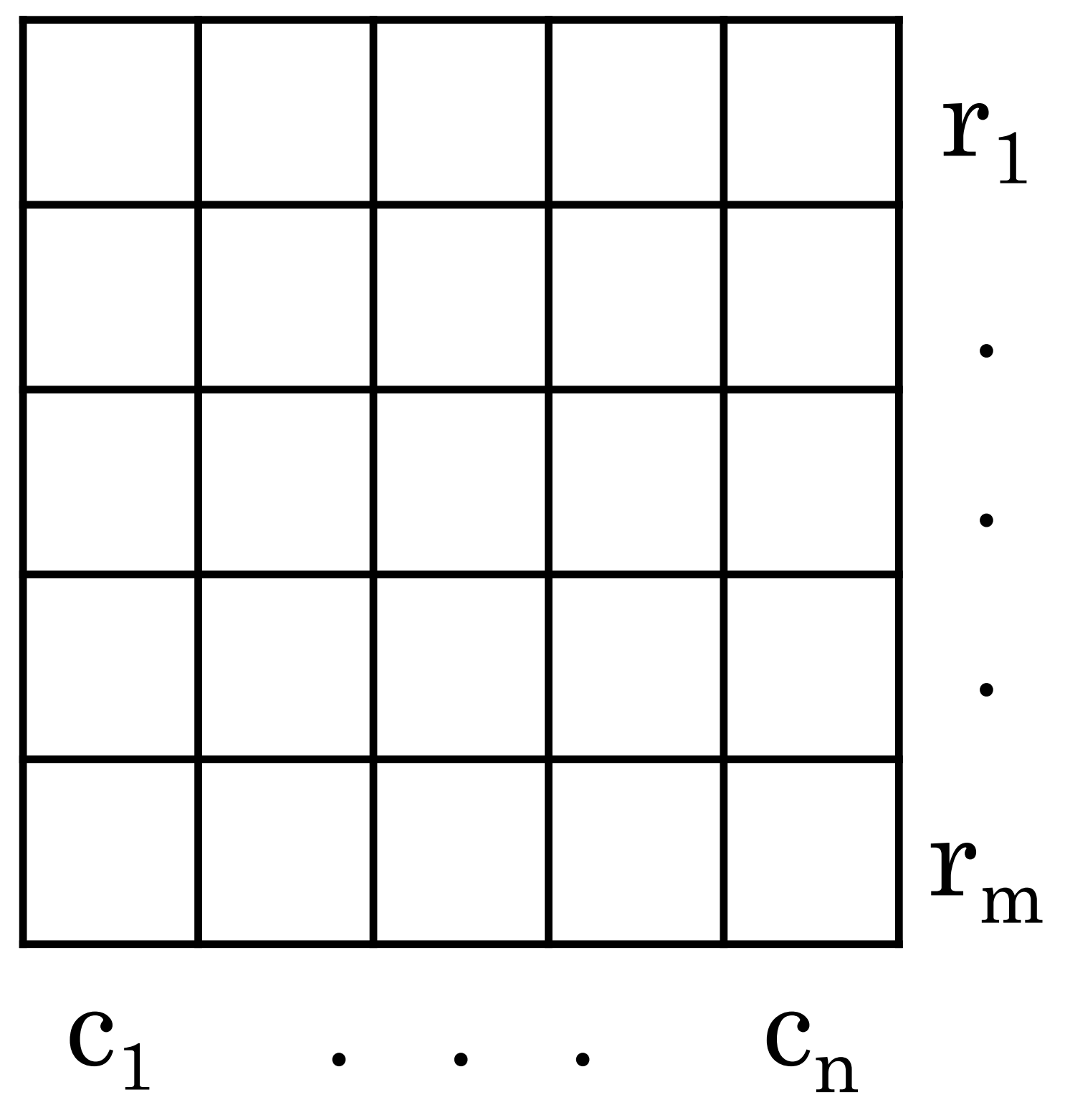
Applications of Randomized Algorithms to Counting Problems

Approximate solution to the Binary Contingency Tables Problem using Markov Chain Monte Carlo methods

Yashil Sukurdeep & Professor Paul Dupuis, Department of Applied Mathematics, Brown University, Providence, RI, USA

