is the index of the first intermediate node
Require: s is the index of the first source node
Require: In case there are no intermediate nodes $i = s$, otherwise $i < s$
Require: $1 < N$
Require: $i, s \in \{0, 1, 2, \ldots, N-1\}$
Require: $\tau[\alpha]$ is the waiting time for node α, $\alpha \in \{i, i+1, i+2, \ldots, N-1\}$
Require: $P[i, j]$ is the probability of branching from node j to node i

```
for all γ ∈ {i, i+1, i+2, ..., s−1} do
  for all β ∈ {γ+1, γ+2, ..., N−1} do
    if P[γ, β] > 0 then
      τ[β] ← (τ[β] + τ[γ]P[γ, β])/(1 − P[β, γ]P[γ, β])
      for all α ∈ {0, 1, 2, ..., N−1} do
        if α ≠ β and α ≠ γ then
          P[α, β] ← (P[α, β] + P[α, γ]P[γ, β])/(1 − P[β, γ]P[γ, β])
        end if
      end for
      P[γ, β] ← 0.0
    end if
  end for
  for all α ∈ {0, 1, 2, ..., N−1} do
    P[α, γ] ← 0.0
  end for
end for
for all α ∈ {s, s+1, s+2, ..., N−1} do
  for all β ∈ {s, s+1, s+2, ..., N−1} do
    if α ≠ β and P[α, β] > 0 then
      P_{α,β} ← P[α, β]
      P_{β,α} ← P[β, α]
      T ← τ[α]
      τ[α] ← (τ[α] + τ[β]P_{β,α})/(1 − P_{α,β}P_{β,α})
      τ[β] ← (τ[β] + T P_{α,β})/(1 − P_{α,β}P_{β,α})
      for all γ ∈ {0, 1, 2, ..., i−1} ∪ {s, s+1, s+2, ..., N−1} do
        T ← P[γ, α]
        P[γ, α] ← (P[γ, α] + P[γ, β]P_{β,α})/(1 − P_{α,β}P_{β,α})
        P[γ, β] ← (P[γ, β] + T P_{α,β})/(1 − P_{α,β}P_{β,α})
      end for
      P[α, β] ← 0.0
      P[β, α] ← 0.0
    end if
  end for
end for
```

Algorithm B.4 Detach node γ from an arbitrary graph $\mathcal{G}_N$

Require: $1 < N$
Require: $\gamma \in \{0, 1, 2, \ldots, N-1\}$
Require: $\tau[i]$ is the waiting time for node i
Require: $Adj[i]$ is the ordered list of indices of all nodes connected to node i via outgoing edges
Require: $|Adj[i]|$ is the cardinality of $Adj[i]$
Require: $Adj[i][j]$ is the index of the jth neighbour of node i
Require: $P[i]$ is the ordered list of probabilities of leaving node i via outgoing edges, $|P[i]| = |Adj[i]|$
Require: $P[i][j]$ is the probability of branching from node i to node $Adj[i][j]$

for all $\beta_\gamma \in \{0, 1, 2, \ldots, |Adj[\gamma]| - 1\}$ **do**
  $\beta \leftarrow Adj[\gamma][\beta_\gamma]$
  $\gamma_\beta \leftarrow -1$
  for all $i \in \{0, 1, 2, \ldots, |Adj[\beta]| - 1\}$ **do**
    if $Adj[\beta][i] = \gamma$ **then**
      $\gamma_\beta \leftarrow i$
      break
    end if
  end for
  if not $\gamma_\beta = -1$ **then**
    $P_{\beta,\beta} \leftarrow 1/(1 - P[\beta][\gamma_\beta]P[\gamma][\beta_\gamma])$
    $P_{\beta,\gamma} \leftarrow P[\beta][\gamma_\beta]$
    $Adj[\beta] \leftarrow \{Adj[\beta][0], Adj[\beta][1], \ldots, Adj[\beta][\gamma_\beta - 1], Adj[\beta][\gamma_\beta + 1], \ldots\}$
    $P[\beta] \leftarrow \{P[\beta][0], P[\beta][1], \ldots, P[\beta][\gamma_\beta - 1], P[\beta][\gamma_\beta + 1], \ldots\}$
    for all $\alpha_\gamma \in \{0, 1, 2, \ldots, |Adj[\gamma]| - 1\}$ **do**
      $\alpha \leftarrow Adj[\gamma][\alpha_\gamma]$
      if not $\alpha = \beta$ **then**
        if exists edge $\beta \to \alpha$ **then**
          for all $i \in \{0, 1, 2, \ldots, |Adj[\beta]| - 1\}$ **do**
            if $Adj[\beta][i] = \alpha$ **then**
              $P[\beta][i] \leftarrow P[\beta][i] + P_{\beta,\gamma}P[\gamma][\alpha_\gamma]$
              break
            end if
          end for
        else
          $Adj[\beta] \leftarrow \{\alpha, Adj[\beta][0], Adj[\beta][1], Adj[\beta][2], \ldots\}$
          $P[\beta] \leftarrow \{P_{\beta,\gamma}P[\gamma][\alpha_\gamma], P[\beta][0], P[\beta][1], P[\beta][2], \ldots\}$
        end if
      end if
    end for
    for all $i \in \{0, 1, 2, \ldots, |P[\beta]| - 1\}$ **do**
      $P[\beta][i] \leftarrow P[\beta][i]P_{\beta,\beta}$
    end for
    $\tau_\beta \leftarrow (\tau_\beta + P_{\beta,\gamma}\tau_\gamma)\, P_{\beta,\beta}$
  end if
end for