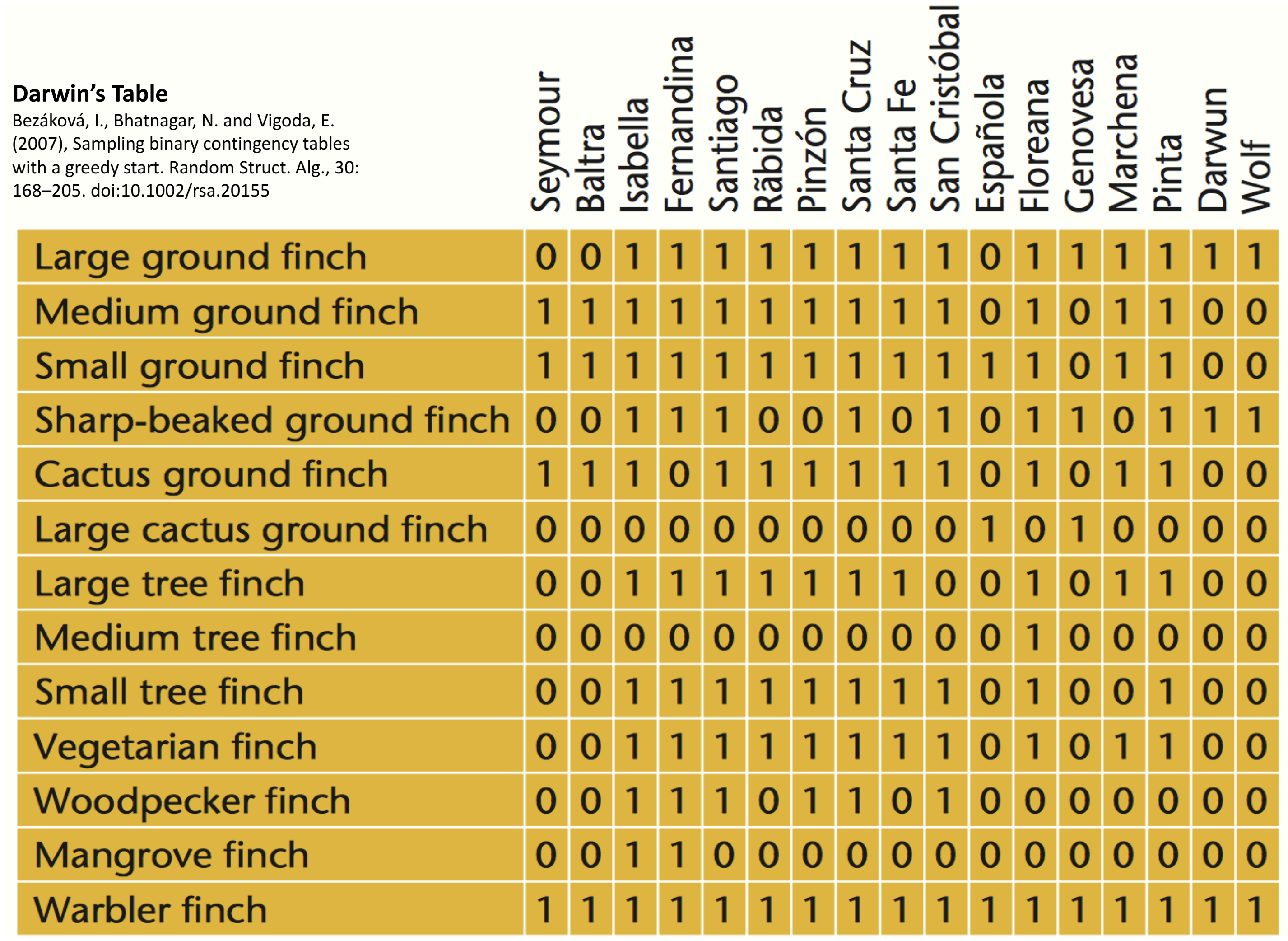
Binary Contingency Tables Problem

The **binary contingency tables problem** is to count the number of $\{0,1\}$-valued $\mathbf{m}$ x $\mathbf{n}$ matrices with row sums and column sums given by two vectors of positive integers, $\mathbf{r} = (\mathrm{r}_1, \ldots, \mathrm{r}_\mathrm{m})$ and $\mathbf{c} = (\mathrm{c}_1, \ldots, \mathrm{c}_\mathrm{n})$ respectively.



Why is this problem of interest?

It is an NP-hard problem, making it interesting from a theoretical perspective. It is also interesting from an applied perspective, with applications in biology, statistics and economics. In fact, the problem stems from Darwin, who collected data on birds living in the Galapagos islands. Darwin presented his data in a binary table, and wanted to know how likely it was that he observed the data that he collected. To find answers to the question, knowledge of the number of binary tables with row and column sums equal to those in Darwin's table would be helpful!



Darwin's Table

Bezáková, I., Bhatnagar, N. and Vigoda, E. (2007), Sampling binary contingency tables with a greedy start. Random Struct. Alg., 30: 168–205. doi:10.1002/rsa.20155

	Seymour	Baltra	Isabella	Fernandina	Santiago	Rābida	Pinzón	Santa Cruz	Santa Fe	San Cristóbal	Española	Floreana	Genovesa	Marchena	Pinta	Darwun	Wolf
Large ground finch	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
Medium ground finch	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Small ground finch	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
Sharp-beaked ground finch	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1
Cactus ground finch	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0
Large cactus ground finch	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Large tree finch	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0
Medium tree finch	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Small tree finch	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0
Vegetarian finch	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
Woodpecker finch	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0
Mangrove finch	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Warbler finch	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Algorithm I: Straightforward MCMC

The idea is to construct a Markov chain on $\mathscr{X}$, the space of all $\mathbf{m}$ x $\mathbf{n}$ binary tables with column sums satisfied, whose target distribution is:

$$\pi(x) = \frac{e^{-\frac{1}{\tau}V(x)}}{Z_\tau} \quad \forall\, x \in \mathscr{X},$$

where $\tau \in \mathbb{R}^{+}$ is the temperature parameter, Z_τ is the normalization constant, and $V(x)$, the energy function, is given by:

$$V(x) = \sum_{i=1}^{m} \left| \sum_{j=1}^{n} x_{ij} - r_i \right|$$

With this choice of energy function, $\pi(x)$ takes its maximal value when $V(x)$ = 0, and $V(x)$ = 0 iff all the row sums are satisfied. Hence, π places higher probability on tables with row sums $\mathbf{r} = (\mathrm{r}_1, \ldots, \mathrm{r}_\mathrm{m})$ and column sums $\mathbf{c} = (\mathrm{c}_1, \ldots, \mathrm{c}_\mathrm{n})$.

For convergence to the desired target distribution π, we evolve a table $\boldsymbol{x}$ from $\mathscr{X}$ for a single dynamical step using the Gibbs move below:

- Pick a column j $\in \{1,\ldots,\mathrm{n}\}$ from $\boldsymbol{x}$ uniformly at random.
- Pick a '1' and a '0' uniformly at random from column j, and swap them to obtain a new table $\boldsymbol{y}$.
- Set the next table in the Markov chain equal to $\boldsymbol{y}$ with probability $\delta = \min\left\{1, e^{-\frac{1}{\tau}V(y)}\right\}$. Otherwise, set it to $\boldsymbol{x}$.

By the ergodic theorem, we can estimate the probability of the set of binary tables with row sums $\mathbf{r}$ and column sums $\mathbf{c}$ by computing the fraction of times that our Markov chain of length T visits binary tables with energy $V(x)$ = 0. We can then convert our probability estimates into 'count' estimates for the number of binary tables with row sums $\mathbf{r}$ and column sums $\mathbf{c}$.

Algorithm II: Parallel Tempering (PT)

PT involves running independent Markov chains at two (or more) temperatures on $\mathscr{X}$, with each chain having target distribution:

$$\pi_i(x) = \frac{e^{-\frac{1}{\tau_i}V(x)}}{Z_{\tau_i}} \quad \forall\, x \in \mathscr{X}, \;\; i = 1{,}2$$

where $\tau_1, \tau_2 \in \mathbb{R}^{+}$ are the temperatures (with $\tau_1 < \tau_2$), Z_{τ_i} are the normalization constants, and $V(x)$ is the same energy function as the one used in algorithm I.

In PT, the global system of Markov chains should converge to the product measure of the stationary distributions: $\pi = \pi_1 \text{ x } \pi_2$. To ensure this, we evolve each individual chain for $\mathbf{K}$ dynamical steps using the Gibbs move from algorithm I. This updates the locations of the two chains to two tables $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$. We then attempt to swap these locations with acceptance probability $\delta = \min\{1, A\}$, where:

$$A = \frac{e^{-\frac{1}{\tau_1}V(x^{(2)})}\; e^{-\frac{1}{\tau_2}V(x^{(1)})}}{e^{-\frac{1}{\tau_1}V(x^{(1)})}\; e^{-\frac{1}{\tau_2}V(x^{(2)})}}$$

We can then compute probability estimates and 'count' estimates just as described for algorithm I.

Performance & Open Questions

Straightforward MCMC gives unbiased, low-variance estimates for low-dimensional problems $(\mathrm{m}, \mathrm{n} < 10)$. PT can handle higher dimensional problems, and reduces variance of estimates. Open questions include:

- Selecting temperatures to optimize performance?
- Effects of the energy landscape on performance?
- Can we improve on the performance of parallel tempering by using infinite swapping (INS)?